meenakshi Srivastava

Meenakshi PhD Thesis.pdf

PhD Thesis

PhD Scholar

Jawaharlal Nehru University

Document Details

Submission ID
trn:oid:::1:3529840373

Submission Date
Apr 7, 2026, 7:24 PM GMT+5:30

Download Date
Apr 7, 2026, 7:29 PM GMT+5:30

File Name
Meenakshi_PhD_Thesis.pdf

File Size
90.5 MB

262 Pages

73,078 Words

406,749 Characters

7% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 8 words)

Exclusions

- 3 Excluded Sources

Match Groups

431 Not Cited or Quoted 7%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

3% Internet sources
6% Publications
2% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **431** Not Cited or Quoted 7%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

3% Internet sources
6% Publications
2% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Student papers	Indira Gandhi Delhi Technical University for Women <1%
2	Internet	link.springer.com <1%
3	Internet	pushkarphd2019.github.io <1%
4	Publication	Himanshu Nandanwar, Rahul Katarya. "Deep learning enabled intrusion detectio... <1%
5	Publication	K. V. Sambasivarao, Anasuya Sesha Roopa Devi Bhima. "Artificial Intelligence, Co... <1%
6	Publication	"Intelligent Vision and Computing", Springer Science and Business Media LLC, 2026 <1%
7	Publication	M. Anu Kiruthika, Angelin Gladston, H. Khanna Nehemiah. "Lightweight dual-stag... <1%
8	Internet	ebin.pub <1%
9	Internet	arxiv.org <1%
10	Internet	dspace.dtu.ac.in:8080 <1%

11	Student papers	
University of North Florida		<1%
12	Internet	
download.bibis.ir		<1%
13	Internet	
www.frontiersin.org		<1%
14	Student papers	
Thapar University, Patiala		<1%
15	Publication	
"Computational Intelligence in Engineering Science", Springer Science and Busin...		<1%
16	Student papers	
University of Sunderland		<1%
17	Publication	
"Data Science, AI and Applications", Springer Science and Business Media LLC, 2025		<1%
18	Publication	
"Smart Data Intelligence", Springer Science and Business Media LLC, 2026		<1%
19	Publication	
Krishnan, Ramesh. "Collaborative Innovation, Life Cycle Analysis and Network Op...		<1%
20	Internet	
repository.rudn.ru		<1%
21	Internet	
dirros.openscience.si		<1%
22	Internet	
thesai.org		<1%
23	Publication	
"Next-Generation Networks and Deployable Artificial Intelligence", Springer Scie...		<1%
24	Publication	
"Machine Vision in Plant Leaf Disease Detection for Sustainable Agriculture", Spri...		<1%

25 Internet
www.ijcaonline.org <1%

26 Internet
dokumen.pub <1%

27 Publication
B. G. Nagaraja, S. Kannadhasan. "Information and Communication Systems", CRC... <1%

28 Publication
Himanshu Nandanwar, Rahul Katarya. "Alpha-Net: A dependable and trustworthy... <1%

29 Publication
Shankar Babu, Mahesh Babu Kota. "Synergies in Smart and Virtual Systems using... <1%

30 Publication
Bhoomika Batra, Vinod Kumar Shukla, Vaibhav Bhatnagar. "Computational Techn... <1%

31 Student papers
Higher Education Commission Pakistan <1%

32 Publication
"Proceedings of International Conference on Innovations in Data Science", Spring... <1%

33 Publication
Paolo Ferro, Harinadh Vemanaboina, Chander Prakash. "Computational Techniqu... <1%

34 Internet
www.isteonline.in <1%

35 Publication
Arshad Ali, Tanzeela Shakeel, Maryam Gulzar, Aamir Wali. "A systematic literatur... <1%

36 Publication
Bal S. Virdee, Tanweer Ali, Jaume Anguera, Suman Lata Tripathy. "Connecting Int... <1%

37 Internet
www.mdpi.com <1%

38 Publication
Patel, Suhasbhai Haribhai. "Automatic Object and Event Detection in Field Hocke... <1%

39 Publication
Arun Kumar Singh, Sandeep Saxena, A.K. Haghi. "Cybernetic Shield - Securing the ... <1%

40 Publication
Mithilesh Kumar Dubey, Kajal Verma, Khalil Ahmed, Sudha Dubey. "Artificial Intel... <1%

41 Publication
Pushpa Choudhary, Sambit Satpathy, Arvind Dagur, Dhirendra Kumar Shukla. "Re... <1%

42 Publication
V. Sharmila, S. Kannadhasan, A. Rajiv Kannan, P. Sivakumar, V. Vennila. "Challeng... <1%

43 Publication
Vinothini Arumugam, Aswiga R V. "Transfer Learning Based Deep Learning Model... <1%

44 Internet
www.tnsroindia.org.in <1%

45 Student papers
Galgotias University, Greater Noida <1%

46 Publication
Maheshbhai, Daxini Bhautik. "A Novel Deep Learning Approach for the Medical I... <1%

47 Internet
e-space.mmu.ac.uk <1%

48 Internet
saucis.sakarya.edu.tr <1%

49 Publication
Himanshu Nandanwar, Rahul Katarya. "TL-BILSTM IoT: transfer learning model fo... <1%

50 Publication
Tom Allen, Andy Alderson, Alan Godfrey, Duane Knudson, James Webster, Kazuya ... <1%

51 Publication
Wubetu Barud Demilie. "Plant disease detection and classification techniques: a c... <1%

52 Publication
"Proceedings of Fifth International Conference on Advances in Computer Enginee... <1%

53 Publication

Arvind Dagur, Sohit Agarwal, Dhirendra Kumar Shukla, Shabir Ali, Sandhya Sharm... <1%

54 Publication

"Advanced Network Technologies and Intelligent Computing", Springer Science a... <1%

55 Publication

Zhang, Tianxiao. "Efficient and Effective Object Detection and Recognition: From ... <1%

56 Internet

dspace.bracu.ac.bd <1%

57 Publication

"Computational Intelligence and Data Analytics", Springer Science and Business ... <1%

58 Publication

"Intelligent Systems", Springer Science and Business Media LLC, 2026 <1%

59 Publication

Anshul Verma, Pradeepika Verma. "Research Advances in Intelligent Computing",... <1%

60 Publication

Kang, Hua. "Towards Ubiquitous Internet of Things Applications with Deep Doma... <1%

61 Publication

Poonam Nandal, Mamta Dahiya, Meeta Singh, Arvind Dagur, Brijesh Kumar. "Pro... <1%

62 Publication

Ruqiang Yan, Jing Lin. "Equipment Intelligent Operation and Maintenance", CRC P... <1%

63 Internet

ijece.iaescore.com <1%

64 Internet

inass.org <1%

65 Internet

ruc.udc.es <1%

66 Publication

"Advanced Communication and Intelligent Systems", Springer Science and Busin... <1%

67 Publication

"Neural Information Processing", Springer Science and Business Media LLC, 2025 <1%

68 Publication

Hritwik Ghosh, Irfan Sadiq Rahat, Md. Mintajur Rahman Emon, Md. Jisan Mashrafi... <1%

69 Publication

Lu, Weiguo. "A Comprehensive Study of Gaussian Mixture Model With Application... <1%

70 Publication

Md. Abdur Rahman, Md. Tofael Ahmed Bhuiyan, Farzan Majeed Noori, Md Zia Ud... <1%

71 Internet

ccsuniversity.blr1.cdn.digitaloceanspaces.com <1%

72 Internet

www.researchgate.net <1%

73 Publication

"AI Technologies for Information Systems and Management Science", Springer Sc... <1%

74 Publication

"Innovative Computing and Communications", Springer Science and Business Me... <1%

75 Publication

"Intelligent and Fuzzy Systems", Springer Science and Business Media LLC, 2025 <1%

76 Publication

Pijush Dutta, Jyoti Sekhar Banerjee, Siddhartha Bhattacharyya, Debashis De et al. <1%

...

77 Publication

Shyam Sundhar, Riya Sharma, Priyansh Maheshwari, Suvidha Rupesh Kumar, T. S... <1%

78 Publication

Sukhpreet Kaur, Amanpreet Kaur, Manish Kumar. "Recent Advances in Computati... <1%

79 Internet

public-pages-files-2025.frontiersin.org <1%

80 Internet

jisem-journal.com <1%

81 Student papers
Addis Ababa Science and Technology University <1%

82 Student papers
Glasgow Caledonian University <1%

83 Publication
Harisu Abdullahi Shehu, Aniebietabasi Ackley, Marvellous Mark, Ofem Ebriba Ete... <1%

84 Publication
Khlefat, Hamza. "Recognizing Counterfeit Brand Logo Using Machine Learning", ... <1%

85 Student papers
Liverpool John Moores University <1%

86 Publication
Sriramulu Bojjagani, V. Dinesh Reddy, Sumalatha Saleti. "Privacy and Security in F... <1%

87 Student papers
University of Sheffield <1%

88 Publication
Ahmed, Abdelrehim H. "Shape from shading under various imaging conditions", ... <1%

89 Publication
Hanxiang Wang, Tri-Hai Nguyen, Tan N. Nguyen, Minh Dang. "PD-TR: End-to-end ... <1%

90 Student papers
VIT University <1%

91 Internet
journal.uob.edu.bh <1%

92 Publication
"Proceedings of International Conference on Artificial Intelligence and Networks... <1%

93 Publication
Fatemi Langroudi, Seyed Hamed. "Tapered-Precision Numerical Formats for Deep... <1%

94 Publication
Li, Jialu. "Learning from Auto-Generated Data to Build Diverse Multi-Modal Model... <1%

95 Publication
Priyadharshini Arputharaj, Kalaivanan Karunanithy. "A review on machine learni... <1%

96 Publication
Suresh Ukarande, Shailesh R. Nikam, Makarand G. Kulkarni, Nandana A. Prabhu e... <1%

97 Publication
Sushil Kumar Singh, Rajendrasinh B. Jadeja, Ashish Khanna, Pushan Kumar Dutta,... <1%

98 Publication
Uvarani Vignesh, Bala Subramanian Chokkalingam. "EnC-SVMWEL: Ensemble App... <1%

99 Internet
www.internationaljournalssrg.org <1%

100 Internet
www.preprints.org <1%

101 Publication
"Artificial Intelligence of Everything and Sustainable Development", Springer Scie... <1%

102 Publication
"Fifth Congress on Intelligent Systems", Springer Science and Business Media LLC... <1%

103 Publication
Abhishek Kumar, Ramesh C. Bansal, Deng Yan, Praveen Kumar. "Microgrid Handb... <1%

104 Publication
Amitabha Chakrabarty, Sarder Tanvir Ahmed, Md. Fahim Ul Islam, Syed Mahfuzul... <1%

105 Publication
Bidishasaimantini Nayak, Satyabhama Dash, Abhyarthana Bisoyi, Debi Prasad Da... <1%

106 Publication
C.H. Chen. "Signal and Image Processing for Remote Sensing", CRC Press, 2024 <1%

107 Publication
Jinlin Ma, Yuetong Wan, Weiqing Min, Ziping Ma, Lidao Tan, Shuqiang Jiang. "Fron... <1%

108 Publication
Khadija Javed, Guy Smagghe, Qi Wang, Humayun Javed, Yong Wang. "Artificial int... <1%

109 Publication

M Amudha, Dr.Brindha K. "Hybrid Paddy disease classification using Optimized St... <1%

110 Publication

R. Sudhakar, R. Sivaraj, M. Vijayakumar. "Intelligent contextual attention mechan... <1%

111 Publication

Tejashwini C. Gadag, D. R. Kumar Raja. "Adaptive deep learning framework for m... <1%

112 Publication

Um, Sara. "Rich, Fair, or Rich and Fair: Gender Bias Mitigation in Image Captionin... <1%

113 Publication

Zeinali Rizi, Behnam. "Enhancing AI Reliability: Probabilistic Inference and Neural... <1%

114 Internet

cdn.jagannathuniversity.org <1%

115 Internet

journal.universitasbumigora.ac.id <1%

116 Internet

tnsroindia.org.in <1%

117 Internet

www.paperdigest.org <1%

118 Internet

www.pnrjournal.com <1%

119 Publication

"Computational Intelligence in Pattern Recognition", Springer Science and Busin... <1%

120 Publication

"Computer Vision and Machine Learning in Agriculture", Springer Science and Bu... <1%

121 Publication

"Harnessing AI to Reshape the Future of Agriculture", Springer Science and Busin... <1%

122 Publication

Ahmed A. Abd El-Latif, Mohammed A ElAffendi, Mohamed Ali AlShara, Yassine Ma... <1%

123 Publication

Asif Shahriar Arnob, Ashfakul Karim Kausik, Zohirul Islam, Raiyan Khan, Adib Bin ... <1%

124 Publication

Cho, Yae Jee. "Faster, Incentivized, and Efficient Federated Learning: Theory and ... <1%

125 Publication

Debatosh Guha, Badal Chakraborty, Himadri Sekhar Dutta. "Computer, Communi... <1%

126 Publication

Gaurav Aggarwal, Rishabh Dev Shukla, Innocent Ewean Davidson, Ramesh Bansal... <1%

127 Publication

Hicham El Akhal, Aissa Ben Yahya, Abdelbaki El Belrhiti El Alaoui. "Positive discri... <1%

128 Publication

Imran Ullah Khan, Haseeb Ali Khan, Jong Weon Lee. "Dual-Stream Architecture En... <1%

129 Publication

Jaskaran Singh, Meenu Gupta, Rakesh Kumar. "Smart Technologies and Intelligen... <1%

130 Publication

Md Taimur Ahad, Noman H. Chowdhury, Ainuddin Ahmed, Angkita Averi Noor, Bo... <1%

131 Publication

Midhun P. Mathew, Sudheep Elayidom, V. P. Jagathy Raj, K. M. Abubeker. "Develo... <1%

132 Publication

Nishu Gupta, Sandeep S. Joshi, Milind Khanapurkar, Asha Gedam, Nikhil Bhave. "... <1%

133 Publication

Prit Mhala, Anushka Bilandani, Sanjeev Sharma. "Enhancing crop productivity wit... <1%

134 Publication

Sanjay Misra, Amit Jain, Manju Kaushik, Chitresh Banerjee, Rakhi Mutha. "Deep L... <1%

135 Publication

Soufiane Haddout, Tonni Agustiono Kurniawan. "Handbook of Artificial Intelligen... <1%

136 Publication

Thangaprakash Sengodan, Sanjay Misra, M Murugappan. "Advances in Electrical ... <1%

137 Publication
Xianyu Zhu, Jinjiang Li, Runchang Jia, Bin Liu, Zhuohan Yao, Aihong Yuan, Yingqiu ... <1%

138 Internet
ijisae.org <1%

139 Internet
norma.ncirl.ie <1%

140 Internet
sscoetjalgaon.ac.in <1%

141 Internet
www.iaeng.org <1%

142 Internet
www.ijisae.org <1%

143 Publication
"Advanced Computational and Communication Paradigms", Springer Science and... <1%

144 Publication
"Communication and Intelligent Systems", Springer Science and Business Media ... <1%

145 Publication
"Computational Intelligence and Image Processing in Agriculture", Wiley, 2025 <1%

146 Publication
"Data Processing and Networking", Springer Science and Business Media LLC, 2025 <1%

147 Publication
"Data Science and Applications", Springer Science and Business Media LLC, 2024 <1%

148 Publication
"Human and Machine Learning", Springer Science and Business Media LLC, 2018 <1%

149 Publication
"Intelligent Computing and Technologies", Springer Science and Business Media ... <1%

150 Publication
"Intelligent Computing", Springer Science and Business Media LLC, 2019 <1%

151 Publication
"Intelligent Systems and Pattern Recognition", Springer Science and Business Me... <1%

152 Publication
"Pattern Recognition and Computer Vision", Springer Science and Business Media... <1%

153 Publication
"Proceedings of Second International Conference on Advances in Computer Engi... <1%

154 Publication
"Proceedings of the Fifth International Conference on Trends in Computational a... <1%

155 Publication
"Recent Advancements in Artificial Intelligence", Springer Science and Business ... <1%

156 Publication
A. Punitha, S. Syedakbar, S. Jeyasudha. "Intelligent and Sustainable Systems: AI, G... <1%

157 Publication
A.V. Senthil Kumar, Ankita Chaturvedi, Atul Bansal, Rohaya Latip. "Artificial Intelli... <1%

158 Publication
Abhilasha Sharma, Vishwas Rathi, Anupam Biswas, Anil Singh, Omer Rana. "Multi... <1%

159 Publication
Ancy Stephen, A. Punitha, A. Chandrasekar. "Optimal deep generative adversarial... <1%

160 Publication
Anthony C. Ijeh, Kevin Curran. "Crime Science and Digital Forensics - A Holistic Vie... <1%

161 Publication
C. K. Sunil, C. D. Jaidhar, Nagamma Patil. "Systematic study on deep learning-base... <1%

162 Publication
Charu Khanna, Sahil Khanna. "Artificial Intelligence and Safety - A Practical Guide... <1%

163 Publication
Dhirendra Kumar Shukla, Shabir Ali, Sandhya Sharma. "Artificial Intelligence and ... <1%

164 Publication
Diana Susan Joseph, Pranav M Pawar, Kaustubh Chakradeo. "Real-time Plant Dise... <1%

165 Publication
Dip Kumar Saha, Mohammad Rasel Ahmed, Tushar Deb Nath, Rounakul Islam Bo... <1%

166 Publication
Gu, Wenchao. "Towards Intelligent Reliable Code Retrieval Based on Code Seman... <1%

167 Publication
Guowei Dai, Zhimin Tian, Jingchao Fan, C.K. Sunil, Christine Dewi. "DFN-PSAN: Mul... <1%

168 Publication
Imran Ahmed, Misbah Ahmad, Haythem Ghazouani, Walid Barhoumi, Gwanggil Je... <1%

169 Publication
Jin, Zengrui. "Adversarial and Reinforcement Learning Based Data Augmentation ... <1%

170 Publication
Lalit Mohan Goyal, Tanzila Saba, Amjad Rehman, Souad Larabi-Marie-Sainte. "Arti... <1%

171 Publication
Md Shofiqul Islam, Sunjida Sultana, Fahmid Al Farid, Md Nahidul Islam et al. "Mul... <1%

172 Publication
Mehta, Kiran. "Synthetic and Augmented Data Methods for Deep Learning-Based ... <1%

173 Publication
Muhammad Mahmood ur Rehman, Jizhan Liu, Aneela Nijabat, Muhammad Fahee... <1%

174 Publication
Nazmul Siddique, Mohammad Shamsul Arefin, Md Zahid Hasan, M Shamim Kaiser... <1%

175 Publication
P. Sajitha, Dr. A. Diana Andrushia, Dr. N. Anand, Dr. M.Z. Naser. "A Review on Mac... <1%

176 Publication
Paul, Nijhum. "Deep Learning Application in Plant Stress Detection.", North Dakot... <1%

177 Publication
Ramjibhai, Tadhani Jaydeep. "Cyber Attack Detection Scheme Using Deep Learnin... <1%

178 Publication
Ravindra R. Patil, Beatriz L. Rodriguez, Dag Røen, Malin Waage, Erik Kyrkjebø, Son... <1%

179 Publication

Rutuja Rajendra Patil, Sumit Kumar. "Rice Transformer: A Novel Integrated Mana... <1%

180 Publication

S. Naveena, K. Kavitha. "chapter 6 From Pixels to Precision", IGI Global, 2025 <1%

181 Publication

S. R. Reeja, Bore Gowda, Y. S. Rammohan, Ganesan Prabu Sankar, G. Jayalatha. "E... <1%

182 Publication

Sajid Ullah Khan, Anas Alsuhaibani, Abdulrahman Alabduljabbar, Fahdah Almarsh... <1%

183 Publication

Sarah M. Alhammad, Doaa Sami Khafaga, Walaa M. El-hady, Farid M. Samy, Khalid... <1%

184 Publication

Shailendra Tiwari, Anita Gehlot, Rajesh Singh, Bhekisipho Twala, Neeraj Priyadars... <1%

185 Publication

Shiva Mehta, Vinay Kukreja, Dibyahash Bordoloi. "Grape Leaf Disease Severity An... <1%

186 Publication

Shweta Mishra, Avneesh Kumar Singh, Pankaj Prajapati. "Challenges and Opport... <1%

187 Publication

Suman Lata Tripathi, Om Prakash Kumar, Allwin Devaraj Stalin, Tanweer Ali. "Inn... <1%

188 Publication

Vasileios Balafas, Emmanouil Karantoumanis, Malamati Louta, Nikolaos Ploskas. ... <1%

189 Publication

Wafae Hamdaoui, Abebaw Degu Workneh, Ahmed El Hilali Alaoui, Meryem Elmou... <1%

190 Internet

conf.kln.ac.lk <1%

191 Internet

csepup.ac.in <1%

192 Internet

downloads.hindawi.com <1%

193 Internet

eprints.qut.edu.au <1%

194 Internet

gehu.ac.in <1%

195 Internet

jastt.org <1%

196 Internet

pmc.ncbi.nlm.nih.gov <1%

197 Internet

repositorio.uneatlantico.es <1%

198 Internet

stax.strath.ac.uk <1%

199 Internet

www.galgotiasuniversity.edu.in <1%

200 Internet

www.nature.com <1%

Design and Development of AI-based Framework for Plant Disease Classification

Thesis submitted to

Delhi Technological University

for the award of the degree of

DOCTOR OF PHILOSOPHY

in

Information Technology

by

Meenakshi Srivastava

(2K21/PHDIT/03)

Under the Supervision of

Dr. Varsha Sisaudia

&

Dr. Jasraj Meena

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly known as Delhi College of Engineering)
NEW DELHI-110042, INDIA
(April, 2026)

Department of Information Technology
Delhi Technological University, Delhi
Delhi, INDIA - 110042

Candidate's Declaration

I declare that the research work presented in this thesis titled **Design and Development of AI-based Framework for Plant Disease Classification** for the attainment of the degree of ***Doctor of Philosophy in Information Technology*** has been conducted by me, under the guidance of ***Dr. Varsha Sisaudia and Dr. Jasraj Meena*** from the Department of Information Technology, Delhi Technological University, Delhi, India. This work has not been previously submitted, either partially or in its entirety, to any other academic institution for any degree or diploma.

I declare that this thesis reflects the ideas I express in my own words. Wherever the ideas or words of others have been incorporated, appropriate citation and referencing have been provided to acknowledge the sources. I also confirm that I have upheld the principles of academic honesty and integrity and that I have not misrepresented, fabricated, or falsified any information, data, facts, or sources in the content of my submission.

(Ms. Meenakshi Srivastava)
(Regn. No. 2K21/PHDIT/03)

This is to certify that the above statement made by the candidate is true to the best of our knowledge and belief.

Dr. Jasraj Meena	Dr. Varsha Sisaudia
(Joint-Supervisor)	(Supervisor)
Associate Professor	Assistant Professor
School of Computer and System Sciences	Department of Information Technology,
Jawaharlal Nehru University, Delhi, India	Delhi Technological University, Delhi, India

Place: Delhi, INDIA
Date: April 2026

Department of Information Technology
Delhi Technological University, Delhi
Delhi, INDIA - 110042

Certificate

This is to certify that the thesis titled **Design and Development of AI based Framework for Plant Disease Classification**, submitted by **Ms. Meenakshi Srivastava**, to the Delhi Technological University, Delhi, for the award of the degree of **Doctor of Philosophy**, is a bonafide record of the research work done by her under our supervision and guidance. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

We, as supervisor and joint supervisor, affirm the authenticity and originality of the research work conducted by Ms. Meenakshi Srivastava and presented in this thesis.

Dr. Jasraj Meena
(Joint-Supervisor)
Associate Professor
School of Computer and System Sciences
Jawaharlal Nehru University, Delhi, India

Dr. Varsha Sisaudia
(Supervisor)
Assistant Professor
Department of Information Technology,
Delhi Technological University, Delhi, India

Place: Delhi, INDIA
Date: April 2026

Dedicated to my family

ACKNOWLEDGMENT

The completion of this doctoral thesis has been a challenging yet incredibly rewarding journey. With profound gratitude and appreciation, I take this opportunity to acknowledge all those who have contributed, in various capacities, to the success of this work.

I would first like to express my deepest gratitude to my thesis supervisor, **Dr. Varsha Sisaudia**, and Joint-Supervisor **Dr. Jasraj Meena** whose continuous guidance, insightful feedback, and unyielding support have been invaluable to the development of this thesis. Their expertise and dedication to my academic progress have inspired me. I am grateful for the countless hours they have spent reviewing my work, providing constructive criticism, and pushing me to reach higher standards. Their encouragement and belief in my abilities, even during the most challenging times, have been crucial in completing this thesis.

I also want to extend my heartfelt thanks to our Head of Department **Prof. Dinesh Kumar Vishwakarma** for their time, thoughtful critiques, and unwavering support. Their constructive feedback has significantly improved the quality of this research, and I have learned much from their expertise.

On a personal note, I want to express my heartfelt appreciation to my family for their unwavering love, patience, and encouragement throughout this journey. To my parents **Mr. Girish Chandra Srivastava** and **Mrs. Meenu Srivastava**, and my sister **Dr. Priyanka Srivastava** , thank you for your constant emotional support and understanding and for always being there when I needed you most. Your sacrifices and belief in my potential have been a continuous source of motivation.

Lastly, I would like to thank my friends **Anita, Manu, Shivani, Lakshita, Akansha, Santosh sir, and Shristi** and all those whose names may not have been mentioned but who have contributed to the success of this thesis in one way or another. Whether through intellectual engagement, moral support, or acts of kindness, your contributions have been deeply appreciated.

April, 2026 Meenakshi Srivastava

ABSTRACT

Agriculture plays a pivotal role in economic growth and food security, with over 60% of India's population relying on it for livelihood. However, crop production faces ongoing obstacles, including pest invasions, environmental variability, and plant leaf diseases, which can severely impact output and quality. Manual disease diagnosis takes time and money and is frequently out of reach for farmers in remote areas. This requires the development of intelligent, automated illness-detection technologies that are both timely and accurate.

This thesis provides a complete framework that combines Artificial Intelligence (AI), Deep Learning (DL), and the Internet of Things (IoT) to identify and classify plant diseases. The work begins with a thorough literature review and a taxonomy of AI and IoT-based techniques to identify key research needs. It then explores transfer learning-based models and the Deep Transfer Learning and Extreme Learning Machine (VESTACK) framework, followed by attention-driven deep learning models such as Agriculture Transfer Learning Vision Transformer (AgriTL-ViT) and Asymmetric and Dilated Convolution ResNet (ADResNet) with Attention, which improve feature extraction and classification efficiency. Furthermore, a Federated Learning (FL) technique based on Convolutional Neural Networks (CNNs) and Transformer encoders is created to enable decentralized, privacy-preserving plant disease diagnostics in real-world agricultural environments.

The suggested methodologies show consistent performance across diverse plant leaf datasets, demonstrating the promise of AI-driven solutions for real-time, scalable, and smart agriculture. This thesis advances precision farming by providing a pathway to early disease identification, reduced crop losses, and enhanced agricultural sustainability. Future directions include expanding the framework for multi-disease identification, integrating cross-device FL, and deploying real-world IoT systems to improve smart agriculture operations.

Contents

Certificate iii

Declaration v

Dedication vii

Acknowledgments ix

Abstract x

List of Tables xv

List of Figures xix

List of Algorithms xxii

List of Acronyms/Abbreviations xxiii

1 Introduction 1

1.1 Overview . . . 1

1.2 What is Plant Leaf Illness Identification? . . . 2

1.2.1 Signs of Bacterial illnesses . . . 2

1.2.2 Signs of Viral Illnesses . . . 3

1.2.3 Fungal Illnesses Symptoms . . . 3

1.2.4 Nutrient Deficiency vs. Plant illnesses . . . 3

1.3 Traditional Approaches to Identifying Plant Leaf Illnesses . . . 5

1.4 Role of AI in Plant Leaf Disease Identification . . . 6

1.5 Motivation of Thesis . . . 7

1.6 Challenges in Plant Leaf Illness . . . 8

1.7 Outlines of the Thesis . . . 10

2 Literature review 13

2.1 Literature Survey on Traditional Approach for Plant Leaf Disease Classification 14
2.1.1 Literature Survey on Deep Learning Approach for Plant Disease Classification . . . 15
2.2 Analysis of Existing Literature Survey . . . 18
2.3 Taxonomy of IoT and AI based Plant Disease Identification . . . 19
2.3.1 Based on Methodology . . . 21
2.3.2 Based on Application . . . 26
2.3.3 Based on IoT Technologies . . . 31
2.4 Analysis of Public Available Datasets . . . 39
2.5 Study and Analysis of Machine and Deep Learning based Existing Techniques for Plant Disease Classification . . . 46
2.5.1 Overview . . . 46
2.5.2 Experimental Setup . . . 47
2.5.3 Experimental Analysis . . . 47
2.5.4 Result Analysis . . . 49
2.6 Research Gap . . . 50
2.7 Research Objectives . . . 50
2.8 Research Contributions . . . 51

3 Transfer Learning Approaches for Plant Leaf Disease Classification 55

3.1 Plant Leaf Disease Detection and Classification using Modified Transfer Learning Models . . . 56
3.1.1 Overview . . . 56
3.1.2 Related Work . . . 56
3.1.3 Preliminary . . . 58
3.1.4 Modified Framework . . . 62
3.1.5 Experimental Setup . . . 64
3.1.6 Experimental Result . . . 64
3.2 A Hybrid VESTACK Model for Plant Disease Classification Using Deep Transfer Learning Approach . . . 80
3.2.1 Overview . . . 80
3.2.2 Related Work . . . 83
3.2.3 Proposed Hybrid Framework . . . 86
3.2.4 Experimental Setup and Result . . . 94
3.2.5 Discussion . . . 109

3.3 Tomato Leaf Disease Detection and Classification using MobileNetV2 and Extreme Learning Method: A Hybrid Approach . . . 113
3.3.1 Overview . . . 113
3.3.2 RELATED WORK . . . 115
3.3.3 Proposed Methodology . . . 116
3.3.4 Experimental Result . . . 118
3.4 Summary . . . 122

4 Attention-Driven Deep Learning Models for Plant Leaf Disease Classification 125

4.1 AgriTL-ViT: A Vision Transformer Model with Attention Techniques for Classification of Plant Leaf Disease . . . 126
4.1.1 Overview . . . 126
4.1.2 Related Work . . . 128
4.1.3 Model Development . . . 132
4.1.4 Experimental Setup and Result . . . 144
4.1.5 Experimental Setup for Dataset . . . 145
4.1.6 Dataset Description . . . 145
4.1.7 Data Pre-processing . . . 145
4.1.8 Hyperparameter . . . 146
4.1.9 Performance Evaluation Metrics . . . 146
4.1.10 Training and Testing Results for Tomato, Maize, and Rice Leaf Dataset 146
4.1.11 Visualisation Results . . . 148
4.1.12 Evaluation Results for Tomato, Maize, and Rice Dataset . . . 150
4.1.13 Generalizability of the proposed model . . . 156
4.1.14 Computational Cost Analysis . . . 159
4.1.15 Ablation Study . . . 160
4.1.16 Comparison with the State-of-the-Art Techniques . . . 163
4.1.17 Discussion on the Proposed AgriTL-ViT Model . . . 166
4.2 ADResNet with Attention: A Multi-Model Approach for Leaf Disease Detection and Pest Classification . . . 170
4.2.1 Overview . . . 170
4.2.2 Related Work . . . 172
4.2.3 ADResNet with Attention: A Lightweight Model . . . 172
4.2.4 Experimental Setup . . . 176

4.2.5 Result Analysis 177
4.2.6 Summary 182

5 Federated Learning based Model for Plant Leaf Disease Classification 183

5.1 Towards Distributed Method for Plant Disease Diagnosis: A Hybrid CNN-Transformer Model with Federated Learning 183
5.1.1 Overview 183
5.1.2 Related Work 185
5.1.3 Proposed Federated Learning with Transformer-augmented CNN (FL-TransCNN) Model 187
5.1.4 Experimental Setup and Results 193
5.1.5 Dataset Description 193
5.1.6 Hyperparameter Table 194
5.1.7 Performance Evaluation Criteria 195
5.1.8 Result Analysis 195
5.1.9 Summary 206

6 Conclusion and Future Scope 207

6.1 Conclusion 207
6.2 Future Scope 208

List of Publications 210

Bibliography 212

List of Tables

2.1 Comprehensive Analysis of AI-based Plant Disease Detection Studies 16
2.2 Analysis of AI and IoT-based Plant Disease Detection Studies 21
2.3 Analysis of ML with IoT in Smart Agriculture (Part 1) 25
2.4 Analysis of Plant Illness Identification using Deep Learning with IoT 27
2.5 Study of Plant Disease Detection Based on Application 29
2.6 (b) Analysis of Plant Disease Detection Based on the Application 32
2.7 (a) Analysis of Plant Disease Detection based on Soil Sensor 34
2.8 (b) Analysis of Plant Disease Identification based on Drone 37
2.9 (c) Analysis of Plant Disease Identification based on Agriculture Robots . . . 39
2.10 Total Number of Images in Each Class 41
2.11 Total Number of Leaf in Each Class 41
2.12 Analysis of Publicly Available Datasets 46
2.13 Analysis Of The Model On Different Parameters 48
2.14 Analysis Of The Model on the Same Parameters for Grape Dataset 48
2.15 Comparative Analysis of Models on different Datasets 49
2.16 Experimental Analysis of Existing Methods for Plant Disease Classification . 50

3.1 Summary of the Latest Literature work 59
3.2 Hyper-Parameter Settings 59
3.3 Training and Testing Outcomes of Leaf Disease Identification Using Mendeley Dataset 66
3.4 Training and Testing Results of Cherry Dataset 68
3.5 Class-wise Performance Evaluation of VGG16 Model Across Twenty-Two Leaf Categories 71
3.6 Class-wise Performance Evaluation of MobileNetV2 Model Across Twenty-Two Leaf Categories 72
3.7 Class-wise Performance Evaluation of DenseNet121 Model Across Twenty-Two Leaf Categories 73

3.8 Class-wise Performance Evaluation of Xception Model Across Twenty-Two Leaf Categories 74
3.9 Class-wise Performance Evaluation of Inception-V3 75
3.10 Weighted Performance Metrics on the Mendeley Dataset 77
3.11 Confusion Matrix–Based Performance Metrics on the Cherry Dataset for Five CNN Models 78
3.12 Results of Weighted Quality Metrics for the Cherry Dataset 80
3.13 Accuracy of Proposed Transfer Learning Models on Unseen Datasets 80
3.14 Comparison of the Modified Transfer Learning Models with State-of-the-Art Techniques on the Mendeley Dataset 81
3.15 Comparison of the Modified Transfer Learning Models with State-of-the-Art Techniques on the Cherry Dataset 81
3.16 Analysis of the Literature Survey on Different Parameters 86
3.17 Hyperparameter Settings Used for the Model 94
3.18 Quantitative Results of Plant Disease Detection using Four Datasets 96
3.19 VESTACK image quality for the Peach, Strawberry, Grapes, and Apple datasets 101
3.20 Classification Results for Detecting Peach Leaf Disease 102
3.21 Ablation Study Score for the VESTACK model for the Peach and Strawberry Datasets 102
3.22 Ablation Study Score for the VESTACK model for Grapes and Apple Datasets 104
3.23 Execution Time (m/sec) of VESTACK (our proposed method) Across Different Datasets 105
3.24 Comparison of the Proposed Model with the State-of-the-Art Techniques on the Peach and Strawberry Dataset 107
3.25 Comparison of the Proposed Model with the State of the Art Techniques on the Grapes Dataset 108
3.26 Comparison of the Proposed Model with the Cutting edge Techniques on the Apple Dataset 108
3.27 Impact of Different Data Augmentation Techniques on Classification Accuracy Across Leaf Datasets 111
3.28 Computational Complexity Comparison of VGG16, ELM, and VESTACK . 112
3.29 Impact of Data Augmentation on Model Performance Across Datasets 113

3.30 Training and Testing Outcomes of Leaf Disease Detection Using Tomato Dataset 120
3.31 Comparison of The Proposed Model With The Existing Techniques On The Tomato Leaf Dataset 121

4.1 An Overview of Literature Work on Plant Disease Detection 131
4.2 Hyperparameter Values for Executing the AgriTL-ViT model 146
4.3 Training and Testing result of Tomato, Maize, and Rice Leaf Dataset 147
4.4 AgriTL-ViT Model Image Quality for Ten Classes of Tomato Dataset 151
4.5 AgriTL-ViT Model Image Quality for Four Classes of Maize Dataset 152
4.6 AgriTL-ViT Model Image Quality for Nine Classes of Rice Dataset 152
4.7 Classification Results for Detecting Peach Leaf Disease. 154
4.8 Accuracy of the Proposed AgriTL-ViT Model on Unseen Datasets 157
4.9 Unseen Crop Testing Accuracy Across Geographic and Climate Zones 159
4.10 Training Time Comparison of Different Models on Leaf Datasets 159
4.11 Score of Ablation Study for the AgriTL-ViT Model for Tomato Dataset . . . 160
4.12 Ablation Study Score for the AgriTL-ViT Model for Maize Dataset 161
4.13 Ablation Study Score for the AgriTL-ViT Model for Rice Dataset 162
4.14 Comparison of AgriTL-ViT Technique with State-of-The-Art Techniques on Tomato leaf Dataset 164
4.15 Comparison of AgriTL-ViT Model with Existing Techniques on Maize Leaf Dataset 165
4.16 Comparison of AgriTL-ViT Model with Existing Techniques on Rice Leaf Dataset 166
4.17 Accuracy of the Proposed Model in Real-World Conditions with Various Augmentation Techniques 168
4.18 Accuracy of AgriTL-ViT on Low-Resolution (128×128) and (200×200) Images Across Different Plant Disease Datasets 169
4.19 Sensitivity Analysis on the Tomato Dataset Showing the Impact of Attention Heads and Embedding Dimensions on Model Accuracy. 170
4.20 Comparison of Different Methods for Plant Disease Classification 172
4.21 ADResNet with Attention Model Architecture Details 176
4.22 Hyperparameter Settings for the Model 177
4.23 Quantitative Results of Leaf Disease Identification Using CCMT Dataset . . 178

4.24 Image Quality Metrics of ADResNet with Attention Model for 23 Classes of CCMT Dataset 179
4.25 Ablation Study Score for ADResNet with Attention Model for CCMT Datasets 180
4.26 Comparison of Models with Constant and Decay Learning Rate 181

5.1 Analysis of the Literature Survey on Different Parameters 187
5.2 Hyperparameter Table 194
5.3 Client-wise Classification Report on the Mango Dataset 196
5.4 Client-wise Classification Report on the Rice Dataset 197
5.5 Macro, Weighted, and Micro Averages per Client for Mango Leaf Dataset . . 198
5.6 Macro / Weighted / Micro Averages per Client for Rice Leaf Dataset 198
5.7 Global Aggregated Model Metrics for Rice and Mago Leaf Dataset 198
5.8 Resource Utilization on Mango and Rice Leaf Datasets 199
5.9 Score of Ablation Study for FL-TransCNN Model for Mango Leaf Dataset . 202
5.10 Score of Ablation Study for FL-TransCNN Model for Rice Leaf Dataset . . . 203
5.11 Comparison of the FL-TransCNN Model with State-of-The-Art Techniques on the Mango Leaf Dataset 204
5.12 Comparison of the FL-TransCNN Model with State-of-The-Art Techniques on the Rice Leaf Dataset 205

List of Figures

1.1 Sample image of bacterial leaf 2
1.2 Sample image of viral leaf disease 3
1.3 Sample images of fungal leaf 4
1.4 AI-based framework for the diagnosis of plant leaf illness 7
1.5 Structure of the thesis 10

2.1 Proposed taxonomy 20
2.2 Sample image of plantvillage dataset 40
2.3 Samples of healthy and unhealthy leaf datasets 42
2.4 Sample images of the Peach, Strawberry, Grapes, and Apple leaf dataset . . 42
2.5 Sample images of Tomato Dataset 43
2.6 Maize or Corn Leaf Dataset 43
2.7 Collection of Rice Leaf 44
2.8 Sample images of the CCMT Dataset 44
2.9 Sample image of Mango Leaf 45
2.10 Sample image of Rice Leaf 45

3.1 VGG 16 Architecture 60
3.2 MobileNetV2 Architecture 61
3.3 Xception Architecture 61
3.4 InceptionV3 Architecture 62
3.5 DenseNet121 Architecture 63
3.6 Framework of plant leaf disease detection & classification 63
3.7 Saliency map of Mendeley Dataset 67
3.8 VGG16 training and validation result analysis 68
3.9 MobilenetV2 training and validation result analysis 69
3.10 DenseNet121 training and validation result analysis 69
3.11 Xception training and validation result analysis 70
3.12 InceptionV3 training and validation result analysis 70

3.13 Saliency map of Cherry Dataset 70
3.14 Confusion matrix chart with mendeley dataset for five cnn model 77
3.15 Confusion matrix chart with cherry dataset for five cnn model 79
3.16 Process of disease identification 81
3.17 Proposed hybrid architecture 87
3.18 VESTACK model training and testing result analysis for peach dataset . . . 96
3.19 VESTACK model training and testing result analysis for strawberry dataset 97
3.20 VESTACK model training and testing result analysis for grapes dataset . . . 97
3.21 VESTACK model training and testing result analysis for apple dataset . . . 98
3.22 Visualisation of lIME analysis on (a) peach (b) strawberry (c) grapes and (d) apple Leaf dataset 99
3.23 Visualisation of sHAPE analysis on (a) peach (b) strawberry (c) grapes and (d) apple Leaf dataset 100
3.24 ROC curve for VESTACK: based on the computed TPR&FPR 103
3.25 Baseline of VGG16 104
3.26 Proposed VGG16 104
3.27 Framework of ELM 105
3.28 VESTACK (our proposed model) 106
3.29 Graphical representation of the proposed model with the existing state of the art techniques on the different parameters 107
3.30 A graphical representation of the suggested model, along with the current baseline techniques on the various parameters 109
3.31 Graphical representation of the VESTACK model with the existing baseline techniques on the different parameters 110
3.32 Accuracy with Error Bars on Peach and Strawberry Leaf Dataset 110
3.33 Accuracy with Error Bars on Grapes and Apple Leaf Dataset 111
3.34 Original leaf image and its augmented versions using Color Jitter, Gaussian Blur, and Random Erasing. 112
3.35 Proposed hybrid framework 117
3.36 Training and testing accuracy of hybrid model 120
3.37 Training and testing loss of hybrid model 121
3.38 Graphical representation of the proposed model with existing state-of-the-art techniques on the different parameters 122

4.1 Structural representation of the modified pre-existing ViT Transformer . . . 132

4.2 AgriTL-ViT model architecture 134
4.3 Proposed ResNet architecture 137
4.4 Training and testing outcome for tomato leaf dataset 147
4.5 Training and testing outcome analysis for maize leaf dataset 147
4.6 Training and testing outcome analysis for rice leaf dataset 148
4.7 Visualisation of grad-cam analysis on (a) tomato (b) maize and (c) rice leaf dataset 149
4.8 Visualisation of lime analysis on (a) tomato (b) maize and (c) rice leaf dataset 150
4.9 Visualisation of shap analysis on (a) tomato (b) maize and (c) rice leaf dataset 151
4.10 Confusion matrices for tomato, maize, and rice leaf datasets using the AgriTL-ViT model 153
4.11 ROC curve for (a) tomato, (b) maize and (c) rice Dataset 155
4.12 Bar graph representation of the AgriTL-ViT model based on FPR, FNR, FOR and FDR for the (a) tomato, (b) maize and (c) rice Datasets 156
4.13 Confusion matrix chart with grapes and apple leaf dataset for model AgriTL-ViT 158
4.14 Results of the Ablation Study for Tomato Leaf Dataset 161
4.15 Results of the Ablation Study for Maize Leaf Dataset 162
4.16 Results of the ablation study for rice leaf dataset 163
4.17 Graphical presentation of the AgriTL-ViT model with state-of-the-art techniques 164
4.18 Graphical presentation of the AgriTL-ViT model with existing base line techniques1 165
4.19 Graphical presentation of the AgriTL-ViT model with existing base line techniques 166
4.20 Original leaf image and its augmented versions using color jitter, gaussian blur, and random erasing 169
4.21 Proposed hybrid model architecture 173

5.1 Overview of the proposed model 189
5.2 Architecture component of proposed model 189
5.3 Part of the FL work 192
5.4 Visualization of SHAP analysis on (a) mango and (b) rice leaf dataset 200
5.5 Visualization of LIME analysis on (a) mango and (b) rice leaf dataset 201

List of Algorithms

3.1 Proposed VESTACK Model 92
3.2 VESTACK Model Testing 93
3.3 VESTACK Model Training and Image Identification 93
3.4 Training and Testing Loop for the Proposed TLMV2-ELM Model 119
4.1 AgriTL-ViT Model 142
4.2 Model Training 143
4.3 Model Testing and Identification 143

List of Acronyms/Abbreviations

AI	Artificial Intelligence
CNNs	Convolutional Neural Networks
ViTs	Vision Transformers
SVMs	Support Vector Machines
DL	Deep Learning
ELM	Extreme Learning Machine
SLFNs	single-layer feed-forward network
FL	Federated Learning
AgriTL-ViT	Agriculture Transfer Learning Vision transformer
IoT	Internet of Things
ML	Machine Learning
LIME	Local Interpretable Model-agnostic Explanations
ADCNN	Asymmetric Dilated Convolution
IoAT	Internet of Agriculture Things
FedAvg	Federated Averaging
MHSA	Multi-Head Self-Attention
TL	Transfer Learning
UAV	Unmanned Aerial Vehicle
AdamW	Adaptive Moment Estimation with Weight Decay

Chapter 1

Introduction

This chapter * *aims to furnish background information regarding the function of AI in the identification of plant leaf diseases and to present a comprehensive introduction to the thesis. The chapter delineates the motivation, aims, and organization of the thesis.*

1.1 Overview

Agriculture is an important sector that is instrumental in the economic growth of any nation. About 60 % of the Indians work in the farming sector, and it generates approximately 17 % of the gross domestic product. Farmers are given various challenges every year, including problems associated with pests, environmental problems, and leaf disease problems, which affect crop production. The supply of food production in the bounded field resources is very hard to satisfy the demand [1]. Consequently, there is an imperative necessity to augment farming output; hence, precision farming and technology-driven farming are essential in the modern era.

Crop illness identification is a crucial field in the agricultural sector that requires attention. Traditionally, scientists used visual examination to determine plant diseases only. This approach requires specialized personnel to keep a constant check on crops, which is not economical in a large farm. Additionally, in most of the countries, the farmers do not have access to the right facilities or professional advice [2]. To overcome these problems, automatic methods of plant diseases detection have been suggested. Plant diseases are detected earlier; hence farmers react in due time, the process becomes more quicker, cost-effective, and dependable. These robots assist in the correct identification of the disease based on machine learning, deep learning, and Internet of Things (IoT) technologies. Plant

*This chapter has been published in: Meenakshi Srivastava, Varsha Sisaudia, and Jasraj Meena, "A Review on Internet of Things and Artificial Intelligence based Plant Disease Detection: Taxonomy, Issues and Challenges," *Journal of Plant Diseases and Protection*, Springer, 132(6), 181. 2025., DOI: https://doi.org/10.1007/s41348-025-01184-0. **(SCIE, IF=2.2)**.

diseases are not easily diagnosed using non-scientific and traditional means, which are mostly imprecise and narrow. Conversely, automated methods of detection make the process to be less time consuming and less effort consuming with a high degree of accuracy. Advanced smart agriculture technologies, including as artificial intelligence, deep learning, and the Internet of Things, can be helpful in recognizing diseases in real time and removing limits associated with data identification and manual monitoring.

1.2 What is Plant Leaf Illness Identification?

Plant leaf illness identification is an aspect of smart farming. It can be defined as the detection of illness in plant leaves, which is normally done by visual inspection or imaging methods. Leaf disease identification is designed to identify plant diseases with high precision and promptly, so as to curb the transmission of the illness and limit the consequences of the disease on the health and productivity. of crops. This may be incorporated by visual inspection of trained personnel, but it can also be done by utilizing various types of technologies like machine learning algorithms, computer vision systems, or handheld devices. These methods are able to scan the leaves images to identify characteristic patterns and signs of various illnesses, including spots [3]. The symptoms are also common in some plants, and they are represented by fungal, bacterial and viral leaf diseases of plants.

1.2.1 Signs of Bacterial illnesses

The illness is known through microscopic pale green spots which look moist with water [4]. The lesions occur and later become dry, dead regions as illustrated in Figure 1.1. The majority of bacterial illnesses can be found on tomato leaves, potato leaves, pepper leaves, apple leaves, and so on.



Figure 1.1: Sample image of bacterial leaf

1.2.2 Signs of Viral Illnesses

37

Plants can show a wide range of symptoms when they are infected with a virus, depending on the type of virus and the type of plant that is infected. Viruses lack discernible warning signs and are often mistaken for nutritional deficiencies and chemical damage. Common vectors of this illness include aphids, leafhoppers, whiteflies, and cucumber beetles [5]. Examine the leaf for yellow or green stripes or spots, as depicted in the Figure 1.2. Leaves may exhibit wrinkling, and development might be inhibited. Viral infections predominantly affect the leaves of tomatoes, cucumbers, cauliflowers, and plums.



Figure 1.2: Sample image of viral leaf disease

1.2.3 Fungal Illnesses Symptoms

The following section delves into plant leaf diseases caused by fungi, as depicted in Figure 1.3. For instance, late blight is caused by fungi. Initially, it manifests as damp, grey-green spots on the lower, more mature leaves. The patches become darker while the fungal problem deteriorates, and white fungal development appears on the bottoms [6]. Fungal infections predominantly affect crops such as basil, maize, rice, and corn.

Although conventional approaches have established a foundation for examining plant leaf diseases, the growing complexity and breadth of agricultural issues demand innovative solutions. This highlights the significance of artificial intelligence in enhancing the efficacy and precision of illness identification.

1.2.4 Nutrient Deficiency vs. Plant illnesses

Nutrient deficits and plant illnesses can produce visually like symptoms on leaves, stems, and fruits, complicating accurate identification in agricultural practices [7]. Nutrient deficiencies



Figure 1.3: Sample images of fungal leaf

arise from inadequate availability or suboptimal absorption of essential macro- and micronutrients, including iron, phosphorus, potassium, nitrogen, and magnesium. Conversely, plant diseases are induced by pathogenic organisms, including fungi, bacteria, viruses, and nematodes. Although originating from different causes, many diseases may exhibit similar visual characteristics, leading to frequent misdiagnosis in field assessments.

- **Visual Similarity of Symptoms:** The visual signs of nutrient deficiency and plant diseases exhibit notable parallels, especially in leaf shape. Nitrogen deficiency frequently leads to the uniform yellowing (chlorosis) of older leaves, resembling the initial phases of viral infections [7]. Interveinal chlorosis, resembling fungal leaf spot infections, results from deficiencies in iron and magnesium. A deficiency in potassium may lead to leaf margin necrosis, often mistaken for bacterial blight or burn-related diseases. Additionally, nutritional stress and disease infections can lead to leaf bending, delayed development, and early leaf drop, making visual differentiation more challenging [8].

  Environmental factors, including sunlight, soil moisture, and the growth stage of the plant, significantly impact the expression of symptoms, intensifying the visual resemblance between nutrient deficiencies and disease indicators. Consequently, visual examination alone may not be adequate to differentiate between the two conditions.

- **Misclassification Issues:** The visual similarities between nutrient deficit symptoms and plant diseases lead to major inaccuracies in both manual and automated diagnosis procedures [8]. Misclassification frequently leads to ineffective corrective measures in

traditional farming practices, such as excessive pesticide treatment when the real problem is nutrient imbalance, or wasteful fertilizer use while the plant is sick with a virus. These wrong actions not only increase production costs, but also have a severe impact on soil health and environmental sustainability.

From the perspective of automated plant disease classification, misclassification poses a significant challenge for image-based and artificial intelligence-driven models. Machine learning and deep learning algorithms trained simply on visual cues may struggle to discern between nutrition-induced stress and disease-related symptoms, especially if datasets do not include a substantial number of nutrient-deficient instances. This constraint can result in lower model accuracy, poor generalization, and incorrect predictions in real-world scenarios.

As a result, resolving nutrient deficiencies and illness misclassification is an important research topic in plant disease classification systems.

1.3 Traditional Approaches to Identifying Plant Leaf Illnesses

Traditional methods to identifying plant illnesses depend significantly on manual research, specialised expertise, and laboratory diagnostic instruments. These methodologies have been employed in agriculture for many years and form the basis for conventional plant health monitoring systems [9]. Traditional approaches, although somewhat effective, often require substantial time and labour and struggle to scale, particularly in extensive agricultural contexts.

The most common traditional approach involves visual inspection, where farmers or agricultural experts recognize diseases by observing visible signs such as leaf discoloration, spots, wilting, lesions, and abnormal growth patterns [10]. This approach necessitates considerable expertise and specialized knowledge, given that numerous plant diseases exhibit comparable visual traits. The precision of visual assessment is significantly affected by the observer's expertise and the surrounding conditions, rendering it subjective and susceptible to human error.

A frequently employed traditional approach is the lab-based test, which involves collecting samples of plants and analysing them using microscopic analysis, growing procedures, or biochemical assays. Pathogens like fungi and bacteria are recognized by analyzing their

morphological structures, while viral infections are often detected using serological or molecular methods [11]. While laboratory methods provide greater accuracy compared to visual inspection, they require specialized equipment, skilled personnel, and longer processing durations. Therefore, these technologies are inadequate for rapid or urgent sickness diagnosis in field environments.

In addition, field trials and manual monitoring are employed to observe the advancement of sickness over time. This process entails the systematic collection of plant samples and the execution of prolonged observation inside controlled or semi-controlled conditions. This technique facilitates the examination of illness activity; however, it is highly expensive and challenging for extensive or widely dispersed farming regions.

Traditional methods of illness identification face challenges due to delayed diagnosis, as symptoms often manifest only after the disease has advanced significantly [12]. At the time identification takes place,irrevocable harm to crop yield and quality may have actually transpired. Moreover, insufficient access to agricultural specialists in rural and remote areas significantly constrains the effectiveness of these initiatives.

In summary, although conventional methods for identifying plant diseases have played a vital role in managing agricultural diseases, their shortcomings regarding accuracy, speed, cost, and scalability highlight the necessity for advanced automated and intelligent systems for disease identification. The limitations identified have led to the adoption of algorithms based on image processing, machine learning, and deep learning to achieve efficient, accurate, and real-time identification of illnesses in plants.

1.4 Role of AI in Plant Leaf Disease Identification

Artificial intelligence (AI) revolutionizes contemporary agriculture by facilitating accurate, rapid, and scalable detection of plant diseases. Conventional disease identification methods, often dependent on expert manual inspection, are laborious, subjective, and inadequate for extensive agricultural operations. AI-driven models, especially those utilizing machine learning and deep learning, can autonomously learn and classify images of diseased leaves with remarkable precision. Support Vector Machines (SVMs), Vision Transformers (ViTs) and Convolutional Neural Networks (CNNs) and have demonstrated exceptional efficacy in identifying moderate illness symptoms inside intricate environments. Farmers may enhance decision-making, minimize crop damage, and boost output by integrating AI into plant

illness classification systems.

The integration of AI with real-time data sources, facilitated by technologies such as sensors and IoT devices, enhances sickness detection capabilities. In agriculture, devices such as cameras, soil moisture sensors, and environmental monitors collect real-time data on plant health, including high-resolution images of leaves. The collected real-time plant leaf data is transmitted to cloud platforms, where AI-driven algorithms detect indications of disease.

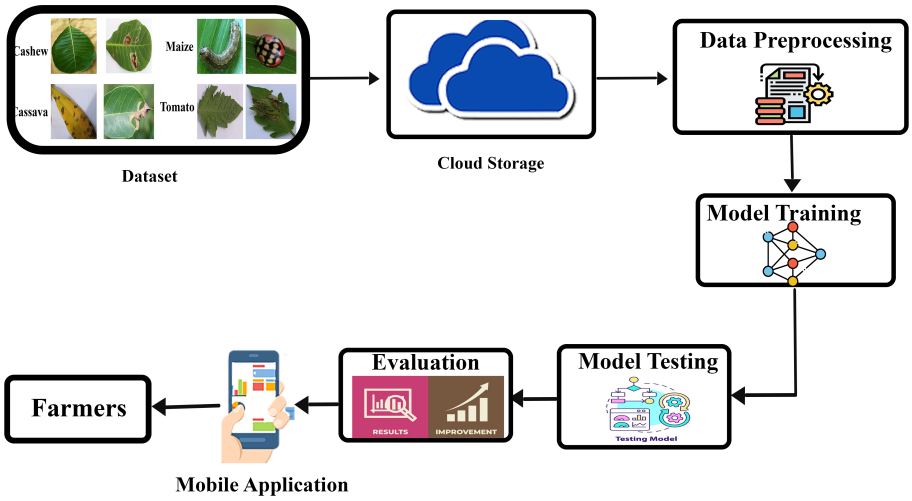


Figure 1.4: AI-based framework for the diagnosis of plant leaf illness

By using AI's capacity to process and analyze vast quantities of varied data, these frameworks provide precise, automated monitoring and prompt intervention, therefore reducing the necessity for manual inspections (Figure 1.4). Advanced systems employing drones and IoT-enhanced cameras may survey vast agricultural fields and identify sick regions with no human involvement. The outcome is a more proactive, data-driven strategy for plant disease management that enhances decision-making, optimizes resources, and ultimately boosts crop output while minimizing losses.

1.5 Motivation of Thesis

Due to heightened agricultural losses from plant diseases, there is an escalating demand for intelligent, scalable, and automated solutions. This study is motivated by the capacity of artificial intelligence to enhance the precision, efficiency, and real-time surveillance of plant illness diagnosis. The key motivations driving this research are summarized as follows:

- **Agricultural Losses :** Every year, a considerable percentage of agricultural yield is lost due to undiscovered or late detected plant illnesses, which have a direct impact

on food security and farmer incomes [1].

- **Growing Food Demand and Agricultural Challenges:** According to the United Nations Food and Agriculture Organization (FAO), the global population is projected to reach 9.1 billion by 2050 [2]. As the need for food escalates, natural resources like arable land and freshwater become ever limited. The proliferation of plant diseases induced by bacteria, fungi, and viruses poses a significant threat to global food safety, as they can severely diminish crop yield and farmer revenue.

- **Limitations of Manual Methods :** Manual inspection of diseases is labour-intensive, time-consuming, error-prone, and unsuitable for large agricultural fields [13].Manual plant illness identification over expert visual inspection is inefficient, imprecise, and time-consuming, mainly in resource-constrained farming surroundings [14].

- **Need for Early and Accurate Identification :** Early detection of plant diseases allows for earlier treatment, prevents crop loss, and provides higher quality and quantity of harvest [15].

- **Supporting Sustainable Agriculture:** AI-based leaf disease monitoring promotes sustainable agriculture by enabling precise disease identification, minimizing unnecessary pesticide use, and conserving essential agricultural resources [16].

- **Empowering Farmers with Technology:** AI-integrated mobile or cloud-based systems can support farmers in making accurate plant disease decisions, enabling more data-driven and informed crop management [17].

- **Lack of Services and High Cost :** AI-based illness classification systems have limited adoption due to a lack of accessible technical assistance and the high cost of required equipment and computational resources [18]. Small, resource-limited farming communities often encounter difficulties in financing these solutions, and the absence of consistent support services further constrains their efficacy and sustainability.

1.6 Challenges in Plant Leaf Illness

- **High Intra-Class and Low Inter-Class Variability:**

  Plant leaf illnesses can vary significantly within the same group due to differences in crop species, leaf maturity, severity of the illness, and environmental factors. The visual differences among various diseases may be minimal [19]. The combination of

substantial intra-class and minimal inter-class heterogeneity renders effective sickness discrimination exceedingly challenging.

- **Early Stage Symptom detection:** The initial symptoms of disease during the early stages of infection are subtle, often appearing as slight color variations or small spots on the leaf [20]. The initial symptoms present challenges in distinguishing them from healthy tissue or natural leaf variations, leading to potential delays or inaccuracies in classification.

- **Impact of Image Acquisition Conditions:** Changes in lighting, shadow patterns, ambient noise, camera clarity, and perspectives all play a crucial role in determining the quality of leaf images [20]. Field-acquired photographs can exhibit complex backdrops and variable lighting, thereby reducing classification accuracy relative to controlled laboratory datasets.

- **Limited and Poorly Annotated Datasets:** Numerous freely accessible plant leaf illness datasets are limited in size and exhibit a deficiency in diversity. In numerous instances, annotations are inadequate or inconsistent, and particular disease categories are underrepresented [21]. The deficiency of adequately labelled data constrains the training and generalisation capacities of deep learning models.

- **Multi-illnesses Viruses on a Specific Leaf:** In practical farming scenarios, a specific leaf may be impacted by many illnesses concurrently. Ambiguous classification arises from overlapping symptoms; however, the majority of current algorithms are tailored for single-label diagnosis [22].

- **Similarity of Illness Symptoms and Abiotic Stress:** Nutrient deficiencies, pesticide injury, and environmental stressors can produce leaf symptoms that mimic those of fungal, bacterial, or viral infections [22]. Differentiating between biotic illness and abiotic stressors is a considerable problem for the precise identification of foliar illness.

- **Model generalization and practical implementation:** Models developed using benchmark datasets may exhibit poor generalization across different crops, regions, or environmental conditions [22]. Moreover, implementing leaf illness identification techniques on data-constrained devices, like cellular devices or IoT nodes, requires a balance between precision and computational performance.

1.7 Outlines of the Thesis

The thesis is split into six chapters, as demonstrated in Figure 1.5.

Chapter 1: Introduction This chapter presents the concept of plant leaf diseases and examines its effects on agriculture. We analyze the significance of AI in addressing these challenges, subsequently outlining the rationale for our research. Additionally, we examine the principal challenges faced in the agriculture sector and conclude the chapter with a brief overview of the thesis framework.

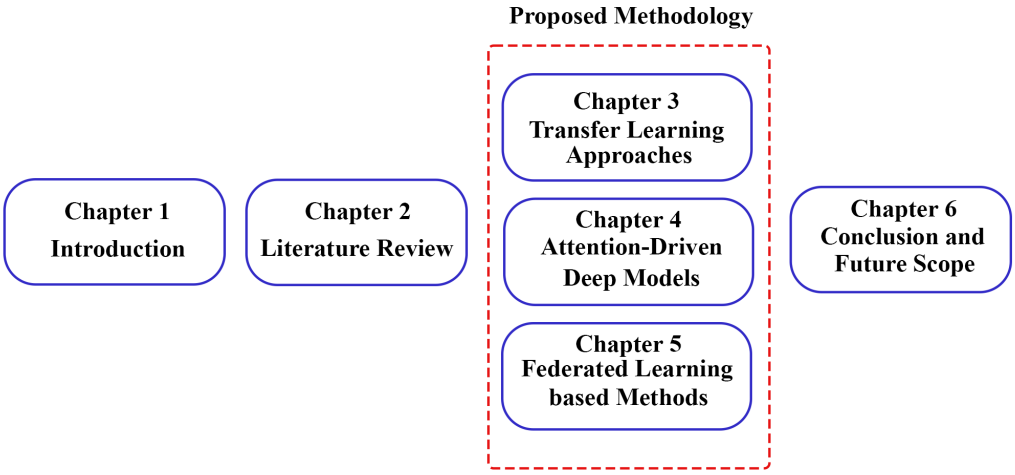


Figure 1.5: Structure of the thesis

Chapter 2: Literature Survey

This chapter presents a comprehensive overview of the thesis. We commence with a summary of the identification of plant leaf diseases and examine the purpose of artificial intelligence in addressing this challenge. A survey of the current research on AI and IoT methodologies follows. Additionally, we offer a taxonomy for classifying advanced plant illness methods for identification. A comprehensive analysis of publicly available datasets is conducted to enhance comprehension of their characteristics and constraints. We identify the principal research gaps according to the survey and dataset evaluation. Subsequently, we shall delineate the principal goals and outcomes of the research. The chapter concludes by encapsulating the principal issues discussed in this thesis.

Chapter 3: Transfer Learning Approaches for Plant Leaf Disease Classification

This chapter analyzes two separate deep learning techniques for the categorization of plant diseases. The initial proposal presents a modified transfer learning model featuring architec-

tural alterations, subsequently assessed on tomato and peach datasets utilizing various pre-trained networks. The second presents the VESTACK framework, which integrates transfer learning with Extreme Learning Machine (ELM) and is evaluated on multiple fruit datasets. The two approaches are delineated individually, highlighting their effectiveness and accuracy of classification.

Chapter 4: Attention-Driven Deep Models for Plant Disease and Pest Classification

This chapter outlines two efficient and effective deep learning methodologies for the classification of plant diseases. The AgriTL-ViT model improves Vision Transformer efficacy by substituting MLP modules with ResNet-based Transformer blocks and integrating a dual multi-head self-attention system. It attains elevated precision on the datasets of tomato, maize, and rice leaves. The second method, ADResNet with Attention, utilizes Asymmetric and Dilated Convolution (ADCNN), InceptionResNetV1, and self-attention to diminish computational complexity while preserving precision. Upon assessment using the CCMT dataset, it outperforms traditional multi-class classification methods. Both models yield encouraging outcomes for real-time, scalable plant illness diagnosis.

Chapter 5: FL based Methods for Plant Leaf Disease Classification

This chapter introduces a streamlined Federated Learning (FL) framework for categorising plant leaf illness in distributed and privacy-sensitive farming environments. The approach integrates CNNs for spatial retrieval of features with transformer encoders for acquisition of longer-term dependency. FedAvg, augmented by Fed-Prox regularization, is employed to manage non-IID data among customers, enhancing stability, while Focal Loss and attention entropy improve class balance and attention sharpness. Experiments on mango and rice leaf datasets provide high accuracies of 97.4% and 94.10%, respectively, surpassing standalone CNN and ViT models in federated settings. This method facilitates scalable, privacy-preserving identification of plant diseases.

Chapter 6: Conclusion and Future Scope

This chapter delineates the thesis's primary results and contributions. This chapter also addresses prospective research avenues.

Chapter 2

Literature review

The objective of this chapter *. *is to provide a structured taxonomy of (Machine Learning) ML, (Deep Learning) DL, and (Internet of Things) IoT-based approaches for plant leaf disease classification with a detailed discussion on algorithms, datasets, and intelligent sensors in depth. This chapter analyses the utilities of existing state-of-the-art and pinpoints existing research gaps.*

While tremendous advancements have been achieved in AI-based agricultural systems, there is still an increasing demand for structured analysis and unified frameworks for plant disease diagnosis. This chapter provides an overview of the literature survey. It then presents a thorough evaluation of existing AI-based techniques in the field, as shown in Table 2.1. The table presents a categorized review of AI-based plant disease detection studies, highlighting five key components: Year, techniques used, datasets, limitations, and journal name & publisher. The literature covers a range of methodologies from traditional machine-learning approaches like SVM and handcrafted features to advanced deep-learning architectures such as CNNs and VGG-based models. Overall, the table shows that while early works depend heavily on color/texture features, more recent research shifts toward deep learning, yet still faces challenges like unbalanced datasets, high-dimensional features, and limitations in real-world generalization.

*The part of this chapter has been published/accepted/communicated in: Meenakshi Srivastava, Varsha Sisaudia, Jasraj Meena, "A review on internet of things and artificial intelligence based plant disease detection: taxonomy, issues and challenges," *Journal of Plant Diseases and Protection*, Elsevier, Volume 132, pp. 181, 2025, DOI: https://doi.org/10.1007/s41348-025-01184-0. **(SCIE, IF=2.2)**

2.1 Literature Survey on Traditional Approach for Plant Leaf Disease Classification

Early stages of plant disease classification relied heavily on handcrafted image features, rule-based systems, and traditional machine-learning approaches. These methods focused on extracting simple visual cues such as color changes, texture variations, and lesion shapes from leaf images, using predefined feature sets to distinguish between healthy and diseased regions. Rule-based decision systems and classical classifiers like SVM or k-NN were commonly used to categorize diseases based on these manually crafted features. However, these traditional approaches were limited in their ability to handle complex disease patterns, environmental variations, and subtle symptom differences, which restricted their effectiveness in producing accurate and robust disease classification across diverse plant species.

Shrivastava V et al. [23] presented an image-based machine learning approach for classifying rice plant diseases using only color features. Since traditional manual identification was slow, subjective, and often inaccessible to farmers, the authors developed an automated system that converts each image into 14 different colour spaces and extracts 172 statistical colour features. The study evaluated seven classifiers and found that the Support Vector Machine with a polynomial (order-2) kernel achieved the best classification accuracy of 94.65%. The results demonstrated that color information alone is highly effective for disease detection, offering a fast, affordable solution for farmers. The major limitation of this study was that it relied solely on colour-based features for disease classification, limiting the model's ability to capture complex visual cues, such as texture and shape, that could improve accuracy. Additionally, the work focused exclusively on rice, limiting the approach's generalizability to other plant species.

Saleem, R et al. [24] presented a novel computerised system for recognising mango leaf diseases, specifically powdery mildew and sooty mould, using a new segmentation method based on leaf vein patterns. Since manual disease detection was slow and expert-dependent, the authors propose a step-by-step approach that includes data augmentation, a unique vein-based segmentation technique, and the extraction of colour and texture (LBP) features. These features were fused using canonical correlation analysis (CCA) and classified using 10 machine-learning algorithms, with cubic SVM achieving the highest accuracy of 95.5%. The major limitations of this study were mainly related to its dataset and methodology. The dataset used was very small, which restricts the model's ability to generalize

to actual variations. Another drawback was that the system relies on handcrafted color and texture data along with SVM classifiers, which might not capture the complex and diverse patterns present in diseased leaves. With recent advances in deep learning, modern CNN-based or transfer-learning models could potentially achieve higher accuracy and more robust performance compared to the traditional feature-engineering approach used in this research work.

Dwivedi, R. et al. [25] suggested a fast and efficient leaf disease detection mechanism using L1-Norm minimization Extreme Learning Machine (L1-ELM) combined with multi-feature extraction and Kuan filtering. Traditional deep learning and computer vision methods often require large datasets, heavy training time, and complex segmentation pipelines, making them unsuitable for real-time agricultural use. To overcome these issues, the authors apply Kuan filtering to remove noise and preserve texture, then compute a rich set of color, texture, and shape features from leaf images. These features were fed into the proposed L1-ELM, which enhances classical ELM by applying L1-norm optimisation to achieve sparser hidden layers, faster learning, and better generalisation. Experiments conducted on peach, strawberry, and UAV maize leaf datasets show that L1-ELM consistently outperforms SVM, KNN, K-means, OP-ELM, and even standard ELM in terms of accuracy, precision, recall, and F-measure. The method achieved high recognition rates with significantly lower training/testing time than deep CNN models. The paper does not compare its method with modern lightweight CNNs like MobileNet or EfficientNet, which serve as stronger baselines for large datasets. The approach also shows limited generalization to real farm conditions, where images include multiple leaves, occlusions, and varying lighting. As a result, the model performs well on controlled datasets but may struggle in real-world agricultural environments.

2.1.1 Literature Survey on Deep Learning Approach for Plant Disease Classification

Early stages of deep-learning-based plant disease classification marked a significant shift from human feature development to automated feature extraction using neural networks. Deep learning models especially Convolutional Neural Networks (CNNs) learned discriminative patterns directly from raw leaf images, capturing complex visual cues such as multi-scale textures, disease progression characteristics, and subtle color variations that traditional methods often failed to detect. Instead of relying on predefined rules or handcrafted fea-

Table 2.1: Comprehensive Analysis of AI-based Plant Disease Detection Studies

Author	Year	Techniques Used	Datasets	Limitations	Journal Name & Publisher
Saleem. R et al. [24]	2021	colour feature, Texture feature technique	Mango Disease Dataset	Relies on handcrafted feature, modern deep learning could achieve higher accuracy.	Applied Science
Chen, J. et al. [26]	2020	INCVGGNet	Rice and Maize	Dataset used in the study is unbalanced and contains overlapping disease symptoms along with cluttered field backgrounds, which leads to a noticeable drop in model performance	Computers and Electronics in Agriculture, Elsevier
Shrivastava V et al. [23]	2021	SVM Classifier	Rice Disease Dataset	Only color features used; ignores shape and texture, reducing robustness in practical settings.	Journal of Plant Pathology, Springer
Bei Mohammadi et al. [27]	2022	SWAP leafNet	Flavia, Malaya Kew	High-dimensional data leads to potential overfitting and increased model complexity.	Expert Systems With Applications, Elsevier
Dwivedi et al. [25]	2021	ELM	Peach, Strawberry Dataset	Random initialization limits ability to capture complex non-linear patterns.	IEEE GRSL, IEEE
Paul S et al. [28]	2023	VGG16,VGG19	Tomato leaf dataset	its performance heavily depends on large, high-quality labeled datasets, making it less effective when training data is limited or collected in real-world farm conditions.	Array, Elsevier
Kaur P et al. [29]	2023	MIR-V2	Tomato leaf dataset	Pre-trained models may inadequately encapsulate the domain-specific visual features of plant diseases, leading to reduced accuracy when utilized on varied actual field images.	Journal of Ambient Intelligence and Humanized Computing, Springer
Khan I et al. [30]	2024	VGGNet, InceptionV3, ResNet50	Maize Leaf Dataset	The limitation of this paper is that it focuses on a curated dataset and refined ResNet50 architecture but does not address aspects like generalizability, real-world deployment, and interdisciplinary integration.	Journal of Agriculture and Food Research, Elsevier
Nawaz M et al. [31]	2024	CoffeeNet	Arabica coffee leaf dataset	Limitation find that the data augmentation strategy effectively tackles overfitting concerns arising from a limited number of photos while ensuring a balanced dataset.	Expert Systems with Applications, Elsevier

tures, CNN-based systems automatically discovered hierarchical representations, leading to more accurate and scalable disease identification. As research progressed, deeper architectures, transfer learning techniques, and large annotated datasets like PlantVillage further enhanced model robustness, enabling reliable classification across diverse crop species and real-world environmental conditions. Despite their computational demands, deep learning approaches significantly improved precision, generalization, and adaptability in modern plant disease classification systems.

Chen, J. et al. [26] presented a deep transfer-learning-based approach for automatic plant disease identification using leaf images. The authors combine a pre-trained VGG-19 network with enhanced components such as Batch Normalization, the Swish activation function, an additional convolutional layer, global average pooling, and two Inception modules, forming a new architecture called INC-VGGN. Instead of training from scratch, the model transfers knowledge from ImageNet and fine-tunes it on rice and maize disease datasets. The method includes image preprocessing, augmentation, and multi-scale feature extraction to handle real-world challenges like uneven lighting and cluttered backgrounds.This paper's limitations are mainly related to data quality, hardware requirements, and dataset size.

Beikmohammadi, A et al. [27] introduced SWP-LeafNET, a novel multistage deep-learning structure for identifying plant species via leaf photos that mimics botanists' decision-making processes. The framework integrates three CNN-based models, S-LeafNET, W-LeafNET, and P-LeafNET, that progressively analyze a leaf's shape, color, and venation. The first two models are lightweight architectures built from scratch to enable fast processing on portable devices, while the third model uses transfer learning with MobileNetV2 for detailed patch-level venation analysis. The system employs stepwise termination conditions, allowing easy samples to be classified early and improving speed by sending only ambiguous cases for deeper analysis. This paper's approach introduced additional system complexity due to its multistage architecture. Unlike a single CNN model, SWP-LeafNET requires the design, training, tuning, and integration of three separate deep-learning models. This increases the overall development effort, demands careful coordination between stages, and makes the training pipeline more time-consuming. As a result, the multistage setup is more complex to manage and optimize compared to end-to-end single-network solutions.

Khan I et al. [30] suggested a transfer learning approach to recognise maize leaf disease to overcome the strength and generalizability limits of the image processing approach. Furthermore, ResNet50 had the highest performance, with 0.87% validation accuracy, 0.90% precision, and 0.99% recall. Transfer learning has the potential to address nuanced plant issues, as evidenced by the innovative maize disease detection technique presented in this research.

Nawaz M et al. [31] introduced the CoffeeNet framework, an improved deep-learning model for detecting and categorizing aberrant coffee plant leaves. Additionally, extensive experimental research and assessments on the Arabica coffee leaf demonstrated the effectiveness of the proposed approach. The results show that the CoffeeNet technique outperforms

other approaches and can handle various types of data distortions. According to the study, the proposed approach's finding is that the data augmentation strategy addresses concerns caused by a shortage of photos and balances the dataset.

Paul S et al. [28] proposed a lightweight custom CNN algorithm and transfer learning-based method to categorize tomato diseases. This paper simulates several tomato leaf diseases using eleven classes, including a healthy one. Also, a study was conducted to determine the appropriate parameters for the suggested model. Further, assessment measures were employed to examine the effectiveness of the suggested and TL-based approaches. Furthermore, using data augmentation approaches, the suggested approach obtained 95.00% accuracy and recall, greater than all other models. Finally, the most effective model was used to create a Web-based system for farmers to recognize tomato leaf illness

Kaur P et al. [29] proposed the Modified InceptionResNet-V2 (MIR-V2) CNN technique, which uses transfer learning to detect diseases in images of leaves of tomatoes. The suggested approach has been trained on a tomato dataset that includes seven disease leaf classifications and one normal leaf sample. Then, proposed models outperform current tomato leaf disease identification techniques and advanced deep learning methods in overall efficacy. A key limitation of this transfer learning–based plant leaf disease detection approach is that pre-trained models may not fully capture domain-specific visual features of plant diseases, resulting in reduced accuracy when applied to diverse real-world field images.

2.2 Analysis of Existing Literature Survey

An analysis of the existing literature surveys is conducted in this section. Table 2.2 presents a detailed literature survey based on many techniques. At first, most of the papers mainly focused on the role of AI-based techniques, such as ML and DL, while some were based on the IoT-based plant disease recognition framework. We can see from the review papers that no research has proposed a clear taxonomy, conducted rigorous dataset analysis, and provided information on IoT sensors. Finally, as per the analysis, researchers only discussed IoT-based plant disease detection-related applications, which are considered only ML, DL, etc. methods. From 2017 to 2025, researchers made major contributions to AI and IoT in agriculture, though their focus areas varied. Most early research, like that by Shoaib M et al. [32] and Yilmaz E et al. [33], focused on IoT frameworks or smart sensors rather than integrating AI with IoT. Studies such as Iqbal Z et al. [34] and Kaur S et al. [35] focused on applications of AI for analysis while ignoring IoT frameworks, datasets, and

AI+IoT solutions. By 2020, researchers like Kour V et al. [36] and Hasan R et al. [37] had advanced the area by incorporating smart sensors and databases, but full AI+IoT frameworks remained unavailable. Moving forward to 2021 and 2022, contributions from Orchi H et al. [38] and Thakur P et al. [39] maintained AI-centric research, although without real IoT integration. A transition occurred in 2023, when works such as R P et al. [40] give a more holistic picture, merging AI with IoT frameworks, smart sensors, and even tackling problems in AI+IoT adoption, marking a huge step toward actual implementation. In 2024 and 2025, research by Dang M et al. [41], Ding W et al. [42], Shoaib M et al. [32], and Yilmaz E et al. [33] focus primarily on AI analysis, taxonomy creation, and datasets, but they mostly ignore incorporating IoT-based frameworks or directly tackling AI+IoT challenges. Finally, our study stands out for being the most complete. Unlike previous studies that focused solely on AI or IoT, our research incorporates AI, IoT frameworks, smart sensors, taxonomy, datasets, and a combined AI+IoT approach. It also investigates AI+IoT-specific challenges and issues that are relevant to real-world agricultural applications. Thus, the literature demonstrates a steady shift from separate AI or IoT studies to more integrated approaches, with our contribution filling a significant research vacuum by offering a unified and thorough AI+IoT-based framework for plant disease detection and smart agriculture systems.

2.3 Taxonomy of IoT and AI based Plant Disease Identification

The taxonomy shown in Figure 2.1 highlights major research and implementation of plant illnesses across time. We divided the taxonomy of IoT-based plant illness diagnosis into three categories: supporting methodology, field of application, and IoT technology. However, the current taxonomy has limitations because it does not fully encompass emerging advancements such as AI-IoT-based approaches, federated learning for safe data processing, and real-time adaptive models. Furthermore, it lacks a standardized testing methodology for comparing various IoT-based approaches. It does not sufficiently address the challenges of integrating heterogeneous IoT devices or the influence of environmental changes on identification accuracy.

The following taxonomy, developed through a thorough review of current literature and case studies, is divided into three subparts: Methodology, Application, and Technology, as explained below:

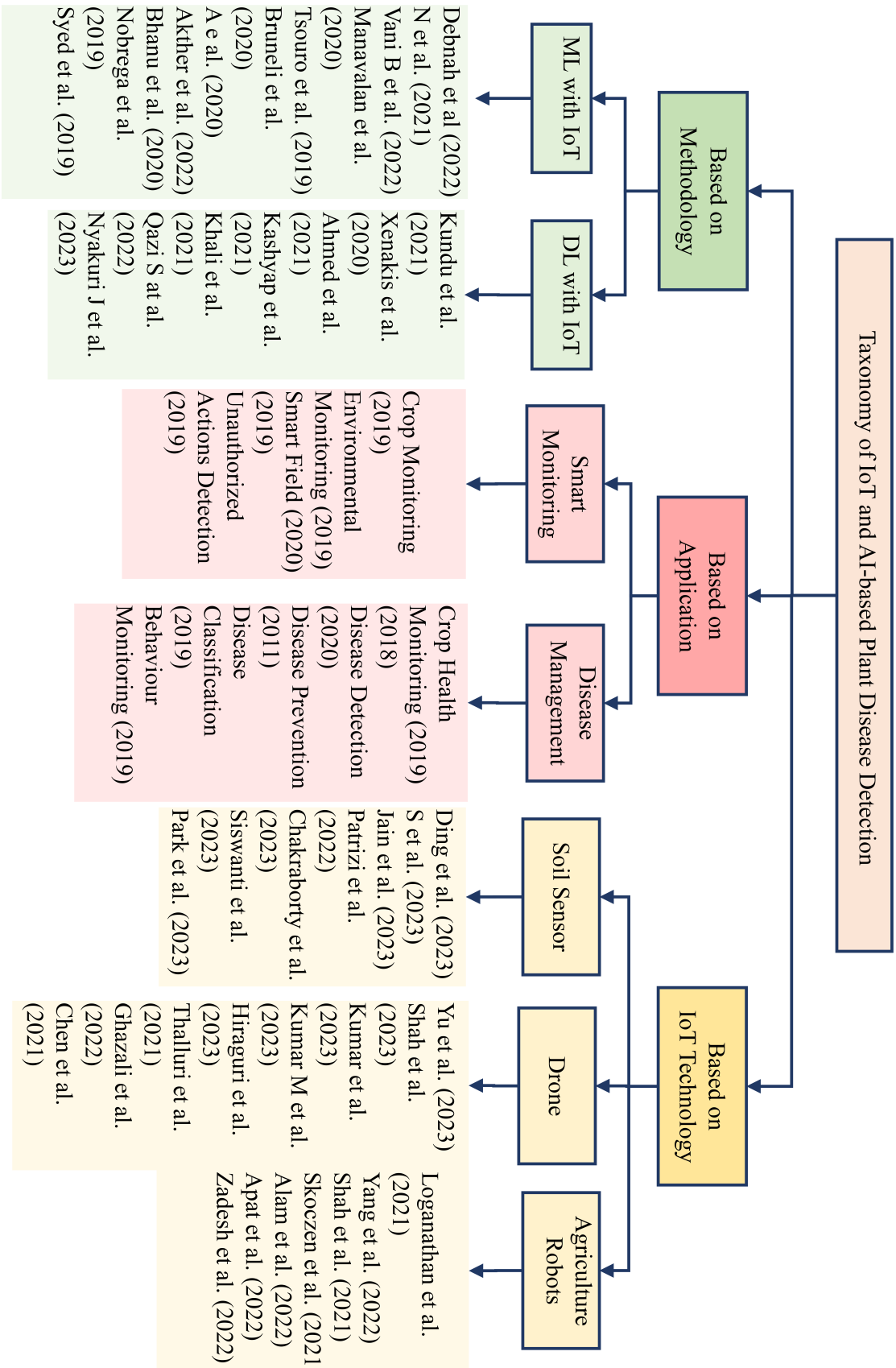


Figure 2.1: Proposed taxonomy

Table 2.2: Analysis of AI and IoT-based Plant Disease Detection Studies

Author	Year	AI Analysis	IoT-based Frame-work	IoT	AI + IoT	Smart Sensors	Taxonomy	Dataset	Challenges
Shoaib M et al. [32]	2025	✓	⊗	⊗	⊗	⊗	✓	⊗	⊗
Yilmaz E et al. [33]	2025	✓	⊗	⊗	⊗	⊗	✓	⊗	⊗
Ding W et al. [42]	2024	✓	⊗	⊗	⊗	✓	⊗	⊗	⊗
Dang M et al. [41]	2024	✓	⊗	⊗	⊗	⊗	✓	⊗	⊗
Ahmad A et al. [43]	2023	✓	⊗	⊗	⊗	⊗	✓	⊗	⊗
R P et al. [40]	2023	✓	✓	✓	⊗	✓	⊗	⊗	✓
Thakur P et al. [39]	2022	✓	⊗	⊗	⊗	⊗	✓	⊗	⊗
Orchi H et al. [38]	2021	✓	⊗	✓	⊗	⊗	⊗	⊗	⊗
Hasan R et al. [37]	2020	✓	⊗	⊗	⊗	⊗	✓	⊗	⊗
Kour V et al. [36]	2020	⊗	⊗	✓	⊗	✓	⊗	⊗	✓
Zhang J et al. [44]	2019	⊗	⊗	⊗	⊗	✓	⊗	⊗	⊗
Iqbal Z et al. [34]	2018	✓	⊗	⊗	⊗	⊗	⊗	⊗	⊗
Kaur S et al. [35]	2018	✓	⊗	⊗	⊗	⊗	⊗	⊗	⊗
Sophocleous M et al. [45]	2017	⊗	⊗	✓	⊗	⊗	⊗	⊗	⊗
Our Study	–	✓	✓	✓	✓	✓	✓	✓	✓

2.3.1 Based on Methodology

In the first category, Debnath et al. [46] proposed an intelligent real-time farming system using ML and IoT. They used the CNN model for the first time on rice paddy. However, the author also developed an application for a farmer through which farmers can monitor the crop from their phones. The model has been trained and evaluated on actual data, which is more complicated. In addition, the model was constructed using the most advanced frameworks, such as TensorFlow and Keras API (Application Programming Interface), making the suggested model highly versatile. This model achieved a high testing accuracy of 97.7%, making it superior to conventional CNN models for disease identification in rice paddy.

Nagasubramanian et al. [47] proposed ensemble learning techniques with SVM and IoT technology to provide quick categorization and tracking systems. In addition, the author collects real-time images by using hyperspectral cameras. Furthermore, Vani B et al. [48]

conduct a literature review on crop yield prediction issues in IoT-based farm applications. In addition, the author also exposes many errors (prediction error, categorization error, and calculation error) in current approaches to predicting plant disease in smart agriculture. The author has mentioned many potential perspectives and problems that can improve accuracy in smart agriculture applications. As per the findings, lacking advanced technologies is a primary challenge in the leaf illness recognition system.

Manavalan et al. [49] investigated the efficacy of standard ML and IoT-based plant illness recognition systems for the five main cash yields: soybeans, maize, barley, wheat, and rice. This paper also discussed the rewards (High productivity, Effectiveness in interpretation and comprehension, etc.) and disadvantages (Large dimensional data is challenging to manage (Gradual convergence, etc) based on the techniques.

Tsouros et al. [50] discussed the benefits and drawbacks of each data-gathering technique and technology utilized in UAV-based precision agriculture. Moreover, irrigation has been optimized using these techniques by automating the procedure based on data collected by sensors, GPS, and other devices. In addition, the authors also reviewed some standard methods for analysing data from remotely sensed images and then talk about the results of each and how they might be utilised in farming operations.

Brunelli et al. [51] proposed an application based on a low-energy structure, powered by a few hundred square cm, to create an emission system that can continuously run a low-power WAN. In addition, this IoT solution also has a sophisticated, lightweight framework for an ML method designed for virtual prototyping. Further, they additionally used a microcontroller and the Intel Movidius Neural Computing Stick to create the neural network. This study has found that the suggested systems are effective and reviewed power usage comments to attain zero energy balance.

Escamilla et al. [52] proposed an ANN application in smart agriculture. It also addresses the development of this model by combining foundational advancements such as IoT and AI. In this paper, most studies concentrated on predicting the microclimate, with a feed-forward network being the better prevalent structure. However, because RNNs are less frequent, studying different architectures and training methodologies is critical to assess their advantages and disadvantages over feed-forward networks. Following that, they identified the most significant findings in this work, which might serve as a guide for those developing revolutionary protected agriculture technology based on Technology 4.0.

Akhter et al. [53] proposed a strategy for predicting apple disease in orchards in the Kashmir Valley using data analytics and ML within an IoT system. A local poll was performed to gather feedback from farmers on current technology and its impact on precision agriculture. The research examine the difficulty of integrating new technology into traditional farming practices. One of the critical advantages is the ability to take immediate action against any potential ailments, such as scabies, if it is accurately and quickly predicted. The author discovered that the work would need to be expanded to cover additional crop-related elements.

Bhanu et al. [54] presented research that examines various ML approaches for an IoT-based intelligent farming system. This study analyses the current state of IoT in farming by examining key literature, hardware breakthroughs, cloud-based services, agricultural applications, and challenges. After reviewing that research, the author only employed irrigation for production in the suggested work, whereas fertility characteristics may be applied in future work.

Nobrega et al. [55] proposed a smart agriculture application, the SheepIT project. It aimed to keep sheep out of vineyards, preventing crop damage. IP-based methods are not appropriate for the needs and limits. However, the authors created a unique connectivity stack to satisfy the project's requirements, from the physical level to the application level. While built for SheepIT, this stack can be extended for larger intelligent farming applications because of its significance to the agricultural environment. The suggested work role is to create a strong communication structure customized to specific agriculture requirements.

Syed, Fahad Kamraan, et al. [56] suggested an ML and IoT-based strategy to maintain soil moisture levels appropriate for agricultural development over a 24-hour period, regardless of the climate. On the other hand, the proposed approach has an intelligent irrigation system that supports effective water management and recommends the optimal crop based on previous soil quality data.

The advanced analysis of ML with IoT in plant disease has been discussed in Table 2.3. The table discusses the use of ML and IoT in plant disease, showing various methodologies and datasets for improving farming methods. Several studies aim to improve disease identification efficiency and precision, with some using specialized datasets such as PlantVillage, etc. There are some performance metrics, such as accuracy, precision, recall, f1score, confusion matrix, and AUC-ROC, that examine various aspects of a model's performance in categorizing images [47]. Precision demonstrates the model's capacity to avoid false positives,

ensuring that healthy plants are not misidentified as ill. Recall assesses how successfully the model identifies diseased plants, reducing the likelihood of undiagnosed infections. The F1 score balances precision and recall, making it especially effective for imbalanced datasets. Accuracy measures overall correctness. In the second category, Kundu N et al. [57] proposed a real-time smart data collection method that incorporates IoT and DL. Further, the author developed a real-time framework to collect parametric data and images from the farm growing pearl millet. It sends the collected information to the Raspberry Pi & the cloud server. In this In the study, the author used two models, Custom-Net and transfer learning, to ensure accurate classification.

Xenakis et al. [58] presented an adaptable IoT robotic system for greenhouse agricultural applications with actual plant disease diagnosis capabilities. Further, the author also describes how plant illnesses can be classified without costly chemical testing and analysis. This study indicates that long training durations and a lack of technology are the main obstacles to these articles.

Ahmad et al. [59] proposed a mobile-based ML system to identify plant leaf diseases. Furthermore, the suggested method employs CNN as the fundamental deep learning algorithm to detect 38 disease categories. Moreover, the author worked on a mobile application that allows farmers to take images of infected plant leaves and quickly determine whether the plant is healthy or unwell.

Kashyap et al. [60] suggested an IoT-based water management method for precision farming. This technique combines feedback and performs better in any place & for any period. However, by managing an irrigation scheduler, the author emphasized fundamental aspects of agriculture, like water conservation and irrigation timing. Based on the preliminary study results, the author aims to develop an advanced irrigation scheduler that predicts rainfall and soil moisture levels to maximize rain and reduce water usage.

Khalil et al. [61] outline the main application of DL in the IIoT. The author first examines several DL methods, such as recurrent neural network, CNN, and Autoencoder, and their application in diverse fields. Then, the author lists several DL application scenarios for IIoT systems, such as smart manufacturing. Finally, the author highlighted several potential future study areas for DL-based IIoT, including intelligent sensing, ultra-reliability, and low latency. A summarized analysis of DL with IoT in farming systems is provided in Table 2.4. The table discusses the use of DL and IoT in plant disease, highlighting various methodologies and datasets to improve farming practices. Several studies aim to improve

turnitin
Page 65 of 279 - Integrity Submission
Submission ID trn:oid:::1:3529840373

Table 2.3: Analysis of ML with IoT in Smart Agriculture (Part 1)

Reference	Techniques	Dataset	Performance Metrics						Objective	Limitations
			Acc.	Prec.	Rec.	F1	CM	AUC		
Debnath [46]	IoT, CNN	Rice paddy	✓	⊗	⊗	⊗	⊗	⊗	Improve reliability of recognizing illnesses.	High costs and low resilience to occlusion.
Nagasubramanian [47]	IoT, ENSVM	Self-collected	✓	✓	✓	✓	⊗	⊗	Ensemble classification benefits.	High load hinders real-time IoT deployment.
Vani B [48]	ML, IoT	—	⊗	⊗	⊗	⊗	⊗	⊗	Review on IoT prediction issues.	No exam of latency/energy tradeoffs.
Manavalan [49]	ML	—	⊗	⊗	⊗	⊗	⊗	⊗	Grain yield illness recognition.	Lacks multi-disease co-occurrence discussion.
Tsouros [50]	ML, IoT	Self-collected	⊗	⊗	⊗	⊗	⊗	⊗	UAV data analysis in agriculture.	No assessment of flight parameter impact.
Brunelli [51]	ML, IoT	—	✓	⊗	⊗	⊗	⊗	⊗	Real-time crop pest monitoring.	Low efficiency in the model.
Escamilla [52]	ANN, RNN	—	⊗	⊗	⊗	⊗	⊗	⊗	ANN application in farming.	High cost and limited scalability.
Akhter [53]	IoT, ML	Apple Scab	✓	⊗	⊗	⊗	⊗	⊗	Precision farming via ANN.	Relies on high-quality sensor data.
Bhanu [54]	ML, IoT	Self-collected	⊗	⊗	⊗	⊗	⊗	⊗	Smart device for precision.	High computational requirements.
Nobrega [55]	IoT, ML	—	⊗	⊗	⊗	⊗	⊗	⊗	M2M interaction for SheepIT.	High costs and integration challenges.
Syed [56]	IoT, ML	—	⊗	⊗	⊗	⊗	⊗	⊗	Low-cost system for intelligent farming.	Flaws in weather forecasts and adaptation.

turnitin
Page 65 of 279 - Integrity Submission
Submission ID trn:oid:::1:3529840373

disease identification efficiency and precision, with some using specialized datasets such as PlantVillage, etc. Performance measures are provided erratically, with accuracy being the most prevalent. Some studies need full metric reporting and use unidentified datasets.

2.3.2 Based on Application

This field is categorized into two subcategories: smart monitoring and disease management. Many subfield types of smart monitoring systems, such as crop, environment, etc., will be examined.

Crop monitoring: An essential part of crop monitoring is tracking how crops grow and produce at all stages of development. Triantafyllou et al. [63] provided supplied the structural elements of an intelligent farming system based on an IoT communication network and WSN characteristics in collaboration with power-saving protocols. The author also discusses using a case study from the DIAS project to strengthen the model's validity and contributions. [64] the mySense environment systematizes data-collecting processes to address typical PA/PV difficulties. It is based on a 4-layer technological architecture that supports end-user software applications, cloud services, networks, and sensor nodes. It makes it possible to increase the adoption of crop monitoring applications by using low-cost, easily deployable, seamless, and integrated technology. [65] The mySense environment standardizes data collection processes to address common PA/PV issues. It is built on a four-tiered technological architecture that supports end-user software applications, cloud services, networks, and sensors. It enables the deployment of agricultural monitoring applications by utilizing low-cost, quickly deployable, smooth, and integrated technology.

Environmental Monitoring: IoT technology is critical for analyzing the physical environment since it gives real-time soil, water, and air data. [66] suggested a technique for adjusting the light settings to manage Brassica Chinensis growth in a lab setting utilizing LEDs. Through an IoT platform, the authors also looked at how light, environment and plant morphology are related. Treatment under various light conditions also had a positive impact on plant output.

Hirsch et al. [67] discuss the development of a soil moisture sensor node that uses capacitive sensing technology to measure the moisture level in the soil. This sensor node is designed for IoT applications in agriculture and home environments. Further, the authors also discuss the testing outcomes, which revealed that the sensor node could accurately measure soil moisture levels in various soil types and conditions. Furthermore, the authors emphasize

Table 2.4: Analysis of Plant Illness Identification using Deep Learning with IoT

Reference	Techniques	Dataset	Performance Metrics					Objective	Limitations
			Acc.	Prec.	Rec.	F1	CM	AUC	
Kundu [57]	IoT, DL	Pearl millet	✓	✓	✓	✓	✓	✓	To enhance disease detection accuracy in agriculture for improved crop management and yield.	The framework is restricted to RGB images using GradCAM and supports only two disease classes, limiting scalability to multi-disease and real-field conditions.
Xenakis [58]	CNN	PlantVillage	✓	⊗	⊗	⊗	⊗	⊗	To integrate CNN-based image recognition with IoT-enabled robotic systems for plant disease diagnosis.	Relies mainly on greenhouse-captured leaf images, reducing robustness under natural field conditions and non-leaf disease symptoms.
Ahmad [59]	CNN	PlantVillage	✓	✓	✓	✓	✓	⊗	To improve disease detection accuracy and provide accessible plant health diagnostics via mobile platforms.	Performance degrades in real-world conditions due to lab-quality images and limited real-time mobile optimization.
Kashyap [60]	IoT, LSTM	—	✓	⊗	⊗	⊗	⊗	⊗	To design an IoT-based precision water management system using feedback-driven learning.	LSTM-based RNN models are computationally intensive and energy-expensive, limiting deployment on low-power IoT devices.
Khalil [61]	IoT, DL	—	⊗	⊗	⊗	⊗	⊗	⊗	To analyze the advantages, trends, and future directions of deep learning in Industrial IoT applications.	High model complexity and strict latency requirements hinder real-world deployment of DL-based IIoT systems.
Quazi [62]	AI, IoT	—	⊗	⊗	⊗	⊗	⊗	⊗	To provide a comprehensive review of AI and IoT applications in smart agriculture.	Lacks detailed implementation strategies, making practical adoption and system design challenging.

the potential advantages of using the sensor to capture huge amounts of soil-level data, which could be examined to develop new insights and improve water management practices. The study's findings suggest that capacitive soil moisture sensing technology can significantly enhance soil levels in agriculture and home environments.

Field Monitoring: Field monitoring is observing and recording data from physical locations or environments to monitor and analyze the status and performance of equipment, systems, or processes Popescu et al. [68] presents a novel approach to precision agriculture monitoring using an advanced system combining wireless sensor networks (WSNs) and UAVs. The system enables farmers to acquire accurate and real-time data on the status of their crops, allowing for more efficient and successful choices. Further, the authors conducted experiments using their system, demonstrating its effectiveness in monitoring crops and detecting anomalies such as pests, diseases, and drought. The outcomes demonstrated that the system could provide accurate and reliable data with high spatial and temporal resolution, enabling farmers to take timely and appropriate actions. Furthermore, the limitation of the paper is that the tests were carried out in a controlled atmosphere and on a small scale. Baseca et al. [69]proposed a smart decision support framework. This framework trains decision rules from data and allows domain experts to change the outcome's rule base.

Unauthorized Actions Detection: Muminov et al. [70] proposed virtual fence. This clever collar simulates an animal's position relative to one or more fence boundaries. It has been utilized to keep goats under control and monitor their well-being without visible physical fencing. The goat's 20% likelihood of receiving an electrical stimulus is only considered if it does not turn or stop in the warning zone. [71] A highly efficient automatic recording device has been developed to detect, record, and wirelessly transmit larvae's acoustic emissions while feeding inside tree trunks. The device is lightweight, approximately half the dimensions of a cell phone, and has been examined in real-world and laboratory settings. The device accurately detects borers and can quickly scan many mulberry and palm trees without human involvement.

This makes it particularly useful in quarantine areas for checking tree trunks. In addition, the device can help reduce the cost of treating and analysing trees using systemic pesticides. A summarised view of these works is obtainable in Table 2.5 and Table 2.6.

The second group covers several types of crop health monitoring, identification of diseases, prevention, and categorization.

Table 2.5: Study of Plant Disease Detection Based on Application

Based on Smart Monitoring Application	Reference	Techniques	Dataset	Performance Metrics						Objective	Limitations
				Accuracy	Precision	Recall	F1-Score	Confusion Matrix	AUC-ROC		
Crop Monitoring	Triantafyllou et al. [63]	IoT	—	⊗	⊗	⊗	⊗	⊗	⊗	To improve crop output and productivity by monitoring crops, soil, and agro-meteorological parameters efficiently and cost-effectively.	The architecture is based on a basic 7-layer IoT/remote sensing stack; however, it does not account for realworld concerns such as latency, bandwidth limits, and sensor fusion delays in large-scale agricultural applications.
	Morais et al. [64]	IoT	—	⊗	⊗	⊗	⊗	⊗	⊗	To offer a complete data gathering, storage, analysis, and visualisation solution for agriculture to aid farmers in making data-driven decisions.	Limitations include the complexity of establishing a publicly accessible monitoring system, which may necessitate more skill than is readily available.
Environmental Monitoring	Harun et al. [66]	IoT	—	⊗	⊗	⊗	⊗	⊗	⊗	To Develop a new and enhanced observation system that uses IoT to optimise Brassica chinensis growth.	The system uses IoT to control indoor lighting via LEDs; however, it does not evaluate sensor calibration drift, ambient light interference, or closed-loop feedback control stability in dynamic situations.
	Hirsch et al. [67]	IoT	—	⊗	⊗	⊗	⊗	⊗	⊗	To design and implement a Capacitive Soil Moisture Sensor Node for the IoT in agriculture and home.	The paper's capacitive soil moisture sensor node is technically limited by insufficient compensation for soil salinity and dielectric variation, which degrade its measurement accuracy across diverse soil types and over long-term deployments.
Field Monitoring	Popescu et al. [68]	UAV, WSN	—	✓	⊗	⊗	⊗	⊗	⊗	The author integrates UAVs and WSNs to provide an intelligent monitoring solution that collects and processes crop health and growth data.	The paper's limitation is the necessity for comprehensive field testing to address issues with sensor coverage and power consumption in the UAV-WSN system.
	Baseca et al. [69]	IoT	Self-collected	✓	⊗	⊗	⊗	⊗	⊗	To develop a smart decision system that analyses crop growth and agricultural data to make informed farming decisions.	Lack of real-time evaluation
Unauthorized Actions Detection	Muminov et al. [70]	IoT, ML	Self-collected	✓	✓	✓	✓	⊗	⊗	To provide a modern alternative to traditional physical fencing and allow for more efficient and effective management of goat herds.	Its reliance on GPS and SVM-based behaviour categorisation, which is prone to positional drift and misclassification of identical activities (for example, walking vs grazing) in dynamic goat movements.s
	Potomitis et al. [71]	IoT	—	⊗	⊗	⊗	⊗	⊗	⊗	To develop a method for monitoring the health of trees using vibroacoustic signals and IoT technology.	High computational overhead.

Crop Health Monitoring: Farmers can significantly increase their productivity with the least effort if crop health parameters are routinely evaluated continuously. Pantazi et al. [14] provided an approach for recognizing illness in leaf samples from various species. It was taught to identify four different illness problems. The overall success rate for all 46 plant condition combinations studied was 95%. Furthermore, the tool may identify and categorize the abovementioned health concerns in plant types that have not been analyzed before. Kim et al. [72] developed a disease identification method for strawberries using IoT for plant disease. The system aims to aid cultivators in observing the development and well-being of their strawberry harvest and anticipate potential ailments that may impact their production. It provides advanced application services by managing farms and their associated devices, data, and models. On the cloud, it can collect, analyse, and forecast information about the agricultural environment.

Disease Detection: Disease must be discovered and diagnosed quickly and adequately to maximize agricultural productivity and reduce both qualitative quantitative losses. Zhao et al. [73] suggested a sophisticated IoT based automated crop disease system that employs deep learning to distinguish items in the wild visualization and MCFN methods traits from over 50,000 field-collected crop disease samples. Further, Experiments on 77 common agricultural illnesses yielded 97.5% detection accuracy.

Disease Prevention: IoT-based controlled environments aid in preventing and managing illnesses that damage agricultural products. Park et al. [74] designed and deployed a WSN-based greenhouse monitoring system. The system is designed to avoid condensation dew, which can negatively impact yield growth and value. Further, the paper demonstrates that the wireless sensor network-based automatic control system can track and regulate the greenhouse surroundings and avoid dew development without user intervention. The study's findings indicate that the presented strategy could forecast and reduce fog precipitation by managing the environmental parameters that cause greenhouse dewdrop formation.

Disease Classification: Several reviewed programs used deep learning models to classify data to assist in detecting plant and animal leaf illness. [5] proposed an IoT-based agriculture decision-control system that may be utilized for real-time applications and big medical and biological datasets, such as plant disease databases. The suggested approach intends to improve the ELM algorithm's accuracy and efficiency by selecting a relevant and compact feature subset from high-dimensional data. However, the tests were done on two distinct datasets, and the results show that the proposed approach outperforms others in terms of

efficacy and computing time.

Behaviour Monitoring: Wearable sensors are becoming increasingly crucial for farm management tracking animal behaviour. B.G et al. [75] developed an IoT-based circuit that may be implanted in a cow's rumen to aid in the identification of acidosis conditions. The developed technology monitors animal behaviour and food parameters.

2.3.3 Based on IoT Technologies

This field is divided into three sub-categories: soil sensors, drones, and agriculture robots. **Soil Sensors:** Soil sensors examine moisture levels at the plant's root area and control the present traditional timer, leading to significant water benefits when correctly placed and operated. Ding et al. [76] introduces SoilId, a soil moisture system that consists of a UAV and a COTS IR-UWB radar. It allows for wide-area soil moisture measurement, avoiding the need for buried battery-powered devices. In addition, the author determined SoilId against real-world conditions such as uncontrollable UAV motion, numerous path interferences, and soil surface covering, and more. After that, testing results show that SoilId can improve the accuracy of RF-based soil moisture monitoring by 50%, with a maximum efficiency of 0.23%. According to this study, the primary challenges for this paper include lower reflectors and a choice of radar.

Sundaresan et al. [77] proposed three systems to select yield-based soil characteristics, automatically water plants and optimal nutrients for each crop variety. In addition, this approach performs best with the following crops: coffee, orange, cotton, apple, etc. Farmers can utilise this technology to make better decisions, decrease manual labour, conserve energy, and boost yield.

Jain et al. [78] suggested a method for drip irrigation utilising IoT and web-based apps. The author also fabricated a web application to support user-specific actions throughout drip irrigation. This web application produces the proper quantity of water via the pump for planting. It can be performed by putting several sensors into the system to monitor temperature and soil moisture levels and transmitting the data to the processor to accurately estimate the plant's water requirements. So, the overall conclusion of this paper is to reduce unnecessary power and water consumption effectively.

Patrizi [79] describes the architecture of a wireless sensor network (WSN) using cheap, low-power devices photovoltaic (PV) sensor nodes that can collect data on the surroundings and soil properties. In addition, the author also proposed an LSTM network to create a simulated

Table 2.6: (b) Analysis of Plant Disease Detection Based on the Application

Based on Disease Management Application	Reference	Techniques	Dataset	Performance Metrics						Objective	Limitations
				Accuracy	Precision	Recall	F1-Score	Confusion Matrix	AUC-ROC		
Crop Health Monitoring	Pantazi et al. [14]	ML	Leaf dataset	✓	⊗	⊗	⊗	⊗	⊗	To propose image features analysis to extract relevant information from leaf images and one-class classifiers to identify diseased leaves based on this information.	ts reliance on handmade LBP image features and oneclass classifiers limits scalability to huge multiclass datasets and impairs resilience under real-world fluctuations.
	Kim et al. [72]	IoT, FaaS	—	⊗	⊗	⊗	⊗	⊗	⊗	To use sensor data collected from IoT devices to monitor the environmental conditions growth of the strawberries.	It establishes a constant critical disease-onset threshold and assumes infinite pathogen presence, which limits its forecast accuracy when pathogen loads vary, and infection dynamics are nonuniform.
Disease Detection	Zhao et al. [73]	IoT, Deep Learning	PlantVillage	✓	✓	✓	✓	✓	⊗	To suggest a sophisticated IoT-based automated crop disease system that employs deep learning to distinguish items in the wild visualisation.	The paper's limitation is the need for massive amounts of labelled data to adequately train the CNN, which can be challenging in agricultural settings.
Disease Prevention	Park et al. [74]	WSN	—	⊗	⊗	⊗	⊗	⊗	⊗	To develop a practical and efficient solution for preventing dew condensation in greenhouse environments.	It employs a static sensor placement and dew-point control method that overlooks spatial heterogeneity within large greenhouses or adaptive sensor reconfiguration to address microclimate variability.
Disease Classification	Kale A et al. [5]	IoT, DLCNN	Nutrient Use Outcome Network Dataset	✓	✓	✓	✓	✓	⊗	To design a mobile app-controlled intelligent hydroponics farming system that uses IoT and AI to optimize plant growth under controlled environmental conditions.	The proposed GA-ELM framework is technically constrained due to high computational overhead during feature subset optimization and the lack of dynamic re-selection, limiting its scalability for real-time IoT smart agriculture data streams.
Behavior Monitoring	Basciftci F et al. [75]	IoT	—	⊗	⊗	⊗	⊗	⊗	⊗	To describe an approach that employs IoT sensors to track the pH content of the rumen fluid in cattle and detect changes indicative of acidosis.	Limitations of the paper include the potential difficulty in effectively detecting acidosis in cattle due to symptom variability and reliance on reliable IoT infrastructure.

soil moisture sensor utilising only data from the node's other transducer. Then, performance evaluation has demonstrated the usefulness of the suggested soft sensing technique. The findings of this study aim to improve the farm's overall resource allocation and modify irrigation methods to reduce water loss.

Chakraborty et al. [80] proposed that MELA solve the issue of a few soil moisture data points. Using this method, the author effectively improved historical monthly moisture level measurements for randomly selected areas in semi-dry Caddo County from 1958-2021. In addition, the author also used the SHAP model with MELA to form a MELA-XAI architecture. This system uses soil moisture to predict annual winter wheat yields, considering typical local precipitation and temperature. In the future, the author suggests implementing this strategy in other parts of Oklahoma and across the United States.

Siswanti et al. [81] proposed a decision tree technique for automatic plant irrigation and monitoring of plants based on soil moisture levels. The decision tree determines the microcontroller's activity, turning the pump on or off for watering plants based on soil moisture. So, the finding achieved with automatic plant irrigation depends on soil moisture levels; specifically, the relay located on the micro-controller activates the water pump when the soil moisture content falls below 50%.

Park et al. [82] aims to create a soil moisture (SM) forecasting technique to guide irrigation decisions based on meteorological conditions. In addition, for time series data, the author also proposed RNN-LSTM techniques to forecast SM. The results show that the model performed well at 10 cm depth, with R2 of 0.999, a loss of 0.022, and a test loss of 0.105. The best outcomes were achieved at 20 cm and 30 cm depths, with R2s of 0.999, 0.016, and 0.098, and R2s of 0.956, 0.057, and 2.883, respectively. A summarized analysis of IoT technology, such as soil sensors, drones, and agriculture robots, in relation to plant disease is shown in Table 2.7-Table 2.9. The table analyzes numerous IoT-based plant disease strategies, including KNN, LSTM, and Decision Trees, primarily focusing on soil moisture and crop management. Most studies report accuracy, but additional performance metrics are often missing. The goals include improved soil moisture measurement, better recognition of plant illness, and improved irrigation systems. However, inconsistencies in metric reporting and undefined datasets are prevalent constraints, indicating the need for uniform assessments.

Drone: A farming drone is mainly used to optimise yields and examine crop growth and cultivation. Yu et al. [83] used AI and surface air nozzles built on drones to test their spraying

Table 2.7: (a) Analysis of Plant Disease Detection based on Soil Sensor

Reference	Techniques	Dataset	Performance Metrics						Objective	Limitations
			Accuracy	Precision	Recall	F1-Score	Confusion Matrix	AUC-ROC		
Ding et al. [76]	SoilId	Data collected by IR-UWB radar	✓	⊗	⊗	⊗	⊗	⊗	The author proposed the SoilId method. It allows for wide-area soil moisture measurement, avoiding the need for buried battery-powered devices.	Using smaller reflectors makes it more challenging for the UAV to align correctly with the measurement site.
Sundaresan et al. [77]	KNN	—	✓	⊗	⊗	⊗	⊗	⊗	To propose three systems to select yield-based soil characteristics, smart watering, and optimal nutrients for each crop variety.	The paper's limitation is the possible difficulty of effectively combining ML with IoT systems and ensuring correct data gathering and analysis for crop yield optimisation.
Jain et al. [78]	IoT	Self-collected data	⊗	⊗	⊗	⊗	⊗	⊗	The author fabricated a web application to support user-specific actions throughout drip irrigation. This web application produces the proper quantity of water via the pump for planting.	The IoT-enabled drip irrigation system needs continuous internet access and lacks fault-tolerant control, resulting in irrigation failures during network interruptions or outages.
Patrizi [79]	LSTM	Self collected data	✓	⊗	⊗	⊗	⊗	⊗	The paper's strategy improves crop disease recognition accuracy and reliability by using many context information sources.	Because of the restricted calibration across varied field circumstances, the deep learning-based virtual soil moisture sensor performs poorly when applied to unseen soil textures and moisture dynamics.
Chakraborty et al. [80]	MELA	Soil Moisture	✓	⊗	⊗	⊗	⊗	⊗	To Proposes the MELA method to solve the issue of sparse soil moisture data.	The limitation of the paper is the difficulty in appropriately estimating soil moisture data over diverse scales and weather circumstances.
Siswanti et al. [81]	Decision Tree	—	⊗	⊗	⊗	⊗	⊗	⊗	Proposes a decision tree technique for automatic irrigation and plant monitoring based on soil moisture levels.	The paper's limitations involve possible errors caused by reliance on soil moisture measurements, which may not correctly account for changing plant demands or climatic conditions.
Park et al. [81]	RNN-LSTM	Soil and weather data	✓	⊗	⊗	⊗	⊗	⊗	Proposed a soil moisture forecasting method to guide irrigation decisions based on meteorological conditions.	Dependency on high-quality data for correct training.

and insect control effectiveness. In addition, drone spraying was used on soybeans and rice to assess coverage, absorption ratio, crop production, and disease effects. In addition, the outcome demonstrated that the plant area with the AI nozzle was 130% greater than that with the flat air nozzle.

Shah et al. [84] proposed an autonomous plant recognition procedure using a synthetic neural network capable of detecting leaves from plants. The Author also addressed the transfer learning techniques for analysing plant diseases using photos taken with a drone or cell phone camera. The overall conclusion of this paper is to identify plant diseases quickly and efficiently, helping farmers of all experience levels make informed decisions about pesticide use and plant health.

Kumar et al. [85] presented a new approach to recognising apples using drone footage and the YOLOv7 framework. In addition, the suggested approach can automate agricultural operations, such as yield calculation and crop health evaluation, by providing accurate data on apple distribution and plant thickness. Also, the proposed model was designed to better identify apples in complex environments. The model recognises essential items and improves feature fidelity for accurate bounding boxes, resulting in high identification accuracy. So, a comparison analysis found that the updated YOLOv7 model achieves accurate results (0.91, 0.96, and 0.92 for precision, recall, and F1-score, respectively).

Kumar M et al. [86] proposed drone technology can optimise pesticide spraying by improving plant illness categorization through picture classification. Further, this approach allows various pesticide-spraying procedures to be used to assess treatment efficacy. Furthermore, analysing the effectiveness of this technology provided valuable insight into the effectiveness of drone-based methods. In the future, drone-based crop disease tracking utilising the NDVI approach will transform crop health monitoring and management.

Hiraguri et al. [87] discussed employing miniature drones to locate and pollinate flowers, replacing bees autonomously. The author also presented our testing results with these drones. The drones were designed to hunt for flowers and pollinate them. In addition, the Author used machine learning to create an AI classifier that can detect flowers.

Thalluri et al. [88] proposed a drone with a Raspberry Pi camera interface to automatically capture and analyse leaf photos for disease detection. Further, the author used the ML algorithm for feature extraction in the leaf dataset. The leaf's properties are recovered by identifying edges and statistical equalisation. Furthermore, the primary benefit of this method is that it provides guidance on effective ways to reduce the spread of illness. According to the findings, the proposed method significantly increases productivity while reducing the time and effort required to complete tasks in agricultural regions.

Ghazali et al. [89] suggested a flaw detection method using multirotor arm vibration utilising artificial intelligence. In addition, this study incorporates various AI approaches, such as fuzzy logic and neuro-fuzzy. Fuzzy logic and neuro-fuzzy methods provide appropriate decision-making. However, the outcome of the neuro-fuzzy method depends on the dataset used, as an overfitted model may lead to incorrect decisions. So, the findings of this study only considered one parameter: multirotor arms for flaw detection.

Chen et al. [90] proposed edge intelligence to detect T. papillose and develop real-time

paths for pesticide-spraying drones. Further, this study utilised two drones: a tiny reconnaissance drone and a larger pesticide-spraying drone. The reconnaissance drone's modest size minimises leaf disruption. Furthermore, the pair of drones and TX2 allows real-time insect detection in orchards. In addition, the outcome demonstrated that the proposed method helps farmers understand insect dispersion and implement preventive measures in real time. The farming drone applies pesticides only where necessary, reducing pesticide use and environmental damage while increasing crop yields.

Agricultural Robots: An agricultural robot is any robotic technology that can enhance the farming process by doing many of the farmer's tedious or labour-intensive tasks. Loganathan et al. [91] proposed Agri Bot, a new agricultural robot, which has been built to intelligently monitor each agricultural field and its related crops using artificial intelligence. This robot is controlled by artificial intelligence and accumulates varied temperatures and levels of soil moisture, inappropriate propositions, which are subsequently transmitted to a remote server via IoT technology.

In addition, the suggested model extracts agricultural details and sends them to the server side for tampering, where the data is received and processed using a new ML approach, the modified convolutional neural method. So, the research conclusion demonstrates the suggested technique's performance regarding both image processing results and the new ML technique, indicating that the process laid out in the study is ready for implementation for farming field monitoring.

Yang et al. [92] presented a residual-like soft actor-critic (R-SAC) method for farming applications, enabling obstacle avoidance and smart path planning for robots. In addition, this research presented an offline professional expertise pre-training strategy to enhance the efficacy of reinforcement learning training and reduce time-consuming research processes. Moreover, this study optimises the algorithm's reward function using multi-step TD error, addressing potential training issues. This paper concludes that our proposed method outperforms previous reinforcement learning methods in both static and dynamic contexts.

Shah et al. [93] introduced a weed identification robot using AI networks to identify weeds in inter-row spaces in fields of crops. Additionally, this robot is presented as a solution for infestations of weeds in agricultural fields. The results show a forecast accuracy of 88–98% compared to the ground truth. For an unidentified dataset of plants and weeds, estimations ranged from 67-95% for plants and 84% to 99% for weeds.

Table 2.8: (b) Analysis of Plant Disease Identification based on Drone

Reference	Techniques	Dataset	Performance Metrics						Objective	Limitations
			Accuracy	Precision	Recall	F1-Score	Confusion Matrix	AUC-ROC		
Yu et al. [83]	IoT	Rice and Soyabean	⊗	⊗	⊗	⊗	⊗	⊗	The author proposed AI and surface air nozzles built on drones to test their spraying and insect control effectiveness.	The research does not adequately examine how droplet evaporation, drift under changeable wind, and canopy turbulence affect spray deposition efficacy for various nozzle types in real-world situations.
Shah et al. [84]	EfficientNet-B3	PlantVillage	✓	✓	✓	✓	✓	⊗	To propose an autonomous plant recognition procedure using a synthetic neural network capable of detecting leaves from plants	Limited dataset robustness, bandwidth/power limitations in the drone-Android connection, and CNN overfitting all degrade accuracy in real-world scenarios.
Kumar et al. [85]	YOLOv7	Apple	✓	✓	✓	✓	⊗	⊗	The author fabricated a web application to support user-specific actions throughout drip irrigation. This web application produces the proper quantity of water via the pump for planting.	High computational load and limited depth accuracy under occlusion/illumination variations restrict real-time drone deployment.
Kumar M et al. [86]	MobileNetv2	Cashew	✓	⊗	⊗	✓	⊗	⊗	The author proposed drone technology can optimise pesticide spraying by improving plant illness categorization through picture classification.	The Limitations of the paper involve possible errors in disease identification and the use of pesticides, which can be attributed to differences in image quality.
Hiraguri et al. [87]	AI Classifier	Tomato	✓	⊗	⊗	⊗	⊗	⊗	To propose employing miniature drones to locate and pollinate flowers, replacing bees autonomously.	Limited flower recognition precision under occlusion/lighting, short drone battery life, and decreased stability in greenhouse conditions.
Thalluri et al. [88]	CNN	Tomato, Potato, Bell Pepper	✓	⊗	⊗	⊗	⊗	⊗	To propose a drone with a Raspberry Pi camera interface to capture and analyse leaf photos for disease detection automatically.	Low-resolution aerial images, high computational loads, and limited drone battery life restrict the accuracy of real-time multidisease detection.
Ghazali et al. [89]	Fuzzy Logic, Neuro-Fuzzy Logic	NA	✓	⊗	⊗	⊗	⊗	⊗	The author suggested a flaw detection method using multirotor arm vibration utilising artificial intelligence.	There is a requirement for diversified and high-quality data for training.
Chen et al. [90]	Tiny-YOLOv3	T. papillosa	✓	⊗	⊗	⊗	⊗	⊗	To propose edge intelligence to detect T. papillose and develop real-time paths for pesticide spraying drones.	Limitations include Potential difficulties in effective pest identification under variable environmental circumstances.

Skoczen M et al. [94] suggested a mechanism for determining accuracy in object identification and plotting via RGB-D camera-based vision. Further, the suggested method uses generated and accurate multimodal data, including photos and 3D representations of the area. Furthermore, the vision analysis pipeline evaluates the initial RGB image to find impediments in the work area. According to the results, the suggested photo and depth information processing pipeline introduces an additional 38 centimetres of distortion.

Alam et al. [95] compare two CNN-based methods, YOLOv5 and Faster R-CNN, to provide a vision-based structure for detecting and classifying tobacco crops and weeds in farms. Both methods are trained and estimated using TobSet, a self-created dataset of tobacco crops and weeds. The dataset includes 7000 photos of tobacco crops and 1000 photographs of empty soil and weeds, shot by hand with digital cameras during a 2-month timeframe. Moreover, the R-CNN-based vision architecture outperformed the YOLOv5-based architecture in accuracy and durability while using less inference duration.

Apat et al. [96] presented a new machine learning method for diverse data surroundings that includes IoT-sensed information about the surroundings, agricultural circumstances, plant characteristics, demands, etc. Further, the author utilised five plants (grain, ragi, gramme, potato, and onion) obtained from the Andhra Pradesh Kaggle database library. The farming robot collects these plants. Furthermore, the author utilised several machine learning and deep learning models, including regression, RNN, LSTM, EM, etc. Among machine learning methods, the method known as Random Forest surpasses the others with 99.27% accuracy in training for crop yield estimation.

Zadesh et al. [97] presented an innovative farming technique for medium-sized businesses (SMEs) using an automated arm and claw on a crane with a gantry to automate sowing, weeding, and harvesting procedures. Moreover, a website-based app was created to help farmers analyse agricultural conditions like temperature, humidity, and moisture in the soil in actual time. They can see the robotic seeding process and track the growth of the corps on an itinerary. Finally, a working model has been tested effectively to assess the overall efficacy of the planned AGRIBOT.

While the previous section identified known methods using a rigorous taxonomy, the next step is to investigate the available datasets that support these approaches and enable performance evaluation.

Table 2.9: (c) Analysis of Plant Disease Identification based on Agriculture Robots

Reference	Techniques	Dataset	Performance Metrics						Objective	Limitations
			Accuracy	Precision	Recall	F1-Score	Confusion Matrix	AUC-ROC		
Loganathan et al. [91]	MCNS, IoT	Leaf	✓	⊗	⊗	⊗	⊗	⊗	AgriBot, a new agricultural robot, has been built to intelligently monitor each agriculture field and its related crops using artificial intelligence.	High computational complexity and cost.
Yang et al. [92]	R-SAC	———	⊗	⊗	⊗	⊗	⊗	⊗	To present an R-SAC method for farming applications, enabling secure avoidance of obstacles and smart planning of robot paths.	High training costs, poor adaptability to dynamic farm environments, and hardware limitations restrict the use of real-time RL-based path planning.
Shah et al. [93]	Faster RCNN	Red radish, Garden cress, Common Dandelion	✓	⊗	⊗	⊗	⊗	⊗	The author proposes and trains a plant and weed identification robot using AI networks to identify weeds in inter-row spaces in fields of crops.	In field situations, accuracy and real-time weed detection are limited by a small dataset, shallow ANN feature extraction, and hardware constraints.
Skoczen M et al. [94]	ML	Synthetic and Real-world	✓	✓	✓	✓	⊗	⊗	The author suggests a mechanism for determining accuracy in object identification and plotting via RGB-D camera-based vision.	Robust obstacle identification in agricultural fields is limited by a low sensing range, high processing load, and limited depth accuracy in sunny conditions.
Alam et al. [95]	YOLOv5, R-CNN	TobSet	✓	⊗	⊗	⊗	⊗	⊗	To compare two CNN-based techniques, Faster R-CNN and YOLOv5, to provide a vision-based structure for detecting and classifying tobacco crops and weeds in farms.	The primary shortcoming of this method is its lower classification accuracy in actual field conditions.
Apat et al. [96]	Decision Trees, LR, LOGR, & NB	Onion, gram, potato, ragi & rice	✓	✓	✓	✓	✓	✓	To present a new machine learning method for diverse data surroundings that includes IoT-sensed information about the surroundings, agricultural circumstances, plant characteristics, demands, etc.	Practical deployment is limited by low adaptation to real-time farm fluctuations, high computer complexity, and limited dataset generalization.
Zadesh et al. [97]	SIFT	Carrot, Weed	✓	✓	✓	✓	✓	⊗	Introduced an automated farming technique using an arm and claw crane system for sowing, weeding, and harvesting.	FLANN lacks generalization across different crops, has a large computational overhead with robotic arm control, and has decreased accuracy under occlusion/lighting change.

2.4 Analysis of Public Available Datasets

After presenting the taxonomy of existing approaches and techniques for plant disease identification, it is critical to investigate the available datasets that serve as the foundation for training, validating, and benchmarking these technologies. This section gives a thorough examination of commonly used public datasets in this field. Additionally, Table 2.12 analyzes numerous additional datasets, including the number of photos accessible and the specific diseases related with each crop.

Plant Village Dataset:

The PlantVillage collection has 54303 healthy and unhealthy leaf pictures, organized into 38 sections based on species and illness. Figure 2.2 shows a sample copy of the dataset. The dataset includes images of 14 crops: apple, blueberry, cherry, grape, orange, peach, pepper,

potato, raspberry, soy, squash, strawberry, and tomato.

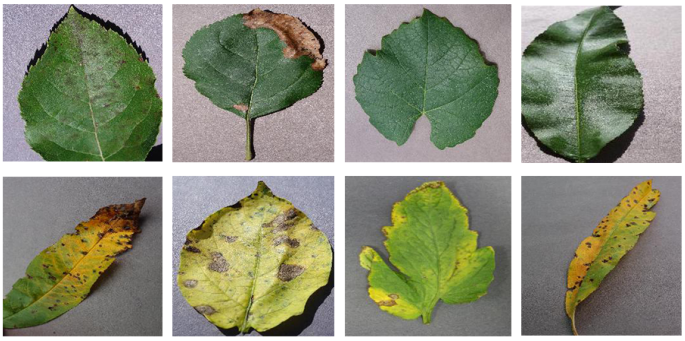


Figure 2.2: Sample image of plantvillage dataset

It includes photos of 17 primary illnesses, four caused by bacteria, two mold (oomycete), two viral, and one mite illness. Twelve varieties of crops also show healthy leaf photos that are free of visible illness. The PlantVillage dataset is a popular image collection utilized to help researchers construct ML algorithms for identifying plant diseases. The following are a few essential characteristics of the dataset:

1. PlantVillage dataset includes photos of healthy and diseased leaves from various plant types and illnesses.
2. Every image is identified with the plant type and particular illness or as healthy, offering helpful information for supervised learning.
3. The dataset has been widely utilized in academic research and is freely accessible, leading to notable agricultural breakthroughs.

Mendeley dataset: The Mendeley dataset is prepared by researchers from Shri Mata Vaishno Devi University in Katra, India [98]. Further, the dataset contains 11 different plants, and each plant has 2 classes, so in total there are 22 classes: healthy and unhealthy leaves, as shown in Table 2.10.

Cherry Dataset: The cherry dataset is a subset of the PlantVillage dataset [99], comprising approximately 2052 images of 1026 healthy leaves and 1026 powdery mildew leaves. Leaf images consist of 2 different plants, classified into 2 classes: healthy and unhealthy leaves, as shown in Table 2.11. Sample copies of Mendeley dataset and Cherry dataset are shown in Figure 2.3a and Figure 2.3b, respectively.

Peach Dataset

Table 2.10: Total Number of Images in Each Class

Class Name	Total Number of Leaves	Class Name	Total Number of Leaves
Alstonia Scholaris (P2) _diseased	256	Jamun (P5) _healthy	279
Alstonia Scholaris (P2) _healthy	254	Jatropha (P6) _diseased	124
Arjun (P1) _diseased	232	Jatropha (P6) _healthy	133
Arjun (P1) _healthy	224	Lemon (P10) _diseased	77
Bael (P4) _diseased	118	Lemon (P10) _healthy	159
Basil (P8) _healthy	148	Mango (P0) _diseased	265
Chinar (P11) _diseased	120	Mango (P0) _healthy	170
Chinar (P11) _healthy	103	Pomegranate (P9) _diseased	279
Gauva (P3) _diseased	142	Pomegranate (P9) _healthy	287
Gauva (P3) _healthy	277	Pongamia Pinnata (P7) _diseased	276
Jamun (P5) _diseased	345	Pongamia Pinnata (P7) _healthy	322
Overall Total Leaves = 4590			

Table 2.11: Total Number of Leaf in Each Class

Class	Number of Leaf Images
Cherry_healthy	1000
Cherry_Powdery	1052
Total	**2052**

The Peach dataset consists of healthy and bacterial spot diseases in peach leaves. These images of the leaf have been taken from the PlantVillage dataset [99]. The peach leaf dataset consists of 4457 leaf images separated into two distinct groups: healthy and bacterial spots. The healthy category includes 2160 images of peaches, whereas the diseased category includes 2297 images of peaches infected with bacterial spots.

Strawberry Dataset

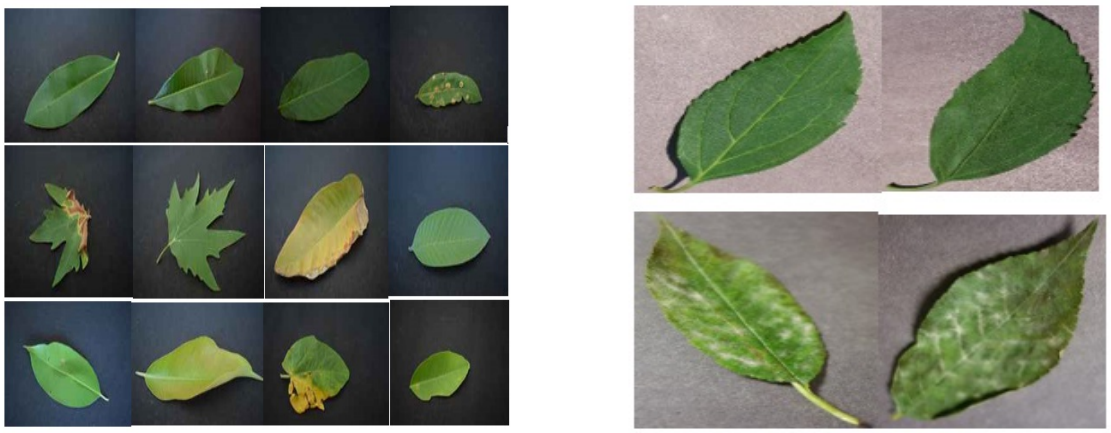
The strawberry leaf dataset consists of 1476 leaf images separated into two distinct groups: healthy and unhealthy leaf_scorch. A sample subset of the Strawberry dataset is shown in Figure 2.4.

Grapes Dataset

The Grapes leaf dataset consists of 4485 leaf images divided into four groups: Grape_Black_rot, Grape_Esca, Grape_healthy, and Grape_Leaf_blight [100]. As demonstrated in Figure 2.4, we chose a set of data of 3 categories of ill grape leaf along with a healthy grape leaf for an overall number of four categories.

Apple Dataset

The Apple leaf dataset comprises 9713 leaf images divided into four groups: Apple Scab, Black Rot, Cedar Apple Rust, and Healthy [101]. As demonstrated in Figure 2.4. We chose



(a) Sample of Healthy and Unhealthy Plant Leaf Dataset (b) Sample of Health and Unhealthy Cherry Leaf Dataset

Figure 2.3: Samples of healthy and unhealthy leaf datasets

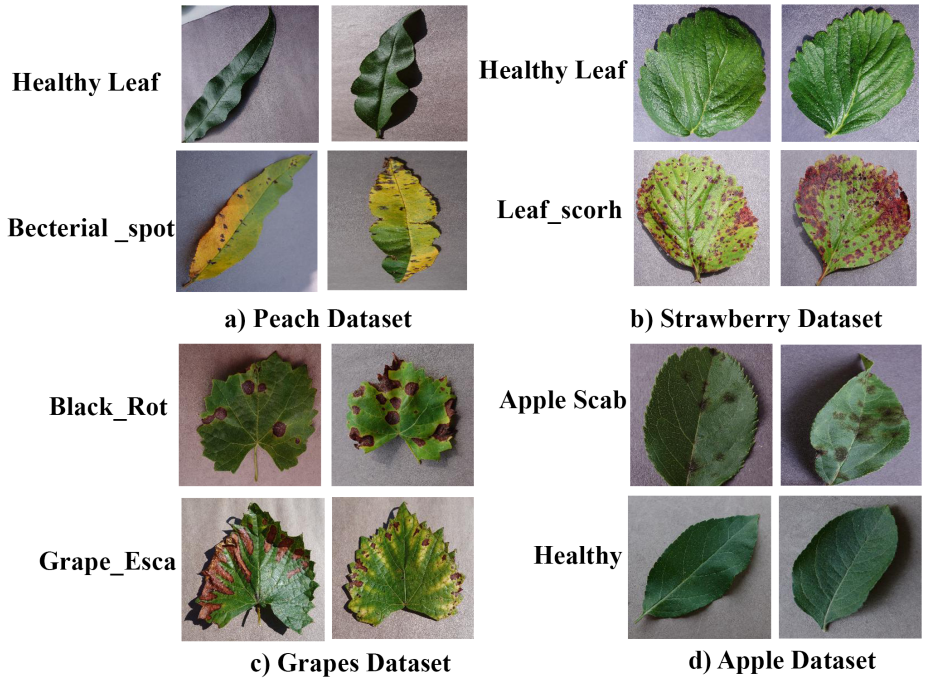


Figure 2.4: Sample images of the Peach, Strawberry, Grapes, and Apple leaf dataset

a set of data of 3 categories of ill apple leaves, along with a healthy apple leaf, for an overall number of four categories.

Tomato Leaf Dataset

This research utilized the Tomato Leaf Illnesses Dataset, which consist of 14529 photos of tomato leaf harmed by ten illnesses. Every class includes target spot, tomato mosaic virus, bacterial spot, tomato spider mites leaf, tomato healthy, early blight, Septoria leaf spot, leaf mould, tomato yellow leaf curl virus, and tomato blight. The dataset is publicaly available

on Kaggle [102]. As seen in Figure 2.5, we selected data from nine groups of unhealthy tomato plants and one healthy tomato leaf for ten groups.

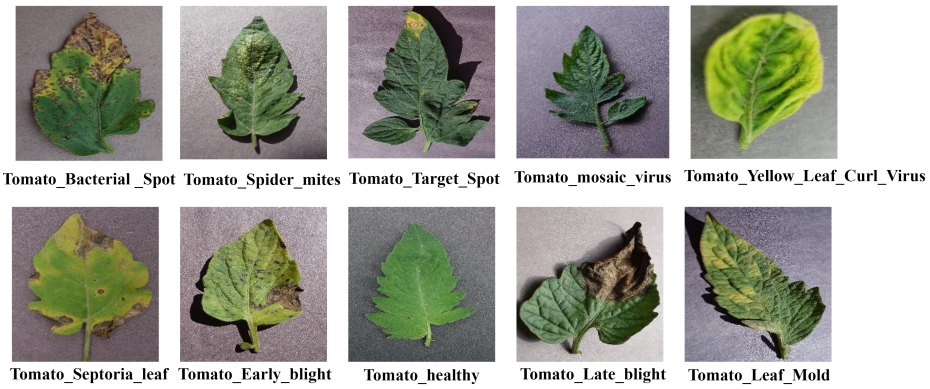


Figure 2.5: Sample images of Tomato Dataset

Maize Leaf Dataset

The maize leaf dataset has total of 4188 images in which 1162 healthy image and 3026 disease leaf images. The maize leaf dataset is available on Kaggle [103]. It contains of 4 classes such as Blight, Gray Leaf Spot, Healthy, Common Rust which is demonstrated in Figure 2.6.

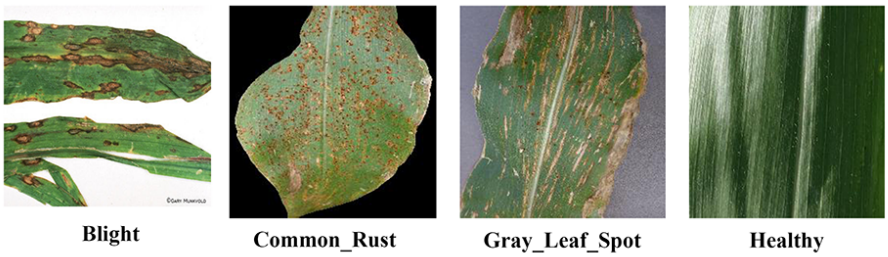


Figure 2.6: Maize or Corn Leaf Dataset

‘

Rice Leaf Dataset

Rice leaf Dataset consist of one Healthy Rice Leaf and eight type of rice disease images which are Sheath Blight, Narrow Brown Leaf Spot, Leaf Blast, Bacterial leaf Blight, Leaf Scald, Brown Spot, Neck_Blast, Rice Hispa. This dataset contains of 11810 images of rice leaf with various disease [104]. Figure 2.7 demonstrated the rice leaf image.

CCMT Dataset: The CCMT dataset [105] consists of 15000 images of 4 crops with 22 classes, five classes of Cashew, 7 classes of Maize, 5 classes of Cassava and 5 classes of Tomato. A sample image of the dataset is demonstrated in Fig. 2.8.

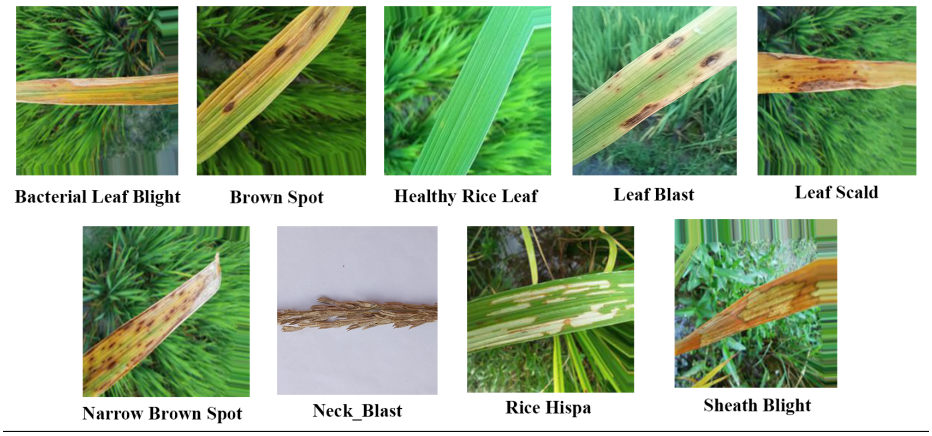


Figure 2.7: Collection of Rice Leaf

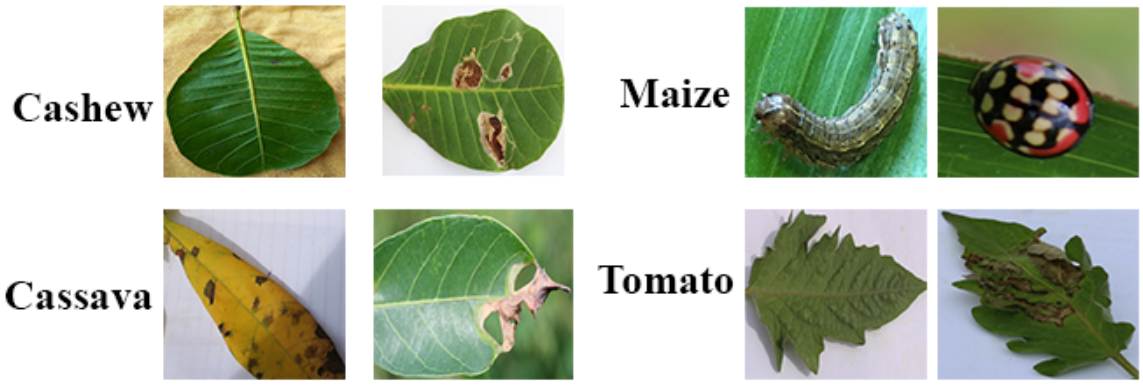


Figure 2.8: Sample images of the CCMT Dataset

Mango Leaf Dataset

The mango leaf dataset has a total of 4000 images, in which each of the eight categories contains 500 images. The mango leaf dataset is available on Kaggle [106]. It consists of 8 classes, such as Anthracnose, Bacterial Canker, Cutting Weevil, Die Back, Gall Midge, Powdery Mildew, Sooty Mould and Healthy, which is demonstrated in Figure 2.9.

Rice Leaf Dataset

The Rice leaf dataset has a total of 2627 images, in which each of the six categories is represented. The Rice leaf dataset is available on Kaggle. It consists of 6 classes, such as Bacterial Leaf Blight, Brown Spot, Healthy, Leaf Blast, Leaf Scald, and Narrow Brown Spot, which are demonstrated in Figure 2.10.

In Table 2.12, the databases contain both healthy and infected crop leaves. The existing datasets have been classified into two groups depending on the number of crops: one crop and several crops. Table 2.12compares datasets in terms of available crops, the quantity of

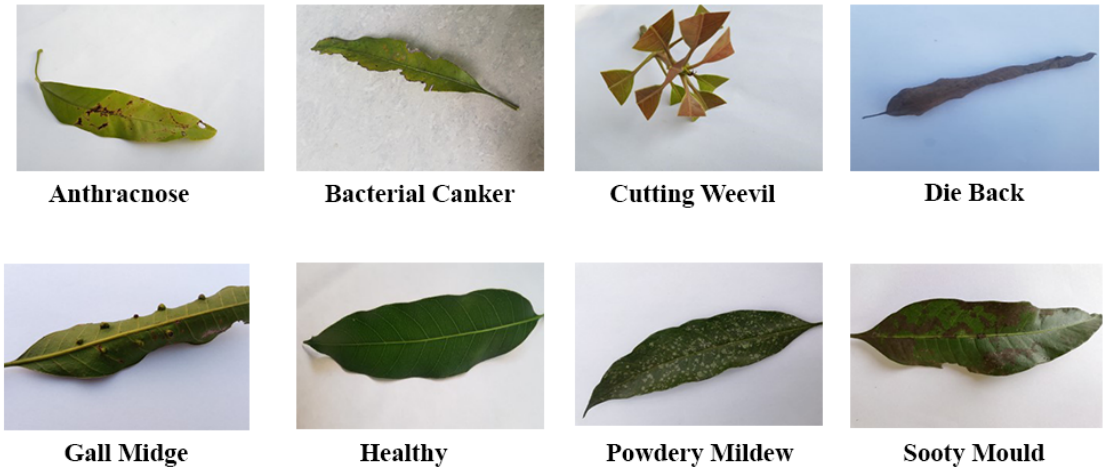


Figure 2.9: Sample image of Mango Leaf

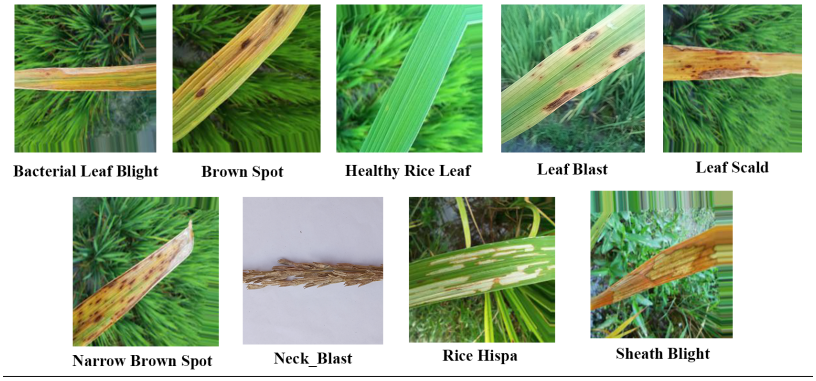


Figure 2.10: Sample image of Rice Leaf

crop images, and the disease coverage of the related dataset. The diversity in datasets is essential for assessing the efficacy and generalizability of AI-driven plant disease detection models. The variations in image quantity, crop types, and disease diversity directly affect the learning capacity of models. Large-size datasets, such as the Plant Village Dataset, which has 54,309 images, provide more training examples, allowing models to understand complicated patterns and generalize more effectively acros plant species and illness instances. Smaller datasets, such as the Wheat Leaf Dataset (407 pictures), may result in overfitting, in which the model performs well on training data but suffers with new, unidentified data. In addition, datasets that cover various crops and disease types, such as the PlantDoc Dataset, improve the model by exposing it to a wide range of properties.

Still, single-crop datasets, such as the Maize Dataset, limit applicability to other plant species. Differences in image quality, resolution, and data-gathering conditions also influence how effectively models manage real-world changes, including illumination, angles, and disease development stages. AI models should be trained on varied, well-balanced datasets

Table 2.12: Analysis of Publicly Available Datasets

Dataset	Year	Crop Available	Number of Images	Disease Covered
Rice Leaf Dataset [107]	2014	Rice	120	Leaf smut, Bacterial leaf blight, Brown spot
Plant Village Dataset [99]	2015	Apple, Corn, Bell Pepper, Orange, Cherry, Potato, Blueberry, Peach, Grape	54,309	Fungal, Bacterial, Mold, Viral
Maize Dataset [108]	2018	Maize	18,222	Northern Leaf Blight
Citrus Fruits and Leaves Dataset [109]	2018	Citrus Leaf	759	Canker, Fungal, Bacterial, Greasy Spot
PlantDoc Dataset [110]	2019	Apple, Blueberry, Cherry, Corn	2,598	Fungal, Bacterial, Mold, Viral
Tomato Leaf Dataset [111]	2020	Tomato	14,531	Late blight, Mold, Bacterial spot, Virus
Wheat Leaf Dataset [112]	2021	Wheat Leaf	407	Viruses, Bacteria, Fungi
Cotton Crop Plant Leaf Dataset [113]	2022	Cotton Crop	2,307	Bacterial Blight, Leaf Spot, Fungal, Virus
Bell Pepper Disease Detection [114]	–	Bell Pepper Leaf	3,901	Fungal, Bacterial

containing numerous plant species, environmental variables, and disease variants to promote generalizability.

2.5 Study and Analysis of Machine and Deep Learning based Existing Techniques for Plant Disease Classification

2.5.1 Overview

Plant diseases pose enormous challenges to world agriculture, resulting in significant crop losses and financial crises. Traditional diagnostic methods are labor intensive and strongly reliant on expert knowledge, making them unsuitable for large-scale use. This work investigates the classification of plant diseases using artificial intelligence (AI) approaches such as machine learning (ML), deep learning (DL), and transfer learning. The study compares approaches based on accuracy, training duration, and computing economy. Special emphasis is given to transfer learning models for their ability to generalize well across diverse datasets while mitigating common challenges like data imbalance, overfitting, and limited annotations. The results indicate that hybrid and deep transfer learning approaches outperform traditional methods in both performance and efficiency. Furthermore, the study outlines critical limitations in current approaches, such as a lack of explainability and deployment constraints on edge devices. The paper concludes with a discussion on future research directions aimed at developing scalable, interpretable, and resource-efficient plant

disease detection systems, ultimately contributing to smarter and more resilient agricultural practices.

Our contributions are summarized below.

1. We analyze machine learning and deep learning-based approaches for identifying plant diseases.
2. This study explores transfer learning methods under consistent and varying performance parameters.
3. We evaluate accuracy and training time to determine the efficiency of the proposed techniques.
4. A comparative experimental analysis is presented for existing plant disease classification methods, highlighting the most effective algorithms for precise leaf disease identification.

2.5.2 Experimental Setup

Our simulations are on a Windows 11 Pro version 22H2 system with an Intel(R) Xeon(R) Silver 4114 CPU @ 2.10GHz, 16GB RAM. We use Python 3.8, Google Colab Pro+ with V100, Keras library, NVIDIA GTX 1070 GPU, and Pytorch to implement the various ML and DL models on different datasets.

2.5.3 Experimental Analysis

This section analyzed multiple transfer learning models using different parameters from our experimental setup and hyperparameter (No. of epochs, Learning Rate, No. of layers, Batch Size).

Table 2.13 presents a detailed comparative analysis of various transfer learning models applied to the Grapes Dataset under different learning rates, illustrating the impact of learning rate changes on model convergence times. In the table, n denotes the number of filters, k^2 represents the kernel size, m^2 is the size of the input, and d is the depth of the input. We selected VGG16, VGG19, Xception, ResNet50, and InceptionV3 for comparison because they are well-known deep-learning models that have already been trained and are good at extracting features. The analysis underscores that deeper models, such as VGG19 and Xception, achieve superior accuracy due to their complex architectures with a higher number

Table 2.13: Analysis Of The Model On Different Parameters

Model	Dataset	No. of Images	No. of Layers	Learning Rate	Accuracy	Training Time (min)	Time Complexity
CNN	Grapes Dataset	4485	8	10^{-4}	94%	~10	$O(n \cdot k^2 \cdot m^2 \cdot d)$
VGG16	Grapes Dataset	4485	16	10^{-3}	98%	~20	$O(n \cdot k^2 \cdot m^2 \cdot d)$
VGG19	Grapes Dataset	4485	19	10^{-4}	96%	~30-80	$O(n \cdot k^2 \cdot m^2 \cdot d)$
InceptionV3	Grapes Dataset	4485	48	10^{-4}	97%	~20-60	$O(n \cdot k^2 \cdot m^2 \cdot d)$
ResNet50	Grapes Dataset	4485	50	10^{-3}	97%	~50	$O(n \cdot k^2 \cdot m^2 \cdot d)$
Xception	Grapes Dataset	4485	71	10^{-4}	96%	~30-70	$O(m^2 \cdot d \cdot k^2 + m^2 \cdot d \cdot p)$

Table 2.14: Analysis Of The Model on the Same Parameters for Grape Dataset

Model	Learning Rate	Accuracy	Training Time (minutes)
CNN	10^{-3}	90%	~10
VGG16	10^{-3}	98%	~20
VGG19	10^{-3}	96%	~10 to 20
InceptionV3	10^{-3}	96%	~40
ResNet50	10^{-3}	97%	~50
Xception	10^{-3}	96%	~30 to 70

of layers. However, these models come with substantial computational requirements, which can limit their feasibility for real-time applications or large-scale deployment scenarios, thus raising questions about their scalability. Conversely, simpler models like CNN demonstrate faster training durations and lower computational costs but achieve relatively lower accuracy, making them more suitable for scenarios where efficiency is prioritized over precision. This study highlights the importance of balancing model performance, computational efficiency, and application-specific requirements when selecting a model.

Table 2.14 compares various transfer learning models with uniform learning rates utilized on the same dataset. As evident, the training time decreases with an increased learning rate. This evaluation examines the accuracy and duration of training for various transfer learning models.

This indicates that deeper models such as VGG16 and ResNet50 attain greater accuracy; however, they necessitate increased training time, which highlights their computational requirements. Simpler models like CNN achieve decent accuracy and facilitate quicker training but may not be suitable for applications that are critical to accuracy. For such applications, advanced models are needed to achieve higher accuracy at the cost of increased computational demand. InceptionV3 and Xception achieve a balance between accuracy and efficiency, though they require significant computational resources. Transfer learning also assists in addressing overfitting by employing pre-trained weights and fine-tuning, as well as addressing data imbalance through approaches such as class weight modification and data augmentation, which improve model generalization.

This table 2.15 compares different machine learning and deep learning models used for

Table 2.15: Comparative Analysis of Models on different Datasets

Model	Dataset	Number of Images	No. of Layers	Learning Rate	Training Time	Time Complexity
SVM Classifier	Grapes Dataset	4485	–	–	13 minutes	$O(n^2 \cdot d)$
KNN	Rice Leaf	11810	–	–	15 minutes	$O(n \cdot d + n \log n)$
CNN	Rice Leaf	11810	5 layers	10^{-3}	15–40 minutes	$O(N \cdot H \cdot W \cdot K^2 \cdot C_{in} \cdot C_{out})$
VGG16	Grapes Leaf	4485	16 layers	10^{-4}	20 minutes	$O(N \cdot H \cdot W \cdot K^2 \cdot C_{in} \cdot C_{out})$
DenseNet201	Potato Leaves	2152	202 layers	10^{-3}	40–70 minutes	$O(N \cdot H \cdot W \cdot K^2 \cdot C_{in} \cdot C_{out})$
VGG16	Potato Leaves	1700	16 layers	10^{-4}	20–40 minutes	$O(N \cdot H \cdot W \cdot K^2 \cdot C_{in} \cdot C_{out})$
HDLCNN	Potato Leaf	920	20 layers	10^{-3}	20 minutes	$O\left(N \cdot \sum(H \cdot W \cdot K^2 \cdot C_{in} \cdot C_{out})\right)$

plant disease categorization. It comprises models such as SVM, KNN, CNN, VGG16, DenseNet201, and HDLCNN, which were tested on datasets such as Grapes, Rice Leaf [115], and Potato Leaf. The amount of pictures, layer count, learning rate, training duration, and theoretical time complexity are all important criteria to consider. Traditional models such as SVM and KNN have simpler designs but are more complex in terms of data size. VGG16 and DenseNet201, for example, provide superior scalability and performance because to organized layer depths and optimal learning rates. HDLCNN uses a hierarchical technique to improve classification. This analysis assists in choosing appropriate models based on computing efficiency and dataset properties.

2.5.4 Result Analysis

This section summarises existing work in plant illness identification and classification using ML and DL-based methods [116]. We have taken some research papers that use ML and DL approaches for comparison examination, as shown in Table 2.16. Additionally, we have run the existing techniques on different datasets in our setup environment, so we got different accuracies based on existing methods. As per Table 2.14, deep learning models like VGG16, DenseNet201, CNN, and Efficient DenseNet perform better than ML-based methods in accuracy.

Plant diseases significantly impact global agriculture and economies, necessitating robust and scalable diagnostic solutions. Artificial intelligence, through machine learning and deep learning approaches, has demonstrated considerable promise in enhancing the accuracy and efficiency of plant disease classification. This study evaluates recent advancements in AI-driven methods, highlighting their strengths and limitations. Transfer learning models showcase potential by addressing challenges such as data imbalance and overfitting. Despite substantial progress, key challenges remain unresolved, including the lack of diverse datasets, segmentation issues, and difficulty distinguishing diseases with similar symptoms. Addressing these gaps requires adopting data augmentation strategies, improved segmentation techniques, and the collection of more comprehensive and diverse plant image datasets.

Table 2.16: Experimental Analysis of Existing Methods for Plant Disease Classification

References	Methods	Dataset	Accuracy
Javidan et al. [117]	SVM classifier	Grapes Dataset	91.30%
Sharma et al. [118]	KNN	Rice Leaf	95.3%
Krishnamoorthy et al. [119]	CNN	Rice Leaf	93.03%
Paymode and Malode [120]	VGG16	Grapes Leaf	98.1%
Mahum et al. [121]	DenseNet201	Potato Leaves	96.03%
Mahum et al. [121]	Efficient DenseNet	Potato Leaves	97.2%
Singh and Yogi [122]	VGG16	Potato Leaves	98%
Kumar and Patel [123]	HDLCNN	Potato Leaf	95.7%

Resolving these limitations will enable the development of highly accurate, scalable, and efficient plant disease detection systems, paving the way for sustainable agricultural innovation.

2.6 Research Gap

Based on the recent literature, the following research gaps are identified:

1. Various state-of-the-art methods have been proposed for plant disease classification, but their performance evaluation is limited to benchmark datasets.

2. In the literature, researchers have suggested various Machine Learning and Deep Learning (DL) models, each having their advantages and limitations. It is observed that most DL-based models are computationally expensive due to high parameter count and face generalisation problem.

3. Deep Learning models often require large number of training samples, and insufficient samples leads to poor performance. However, approaches that can overcome this data requirement while maintaining high performance is underexplored.

2.7 Research Objectives

1. To study and analyze various existing techniques for plant disease classification.

2. To investigate a transfer learning-based approach for plant disease classification.

3. To propose an improved AI based framework for plant disease classification.

2.8 Research Contributions

This thesis examines plant disease classification and identifies research gaps. In this thesis, three research objectives were stated in Section 2.6. We discuss each objective individually below as a contribution to the thesis:

(I) **Objective 1:** *To study and analyze various existing techniques for plant disease classification.*

We propose a new taxonomy for IoT and AI-based plant disease recognition, categorising it based on methodology, application, and technology. Additionally, we conduct a thorough analysis of ML, DL, and IoT in plant disease detection and evaluate publicly available datasets used in this domain. Furthermore, we examine the role of IoT sensors in plant disease recognition systems and discuss key open issues and challenges in IoT and AI-based approaches. Finally, we have suggested some future research directions from AI-based approaches to AI-IoT integration.

(II) **Objective 2:** *To investigate a transfer learning-based approach for plant disease classification.*

This research objective advances plant leaf disease detection by presenting three transfer learning-based models that prioritize accuracy, scalability, and generalization across several datasets and plant species.

Firstly, We Developed modified transfer learning models by combining two drop out layer with different size, linear and relu layer and adaptive average pooling layer. This model is more accurate, scalable in effectively identifying and classifying leaf disease. To prove the efficiency of the modified models in case of unseen datasets, we have tested MobilNetV2, VGG16, InceptionV3 and Xception on tomato and peach dataset. This shows that the modified models have much better accuracy to detect leaf diseases even for the plants not involved in the dataset.Performance was examined using binary and multi-class classification metrics, and the findings outperformed state-of-the-art approaches on the Mendeley and Cherry datasets.

Second, we introduced VESTACK, an efficient and scalable hybrid model that combines VGG16 with Extreme Learning Machine (ELM) to improve illness pattern detection and generalization. Using ELM's quick learning and minimal variable adjustment capabilities, the model achieved high classification results on publically available

datasets such as peach, strawberry, grape, and apple leaves. The model was tested using binary and multi-class classification metrics, and it consistently outperformed numerous cutting-edge methods.

Third, we proposed TLMV2-ELM, a lightweight hybrid model that uses MobileNetV2 and an ELM classifier to classify illnesses in tomato leaves. This model was assessed using a multi-class classification method, which classified ten different disease categories. The model was tested on the PlantVillage tomato dataset, and its performance was compared to other methodologies, revealing good accuracy and computational efficiency.

Collectively, these contributions provide a complete method to plant disease classification based on transfer learning and hybrid neural models, with strong performance across numerous crops, datasets, and classification conditions.

(III) **Objective 3:** *To propose an improved AI based framework for plant disease classification.*

We introduce two deep learning-based algorithms that aim to improve plant disease diagnosis using efficient and accurate image analysis techniques.Firstly, We proposed the "AgriTL-ViT" model to increase a computing efficiency of the Vision Transformer (ViT) model for recognizing illness of plant over image analysis. The model is made up of modified encoder blocks known as Transformer ResNet blocks. Transformer ResNet blocks use ResNet modules rather than MLP modules to extract hierarchical characteristics from input images successfully.To enhance the model's ability to capture both local and global contextual information, we included a dual multi-head self-attention mechanism. This update improves the model's expressive strength and adaptability, allowing it to focus on complex visual characteristics that are important for reliably recognizing disease symptoms.It shows good generalization across many datasets and stays durable under field conditions due to tailored data augmentation algorithms such as ColorJitter and GaussianBlur. Grad-CAM, LIME, and SHAP improve its interpretability, allowing domain specialists to better comprehend the regions that influence forecasts. The model performs consistently even on low-resolution photos, demonstrating its potential for real-world agricultural applications.

Second, we create a lightweight model that uses Asymmetric Dilated Convolution (ADCNN), InceptionResNetV1, and a self-attention mechanism to decrease computational complexity and improve feature learning. This architecture effectively localizes

disease-specific areas while keeping the parameter count minimal. When tested on the CCMT dataset, the model outperforms other cutting-edge approaches, scoring 97.7% accuracy, 97.5% precision, 97.2% recall, and 96.7% F1-score. Together, these models help to produce scalable, interpretable, and high-performance plant disease recognition solutions for both controlled and field settings.

Third, we propose federated learning based framework for plant leaf disease classification in distributed agricultural environments.The framework combines CNN-based feature extraction, Transformer encoders, and federated optimization techniques such as FedAvg and FedProx to ensure accurate, privacy-preserving, and efficient collaborative training across distributed clients.

Based on the findings from the literature study, the following chapter examines various transfer learning approaches used to classify plant leaf diseases, providing a more in-depth look at their design and evaluation.



Chapter 3

Transfer Learning Approaches for Plant Leaf Disease Classification

*The purpose of this chapter * is to provide a transfer learning-based hybrid model of plant leaf disease classification that makes use of pre-trained deep convolutional neural networks and effective classification systems. These structures take advantage of the experience gained from large image data sets and apply it to agricultural disease detection problems, allowing for reliable and robust performance despite a lack of training data. This would improve feature representation while reducing overfitting and computing loads. Such transfer learning models are able to maximize training efficiency and model generalization and strike a balance between accuracy and computation cost. To prove the efficiency of the suggested frameworks on real-world agricultural settings, it is necessary to compare them to state-of-the-art solutions in terms of multiple performance indicators, such as classification accuracy, training time, model complexity, and loss values. This overall assessment shows that transfer learning tactics can be effectively applied to the sound and resource-conscious plant disease classification systems.*

*The part research work covered in this chapter has been published/accepted/communicated in: Meenakshi Srivastava, Jasraj Meena, "Plant Leaf Disease Detection and Classification using Modified Transfer Learning Models," *Multimedia Tools and Applications*, Springer, vol. 83, pp.38411–38441, 2024, DOI: https://doi.org/10.1007/s11042-023-16929-y.
&
Meenakshi Srivastava, Varsha Sisaudia, and Jasraj Meena, "A Hybrid VESTACK Model for Plant Disease Classification Using Deep Transfer Learning Approach," *Computers and Electrical Engineering*, Elsevier, 2025. **(SCIE, IF= 4.9)**

3.1 Plant Leaf Disease Detection and Classification using Modified Transfer Learning Models

3.1.1 Overview

This paper initially proposes a transfer learning based framework for plant leaf disease detection and classification. These five Deep Convolutional Neural Network models (VGG-16, MobileNetV2, Xception, InceptionV3, and DenseNet121) are used to detect and classify plant diseases. Two datasets have been considered; First from Mendeley (plant leaf dataset), having 4590 leaf images split into twenty-two classes. Second from PlantVillage (Cherry Dataset), having 2052 leaf images split into two classes. Further, apply preprocessing techniques such as data augmentation, resizing, and rescaling to mitigate overfitting. Then, testing and training of the proposed models have been on leaf images. To evaluate the quality and quantity of proposed models, a quality matrix was used, and the results show that, for dataset1, MobileNetV2 outperforms other existing models with 98.9% accuracy. For the Cherry Dataset, DenseNet121 outperforms other existing models with 99.9% accuracy.

3.1.2 Related Work

Technology enabled agriculture is one of the best approaches to increase production and feed population of the World. Farmers are annually faced with various challenges caused by pest-related problems, environmental issues, and leaf disease problems, which hinder the quality of the various crops. Due to leaf disease, the plant will not grow properly. So, to overcome this problem, recent technologies has been used such as ML and DL for plant disease detection and classification. In the literature survey, our main focus is to go through a number of research papers for plant disease detection and classification. Some of the researchers (Sujatha, Radhakrishnan, et.al. [124], Samie, et al. [125], Sinha et al. [126]) analyse issues and challenges related to smart agriculture, and they also discuss ML, DL and IoT-based models. According to a review of surveys, there is a lot of effort being made to create this technology, which can be applied to boost plant and livestock productivity and operational effectiveness. As per our detailed studies on this topic, the leaf disease-based research papers have been classified into three main categories: ML with DL, ML with IoT, and DL with IoT, as shown in Fig. 1. In the first category, Garima Garg, et.al [127]

proposed different ML and DL techniques with and without data augmentation. Their proposed application provides a recommendation system that will help the farmer to identify the weather conditions, soil conditions, and disease detection of the field. Further, Ouhami, Maryam, et al. [128] has used the data fusion technique to detect plant disease. Their proposal demonstrates the value of applying intelligent information fusion methodologies to mix data from several sources to improve plant health prediction and outlines the significant hurdles in this field. In addition to this paper, Pham et al. [129] suggested mango leaf illness detection by an ANN model to find micro disease blobs in the mango leaf. In their proposal, out of 114 features, they have considered 81 features to perform a wrapper-based feature selection method for leaf disease. D Stevens, et.al [130] examine the most recent AI methods used in the smart farming system. This paper not only demonstrates the useful application of AI at a different layer of smart farming but also offers potential avenues for further study in this area. The author also suggested that deep learning algorithms outperformed ML due to recent technological advancements that can effectively analyse massive amounts of data and make immediate, intelligent conclusions equivalent to human decisions. However, several ML with DL papers have been proposed to detect plant disease. Still, the primary issues with these techniques are a lack of datasets, insufficient accuracy, and the complexity of ML techniques compared to traditional methods. For better accuracy and analysis on real data, some researchers proposed machine learning with IoT models. Based on the above category, Gayathri Nagasubramanian et al. [131] proposed an ensemble learning-based SVM with IoT for early classification and monitoring of plant disease detection.

Further, Shrivastava, Vimal K, et al. [132] gave a rice plant detection and classification system using a colour feature based on ML models. Out of 172 features, they considered 14 different colour spaces in, which they used 4 characteristics for each. The result shows that it will help farmers enhance the quality & amount of their yield. In addition to this paper, Kumar, Manish, et al. [133] suggested a machine learning methods to develop a system that predicts various fungal infections for plant disease detection. This study shows how sensors can provide insights into numerous abiotic variables that could support the detection of plant illnesses. Although several ML with IoT papers have been proposed to detect plant disease, the main disadvantage of ML with IoT techniques is overfitting problems, fine-tuning issues and Conventional approaches falling short in assessing the disease's severity. In the third category, some researchers like Wei Tan, Jing, et al. [134] give a brand-new CNN-based approach called D-Leaf. In which they worked on a plant species classification system and applied CNN morphological techniques for feature extraction. In 2021, Kundu, Nidhi, et

al. [135] created an automatic data collecting machine and classifier architecture using IoT and several deep learning models. Their study shows that their developed IoT device is affordable and the proposed framework helps farmers in increasing yield productivity and quality of the plants. Furthermore, Gonzalez-Huitron, Victor, et al. [136] suggested a tomato leaf identification system via a microcontroller, i.e., Raspberry Pi, and different IoT sensors. They worked on a leaf classification system and developed a basic IoT-based framework. Mishra et al. [137] presented the rider neural network with the sine-cosine algorithm as a unique illness classifier. The SCA model in this case modifies the ROA algorithm to regulate the location update. For the experiment, the three Internet of Things systems are taken with a total number of nodes such as 50,100,150. The overall work of the classifier is validated using the criteria such as specificity, accuracy and efficiency of nodes on various Internet of Things systems. As a result, the suggested algorithm aids farmers in quickly identifying the infected plants on their property. Further, in addition to the above three categories, the latest research paper has been evaluated using parameters like dataset, classifier, accuracy, and method in Table 3.1. From the literature analysis, we have seen that most of the research papers are applied to datasets, such as Rice, Pepper, Citrus Leaf, and Wheat crops for leaf disease detection. The motivation behind choosing the two datasets is to provide a large collection of images and to give organised data required to train a classification model. The first dataset is from Mendeley (Plant Leaf). This Dataset contains plants like Mango, Guava, Bael, Jamun, Basil, Chinar, etc. They provide a variety of elements that enrich the environment. These plants play a significant role in limiting climate change and carbon emissions. The second dataset that is used for evaluation has two research paper [138] [139], but its dataset size and accuracy are not sufficient for identifying the disease.

So, in this research, a DL method is used to identify plant illness detection.

3.1.3 Preliminary

We have proposed five modified transfer learning models (Vgg16, InceptionV3, MobileNetV2, Xception, DenseNet121) for leaf disease detection and classification by modifying selected layers, as described in the following subsections [148]. For the training of the above models, hyperparameters and a pre-trained ImageNet model have been used [149]. The details of the hyperparameters are shown in Table 3.2. For context, the main ideas behind the five transfer learning architectures are briefly explained.

VGG 16 Architecture:

Table 3.1: Summary of the Latest Literature work

Author	Dataset	Classifier	Method used	Performance Indices utilized
Garg, Garima, et al. [127]	PlantVillage	SoftMax	SRCNN, MobileNet V2 DA, Inception V3 without DA, Inception V3 with DA, ResNet 34	Accuracy
Nagasubramanian, Gayathri, et al. [131]	Own-dataset	SVM	ML- ENSVM, CNN, naïve Bayes, K-nearest neighbors	Precision, Recall, Specificity
Shrivastava, Vimal K, et al. [132]	Rice Plant Dataset	SVM	Machine learning	AUC, F-Score
Sujatha, Radhakrishnan, et al. [124]	Citrus leaf disease dataset	SVM, SoftMax	Machine Learning and Deep learning	Precision, recall, F1 score
Roy, Saurav, et al. [140]	Own Rice Dataset	SVM	Machine Learning	Accuracy
Karthickmanoj, et al. [141]	Own dataset	SVM	Support vector machine	Precision, Recall and F1 score
Ahmad, Mobeen, et al. [142]	Pepper dataset	SoftMax	Transfer Learning	Precision, Recall and F1 score
Khattak, Asad, et al. [143]	Citrus and PlantVillage dataset	SoftMax	CNN	Precision, Recall
Udutalapally, Venkanna, et al. [144]	PlantVillage dataset	SoftMax	CNN	Accuracy
Nalini, Shankaranarayanan, et al. [145]	Own-dataset	SoftMax	Deep neural network	Accuracy, Precision, Recall
Sharma, Ashutosh,et al. [146]	Wheat crop dataset	–	Genetic algorithm, ANN	Sensitivity, Specificity, precision, F-measure
Singh, Uday Pratap, et al. [147]	Mango leaf dataset	SoftMax	MCNN	Classification Accuracy

Table 3.2: Hyper-Parameter Settings

Hyper-parameter	Values
Classes	22
Optimizer	Adam
Epochs	10
Learning Rate	0.01
Batch Size	32

The VGG16 model has multiple applications in different domains, such as image detection and localization, disease diagnosis, etc. So, this model is chosen because it fits in these domains. By using the pre-trained VGG16 model along with two dropout layers at the end to improve the results of our accuracy, which is presented in Figure 3.1. Initially, the size of the image (224*224*3) was passed through 13 convolution layers and five max pooling layers. Where each CNN layer has a filter size of 3*3 and a stride of 1 [150]. When the image passes through a max pooling layer its CNN becomes half of its previous CNN [151]. This could be evaluated by Equation 3.1.1 and also presented in this paper:

$$\left(\frac{m + 2e - g}{\text{Stride}}\right) + 1 \tag{3.1.1}$$

Where, m+2e = image size, g= Max pooling Filter size, Stride = 2

Once the image goes through the above-mentioned process, it is passed through a flattened layer followed by a dense layer and two dropout layers. Further, it will produce a flattened or vectorised image that generates the required output image.

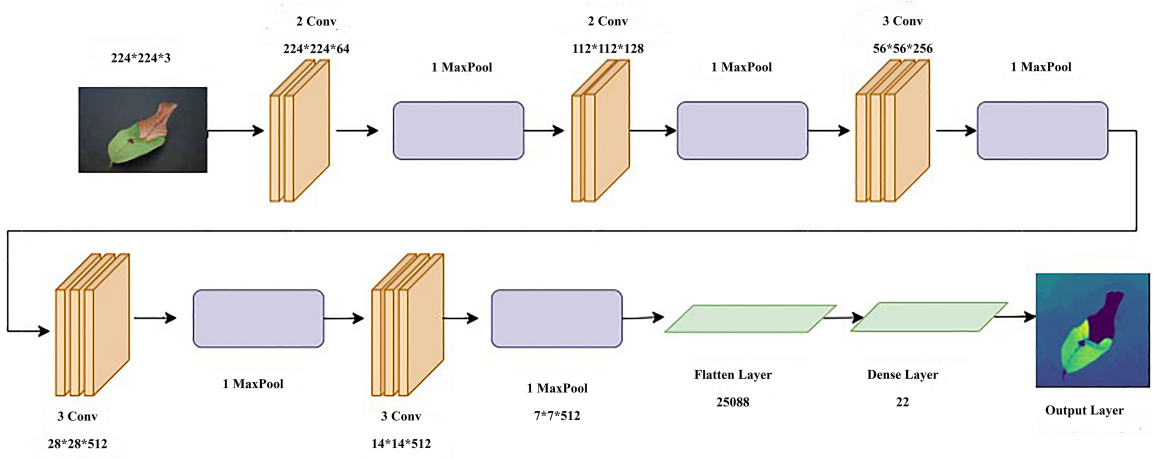


Figure 3.1: VGG 16 Architecture

MobileNetV2:

Here, the MobileNet V2 model effectively reduces the depth of the final image with minimal variation. MobileNetV2 architecture, with two dropout layers and a linear and ReLU layer at the end to enhance accuracy, as shown in Figure 3.2. Initially, the input image size is 224, 224, 3. The input image is passed through 3 convolution layers and 7 Bottleneck layers. When an image passes through a series of bottlenecks, its size changes with the number of layers in each bottleneck [152]. Then the image passes through an average pooling layer, a dense, flattened layer, and two dropout layers to produce the final output image. Finally, it concludes that MobileNetV2 is preferred over VGG16 because it is 30 times smaller and about 10 times faster, with similar accuracy. It is more effective than VGG16 at collecting information.

Xception:

Xception model is mainly preferred for object detection, image classification, etc. By using a pre-trained Xception model along with one average pooling layer, one linear & relu layer, and one dropout layer at the end to increase the accuracy, which is presented in Figure 3.3. Initially, using the Xception model, which effectively reduces the depth of CNN with a lightweight variation to give the final output. Xception is another form of inception block consisting of a 1*1 bottleneck with various kernel sizes, after which it focuses on those

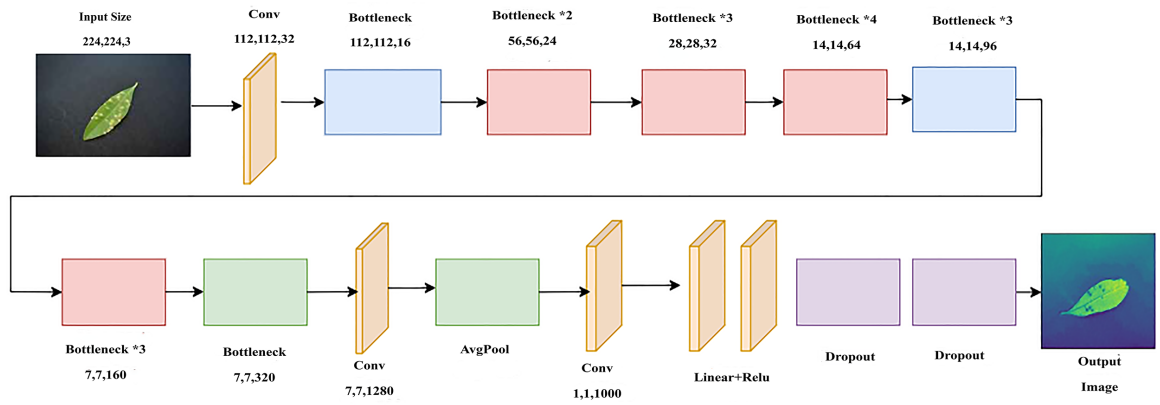


Figure 3.2: MobileNetV2 Architecture

modules whose outputs are carried forward. There are three layers: Input Flow, Middle Flow, and Exit Flow. The input image passes through the input flow, then the middle flow, which is repeated 8 times, and finally the exit flow to produce the output image [153].

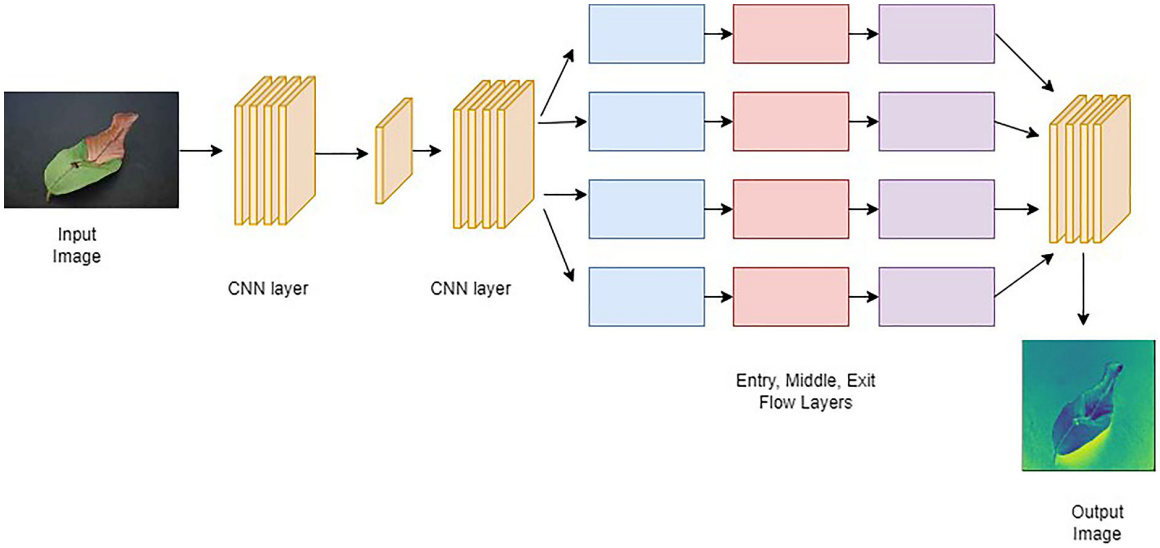


Figure 3.3: Xception Architecture

InceptionV3 Architecture:

InceptionV3 is used only for segmentation, detection, classification, and related tasks. Inception V3 enables us to use multiple filter sizes within a single image block, rather than being limited to a single filter size, which can then be concatenated and passed to the next layer. InceptionV3 model with one average pooling layer and two dropout layers at the end to improve overfitting and accuracy, as shown in Figure 3.4. Initially, the given input image size is 229*229*3. The image passes through 7 CNN layers with a filter size of 3*3. The CNN extracts less critical information from the image and feeds the processed image into the max pooling layer. The Max Pool layer decreased the image dimensions. After that, the

Inception A, B, and C architectures with Reduction A, B Block, Average Pooling layer, and dropout layers have been used. The processed image passes through these parameters to capture the image in parallel [153]. Finally, an auxiliary classifier is applied to the processed image to illustrate the issue of vanishing ingredients.

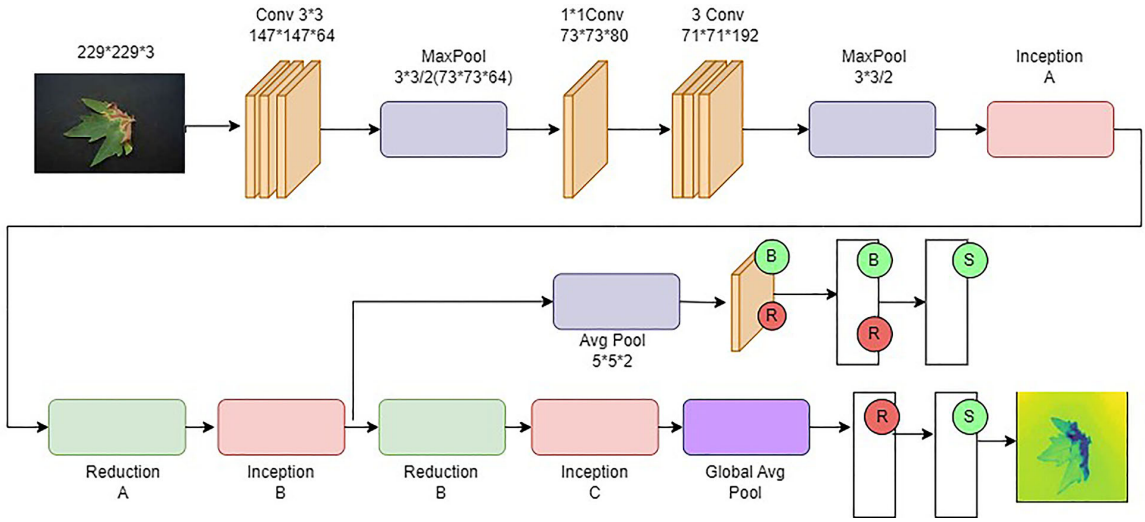


Figure 3.4: InceptionV3 Architecture

DenseNet121:

DenseNet121 has applications across multiple areas, including general classification and image classification. Densenet121 model with two dropout layers, one average pooling layer, and one linear & relu layer at the end to enhance accuracy, as shown in Figure 3.5. Firstly, the image size of 224*224*3 is given as input. The image passes through one CNN layer with a filter size of 7*7 and a stride of 2. The Convolution layer extracted the less important information and passed the processed image to the max pool layer. The main objective of the max pooling layer is to reduce the dimensions of the feature map [154].

In the above architecture, a max pooling layer with dimensions 3×3 and a stride of 2 has been used. After that, the image goes through 4 Dense blocks and 3 Transition layers. The primary function of the transition layer is to convert the feature map size via the CNN and max pooling. After that, the output image passes through the softmax function with a linear & relu layer, flattened layer, dropout and dense layer to produce the final output image.

3.1.4 Modified Framework

In this section, a leaf disease detection and classification framework has been modified to help farmers detect plant diseases. The modified framework is shown in Figure 3.6.

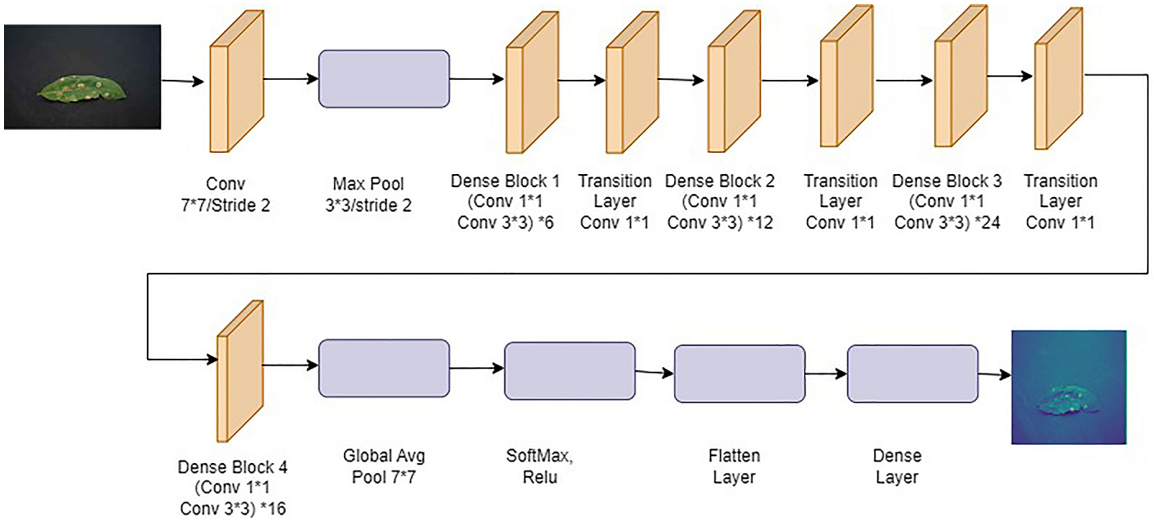


Figure 3.5: DenseNet121 Architecture

This framework primarily focuses on converting ordinary farming into technology-enabled farming that leverages cutting-edge technologies such as DL [155].

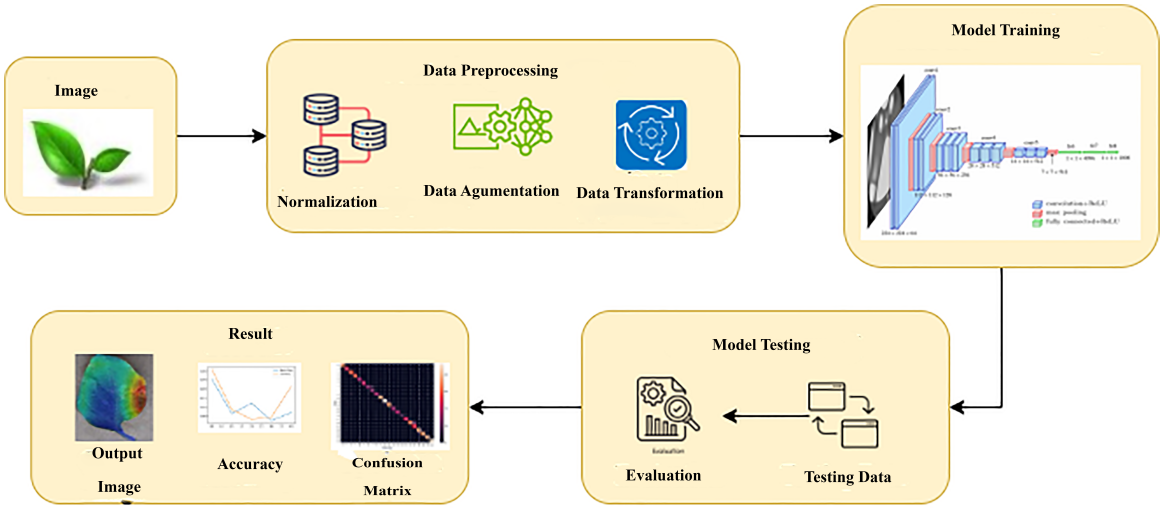


Figure 3.6: Framework of plant leaf disease detection & classification

Initially, an image from the datasets is used as input. In the next step, splitting the dataset into 70% for training and 30% for testing. In addition, data preprocessing techniques such as normalisation, resizing, rotation, and flipping have been applied. By using several augmentation approaches, more training samples are used to improve generalization and reduce overfitting [156]. Overfitting occurs when an ML model performs well on training data but fails to generalise to previously unseen data. Then, the hyperparameters for the training stage are presented, along with a brief overview of the transfer learning architecture used for plant disease detection. Modified transfer learning models have been developed by combining two dropout layers with different sizes, one linear & relu layer and one adaptive

average pooling layer. This model is more accurate and scalable in effectively identifying leaf disease. Then, testing is performed on five transfer learning models for the Mendeley and Cherry datasets. Finally, evaluate the quality criteria of the proposed model (VGG16, MobileNetV2, InceptionV3, Xception, DenseNet121) on both datasets.

In the next section, we detail the experimental setup and methodologies used to evaluate the proposed model's performance.

3.1.5 Experimental Setup

All the experiments have been performed on Windows 10 OS, with 8 GB RAM, and an Intel Core i5 processor. For the implementation of five proposed models (VGG16, MobileNetV2, InceptionV3, Xception, DenseNet121) on both datasets, use Google Colab Pro, Python version 3.8 tool, PyTorch, NVIDIA GTX 1070 GPU, and OpenCV 3.4.2.

3.1.5.1 Dataset Description and Data Augmentation

In this section, five modified deep learning models (Vgg16, InceptionV3, MobileNetV2, Xception, DenseNet121) have been applied. Training for these models is performed on the Mendeley and Cherry datasets, as discussed in Chapter 2, Section 2.4.

This dataset is also split into 70% training and 30% testing to evaluate the modified model's performance. Data augmentation techniques, such as horizontal flips and rotations, have been applied at four distinct angles to avoid overfitting in the models. Image rotation allows the models to be free of the object's orientation. When the image is rotated, the pixel values shift outside the image, leaving blank pixels in the image. So, this black pixel value can be filled in several ways, such as with a constant or the closest pixel value [157].

3.1.6 Experimental Result

In this section, the performance metrics to be applied are first discussed, followed by the outcomes for each model, including classes and classified examples. After this process, the models are evaluated using the performance matrices. Finally, conclude by presenting the overall measurements for each model, which include the descriptions, potential causes, and the performance of best model for Dataset 1 and Dataset 2.

3.1.6.1 Quality Criteria

The accuracy, precision, F1-score, and recall matrices are used to evaluate the performance of each model for leaf illness identification and classification. The definitions of these quality matrices are provided below: Accuracy: Accuracy is the percentage of correctly categorised images of all samples. "The ratio of correct predictions to all predictions made" is called accuracy, and it is shown in Equation 3.1.2.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1.2}$$

Where TP is for true positive, TN is for true negative, FP is for false positive, and FN is for false negative.

Precision: The proportion of accurately categorized positive samples (True Positive) to the whole amount of correctly categorized data is referred as precision. The number of discovered objects that are relevant depends on a model's precision. It is determined through splitting the actual positives from the total positives. Precision is calculated using Equation 3.1.3.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.1.3}$$

Recall: The accuracy of instances that are predicted positively, reflecting how many were accurately identified, is known as recall or sensitivity. Recall is the proportion of all relevant results that our algorithm accurately identified as being relevant. Recall is calculated using Equation 3.1.4 [158].

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.1.4}$$

F1-Score: It is one of the most fundamental evaluation metrics used in machine learning. It simply summarizes a model predictive power through integrating precision and recall, two metrics that would typically be in conflict. F1-Score is calculated using in Equation 3.1.5.

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.1.5}$$

Support: The number of true response samples in each goal-value category can be used to calculate support.

3.1.6.2 Training and Testing Result for Mendeley Dataset

In this section, our obtained results are analyzed from training and testing for the Mendeley Dataset [98]. Initially, split the dataset into two parts, such as 70 % for training and 30% for testing. Then, pre-processing using Data augmentation techniques such as rotation, flipping, etc., is performed to enhance image quality. Further, the pre-processed data is used to train and test the five models (Vgg16, MobileNetV2, Xception, InceptionV3, and DenseNet121). The obtained results are shown in Table 3.3. Further, Table shows that MobileNetV2 has 0.8%, 1%, 2.2%, and 0.4% higher accuracy as compared to VGG16, DenseNet121, Xception and InceptionV3 respectively. It is also identified that MobileNetV2 outperform to other four models.

Table 3.3: Training and Testing Outcomes of Leaf Disease Identification Using Mendeley Dataset

Architecture	Train Acc. (%)	Train Loss	Test Acc. (%)	Test Loss
VGG16	97.90	0.04	98.10	0.03
MobileNetV2	98.60	0.04	98.90	0.02
DenseNet121	97.60	0.15	97.90	0.02
Xception	96.90	0.07	96.70	0.04
InceptionV3	98.40	0.06	98.50	0.02

MobileNetV2 has higher accuracy because, during model training, the training and test accuracies are quite high. It is also more effective than the other four models at collecting information. So that's why MobileNetV2 is preferred over the other four models.

Furthermore, the qualitative results using Saliency maps has been shown in Figure 3.7. It mainly depicts the areas of activation for the trained classes. The main aim of these maps is to conclude whether the model is learning and extracting relevant patterns for the images to be trained [159]. No additional annotation is required for these maps as they are directly extracted using five proposed models on the labels of the images. Also, estimation of computing the saliency maps for a particular image is done for a single class, as well as it mainly requires only one single pass of the backpropagation.

Figure 3.7 shows the testing dataset that exposes different regions of the leaves where the model is activated for their respective classes. It can be identified from Figure 3.7 that the result of MobileNetV2 has better activated image than other models. It also demonstrates that when there are anomalies in the regions of the leaves, there are chances

of higher activation in that area. Two models have a few inconsistencies that lead to the classification error. For distinct class of healthy leaves, it is exhibited that MobileNetV2 usually has higher certainty than other methods, which tends to maximize anomaly score near panels containing petioles with different shapes. For some categories, it is exhibited that MobileNetV2 continues to work vigorously but the other approaches do not classify shades as diseases correctly. It is also verified that MobileNetV2 model can recognize disease areas even in those cases when they are not clear whereas other methods cannot perform accurately to identify diseases.

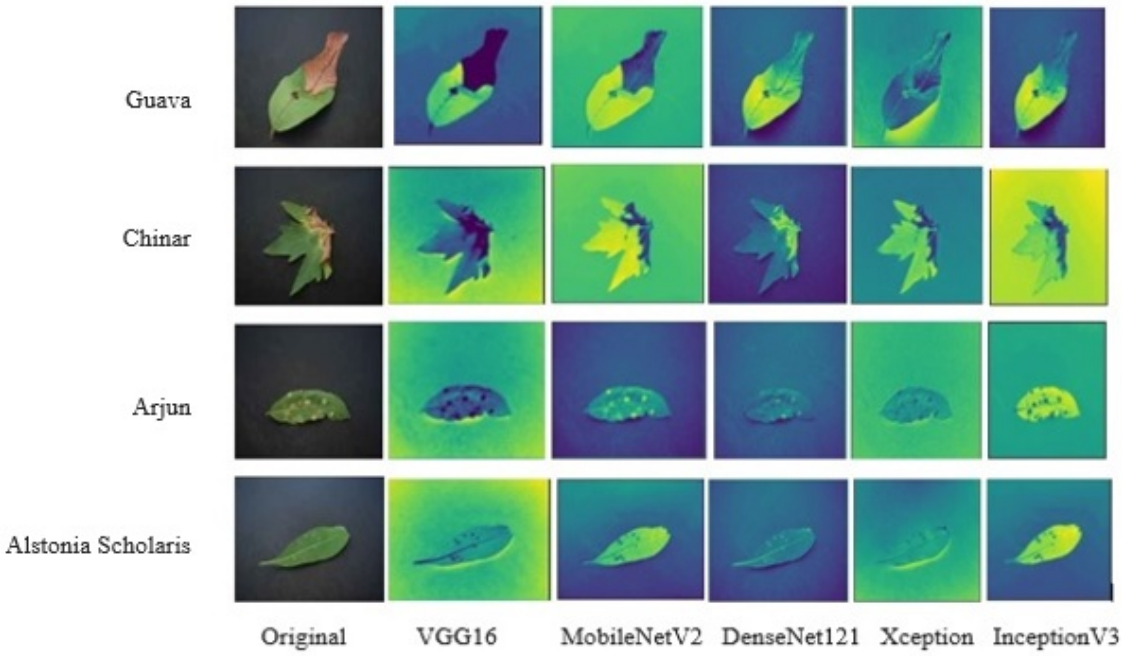


Figure 3.7: Saliency map of Mendeley Dataset

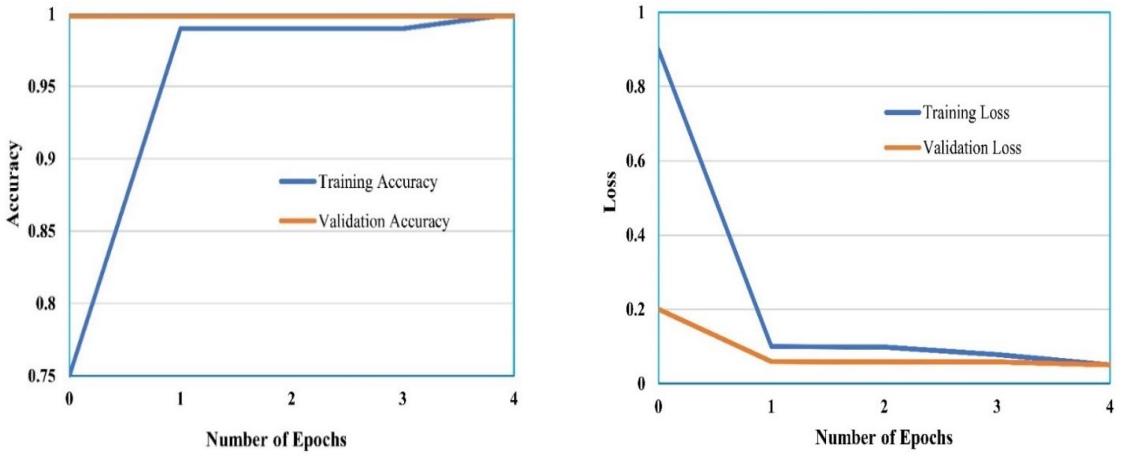
3.1.6.3 Training and Testing Result for Cherry Dataset

In this section, our obtained results from training and testing for the Cherry Dataset [99] are analyzed. Initially, split the dataset into two parts, such as 70 % for training and 30% for testing. Further, pre-processing, training, and testing are done similarly to the Mendeley Dataset discussed in section 3.1.6.2. The obtained results are shown in Table 3.4. Table 3.4 shows that DenseNet121 has the highest test accuracy among the five models (Vgg16, MobileNetV2, Xception, and InceptionV3). However, DenseNet121 has 0.2%, 0.1 %, 0.3%, 0.4% higher accuracy as compared to VGG16, MobileNetV2, Xception and InceptionV3 respectively. DenseNet121 has reduced depth with minimal variation, which is why it performs well compared to the other four models.

Table 3.4: Training and Testing Results of Cherry Dataset

Architecture	Train Acc. (%)	Train Loss	Test Acc. (%)	Test Loss
VGG16	99.50	0.01	99.70	0.01
MobileNetV2	99.40	0.06	99.80	0.01
DenseNet121	99.80	0.02	99.90	0.04
Xception	99.60	0.03	99.60	0.04
InceptionV3	99.70	0.03	99.50	0.03

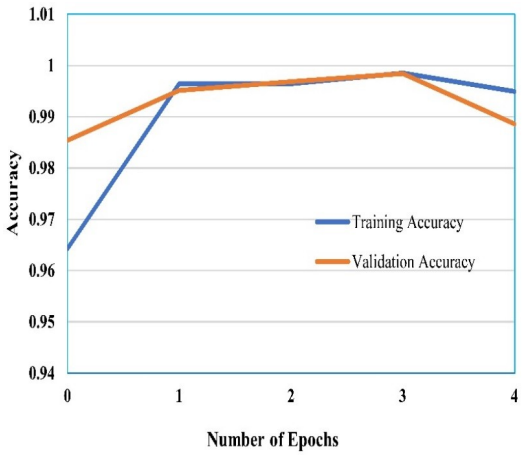
Also, the performance of the five modified transfer learning models is analysed by showing their relationships between accuracies (training and testing) and losses (training and testing), similar to Mendeley Dataset. The obtained results are depicted in Figure 3.8-Figure 3.12. From Figure 3.8(a) to Figure 3.12(a), it is identified that when the number of epochs is less, the accuracy is also less or near zero; whenever the number of epochs increases, the accuracy increases [160]. More variation in testing accuracy has been observed for DenseNet121 than for the other four models. Further, the performance of these models for testing and training loss is shown in Figure 3.8(b) - Figure 3.12(b). The testing loss of DenseNet121 shows greater variation than that of the other four models. Moreover, the qualitative results using Saliency maps are presented in Figure 3.13. Figure 3.13 shows the test data set for the cherry dataset that exposes the different regions of the leaves where the model is activated for their respective classes.



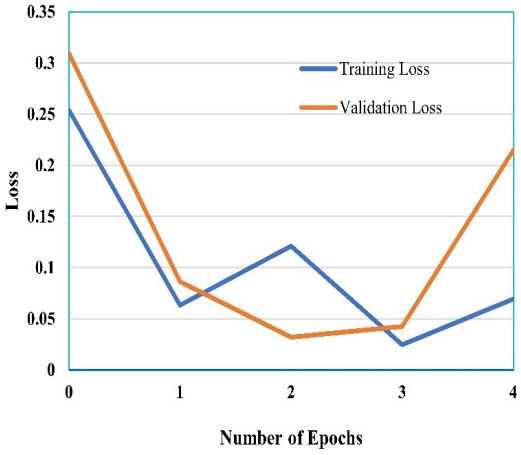
(a) Accuracy of Training and Testing **(b)** Loss of Training and Testing

Figure 3.8: VGG16 training and validation result analysis

The results of the DenseNet121 column have a better-activated image than the four models. This shows that when there are anomalies in the regions of the leaves, there are chances of higher activation in that area as discussed in section 3.1.6.3.

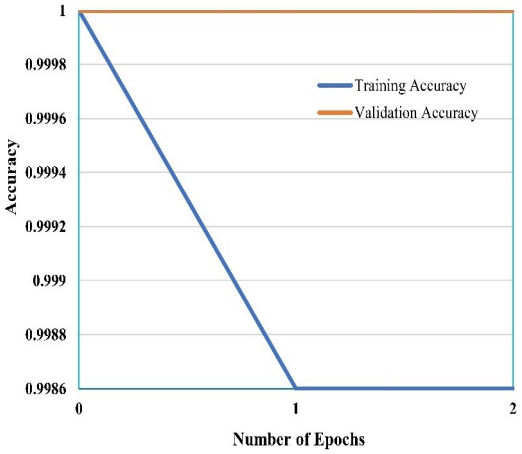


(a) Accuracy of Training and Testing

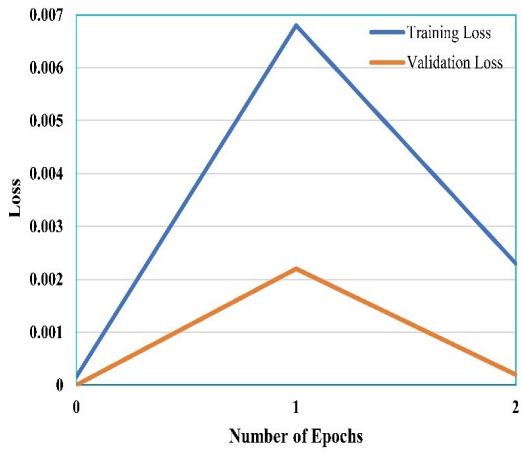


(b) Loss of Training and Testing

Figure 3.9: MobilenetV2 training and validation result analysis



(a) Accuracy of Training and Testing

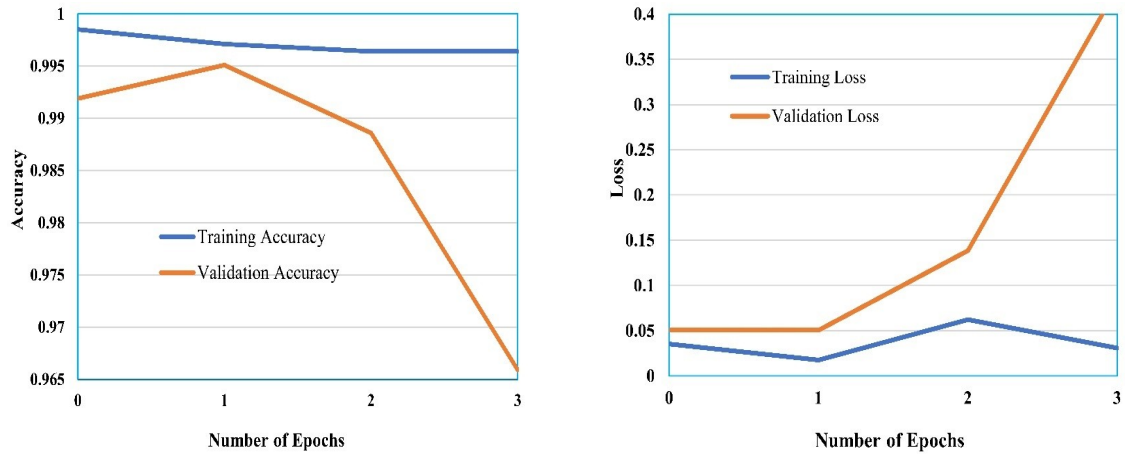


(b) Loss of Training and Testing

Figure 3.10: DenseNet121 training and validation result analysis

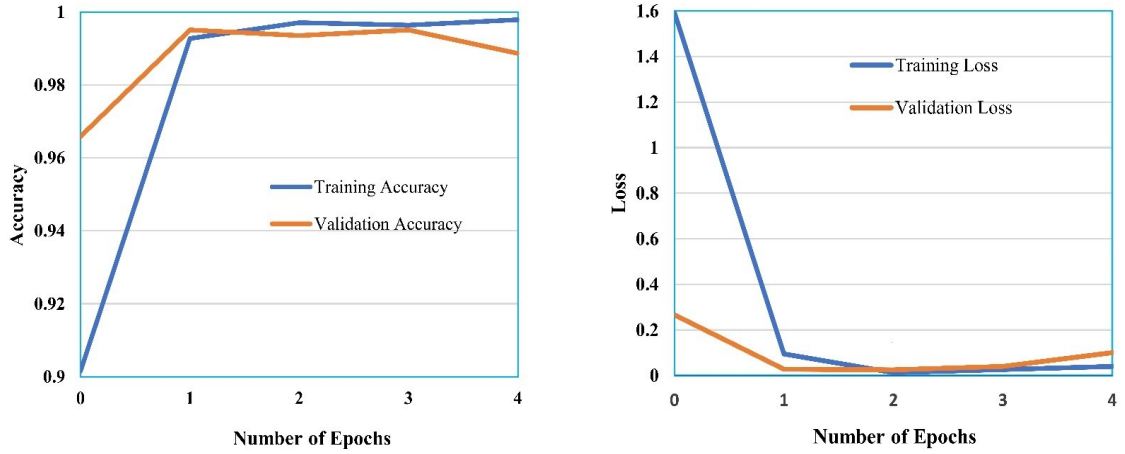
3.1.6.4 Evaluation Results for Mendeley Dataset

Firstly, this section presents the result obtained for performance metrics for each of the twenty-two classes of the Mendeley Dataset [98]. Then, it is probable to study the quality of the classifiers in detail, to detect and classify these labels is a major challenge in the dataset. The quality evaluations and the number of tables being tested for every level are listed in Table 3.5- Table 3.9. Further, Table 3.5-Table 3.9 also shows the value of recall, F1 score, support and precision of all 22 classes for five transfer learning models. In Table 3.5, for Lemon (P10) diseased for the VGG16 model has a higher value of precision and a low recall regarding quality metrics of the 22 class. This implies that the response contained no false positives value and suggests that images from different classes were incorrectly categorized as Lemon (P10) diseased.



(a) Accuracy of Training and Testing **(b)** Loss of Training and Testing

Figure 3.11: Xception training and validation result analysis



(a) Accuracy of Training and Testing **(b)** Loss of Training and Testing

Figure 3.12: InceptionV3 training and validation result analysis

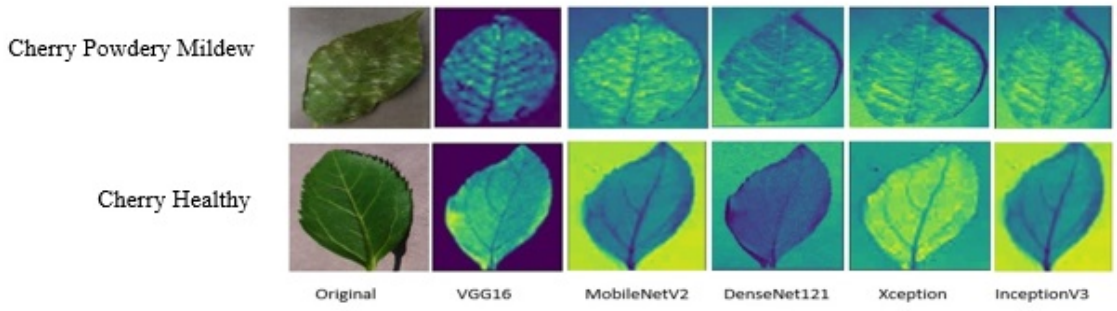


Figure 3.13: Saliency map of Cherry Dataset

A similar response is observed for the Lemon(P10) for the MobileNetV2 model as shown in Table 3.6. Further, for Guava (P3) diseased for MobileNetV2 has high precision and low recall is observed.

Table 3.5: Class-wise Performance Evaluation of VGG16 Model Across Twenty-Two Leaf Categories

Class	Precision	Recall	F1-score	Support
Alstonia Scholaris (P2) Diseased	0.97	0.96	0.96	79
Alstonia Scholaris (P2) Healthy	0.98	0.80	0.92	54
Arjun (P1) Diseased	0.95	0.99	0.97	70
Arjun (P1) Healthy	0.97	0.96	0.96	66
Bael (P4) Diseased	0.94	0.94	0.94	36
Basil (P8) Healthy	0.90	0.98	0.94	45
Chinar (P11) Diseased	0.96	0.95	0.92	36
Chinar (P11) Healthy	0.65	0.65	0.65	31
Guava (P3) Diseased	0.84	0.86	0.85	43
Guava (P3) Healthy	0.91	0.90	0.90	84
Jamun (P5) Diseased	0.92	0.74	0.82	104
Jamun (P5) Healthy	0.96	0.94	0.84	84
Jatropha (P6) Diseased	0.95	0.91	0.93	38
Jatropha (P6) Healthy	0.98	0.94	0.90	40
Lemon (P10) Diseased	0.90	0.38	0.53	24
Lemon (P10) Healthy	0.90	0.88	0.76	48
Mango (P0) Diseased	0.98	0.99	0.98	80
Mango (P0) Healthy	0.98	0.94	0.96	52
Pomegranate (P9) Diseased	0.87	0.87	0.87	82
Pomegranate (P9) Healthy	0.82	0.94	0.88	87
Pongamia Pinnata (P7) Diseased	0.93	0.94	0.93	83
Pongamia Pinnata (P7) Healthy	0.95	0.95	0.95	97

However, a low precision and a higher recall value has been seen for the Lemon(p10) diseased and Lemon (P10) healthy classes for the DenseNet121 as shown in Table 3.7. However, a low F1 score value has been seen for Lemon (10) healthy class for the DenseNet121.

In the Table 3.8. Pomegranate(P9) diseased for the Xception model has higher precision and low recall. In addition, a low F1 score value has been seen for Lemon(P10) diseased for Xception model.

Further, from the Table 3.9 Bael(P4) disease for the InceptionV3 model has high F1 score as compared to the other four models.

It is observed from the evaluation result that for recall and precision of all five modified

Table 3.6: Class-wise Performance Evaluation of MobileNetV2 Model Across Twenty-Two Leaf Categories

Class	Precision	Recall	F1-score	Support
Alstonia Scholaris (P2) Diseased	0.98	0.97	0.98	79
Alstonia Scholaris (P2) Healthy	0.96	0.88	0.92	54
Arjun (P1) Diseased	0.94	1.00	0.97	70
Arjun (P1) Healthy	0.99	0.98	0.95	66
Bael (P4) Diseased	0.94	0.92	0.93	36
Basil (P8) Healthy	0.98	0.93	0.95	45
Chinar (P11) Diseased	0.93	0.69	0.79	36
Chinar (P11) Healthy	0.97	0.87	0.92	31
Guava (P3) Diseased	0.96	0.95	0.95	43
Guava (P3) Healthy	0.96	0.95	0.91	84
Jamun (P5) Diseased	0.95	0.89	0.92	104
Jamun (P5) Healthy	0.90	0.90	0.90	84
Jatropha (P6) Diseased	0.91	0.81	0.86	38
Jatropha (P6) Healthy	0.81	0.92	0.87	40
Lemon (P10) Diseased	0.93	0.58	0.72	24
Lemon (P10) Healthy	0.97	0.96	0.91	48
Mango (P0) Diseased	0.98	1.00	0.88	80
Mango (P0) Healthy	0.98	0.99	0.95	52
Pomegranate (P9) Diseased	0.98	0.95	0.96	82
Pomegranate (P9) Healthy	1.00	0.99	0.99	87
Pongamia Pinnata (P7) Diseased	0.98	0.99	0.99	83
Pongamia Pinnata (P7) Healthy	0.98	0.96	0.96	97

transfer learning models are performed closely to detect and classify healthy and unhealthy leaves for the Mendeley Dataset [98]. The evaluation results also presented by using the confusion matrix of the five proposed DL models. Figure 3.14 shows the Confusion Matrix graph, which makes it easy to classify the twenty-two classes that produce inaccuracies in the trained model. In the confusion matrix, the number of pictures forming either true or false positive is obtained for every class, providing a visual representation of the recall and precision values that were determined for every disease and model. For Jamun(P5) diseased, the MobileNetV2 model performs best because it predicts 96 percent correctly compared to VGG 16, MobileNetV2, Xception, and InceptionV3.

In Table 3.10, all the outcomes are represented for the classifiers one after another along

Table 3.7: Class-wise Performance Evaluation of DenseNet121 Model Across Twenty-Two Leaf Categories

Class	Precision	Recall	F1-score	Support
Alstonia Scholaris (P2) Diseased	1.00	0.82	0.90	79
Alstonia Scholaris (P2) Healthy	0.97	0.93	0.95	54
Arjun (P1) Diseased	0.96	0.97	0.96	70
Arjun (P1) Healthy	1.00	0.39	0.57	66
Bael (P4) Diseased	1.00	0.94	0.92	36
Basil (P8) Healthy	0.97	0.87	0.92	45
Chinar (P11) Diseased	0.96	0.96	0.97	36
Chinar (P11) Healthy	0.92	0.94	0.99	31
Guava (P3) Diseased	0.94	0.95	0.99	43
Guava (P3) Healthy	0.96	0.95	0.89	84
Jamun (P5) Diseased	0.90	0.92	0.91	104
Jamun (P5) Healthy	0.97	0.99	0.97	84
Jatropha (P6) Diseased	0.89	0.82	0.85	38
Jatropha (P6) Healthy	0.94	0.80	0.86	40
Lemon (P10) Diseased	0.97	0.95	0.92	24
Lemon (P10) Healthy	0.98	0.96	0.90	48
Mango (P0) Diseased	0.99	0.99	0.93	80
Mango (P0) Healthy	1.00	0.98	0.99	52
Pomegranate (P9) Diseased	0.97	0.95	0.96	82
Pomegranate (P9) Healthy	0.92	0.98	0.95	87
Pongamia Pinnata (P7) Diseased	0.95	1.00	0.98	83
Pongamia Pinnata (P7) Healthy	0.97	0.97	0.97	97

the training duration need for this phase. The weighted accuracy acquired from the five models is a number that indicates the percentage of positive and negative actual cases that the model adequately classified. Further, Table 3.10 shows that MobileNetV2 has the highest precision, F1 score, accuracy and recall compared to the other four models (Vgg16, DenseNet121, Xception, and InceptionV3). However, MobileNetV2 has 0.8%, 1%, 2.2%, 0.4% higher accuracy, as compared to VGG16, Densenet121, Xception and InceptionV3 respectively. Further, MobileNetV2 has 2%, higher precision, as compared to VGG16, and Xception, respectively. However, it gives similar performance as compared to Densenet121, and InceptionV3. Furthermore, MobileNetV2 has 1% higher recall as compared to VGG16, Densenet121, Xception and InceptionV3 respectively. Finally, MobileNetV2 has 1% higher

Table 3.8: Class-wise Performance Evaluation of Xception Model Across Twenty-Two Leaf Categories

Class	Precision	Recall	F1-score	Support
Alstonia Scholaris (P2) Diseased	0.92	0.92	0.92	79
Alstonia Scholaris (P2) Healthy	0.95	0.93	0.94	54
Arjun (P1) Diseased	0.97	0.96	0.96	70
Arjun (P1) Healthy	0.94	0.91	0.92	66
Bael (P4) Diseased	0.97	0.89	0.93	36
Basil (P8) Healthy	1.00	0.97	0.93	45
Chinar (P11) Diseased	0.97	0.96	0.96	36
Chinar (P11) Healthy	0.94	0.90	0.95	31
Guava (P3) Diseased	0.96	0.95	0.92	43
Guava (P3) Healthy	0.93	0.82	0.87	84
Jamun (P5) Diseased	0.94	0.97	0.90	104
Jamun (P5) Healthy	0.98	0.97	0.97	84
Jatropha (P6) Diseased	0.85	0.87	0.86	38
Jatropha (P6) Healthy	0.85	0.85	0.85	40
Lemon (P10) Diseased	0.91	0.92	0.92	24
Lemon (P10) Healthy	0.94	0.90	0.91	48
Mango (P0) Diseased	0.97	0.99	0.86	80
Mango (P0) Healthy	0.97	0.96	0.95	52
Pomegranate (P9) Diseased	1.00	0.93	0.95	82
Pomegranate (P9) Healthy	0.91	0.90	0.90	87
Pongamia Pinnata (P7) Diseased	0.91	0.96	0.94	83
Pongamia Pinnata (P7) Healthy	0.88	0.98	0.93	97

F1 score as compared to VGG16, Densenet121, Xception and InceptionV3 respectively. So, from the result analysis, it is identified that MobileNetV2 performs best performance as compared to the other four models. MobileNetV2 is mainly preferred over other four models because its 30 time smaller and 10 time faster in term of accuracy. It is more adequate than other four models at collecting information.

3.1.6.5 Evaluation Results for Cherry Dataset

This section presents the performance metrics for each of the two classes of the Cherry Dataset [99]. A detailed analysis of the classifier performance and quality evaluation is conducted, following a methodology similar to that used for the Mendeley Dataset.

Table 3.9: Class-wise Performance Evaluation of Inception-V3

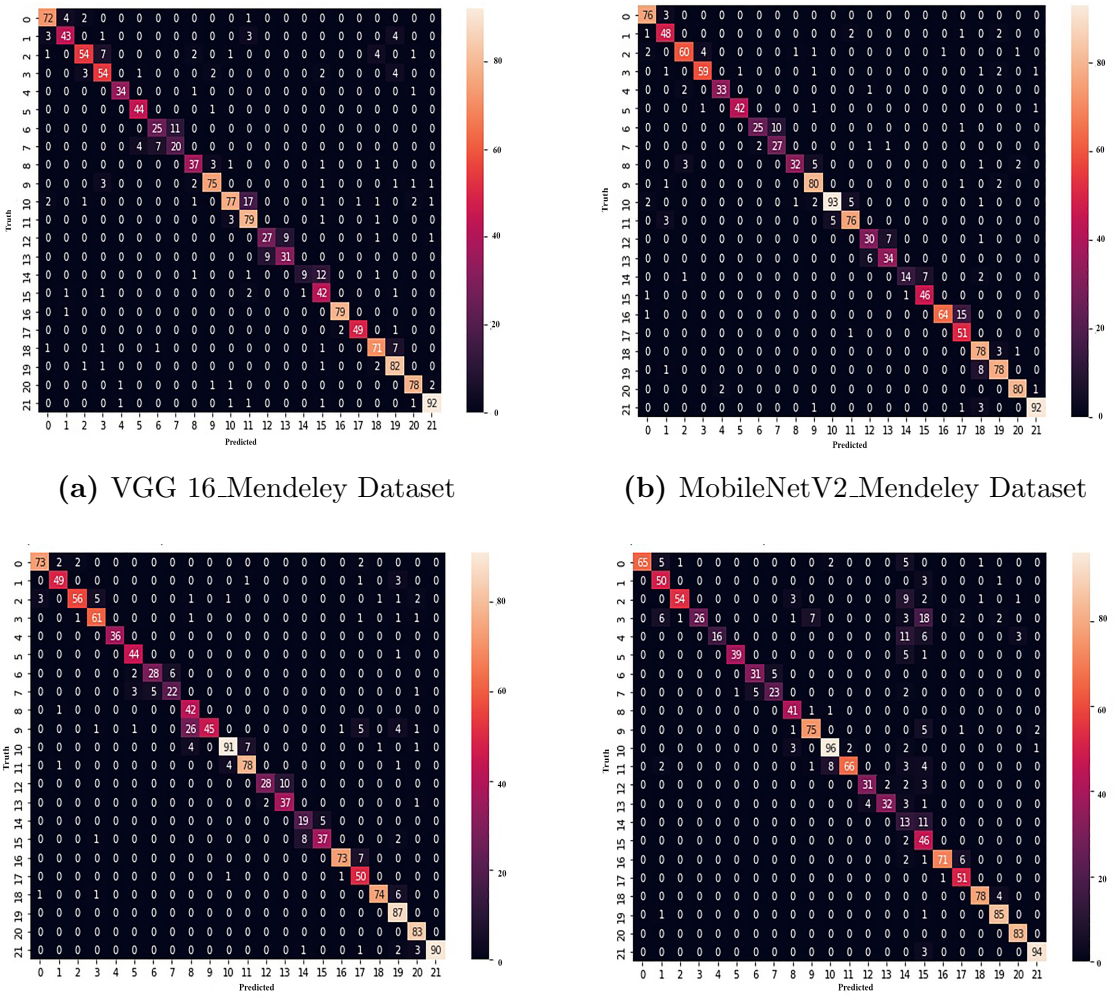
Class	Precision	Recall	F1-score	Support
Alstonia Scholaris (P2) Diseased	0.95	0.92	0.94	79
Alstonia Scholaris (P2) Healthy	0.92	0.91	0.92	54
Arjun (P1) Diseased	0.95	0.90	0.97	70
Arjun (P1) Healthy	0.88	0.92	0.90	66
Bael (P4) Diseased	1.00	1.00	1.00	36
Basil (P8) Healthy	0.98	0.98	0.93	45
Chinar (P11) Diseased	0.85	0.78	0.81	36
Chinar (P11) Healthy	0.93	0.91	0.91	31
Guava (P3) Diseased	0.97	0.98	0.92	43
Guava (P3) Healthy	1.00	0.94	0.90	84
Jamun (P5) Diseased	0.94	0.93	0.91	104
Jamun (P5) Healthy	0.91	0.93	0.92	84
Jatropha (P6) Diseased	0.93	0.94	0.82	38
Jatropha (P6) Healthy	0.79	0.93	0.85	40
Lemon (P10) Diseased	0.90	0.91	0.92	24
Lemon (P10) Healthy	0.93	0.92	0.92	48
Mango (P0) Diseased	0.97	0.91	0.94	80
Mango (P0) Healthy	0.95	0.96	0.96	52
Pomegranate (P9) Diseased	0.97	0.90	0.94	82
Pomegranate (P9) Healthy	0.81	1.00	0.89	87
Pongamia Pinnata (P7) Diseased	0.89	1.00	0.94	83
Pongamia Pinnata (P7) Healthy	1.00	0.93	0.96	97

Table 3.11 presents the values of precision, recall, F1-score, and support for all classes across five transfer learning models.

For the *cherry_powdery_mildew* class, the Xception model exhibits a higher precision value but a lower recall. This indicates that the model produces very few false positives; however, a significant number of true samples are not correctly identified, leading to missed detections. A similar behavior is observed for the *cherry_powdery_mildew* class using the DenseNet121 model, which also demonstrates high precision and low recall.

In contrast, a lower precision and higher recall are observed for the *cherry_healthy* class under the Xception model, indicating that the number of false positives is relatively high for this class. These results highlight the class-wise variation in model performance and emphasize the trade-off between precision and recall across different transfer learning architectures.

It is observed from the evaluation result that for recall and precision of all five transfer learning models are perform closely to detect and classify healthy and unhealthy leaf for the



(a) VGG 16_Mendeley Dataset

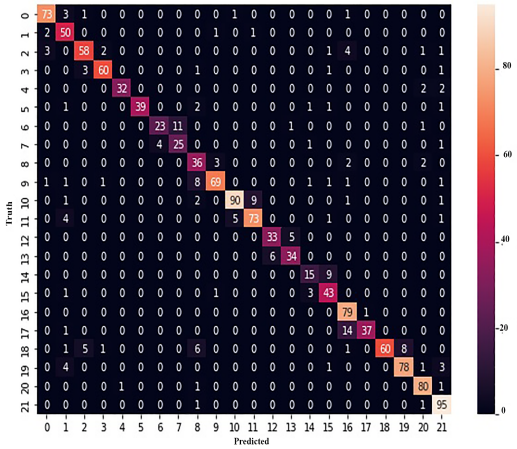
(b) MobileNetV2_Mendeley Dataset

(c) InceptionV3_Mendeley Dataset

(d) DenseNet121_Mendeley Dataset

Cherry dataset [99].

Figure 3.15 shows the Confusion Matrix graph, which makes it easy to classify the two classes that produce inaccuracies in the trained model. In the confusion matrix, the number of pictures forming either true or false positive is obtained for every class, providing a visual representation of the recall and precision values that were determined for every disease and model. For Cherry dataset, the MobileNetV2 model performs best because it predicts 99 percent correctly compared to VGG 16, MobileNetV2, Xception, and InceptionV3. In Table 3.12, all the outcomes are represented for the classifiers one after another along the training duration required for this generation. The weighted accuracy acquired from the five models is a number that indicates the percentage of positive and negative actual cases that the model adequately classified [161]. Further, Table 3.12 shows that MobileNetV2 has the highest precision, F1 score, accuracy and recall compared to the other four models (Vgg16, DenseNet121, Xception, and InceptionV3). However, MobileNetV2 has 0.1%, 0.1%, 0.2%, 0.3% higher accuracy, 3%, 2%,1%, 1% higher precision, 2%, 1%, 1%, 1% higher recall



(e) Xception_Mendeley Dataset

Figure 3.14: Confusion matrix chart with mendeley dataset for five cnn model

Table 3.10: Weighted Performance Metrics on the Mendeley Dataset

Architecture	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
VGG16	98.10	95.00	96.00	95.00
MobileNetV2	98.90	97.00	97.00	96.00
DenseNet121	97.00	96.00	96.00	96.00
Xception	96.90	95.00	96.00	96.00
InceptionV3	98.40	97.00	96.00	95.00

and 3%,1%, 2%, 1% higher F1 score as compared to VGG16, Densenet121, Xception and InceptionV3 respectively. MobileNetV2 has much fewer parameters and lower processing complexity while maintaining competitive accuracy. So, MobileNetV2 performs very well as compared to the other four models.

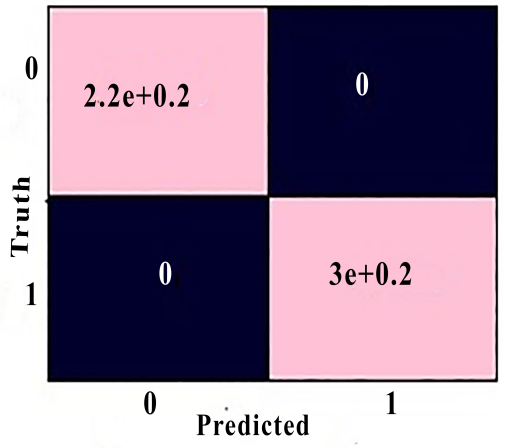
To prove the efficiency of the proposed work in case of unseen datasets, MobilNetV2, VGG16, InceptionV3 and Xception models were tested on tomato and peach dataset [99], for which no training was provided. The model is not overfitted as it performs leaf disease detection with good accuracy on new dataset as well. This proves generalization of the model to detect leaf diseases for plants not included in the dataset. The result for the same has been tabulated in the Table 3.13.

3.1.6.6 Comparison with the State-of-the-Art Techniques

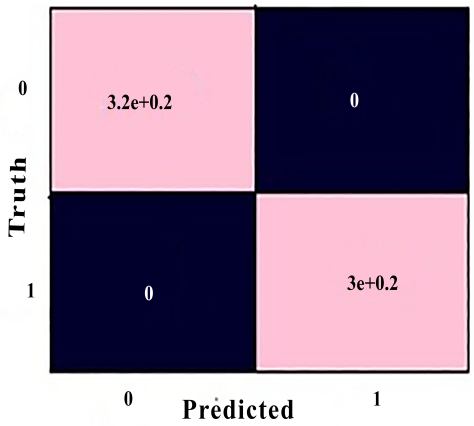
This section compares the proposed transfer learning model to different cutting-edge strategies for identifying and classifying leaf disease using the Mendeley and Cherry Datasets. Table 3.14 and Table 3.15 display the performance results for existing techniques([147] , [162]- [163]) over accuracy. In terms of overall accuracy, the modified TL models outper-

Table 3.11: Confusion Matrix–Based Performance Metrics on the Cherry Dataset for Five CNN Models

Model	Class	Precision	Recall	F1-score	Support
VGG16	Cherry_healthy	1.00	0.99	0.99	316
	Cherry_powdery_mildew	0.99	1.00	0.99	300
MobileNetV2	Cherry_healthy	1.00	0.99	0.99	316
	Cherry_powdery_mildew	0.99	1.00	0.99	300
DenseNet121	Cherry_healthy	0.90	1.00	0.95	316
	Cherry_powdery_mildew	1.00	0.89	0.94	300
Xception	Cherry_healthy	0.76	1.00	0.86	316
	Cherry_powdery_mildew	1.00	0.66	0.80	300
InceptionV3	Cherry_healthy	0.86	1.00	0.93	316
	Cherry_powdery_mildew	1.00	0.83	0.91	300



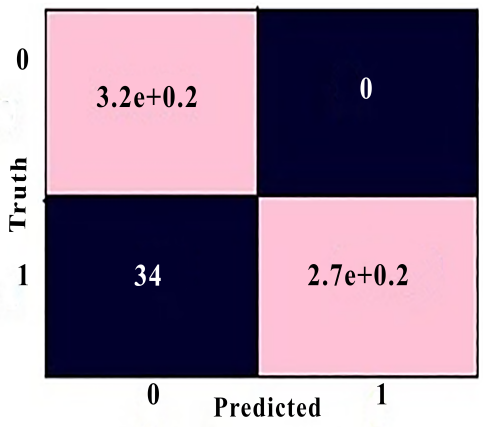
(a) Vgg16_Cherry Dataset



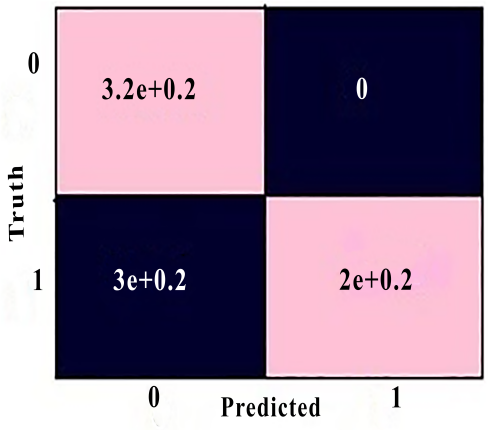
(b) MobileNetV2_Cherry Dataset

form other models for the classification of Mendeley and Cherry datasets. One significant benefit of modified TL models is that it reduces the difficulty of intended notion by utilising a buried node section to aid in classification and identification of anomalous issues.The results shows that for Mendeley Dataset, our modeified TL models have highest accuracy 98.9% which is 1.8-8.8 % better than the other cutting-edge technologies.For the Cherry Dataset, our modeified TL models have highest accuracy 99.9% which is 0.3-0.9 % better than the other existing technologies. Because it remove overfitting and accuracy problem to identify disease in leaf.

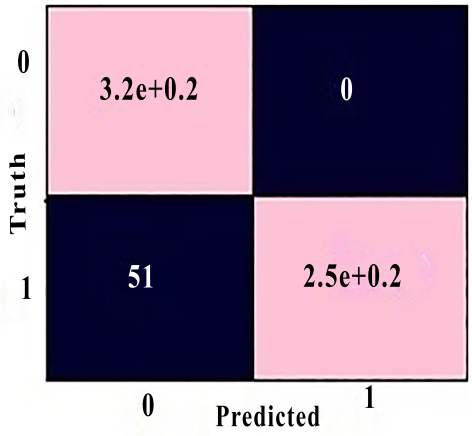
Deep learning enabled agriculture is one of the critical demands of the modern era, which increases farming productivity, sustainability, and profitability. Every year farmers face



(c) DenseNet121_Cherry Dataset



(d) Xception_Cherry Dataset



(e) InceptionV3_ Cherry Dataset

Figure 3.15: Confusion matrix chart with cherry dataset for five cnn model

many difficulties like pest-related problems, environmental issues, and leaf disease problems in producing a wide range of crops. So, to overcome the leaf disease problem, plant disease detection and classification framework have been proposed. Then, apply five deep CNN models (Vgg16, MobileNetV2, Xception, InceptionV3, and DenseNet121) to detect and classify plant diseases. Furthermore, the two datasets are considered; First, Mendeley (plant leaf dataset), which have 4590 leaf images split into twenty-two classes. Second from PlantVillage (Cherry Dataset), which have 2052 leaf images split into two classes. In our experiment, firstly applied pre-processing methods like data augmentation (resizing, rescaling) to remove the problem of overfitting. Then the testing and training of the five proposed models on leaf images was performed. Quality metrics have been used to calculate the quality and quantity of proposed models. The result shows that for dataset1 (Mendeley Dataset), MobileNetV2 has the highest testing accuracy with 98.9%, which is 0.8-0.4% higher than the other four models (Vgg16, Xception, InceptionV3, and DenseNet121). How-

Table 3.12: Results of Weighted Quality Metrics for the Cherry Dataset

Architecture	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
VGG16	99.70	95.00	96.00	96.00
MobileNetV2	99.80	98.00	98.00	99.00
DenseNet121	99.70	96.00	97.00	98.00
Xception	99.60	97.00	97.00	97.00
InceptionV3	99.50	97.00	97.00	98.00

Table 3.13: Accuracy of Proposed Transfer Learning Models on Unseen Datasets

Model	Tomato Dataset Accuracy (%)	Peach Dataset Accuracy (%)
MobileNetV2	97.80	99.20
VGG16	95.00	98.20
InceptionV3	98.20	98.30
Xception	95.80	98.90

ever, MobileNetV2 performs 0.8-0.4% better than the four models for weighted accuracy. For the Cherry Dataset, DenseNet121 has the highest testing accuracy with 99.9%, which is 0.2-0.4 % higher than the other four models (Vgg16, MobileNetV2, Xception, and InceptionV3) and MobileNetV2 performs 0.1-0.4% higher than the other four transfer learning models for weighted accuracy. In Future will focus on the enhancement of training dataset with some additional test images. Further, we will also expand the system's capabilities and add pest detecting tools.

3.2 A Hybrid VESTACK Model for Plant Disease Classification Using Deep Transfer Learning Approach

3.2.1 Overview

Early plant disease detection prevents production losses while maintaining agricultural yield and quality. Farmers have traditionally relied on manual techniques to detect leaf diseases, which is quite time-consuming [166]. Their manual methods need much labour, increasing overall yield production/plant disease detection costs and presenting usability issues.

To address the above issue, machine-learning and deep learning-based solutions have been proposed in the literature for plant disease detection and classification [167]. Most of the

Table 3.14: Comparison of the Modified Transfer Learning Models with State-of-the-Art Techniques on the Mendeley Dataset

Method	Accuracy (%)
MCNN [147]	97.10
Jha *et al.* [162]	96.30
Russel *et al.* [164]	90.02
VGG16	98.10
MobileNetV2	98.90
DenseNet121	97.90
Xception	96.70
InceptionV3	98.50

Table 3.15: Comparison of the Modified Transfer Learning Models with State-of-the-Art Techniques on the Cherry Dataset

Method	Accuracy (%)
Zhang *et al.* [165]	99.60
Sengar *et al.* [163]	99.00
VGG16	99.70
MobileNetV2	99.80
DenseNet121	**99.90**
Xception	99.60
InceptionV3	99.50

techniques in the literature use a similar process to detect and classify plant disease, as presented in Figure 3.16. The initial step is identifying plants that might have been infected with the illness. After identifying plants, images of these infected plants must be captured using unmanned aerial vehicles (UAVs), mobile devices, and photographic tools. In the second step, various pre-processing approaches, such as image cropping, smoothing, enhancement, etc., are used to reduce noise from images. In the next step, the different features of the plant were retrieved, which is crucial for determining the item. After feature extraction, the processed dataset images are categorised using a neural network.

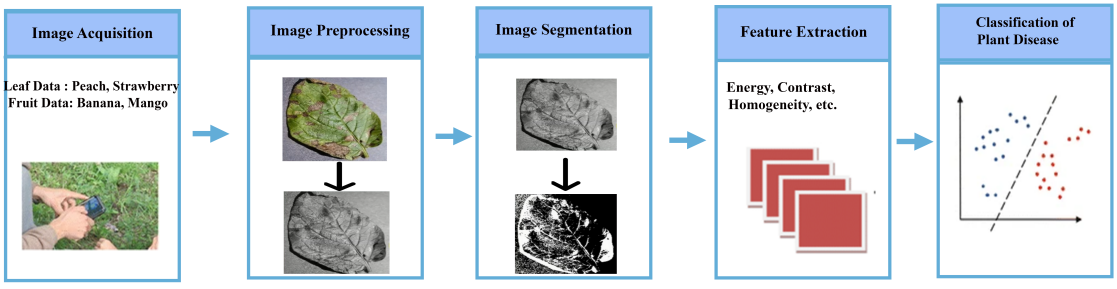


Figure 3.16: Process of disease identification

In the literature, numerous researchers have employed machine learning methods like Support Vector Machines (SVM), Extreme Learning Machines (ELM), etc, to identify plant diseases. ELM is a developing concept for a single-layer feed-forward networks (SLFNs) that exhibit greater generalisation efficiency and quicker learning than gradient-based strategies, and calculate hidden node parameters without human interaction [168]. However, these approaches do not address the various challenges farmers face, such as monitoring soil health, diagnosing leaf diseases, and detecting weeds, among others, in the current context [169]. Due to its ability to address various challenges and achieve greater accuracy, deep learning has recently gained popularity over traditional machine learning methods for detecting and classifying plant diseases. Deep learning methods are very useful for identifying plant diseases because they automatically extract features from raw data, enabling more accurate detection of complex patterns. They successfully handle diverse datasets, enabling robust performance across varying conditions and real-time detection via an IoT connection [170]. The core disadvantage of deep learning approaches is their long training times, typically much longer than machine learning methods [171].

In addressing the challenges of complexity and overfitting, we introduce the VESTACK model, which integrates deep transfer learning utilising VGG16 alongside an Extreme Learning Machine (ELM). This hybrid model aims to enhance generalisation and efficiency in recognising illnesses in plant leaves. Using ELM's rapid learning abilities and limited parameter modifications, the VESTACK model addresses the limitations of earlier methods, leading to enhanced agricultural output [172].

We employ a VGG16 model grounded in transfer learning to detect patterns of leaf illnesses, subsequently conducting various feature set evaluations and utilizing extreme learning machine methodologies. Below is a summary of our contributions:

1. Developing an efficient and scalable transfer learning-enabled hybrid model for identifying and classifying leaf disease.

2. We propose a VESTACK model incorporating VGG16 and ELM with improved generalisation capacity to identify disease patterns in a leaf because ELM inherits fundamental qualities such as minimum variable adjustment and quick learning.

3. The model performance is evaluated using a binary and multi-class classification technique that examines numerous metrics to measure its utility in distinguishing between the multi-classes.

4. The suggested model is assessed and tested using the publicly available Peach, Strawberry, Grapes, and Apple leaf datasets and compared with cutting-edge techniques based on different parameters.

3.2.2 Related Work

This section focuses on different AI and ML based solutions proposed in the literature to identify plant diseases. Initially, we analysed the literature based on various methodologies and identified the key challenges faced while detecting plant disease. Finally, we analyse their research based on the datasets and different parameters.

Le et al. [173] created an approach for detecting and classifying crop and plant diseases. Morphological procedures were used to eliminate noise from the plan images. Then, disease regions and features were extracted using the modified regional digital pattern methodology with outline masque and the coefficient k method. Based on experimental data, the k-FLBPCM technique achieved the highest accuracy of 98.63% in identifying morphologically related plants. Bhatia A. et al. [174] used the Extreme Learning Machine (ELM) algorithm to predict plant diseases on the Tomato Powdery Mildew Disease (TPMD) dataset. The authors employed a number of resampling techniques, including IMPS, SMOTE, RUS, and ROS, to address the imbalance in the dataset. The models were trained using the original and balanced datasets, and Classification Accuracy (CA) and Area Under the Curve (AUC) were used to gauge how effective the models were. With an AUC of 88.57% and a classification accuracy of 89.19%, IMPS fared better than the other resampling techniques, proving its usefulness in handling unbalanced datasets.

Abbas A. et al. [175] utilized a data-driven simulation approach to generate artificial images of tomato plant leaves for disease identification. A DenseNet-based classification model was trained using transfer learning (TL) on simulated and real-world images to categorize tomato leaves based on disease type. Aswinikumar S. et al. [176] introduced OMNCNN, an automated method for identifying and classifying plant leaf diseases using advanced computational techniques. The approach involves four steps: image enhancement through bilateral filtering, segmentation using Kapur's thresholding, feature extraction with a MobileNet-based framework optimized by the Emperor Penguin Optimizer (EPO), and classification via an Extreme Learning Machine (ELM). Multiple experiments demonstrated that the OMNCNN model outperformed existing techniques, achieving high precision (0.985), recall (0.9892), accuracy (0.987), F-score (0.985), and kappa (0.985). This study highlights the

effectiveness of automated methodologies in plant disease diagnosis, showcasing improved accuracy and reliability.

Sood S et al. [177] suggested Convolutional Neural Networks (CNNs) to identify grape leaf illnesses in their early stages. The model was trained and evaluated using the publicly available PlantVillage dataset, including transfer learning with pre-trained architectures like VGG16, ResNet50, InceptionV3, and DenseNet121 for comparison. Gaussian noise features were used in the evaluation process to overcome the problem of overfitting. The suggested model outperformed the existing transfer learning models, with an accuracy of 99.88%. The work emphasises the importance of Gaussian noise in improving the performance of CNN for grape leaf illness identification.

Kini A. et al. [178]proposed an advanced pattern recognition method incorporating optimized classification models for disease prediction in black pepper leaves. A newly curated dataset of black pepper plants was developed under expert supervision, covering diseases such as yellowing, phytophthora, slow wilt, and anthracnose. Multiple computational models, including Inception V3, GoogleNet, SqueezeNet, and ResNet18, were fine-tuned, achieving classification accuracies between 99.1% and 99.7%. Among these, the ResNet18 model achieved the highest accuracy of 99.67%, outperforming others. This study presents an innovative application of deep learning for the early recognition of illnesses, contributing to progress in smart farming. Khan I et al. [179] suggested a transfer learning approach to recognise maize leaf disease to overcome the strength and generalizability limits of the image processing approach. Furthermore, ResNet50 had the highest performance, with 0.87% validation accuracy, 0.90% precision, and 0.99% recall. Transfer learning has the potential to address nuanced plant issues, as evidenced by the innovative maize disease detection technique presented in this research. Rezaei et al. [180] introduced a deep learning framework that goes beyond standard disease recognition to perform a variety of plant disease analysis tasks, including annotation, lesion segmentation, and severity assessment. The study demonstrated successful disease severity classification with a high accuracy of more than 82 percent by introducing vision transformers, lightweight salient object detector models, and new ABCD inspired morphological features concatenated with texture descriptors, pointing to a new direction for assessing interpretable and fine-grained plant diseases.

Chug et al. [181] introduced an advanced classification approach for plant disease detection. This study utilized five classification models and eight versions of an optimized feature extraction framework (B0–B7). Additionally, two publicly available datasets were used for

training and evaluation. The proposed method enables farmers to identify and address plant diseases efficiently, thereby reducing significant crop losses. The study also highlighted that many existing models overlook challenges related to overfitting and generalisation.

Mohammadi et al. [182] proposed a leaf classification approach inspired by botanical characteristics, utilising an automated pattern recognition system to identify plant species. Three models were developed: one designed from the ground up and another based on an optimised feature extraction framework. The method was evaluated using the Flavia and MalayaKew datasets, achieving high classification accuracy of 99.67% and 99.81%, surpassing traditional feature-based techniques. This approach eliminates manual feature extraction and enhances efficiency using a streamlined structure with fewer parameters. The proposed SWP-LeafNET model also operates faster and remains accessible, making it a reliable and practical solution for plant species identification.

Aqel D et al. [183]introduced an automated plant disease detection and classification technique that incorporates segmentation, texture-based feature extraction, and optimization-driven selection methods before applying a classification model. The approach enhances input parameters to improve performance. The model achieved an accuracy of 94%, an F-score of 95%, and an AUC of 96% on a dataset of 73 leaf images representing four diseases: Bacterial blight, Anthracnose, Cercospora leaf spot, and Alternaria alternata. The results indicate that this method effectively supports precise plant disease diagnosis.

Liu L et al. [184] presented a DL method to identify pests in leaves. This method uses a global and local feature to recognise pests in leaves. A localised active region proposal network was introduced to accurately identify pests by enhancing contextual and attentional information. The suggested model was evaluated using a 16 pest categories dataset, which produced an average precision (mAP) of 75.03%, exceeding the efficacy of Faster R-CNN and feature pyramid networks.

Further, Table 5.1 summarises the existing methods based on parameter datasets, performance metrics, and limitations. Table 5.1 clearly shows that most approaches are not scalable and have a low computational efficiency. As a result, our objective is to make attempts to improve the model's efficacy and complexity.

The following research gaps from the literature have been found and are listed below.

1. Numerous current techniques, especially deep learning models, possess intricate architectures that elevate computational requirements, resulting in inefficiency and chal-

Table 3.16: Analysis of the Literature Survey on Different Parameters

Author	Dataset	Method Used	Performance Metrics	Limitations
Le et al. [173]	Bccr-segset, Can-red	SVM	Accuracy, Precision, Recall, F1-score	Scalability and computational efficiency were not adequately addressed.
Bhatia et al. [174]	Tomato Powdery Mildew	ELM	CA, AUC	Model lacks robustness and consistency.
Abbas et al. [175]	PlantVillage	DenseNet121	Accuracy	Limited generalization to real-field agricultural environments.
Aswinikumar et al. [176]	Tomato Leaf Disease Dataset	Optimized Mobile CNN	Precision, Recall, F1-score, Kappa, Accuracy	Limited validation under real-world conditions.
Sood et al. [177]	Grapes Leaf Dataset	VGG16, ResNet50, InceptionV3, DenseNet121	Accuracy	Requires extensive training iterations, reducing efficiency.
Kini et al. [178]	Black Pepper Leaf Dataset	ResNet18	Accuracy	Performance highly dependent on dataset diversity and quality.
Khan et al. [179]	Maize Leaf Dataset	VGGNet, InceptionV3, ResNet50, InceptionResNet50	Accuracy, Precision, Recall	Real-world deployment challenges not sufficiently discussed.
Chug et al. [181]	IARI-TomEBD	HDL	Precision, Recall, F1-score, Accuracy	Overfitting and generalization issues not comprehensively analyzed.
Bei Mohammadi et al. [182]	Flavia, Malaya Kew	SWAP LeafNet	Accuracy	High-dimensional feature space increases complexity and overfitting risk.
Aqel et al. [183]	Plant Leaf Dataset	BDA-ELM	Sensitivity, Specificity, Precision, F-measure, Accuracy	Limited dataset size may impact generalization.
Liu et al. [184]	Pest Dataset	GaFPN, LaRPN	Accuracy, Precision, Recall	Difficulty in detecting small-scale objects.

lenges scaling for extensive datasets or real-time applications.

2. Machine learning and hybrid models often exhibit overfitting to datasets, resulting in inadequate generalisation to diverse datasets or varying environmental conditions, compromising their robustness and flexibility.

3.2.3 Proposed Hybrid Framework

This proposed framework addresses the critical challenge of leaf disease detection and classification by integrating advanced deep learning and machine learning methodologies. Leveraging the strengths of transfer learning from DL and the efficiency of Extreme Learning

Machine (ELM) from ML, the proposed VESTACK model is introduced. This model is designed to classify and identify leaf diseases across diverse datasets with improved accuracy and reduced computational complexity. By utilising sparse hidden nodes, the VESTACK model ensures simplicity in design while maintaining robust performance. A detailed step-by-step description of the proposed approach and a visual representation of the framework are provided in Figure 3.17.

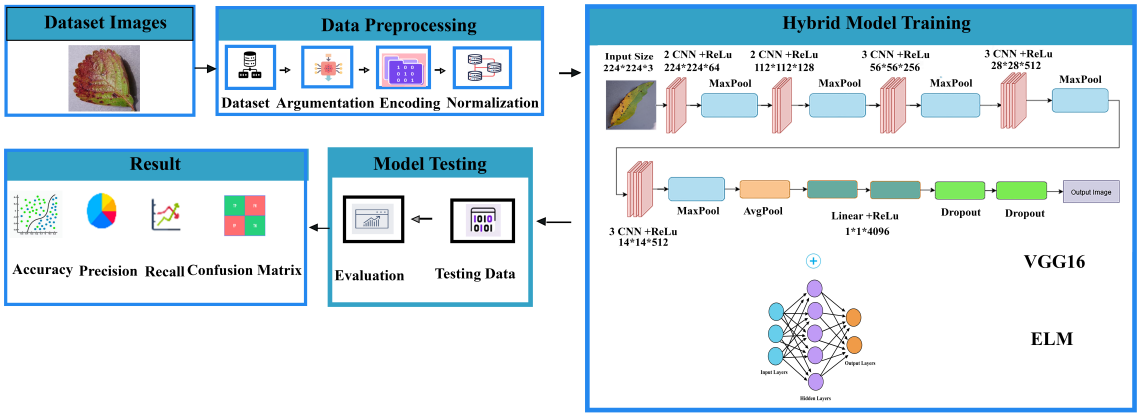


Figure 3.17: Proposed hybrid architecture

The foundation of the VESTACK model lies in VGG16, a widely adopted convolutional neural network (CNN) architecture developed by the Visual Geometry Group at the University of Oxford [185]. VGG16 is renowned for extracting hierarchical features using small-sized CNN filters with a stride of 1, paired with max-pooling layers. It processes input images of size 224×224×3 and comprises layers for convolution, pooling, linear transformations, ReLU activation, and dropout [186]. To enhance the baseline VGG16 model, we incorporated two additional dropout layers of varying sizes and two fully connected linear layers with ReLU activation, thereby improving generalisation and reducing the risk of overfitting. These modifications ensure the effective extraction of high-level features from input images.

To complement the feature extraction capabilities of VGG16, we integrate the Extreme Learning Machine (ELM) into the framework. ELM is a single-layer feed-forward network known for its rapid training speed, simplicity, and excellent generalisation performance [187]. ELM efficiently maps the extracted features to the target classification labels by analytically determining output weights and using sparse hidden node selection. Its suitability for handling large-scale, high-dimensional datasets further strengthens the hybrid approach.

The proposed hybrid model, termed VESTACK, combines the strengths of the enhanced VGG16 and ELM to achieve superior accuracy and efficiency in image classification tasks.

In this framework, the CNN layers of VGG16 are used to extract high-level features from input images. These features are subsequently fed into the ELM, which learns the mapping between the extracted features and the classification task's target labels. This hybrid methodology utilizes the hierarchical feature extraction proficiency of VGG16 alongside ELM's rapid learning and simplicity. Furthermore, the VESTACK model utilizes sparse hidden node selection, diminishing complexity and improving efficacy in disease diagnosis and categorization tasks. Figure 3.17 illustrates the architecture of the hybrid model and showcases its effective layer-wise configuration.

Each layer's detailed explanation is given below: The VESTACK model is built of numerous layers that do specific tasks stepwise.

3.2.3.1 Input Layer

The input to the model is a color image of size 224×224 with three RGB channels. It is represented as:

$$I_m \in \mathbb{R}^{224\times224\times3} \tag{3.2.1}$$

The input tensor can be expressed as:

$$I_m = I_m(h, w, c) \tag{3.2.2}$$

where h denotes the image height, w denotes the image width, and c represents the RGB channels.

3.2.3.2 First Convolutional Layer (Conv2D-1)

The first convolutional layer applies 64 filters of size 3×3. This layer extracts low-level spatial features from the input image.

$$F_1 = \text{Conv}(I_m, K_1) + b_1 \tag{3.2.3}$$

where $K_1 \in \mathbb{R}^{3\times3\times3\times64}$ represents the convolution kernels, and b_1 is the bias term. This layer contains 1,792 trainable parameters.

3.2.3.3 ReLU Activation (ReLU-2)

The Rectified Linear Unit (ReLU) activation function is applied element-wise:

$$F_2 = \max(0, F_1) \tag{3.2.4}$$

where F_2 is the activated feature map.

3.2.3.4 Second Convolution Block (Conv2D-3 and ReLU-4)

Another convolution layer followed by ReLU activation is applied:

$$F_3 = \text{Conv}(F_2, K_2) + b_2 \tag{3.2.5}$$

$$F_4 = \max(0, F_3) \tag{3.2.6}$$

where K_2 and b_2 denote the kernel and bias of this layer.

3.2.3.5 Max Pooling Layer (MaxPool-5)

A max pooling operation with kernel size 2×2 and stride 2 reduces spatial dimensions:

$$F_5 = \text{MaxPool}(F_4, 2 \times 2, 2) \tag{3.2.7}$$

This operation reduces the feature map size to 112×112.

3.2.3.6 Intermediate Convolutional Blocks

The subsequent VGG16 blocks follow a repeated structure of convolution, activation, and pooling operations:

$$F_i = \text{Conv}(F_{i-1}, K_i) + b_i \tag{3.2.8}$$

$$F_{i+1} = \max(0, F_i) \tag{3.2.9}$$

Max pooling is applied after specific convolutional blocks to progressively reduce spatial dimensions.

3.2.3.7 Final VGG16 Layers

The final layers consist of Adaptive Average Pooling and Fully Connected (FC) layers.

$$F_{32} = \text{AdaptiveAvgPool}(F_i) \tag{3.2.10}$$

Fully connected operations are defined as:

$$F_{33} = W_{33}F_{32} + b_{33} \tag{3.2.11}$$

$$F_{34} = \max(0, F_{33}) \tag{3.2.12}$$

where W_{33} is the weight matrix and b_{33} is the bias vector.

Dropout layers are applied during training to prevent overfitting by randomly deactivating neurons.

The final linear layer produces the output feature vector:

$$F_{39} = W_{39}F_{38} + b_{39} \tag{3.2.13}$$

3.2.3.8 Extreme Learning Machine (ELM)

The extracted deep features from VGG16 are fed into the Extreme Learning Machine (ELM) classifier.

The ELM is modeled as a single-layer feedforward neural network:

$$O = g(W_{\text{ELM}}F_{39} + b_{\text{ELM}}) \tag{3.2.14}$$

where:

- W_{ELM} represents the randomly initialized weight matrix,
- b_{ELM} is the bias term,
- $g(\cdot)$ denotes the activation function (e.g., sigmoid or linear),
- O is the final predicted output label.

Thus, the proposed VESTACK hybrid model integrates deep feature extraction using VGG16 with fast classification via Extreme Learning Machine.

In summary, the VESTACK model enhances performance on image classification tasks by integrating the feature extraction capabilities of VGG16 with the ELM method's quick, practical training. Algorithm 3.1 build and train the VESTACK model for image categorization. The algorithm begins with two primary inputs, labels and a training dataset, and one output, which can identify images into predetermined classes. Line 3 shows that the purpose of this function is to construct a hybrid model by performing the following steps. After that, Lines 9-10 show that the training loop starts for each epoch. Lines 11–16 pass the image via the VGG16 model to extract features. In addition, calculate the hidden layer using the formula given in Line 13. Then, calculate the loss function using the cross-entropy. Further, the trained VESTACK model is created after all epochs have been finished. Furthermore, the algorithm returns the trained model, which can subsequently be used to identify images in the provided classes.

Algorithm 3.2 describes the procedure for identifying a test image with an already trained VESTACK model, including an ELM with a VGG16 Model. Algorithm II starts with two inputs, the test image and train hybrid model, and one output to generate a predicted class label for the test image. It extracts features using the VGG16 model, and the ELM element further analyses those characteristics to determine the class label, which is shown in Lines 3–7. After that, the SoftMax function is applied, which converts the result into probabilities of class and returns the anticipated class label for the provided test images.

Algorithm 3.3 is mainly used for hybrid model training and image identification. This Algorithm III initially utilizes the training dataset to train the VESTACK model and then applies the trained model to categorize a set of test images. It uses the two already established algorithms to train and identify images.Algorithm III is mainly used for hybrid model training and image identification. This Algorithm III initially utilizes the training dataset to train the VESTACK model and then applies the trained model to categorize a set of test images. It uses the two already established algorithms to train and identify images.

The time complexity of the VESTACK model depends on several components, primarily influenced by the structure and size of the VGG16 model, the ELM, and the dense layers. The complexity of the CNN and pooling layers in VGG16 is

$$O(C \cdot h_{vgg} \cdot W_{vgg})$$

Algorithm 3.1 Proposed VESTACK Model

Require: Labels, Training Dataset
Ensure: Trained VESTACK Model

Function: Build VESTACK Model
Define Input shape
Create the VGG16 model with pre-trained weights on a dataset
Freeze the VGG16 layer weights to prevent them from being modified during training
Set the total number of hidden nodes (H) and output classes (Y) for the ELM algorithm
Load the training dataset as a collection of pictures I with labels J
Set the weight matrix of the ELM $K \in \mathbb{N}_{(H \times Y)}^{(\Lambda)}$
Set the bias vector of the ELM model $B \in \mathbb{N}_{Y}^{(\Lambda)}$ to random values
for each epoch **do**
    for each image I_i and related label J_i in the training dataset **do**
        To extract features, process the image I_i through the VGG16 model:
        $V_i = \text{VGG16}(I_i)$
        To flatten the retrieved features:
        $\alpha_i = \text{flatten}(fl_i)$
        Determine the hidden layer output:
        $H_i = \alpha_i K + B$
        $H_i = AF(H_i)$
        Calculate the loss:
        $Loss_i = \text{crossentropy}(U_i, TL_i)$
        To minimize the loss, update the weight matrix K and bias vector B using an optimization algorithm:
        $K, B = Up\_We(K, B, \alpha_i, H_i, Y_i)$
    end for
end for
Return the trained VESTACK Model

which is relatively higher for larger input dimensions (h_{vgg} and W_{vgg}) and more channels (C).

The fully connected layer adds a lower complexity term,

$$O(h_{vgg} \cdot neurons)$$

as it depends on the number of neurons but scales less dramatically than the convolutional layers. Flattening the layer contributes minimally with

$$O(h_{vgg})$$

making it a very low-complexity operation.

Algorithm 3.2 VESTACK Model Testing

Require: Trained Hybrid Model, Test Image
Ensure: Classify class label for the test image
1: Load the trained ELM weights and the pre-trained VGG16 model
2: Take a test image and classify it
3: To extract features, run the image through the VGG16 model:
4: $ef = \text{VGG16}(test\_img)$
5: Flatten the retrieved features
6: Calculate the hidden layer output
7: Add the activation layer to the hidden layer
8: Calculate the output matrix
9: To obtain the expected class probabilities, use the SoftMax function:
10: $C = \text{SoftMax}(output\_layer\_matrix)$
11: Add the test image to the most likely class:
12: $expected\_label = \arg\max(U_i)$
13: Return the classified class label

Algorithm 3.3 VESTACK Model Training and Image Identification

Require: Labels, Training Dataset, Number of test images
Ensure: Classify class label for the test image
1: Function: Train the hybrid model and apply it to test images
2: Train VESTACK Model
3: $Trained\_Model = \text{VESTACK}(Labels, Training\ Dataset)$
4: Identify test images with a trained model
5: $Predicted\_Labels = [\,]$
6: **for** each $test\_image$ in the List of test images **do**
7: $Predicted\_labels = \text{Categorize\_Test\_Image}(Trained\_Model, test\_image)$
8: Append $predicted\_label$ to $Predicted\_Labels$
9: **end for**
10: Return $Predicted\_Labels$

In the ELM operation, the linear layer introduces a moderate complexity of

$$O(h_{vgg} \cdot h_{elm})$$

which increases with the size of the VGG16 output and the number of hidden nodes in ELM. The ReLU activation adds another term

$$O(h_{elm} \cdot m)$$

where m is the ELM's output size, and this can vary depending on the architecture.

Overall, the VESTACK model has a time complexity that is dominated by the convolutional operations in the CNN.

3.2.4 Experimental Setup and Result

In this section, we discuss the dataset experimental setup and provide a brief overview of the pre-processing technique employed. Next, the results are generated by considering various parameters to evaluate the strengths and weaknesses of the VESTACK model.

3.2.4.1 Dataset Description

We tested our algorithm against the implementations discussed in research work [170] over four openly available image datasets: Peach, Strawberry, Grapes, and Apple, which is shown in Figure 2.4. A brief description of the datasets and their composition is discussed in Chapter 2, Section 2.4.

We train 70% of the datasets and test 30% to evaluate the VESTACK model. We have enhanced the data with horizontal flips and rotations at different angles to prevent overfitting. Image rotation gives models more leeway to ignore object orientation. Rotating the image shifts pixels beyond the image, leaving an empty pixel value. Various methods are available to fill the black pixel value, including constant and nearest values.

Hyperparameter

A comprehensive hyperparameter configuration Table 3.17 has been added, listing the batch size, optimizer, learning rate, number of epochs, dropout rate, and the number of hidden nodes used in the ELM classifier. This provides transparency and supports reproducibility of the proposed VESTACK framework.

Table 3.17: Hyperparameter Settings Used for the Model

Hyperparameter	Value
Number of Classes	2
Input Image Size	224×224
Optimizer	Adam
Epochs	10, 20, 25
Learning Rate	1×10^{-4}

3.2.4.2 Performance Evaluation Criteria

Precision, recall, and F1-score were used to assess the VESTACK model, as previously described in section 3.1.6.1 above. These measurements are used in this section to assess

the model's performance [168] [188].

The following section outlines some more performance metrics that must be implemented. Following this, the VESTACK model is evaluated using performance metrics such as confusion matrix, FNR, FPR and FDR.

False Positive Rate (FPR)

The fraction of instances in the data that were wrongly reported as positive is known as FPR.

$$\text{FPR} = \frac{FP}{FP + TN} \tag{3.2.15}$$

The above equation describes the total number of negative instances mistakenly classified as positive cases, divided by the overall number of negative cases (i.e., regular data).

False Negative Rate (FNR)

It is calculated by dividing the number of false negatives by the total number of truly positive samples.

$$\text{FNR} = \frac{FN}{TP + FN} \tag{3.2.16}$$

False Omission Rate (FOR)

The frequency of incorrect-negative values relative to the total number of negative values predicted as incorrect and accurate is known as the false omission rate.

$$\text{FOR} = \frac{E}{E + F} \tag{3.2.17}$$

Where, E = False Negative, F = True Negative.

False Discovery Rate (FDR)

The percentage of people with positive test results who have the disease is known as the false discovery rate.

$$\text{FDR} = \frac{C}{D + C} \tag{3.2.18}$$

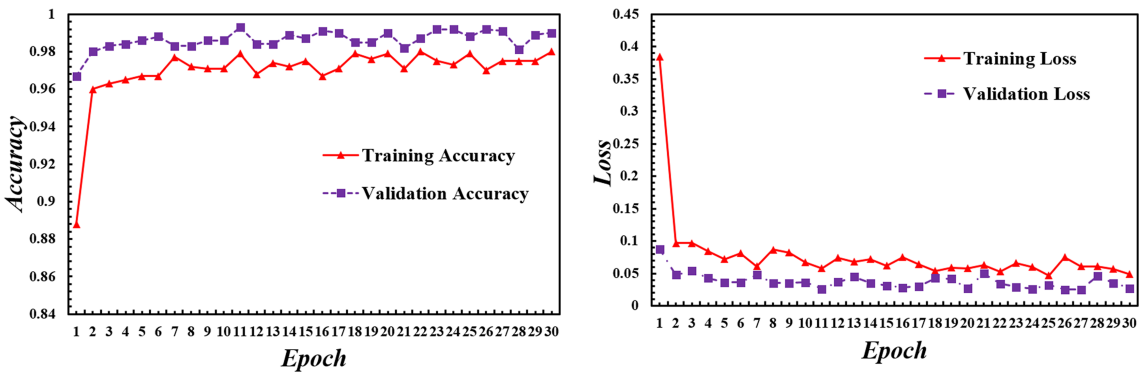
Where, C = False Positive, D = True Positive.

3.2.4.3 Training and Testing Result for Peach, Strawberry, Grapes, and Apple Datasets

This section examined the training and testing for the peach, strawberry, grapes, and apple datasets. Initially, we divided the data set into 70% training and 30% testing. Next, data augmentation methods are applied to improve the visual quality of images, including rotation, zooming, flipping, and so forth. Moreover, the hybrid model (VESTACK) is trained and tested using the pre-processed data. The model was initially run for 10, 20, and 25 epochs to analyze its performance, and it was observed that the results remained approximately consistent across these epochs. Consequently, the model was finalized and run for 30 epochs to ensure stability and completeness of training. The obtained result for 30 epochs is shown in Table 3.18. The proposed VESTACK model achieved a high accuracy of 99.1% for the peach dataset, 99.8% for the strawberry dataset, 98.9% for the grape dataset, and 97.7% for the apple dataset, as demonstrated in Table 3.18.

Table 3.18: Quantitative Results of Plant Disease Detection using Four Datasets

Dataset	Model	Training Accuracy (%)	Training Loss	Testing Accuracy (%)	Testing Loss
Peach	VESTACK	99.00	0.071	99.19	0.028
Strawberry	VESTACK	99.53	0.017	99.84	0.003
Grapes	VESTACK	98.40	0.081	98.90	0.091
Apple	VESTACK	97.60	0.061	97.70	0.072

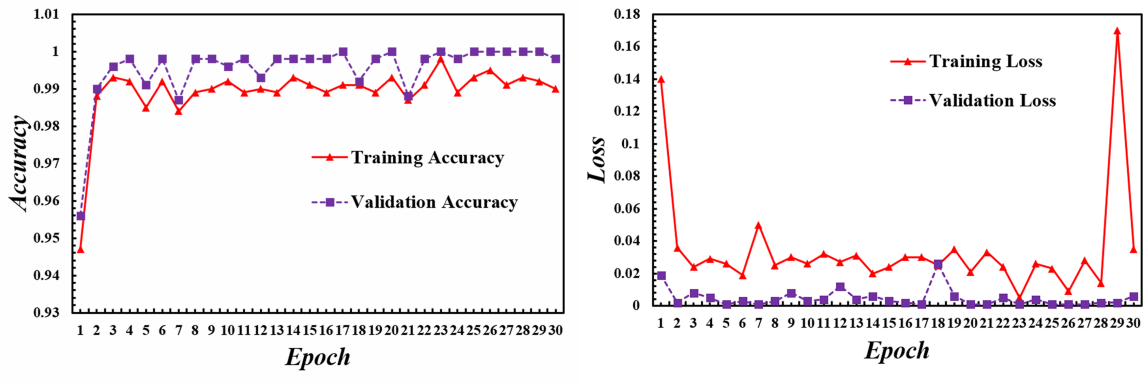


(a) Training and Testing Accuracy (b) Training and Testing Loss

Figure 3.18: VESTACK model training and testing result analysis for peach dataset

We have also analysed the performance of the proposed VESTACK hybrid model by showing their relationship between the accuracy of training & testing as well as training & testing loss as shown in Figure 3.18- Figure 3.21. In Figure 3.18(a)-Figure 3.21(a), the horizontal axis indicates the number of epochs and the vertical axis indicates accuracy. As the number

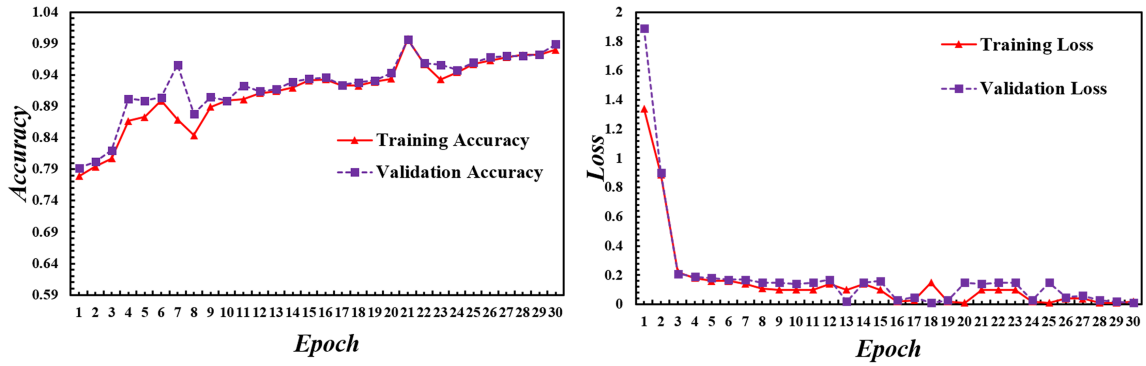
of epochs decreases, the accuracy tends to be low or near zero. Conversely, when the number of epochs increases, the accuracy improves. It usually demonstrates the rising pattern as the system learns from the training data.



(a) Training and Testing Accuracy (b) Training and Testing Loss

Figure 3.19: VESTACK model training and testing result analysis for strawberry dataset

Further, the outcome of such models for validation and training loss is demonstrated in Figure 3.18 (b)- Figure 3.21 (b). When the number of epochs in the initial stage is near zero, a loss is high; with increasing epochs, the loss becomes lesser. It aids in determining how successfully a model generalises to novel, previously unknown data.

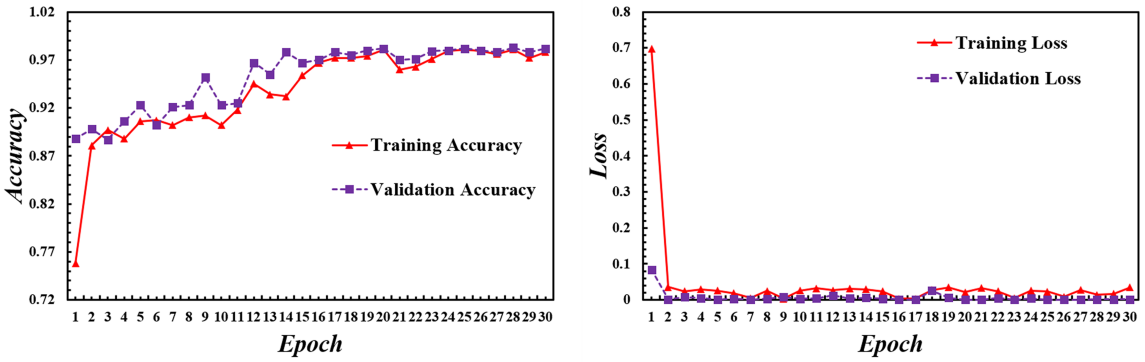


(a) Training and Testing Accuracy (b) Training and Testing Loss

Figure 3.20: VESTACK model training and testing result analysis for grapes dataset

3.2.4.4 Visualisation Results

This section explains procedures for visual interpretation to identify disease-affected plant leaves. To enhance interpretability, three well-known visualization approaches (Grad-CAM, LIME, and SHAP) were utilized, as mentioned in the following sections.



(a) Training and Testing Accuracy (b) Training and Testing Loss

Figure 3.21: VESTACK model training and testing result analysis for apple dataset

1. LIME Visualization

The LIME approach (Local Interpretable Model-agnostic Explanations) is used to better understand the proposed model's decision-making process by identifying the image regions that most contribute to classification outcomes. Figure 3.22 highlights the yellow areas in the LIME visualizations to indicate the super pixels that the model focuses on when detecting illnesses in peach, strawberry, grape, and apple leaves. These highlighted patches are strongly associated with apparent illness manifestations, such as lesions, spots, and discolouration, suggesting that the model focuses on disease-related areas rather than background or irrelevant areas. LIME-based explanations provide understandable, visually interpretable explanations, allowing domain experts to better comprehend and corroborate the model's predictions. This interpretability will also boost confidence in the proposed model's reliability and effectiveness in detecting plant diseases across a variety of crop categories.

2. SHAPE Visualization

To provide a clear understanding of how the model arrives at its disease classification decisions, the SHAP (Shapley Additive Explanations) tool is used to visualise regions of the input image that contribute most to the model's decisions. The red-highlighted parts of the SHAP heatmaps (Figure 3.23) indicate regions that have contributed positively to the model's prediction, while the blue areas indicate regions that have contributed negatively. In all the analysed data sets, including the Peach (a), Strawberry (b), Grapes (c) and Apple (d) datasets, the SHAP visualisations are sensitive to highlighting the areas of the plants that are afflicted with diseases, including lesions, spots, yellowing, and discolouration, which are important signs of diseases in plants. The model prioritises symptomatic areas by assigning fine-grained attributes to the various regions of the image, which is beneficial when

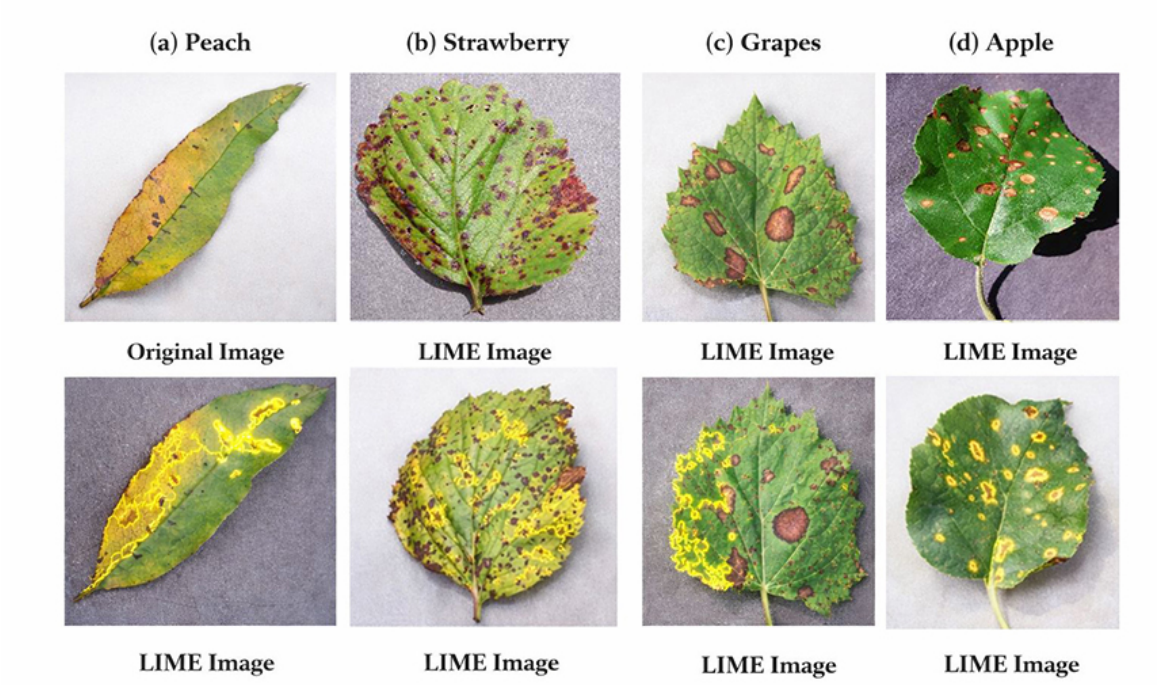


Figure 3.22: Visualisation of lIME analysis on (a) peach (b) strawberry (c) grapes and (d) apple Leaf dataset

focusing on aspects of the disease rather than background information. The explanations of these SHAP values are very complete and intuitive, providing a clear understanding of how the proposed model processes input images to interpret them, thereby improving the transparency, reliability, and trustworthiness of its predictions for diagnosing agricultural diseases.

3.2.4.5 Evaluation Result for Peach, Strawberry, Grapes, and Apple Datasets

Firstly, this section gives the outcome metrics collected for each binary and multiclass category dataset. The quality evaluations and the quantity of datasets examined at each level are listed in Table 3.19. Table 3.19 presents the evaluation results of the proposed models for binary and multiclass leaf disease. The Recall, F1-Score and Precision are evaluation criteria on a 0 to 1 scale. A high recall and precision suggest that the models correctly differentiate between distinct classes, and a high F1-Score indicates a good decision in terms of recall and precision.

In Table 3.20, we have also generated weighted and macro values to assess the model's total outcome on four datasets. In contrast, the macro value evaluates the classifier's overall performance using a more familiar category label, whereas the weighted average accounts for

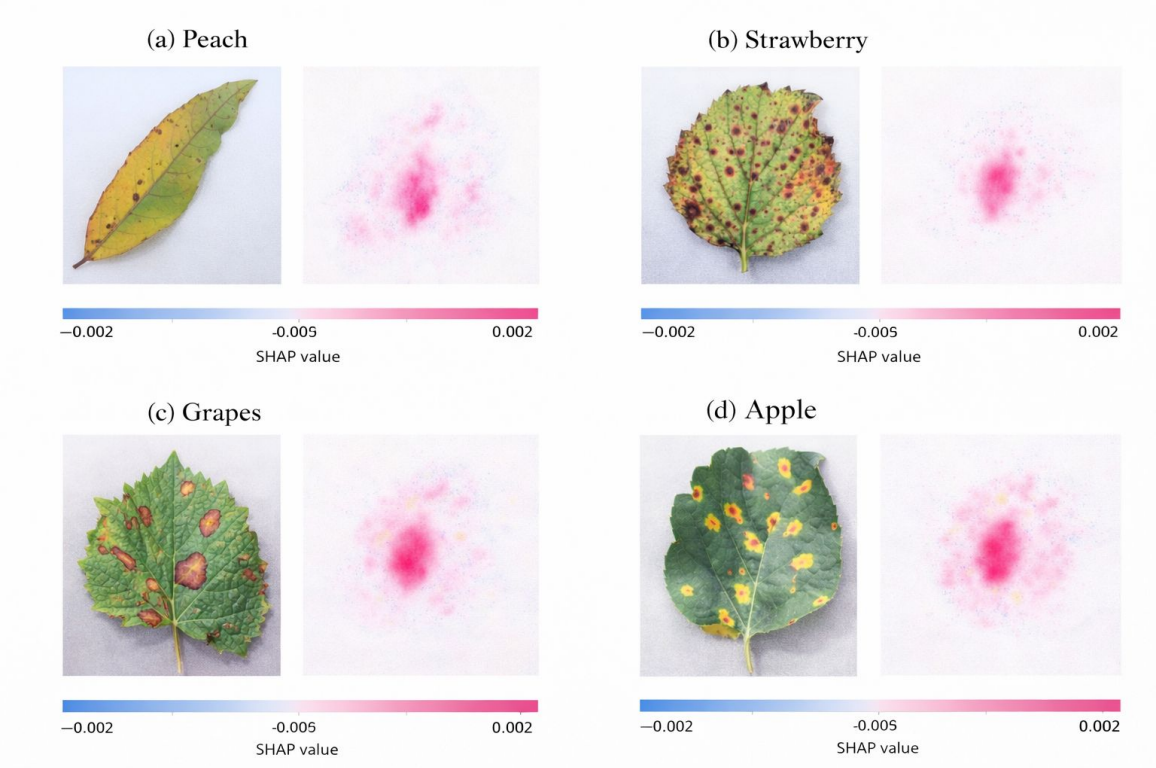


Figure 3.23: Visualisation of sHAPE analysis on (a) peach (b) strawberry (c) grapes and (d) apple Leaf dataset

class imbalance. The macro-average scores across all datasets (peach, strawberry, grapes, and apple) are 0.99, 0.998, 0.984, and 0.976, respectively, indicating that the suggested approach works well across all categories, ensuring that efficiency in smaller categories is not dominated by efficiency in larger classes [189]. Figure 3.24 depicts the Receiver Operating Characteristic (ROC) curve, which shows the connection between precision and recall at different thresholds for recognizing values that are positive in our method [190] [191]. This graph is defined by FPR and TPR on the X and Y axes, respectively, showing the relationship between specificity and sensitivity . The ROC line shows a position in the top-left region of the coordinate (0, 1), denoting 0.99 false-negative rate and 0.99 false-positive rate in Figure 3.24(a). Similarly, for Figure 3.24(b) ROC line shows 1.00 no false negative rate and 1.00 no false positive rate.

Like, in Figure 3.24(c)- Figure 3.24(d), the ROC line shows 0.98 and 0.97 with no false negative rate and 0.98 & 0.97 with no false positive rate. So, the Y-axis is called better classification [192].

Table 3.19: VESTACK image quality for the Peach, Strawberry, Grapes, and Apple datasets

Dataset	Class	Precision	Recall	F1-Score	Support
Peach	Healthy	0.995	0.992	0.994	690
	Bacterial_spot	0.983	0.990	0.986	300
Strawberry	Healthy	0.997	1.000	0.998	333
	Leaf_scorch	1.000	0.996	0.998	300
Grapes	Grape_Black_rot	0.985	0.891	0.936	369
	Grape_Esca	0.971	0.986	0.945	378
	Grape_healthy	1.000	1.000	1.000	396
	Grape_Leaf_blight	0.995	1.000	0.997	423
Apple	Apple Scab	0.975	0.901	0.956	369
	Black Rot	0.971	0.986	0.945	378
	Cedar Apple Rust	0.989	1.000	1.000	396
	Healthy	0.978	1.000	0.997	423

3.2.4.6 Ablation Study

This section presents the ablation study conducted to analyze the contributions of various components in the proposed framework and their impact on the overall system performance. The evaluation focuses on four experimental cases to investigate the effectiveness of individual model variations. The details of the cases are as follows:

1. **Case P: Train the Dataset employing the pre-trained VGG16**

   This case evaluates the baseline performance of the standard pre-trained VGG16 model, leveraging transfer learning without any modifications, which is shown in Figure 3.25.

2. **Case Q: Train the Dataset utilizing the Proposed VGG16**

   This case examines the enhancements made to the original VGG16 architecture, including adding dropout layers and ReLU-based fully connected layers, to analyze their impact on generalization and accuracy which is shown in Figure 3.26.

3. **Case R: Train the Dataset using an Extreme Learning Machine (ELM)**

   This case assesses the standalone performance of ELM in mapping features to classification labels, providing insight into its capability to handle the dataset efficiently

Table 3.20: Classification Results for Detecting Peach Leaf Disease

Dataset	Metric Type	Precision	Recall	F1-Score
Peach	Macro Avg	0.9895	0.9914	0.9905
	Weighted Avg	0.9919	0.9919	0.9919
Strawberry	Macro Avg	0.9985	0.9983	0.9984
	Weighted Avg	0.9984	0.9984	0.9984
Grapes	Macro Avg	0.987	0.969	0.987
	Weighted Avg	0.987	0.977	0.986
Apple	Macro Avg	0.975	0.986	0.956
	Weighted Avg	0.977	0.975	0.957

i.e shown in Figure 3.27.

4. **Case S: Train the Dataset using VESTACK (our proposed model)**

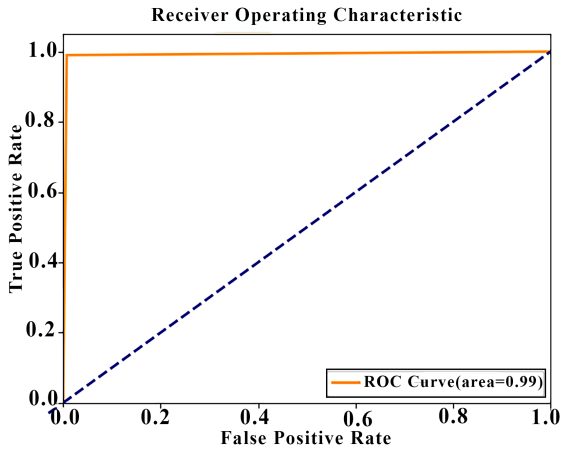
   This case evaluates the combined approach, where the modified VGG16 extracts features, and ELM maps these features to target labels. The goal is to understand how this integration impacts overall performance which is shown in Figure 3.28.

This extensive evaluation of individual cases aims to thoroughly investigate each method variation's impact and effectiveness. The goal is to thoroughly understand their responsibilities and the implications for the system's overall effectiveness.

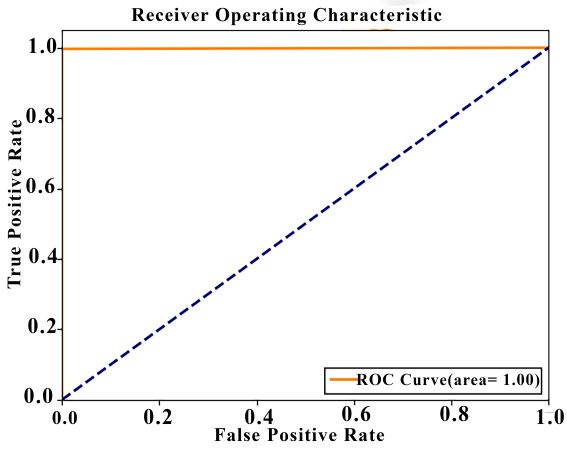
Table 3.21: Ablation Study Score for the VESTACK model for the Peach and Strawberry Datasets

Ablation Study Cases	Case	Precision		Recall		Accuracy		Time (m/sec)	
		Peach	Strawberry	Peach	Strawberry	Peach	Strawberry	Peach	Strawberry
Train the dataset with VGG16	P	0.940	0.937	0.923	0.917	0.940	0.944	10m/11sec	10m/3sec
Train the dataset with Proposed VGG16	Q	0.962	0.964	0.975	0.970	0.971	0.978	8m/2sec	9m/10sec
Train the dataset with ELM	R	0.963	0.953	0.975	0.968	0.969	0.961	9m/4sec	10m/6sec
Train the dataset with VESTACK (Proposed Method)	S	0.983	0.999	0.990	0.996	0.991	0.998	13m/48sec	12m/23sec

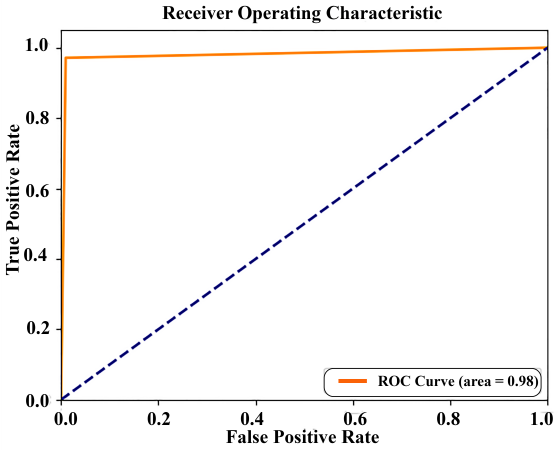
From the existing literature survey, we found that few works are available on the VGG16 pre-trained model, and it also performs well on smaller datasets. So, when we trained the VGG16 pre-trained model (Case P) on our datasets (Peach, Strawberry, Grapes, and Apple), we got results as shown in Table 3.21 and Table 3.22. Additionally, we trained the datasets on the proposed VGG16 model, which can address the problems of the VGG16 models, such as overfitting and generalisation problems, after adding two dropout layers of varying sizes and two linear layers with ReLU for better performance in results. As



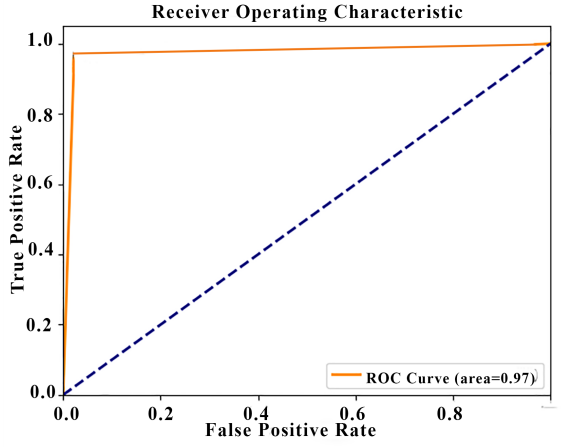
(a) ROC curve for peach dataset



(b) ROC curve for strawberry dataset



(c) ROC curve for grapes dataset



(d) ROC curve for apple dataset

Figure 3.24: ROC curve for VESTACK: based on the computed TPR&FPR

per Table 3.21 and Table 3.22, we see that Case Q (Proposed VGG16) perform better as compared to pre- trained VGG16 model in terms of (precision (0.02%), recall(0.05%), accuracy(0.03%) for Peach), (precision(0.02%), recall(0.05%), accuracy(0.03) for Strawberry), (precision(0.05%), recall(0.06%), accuracy(0.03%) for Grapes) and (precision(0.06%), re-

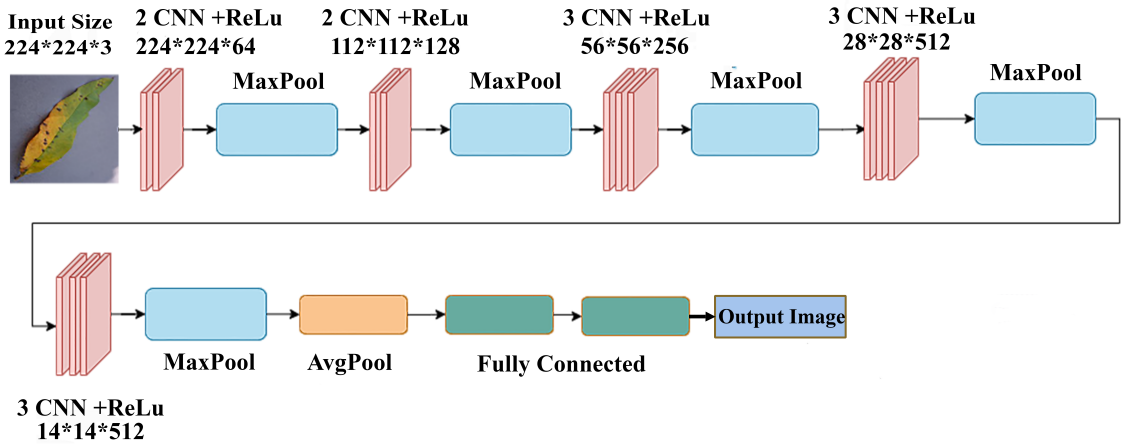


Figure 3.25: Baseline of VGG16

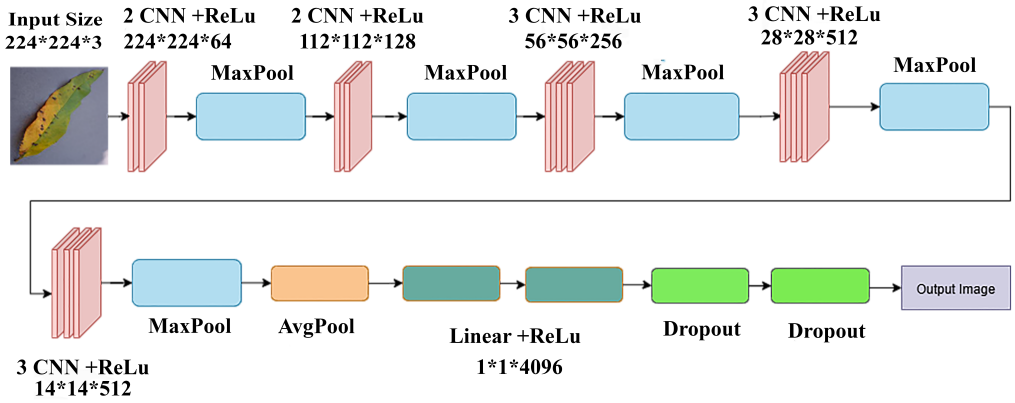


Figure 3.26: Proposed VGG16

call(0.03), accuracy(0.03) for Apple dataset). It is important to note that, in the existing literature, frequently used machine learning models are intrinsically complex in their architecture, further adding to the overall complexity of the classification problem. To overcome this issue, we train the dataset using the ELM model. Case R (ELM model) performs well compared to Case P and Case Q, as shown in Table 3.21 and Table 3.22.

Table 3.22: Ablation Study Score for the VESTACK model for Grapes and Apple Datasets

Ablation Study Cases	Case	Precision		Recall		Accuracy		Time (m/sec)	
		Grapes	Apple	Grapes	Apple	Grapes	Apple	Grapes	Apple
Train the dataset with VGG16	P	0.882	0.891	0.900	0.923	0.920	0.923	8m/10sec	10m/2sec
Train the dataset with Proposed VGG16	Q	0.941	0.956	0.962	0.954	0.955	0.959	6m/9sec	7m/12sec
Train the dataset with ELM	R	0.964	0.935	0.956	0.935	0.964	0.941	8m/12sec	7m/16sec
Train the dataset with VESTACK (Proposed Method)	S	0.987	0.978	0.977	0.975	0.989	0.977	12m/16sec	15m/48sec

After that, VESTACK (our proposed hybrid method) was used for training on the datasets to reduce the architecture's complexity and improve accuracy. Case S (VESTACK) perform better in terms of (precision (0.04%-002%), recall (0.07%-0.02%), accuracy (0.05%-0.02%)

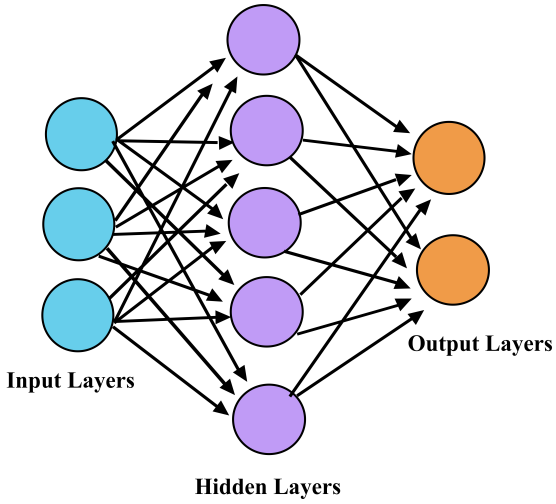


Figure 3.27: Framework of ELM

for peach and strawberry dataset) and (precision (0.10%-0.02%), recall (0.07%-0.01%), accuracy (0.06%-0.02%) for Grapes), (precision (0.08%-0.02%), recall (0.05%-0.02%), accuracy (0.05%-0.01%) for Apple). We have also evaluated the time required for training in different cases (P, Q, R, S). It displays the time required for each model to complete training, as shown in Table 3.21 and Table 3.22. The result shows that the VESTACK model produces consistent, outstanding performance across various settings, demonstrating its reliability and efficacy. Its high recall, precision, accuracy, and least computational time makes it a valuable tool for data analysis and decision-making in farming industries.

Table 3.23 shows the time taken to run the proposed VESTACK model on the Peach, Strawberry, Grapes, and Apple datasets. The findings reveal a steady, controllable execution time across dataset sizes, suggesting a stable computational pattern for the model. It is worth noting that the model works more effectively with smaller datasets like Strawberry and Grapes, whereas the computation time with large datasets like Apple remains reasonable. This discussion shows that VESTACK can be computationally implemented and may be applicable in practice to real-life agricultural scenarios.

Table 3.23: Execution Time (m/sec) of VESTACK (our proposed method) Across Different Datasets

Model	**Time (m/sec)**			
	Peach	Strawberry	Grapes	Apple
VESTACK (our proposed method)	13m/48sec	12m/23sec	12m/16sec	15m/48sec

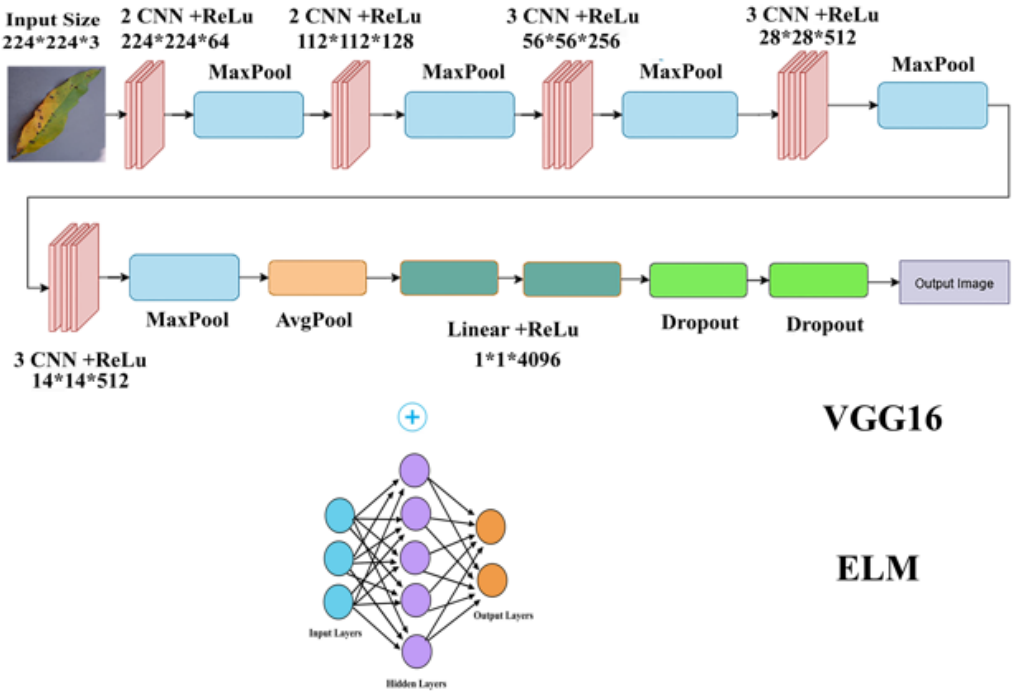


Figure 3.28: VESTACK (our proposed model)

3.2.4.7 Comparison with the State-of-the-Art Techniques

This section compares the proposed hybrid model (VESTACK) with different cutting-edge strategies for identifying and classifying leaf disease using the binary class datasets(peach [193], strawberry [193]) and Multiclass datasets (grape[31], apple [185]). Table 3.24 displays the performance results for these existing techniques([170], [194], [195], [196], [197]) over a variety of criteria. In terms of precision, recall, and overall accuracy, the suggested hybrid method (VESTACK) outperforms existing methods for the binary classification of peach and strawberry datasets. The proposed solutions perform better because of their hybrid phase strategy. One significant benefit of VESTACK is that it decreases the challenge of the desired concept by using a concealed node section to help categorise and identify anomaly issues. Further, it outperforms deep learning-based processes regarding generalization and pre-trained models. The results show that for the Peach Dataset, our VESTACK model has the highest precision of 98.3% which is 0.4-13.1% better, the highest recall of 0.99% which is 0.1-13.7% better, and the highest accuracy of 99.1% which is 0.6-12.5% higher than all baseline methods. For the Strawberry Dataset, our modified VESTACK model has the highest precision of 99.9%, which is 3-15.1% better; the highest recall of 99.6%, which is 1.5-14.9% better; and the greatest accuracy of 99.8%, which is 2.1-14.9% more accurate than all existing methods. Because it eliminates the accuracy and model-complexity issues associated with leaf illness identification. It provides an informative study that compares

the VESTACK model's output with the most reliable method for identifying leaf diseases.

Table 3.24: Comparison of the Proposed Model with the State-of-the-Art Techniques on the Peach and Strawberry Dataset

Model	Precision		Recall		Overall Accuracy	
	Peach	Strawberry	Peach	Strawberry	Peach	Strawberry
VESTACK (our proposed model)	0.983	0.999	0.990	0.996	0.991	0.998
L1_ELM [170]	0.979	0.970	0.989	0.981	0.985	0.977
OP-ELM [170]	0.971	0.965	0.983	0.976	0.979	0.972
ELM [170]	0.963	0.953	0.975	0.968	0.969	0.961
Pantazi et al. [194]	0.940	0.937	0.955	0.949	0.950	0.944
Tetila et al. [195]	0.972	0.966	0.989	0.982	0.986	0.978
Sun et al. [196]	0.911	0.905	0.923	0.917	0.919	0.908
Kaur et al. [197]	0.852	0.848	0.853	0.847	0.860	0.849

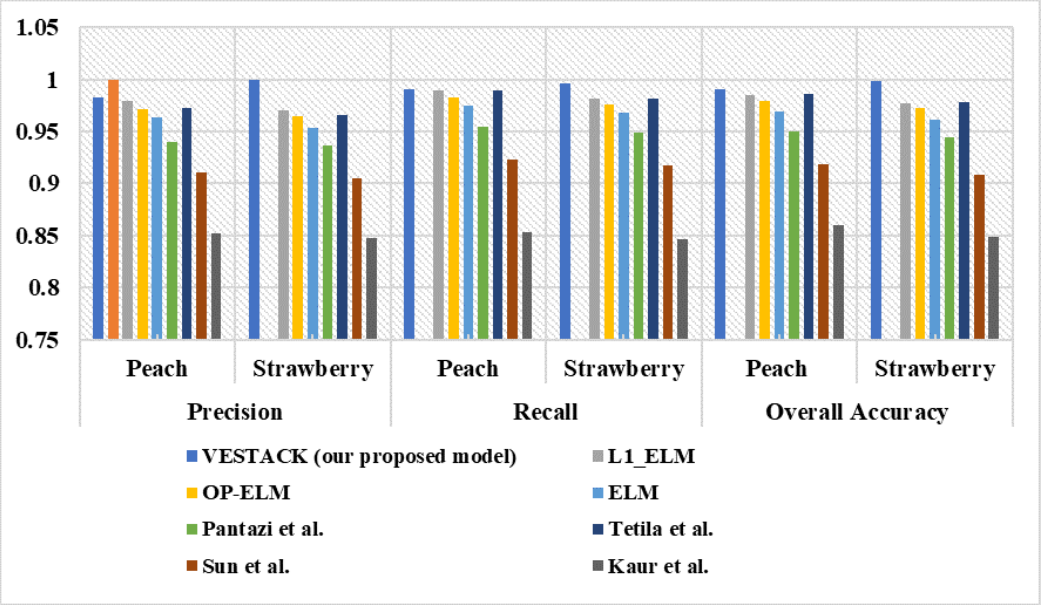


Figure 3.29: Graphical representation of the proposed model with the existing state of the art techniques on the different parameters

Table 3.25 and Table 3.26 displays the performance results for these existing techniques([198], [199], [200], [201], [202], [203], [204], [205]) over a variety of criteria. In terms of precision, recall, and overall accuracy, the suggested hybrid method (VESTACK) outperforms existing methods for the multiclass classification of the Grapes and Apple datasets. The outcomes demonstrate that for Grapes Dataset, our modified VESTACK model has the highest precision 98.7% which is 0.23-0.02% better, highest recall 97.7% which is 0.23-0.02% better, highest accuracy 98,9% which is 0.19-0.02% higher than the all cutting-edge methods.

For the Apple Dataset, our modified model has the highest precision 97.8% which is 0.04-

Table 3.25: Comparison of the Proposed Model with the State of the Art Techniques on the Grapes Dataset

Model	Grapes Dataset		
	Precision	**Recall**	**Overall Accuracy**
Vgg16 [198]	0.964	0.956	0.964
YOLOv5-CA [199]	0.885	0.837	0.895
Faster R-CNN [200]	0.750	0.744	0.791
GrapeNet [201]	0.881	0.774	0.862
Modified Vgg16 (Our Model)	0.941	0.962	0.955
VESTACK (our proposed model)	0.987	0.977	0.989

0.007% better, highest recall 97.5% which is 0.01-0.04% better, highest accuracy 97.7% which is 0.04-0.008% higher than the all existing methods.

Table 3.26: Comparison of the Proposed Model with the Cutting edge Techniques on the Apple Dataset

Model	Apple Dataset		
	Precision	**Recall**	**Overall Accuracy**
VGG-ICNN [202]	0.955	0.964	0.942
PlantXViT [203]	0.935	0.935	0.935
LIU, Song et al. [204]	0.971	0.962	0.962
Suguna, R et al. [205]	0.967	0.964	0.969
Modified Vgg16 (Our Model)	0.956	0.954	0.959
VESTACK (our proposed model)	0.978	0.975	0.977

Figure 3.29- Figure 3.31 shows the comparison results of the suggested model's accuracy, precision, and recall with the baseline research work, presented as a bar graph. Overall, the suggested hybrid approach (VESTACK) performs better than the currently used methodologies and can classify different leaf diseases.

We have now explicitly clarified in the revised manuscript that the comparison results reported for existing methods are directly taken from their respective published papers and were not re-executed by the authors. This clarification has been clearly stated in the result tables and corresponding text to ensure transparency.

To further demonstrate the robustness and stability of the proposed VESTACK model, error-bar–based performance comparisons have been included for the Peach, Strawberry,

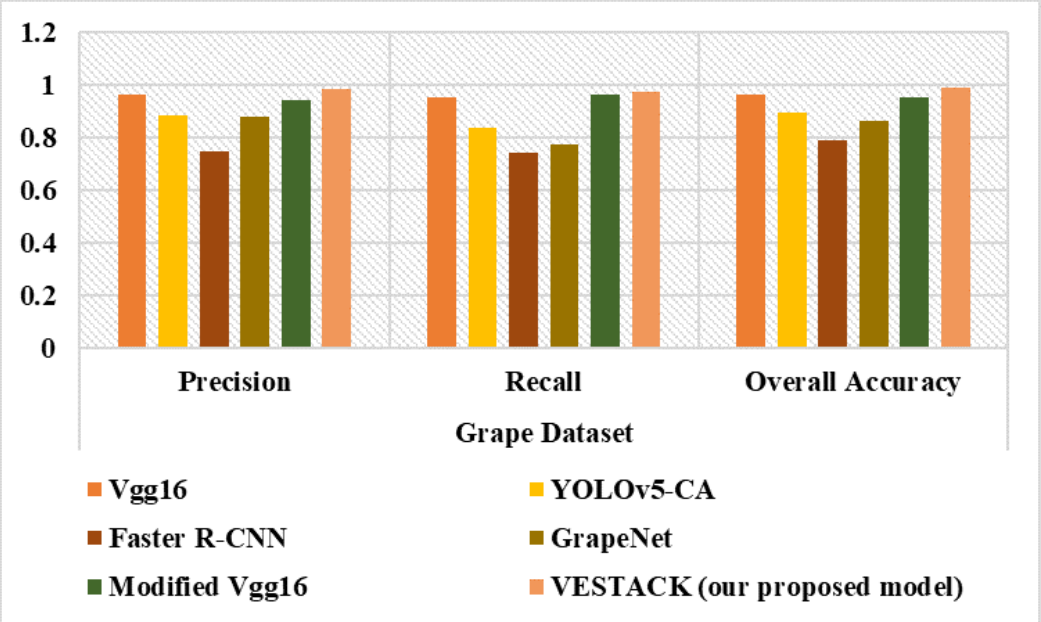


Figure 3.30: A graphical representation of the suggested model, along with the current baseline techniques on the various parameters

Grapes, and Apple datasets. The error bars show performance variation across multiple runs, highlighting VESTACK's consistency compared to existing methods. Across all datasets, VESTACK exhibits minimal variance with higher mean accuracy, confirming reliable generalization. These results strengthen the claim of improved accuracy, with stable, repeatable performance across varying conditions.

3.2.5 Discussion

3.2.5.1 Environmental Constraints

To overcome the restriction of using curated datasets that might not be representative of the diversity of real-world agricultural environments a comprehensive data augmentation plan was followed. This is of particular concern due to the difference in dataset sizes as relatively smaller datasets, like Strawberry (1,476 images) or larger ones, such as Apple (9,713 images), could affect the model generalization in uncontrolled field conditions.

The augmentation framework was designed to simulate common real-world challenges encountered in agricultural fields, including illumination changes, background clutter, and partial occlusions. Specifically, ColorJitter, GaussianBlur, and RandomErasing were applied individually during training to emulate lighting variation, sensor noise, and occlusion effects, respectively (Figure 3.34). In addition, a combined augmentation setting was evaluated to reflect more complex environmental conditions.

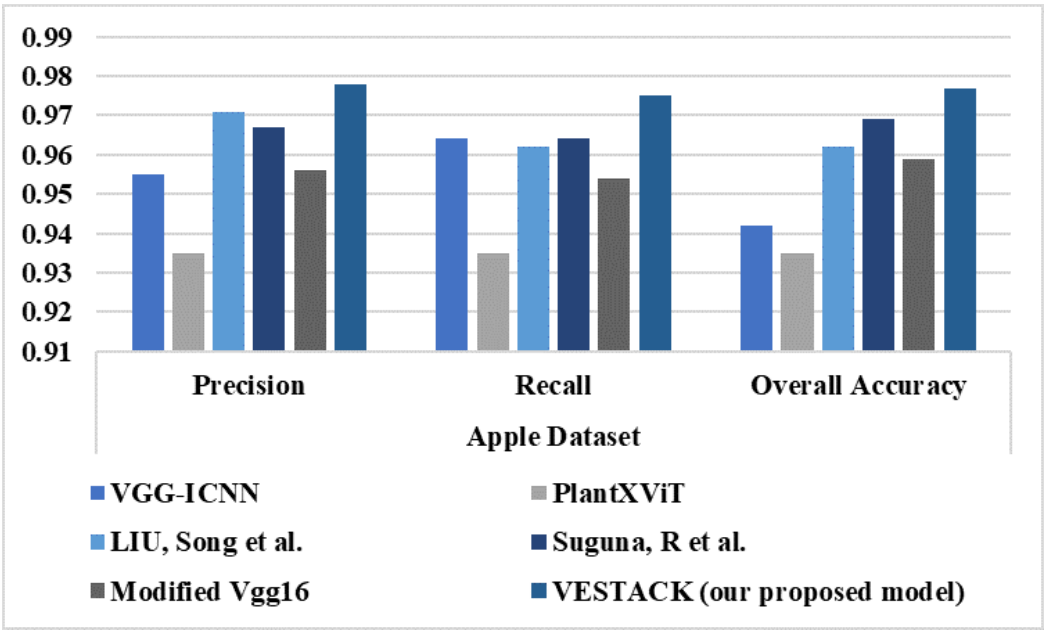
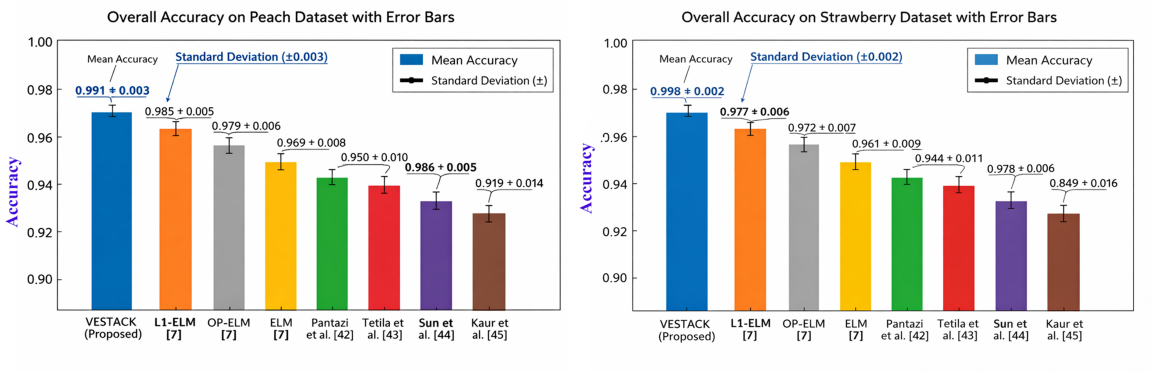


Figure 3.31: Graphical representation of the VESTACK model with the existing baseline techniques on the different parameters

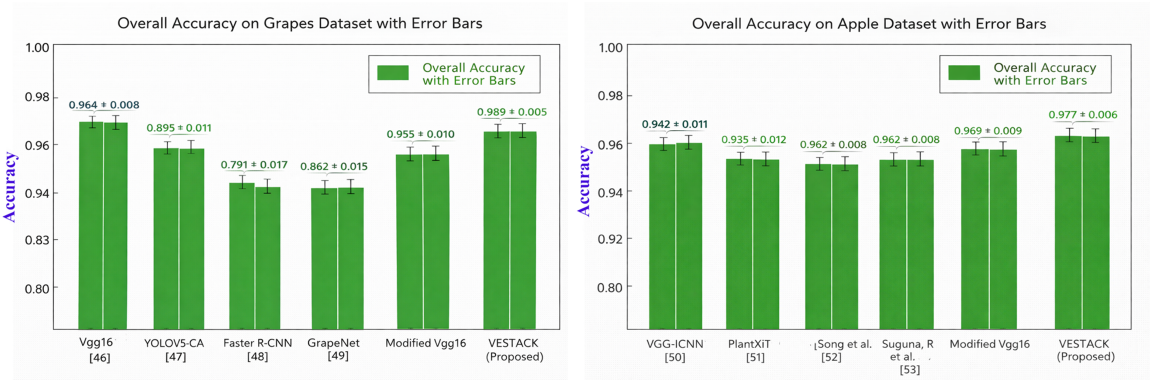


(a) Peach (b) Strawberry

Figure 3.32: Accuracy with Error Bars on Peach and Strawberry Leaf Dataset

To assess robustness under realistic scenarios, a limited-scale evaluation was conducted using real-world field images captured under uncontrolled agricultural conditions. The proposed model, VESTACK, was tested on these images to analyze how different augmentation strategies impact performance across datasets of varying sizes.

Table 3.27 presents the testing accuracy of the proposed VESTACK model under real-world conditions using different augmentation techniques on the Peach, Strawberry, Grapes, and Apple datasets. This analysis highlights the individual and combined contribution of augmentation methods in improving model resilience to real-world noise and variability.



(a) Grapes (b) Apple

Figure 3.33: Accuracy with Error Bars on Grapes and Apple Leaf Dataset

Table 3.27: Impact of Different Data Augmentation Techniques on Classification Accuracy Across Leaf Datasets

Augmentation	Accuracy			
	Peach Leaf Dataset	Strawberry Leaf Dataset	Grapes Leaf Dataset	Apple Leaf Dataset
Only ColorJitter	0.821	0.784	0.806	0.839
Only GaussianBlur	0.807	0.768	0.792	0.824
Only RandomErasing	0.834	0.798	0.819	0.856
Combine	0.861	0.823	0.847	0.879

3.2.5.2 Practical Deployment and Limitations of the VESTACK Framework

The proposed VESTACK architecture has a high likelihood of being used in real-world agricultural contexts because the accuracy-computing efficiency trade-off is well balanced. In a real-world setting, farmers can use their cell phones to snap leaf photos and receive rapid disease detection results. VESTACK is also compatible with IoT-enabled sensor systems and UAV-based imaging systems, enabling large-scale crop surveillance, early disease detection, and decision-making in precision agriculture. Although it performs well, there are downsides to consider. The datasets used in this research are publicly available, which can introduce bias and limit the study's extrapolation to unobserved field situations. Furthermore, ELM classifier performance depends solely on feature scaling, which requires careful preparation. In addition, large-scale field testing has yet to be conducted, even though large-scale experiments were conducted under controlled conditions. The need to overcome these restrictions through on-field performance with IoT and UAV platforms is an important area of future research.



Figure 3.34: Original leaf image and its augmented versions using Color Jitter, Gaussian Blur, and Random Erasing.

Table 3.28: Computational Complexity Comparison of VGG16, ELM, and VESTACK

Model	Parameters (Millions)	FLOPs (GigaFLOPs)	Remarks
VGG16 (Baseline)	138.3 M	15.5 GFLOPs	High computational cost due to fully connected layers
Standalone ELM	2.1 M	0.18 GFLOPs	Lightweight but limited feature representation
VESTACK (Proposed)	21.6 M	6.4 GFLOPs	Balanced complexity with superior accuracy

3.2.5.3 Computational Complexity Comparison of VGG16, ELM, and VESTACK

To support the point about reduced computational complexity, we provide explicit summaries of the number of parameters and FLOPs for VGG16, the standalone ELM, and the proposed VESTACK model. As Table 3.28 shows, VESTACK can be trained on substantially less expensive processors than the baseline VGG16, and its feature learning is more detailed than that of a standalone ELM. This demonstrates that VESTACK cannot achieve a good balance between processing power and categorisation.

3.2.5.4 Impact of Data Augmentation on Model Performance Across Datasets

In the case of data augmentation, we have elaborated on the description of the augmentation strategies that are applied, such as rotation angles with suitable padding after rotation, colour perturbation ranges, and random occlusion settings. We also provide a comparative study of the accuracy of training with augmented images and training accuracy with non-augmented images on the Peach, Strawberry, Grapes, and Apple datasets. As illustrated in Table 3.29, the accuracy of training on non-augmented images is significantly lower due to low intra-class variance and increased susceptibility to overfitting. Conversely, augmented image training always has a significantly higher accuracy level on all types of data, which proves that the training of data augmentation is an effective tool to diversify features and

Table 3.29: Impact of Data Augmentation on Model Performance Across Datasets

Dataset	Augmented Images (%)	Non-Augmented Images (%)
Peach	99.10	92.14
Strawberry	99.53	93.02
Grapes	98.92	91.48
Apple	97.94	90.76

stabilize the learning procedure. Deep learning-enabled agriculture is one of the modern era's significant demands, increasing farming production, sustainability, and profitability. Every year, farmers encounter numerous challenges in growing a diverse range of crops, including pest-related issues, environmental concerns, and leaf disease issues. Therefore, this paper offers a hybrid transfer learning strategy employing the VESTACK model to recognise diverse plant diseases to handle model complexity and high dimensionality data. The model is trained and tested on the peach, strawberry, grapes, and apple datasets. The initial stage is to evaluate a complete set of features capable of properly classifying the class of leaf images. Then, in the hidden layer, we employ VESTACK, which allows us to identify sparse output weights. This feature aids in reducing the model's complexity. Our approach is distinct because it maintains reduced computing time, enhanced generalization, and sparsity. This work shows the effectiveness of the VESTACK method, which achieves an identification rate of 99.1% for the peach dataset,99.8% for the strawberry, 98.9% for the grapes, and 97.7% for the apple dataset compared to the baseline methods. In the future, the suggested VESTACK model will be evaluated on different plant disease datasets. We will also try to develop a real environment to test the proposed hybrid model on real-time data.

3.3 Tomato Leaf Disease Detection and Classification using MobileNetV2 and Extreme Learning Method: A Hybrid Approach

3.3.1 Overview

Tomatoes are an essential commercial crop farmed worldwide. It is susceptible to a variety of diseases, reducing tomato quality and output while also incurring major economic

losses [206]. Tomatoes, as a major vegetable and commercial crop around the world, have tremendous financial and edible value. However, based on the Food and Agriculture Organisation of the United Nations survey data, tomato disease is the primary cause of the drop in worldwide tomato cultivation, with a yearly loss rate of up to 10% [207]. The presence of tomato diseases not only causes significant economic losses to several farmers but also poses a danger to agricultural security. Many tomato infections begin in the leaf and move throughout the plant. Thus, identifying tomato leaf illnesses is the first step in treating its diseases. Currently, expert diagnosis and observation experience are used to detect tomato illnesses early, but these two procedures are lengthy, costly, and inefficient, which might lead to subjective errors. As a result, it is critical and urgent to develop an intelligent and effective system for identifying tomato illnesses [208]. Machine learning is often used to detect leaf diseases since it is an intelligent and effective technology. To identify tomato leaf diseases using machine learning, hand-crafted feature extraction from leaf photos is supplied into the classification system [209]. In addition, the traditional extraction of features method is still not up to mark, which reduces the efficacy of the tomato leaf illness diagnosis. With the rise of deep learning, convolutional neural networks (CNN), recognized as an impartial and scientific technique, have been commonly employed in the detection of tomato leaf illness. As a from-start-to-finish neural network, the input of a convolutional neural network is the initial image data, and the output is the outcome of classification after training via the model [210]. The application of deep learning architectures in image-based plant disease prediction and classification is extremely prevalent. Till now, several studies have employed various deep-learning architectures for plant disease detection via photos of sick leaves. Furthermore, deep learning (DL) is more popular among researchers than ML since it includes feature extraction layers. However, the classification layer of a DL model necessitates the use of a large amount of processing power and resources, which is not the case with typical ML techniques [211]. According to the above research, while cutting-edge methods may demonstrate ideal performance, they are computationally costly and practically impractical for real-time processing. The extreme learning machine (ELM) is a developing theory for single-layer feed-forward networks (SLFNs) that exhibits greater generalization efficiency and faster learning than gradient-based methods, as well as the production of hidden node parameters without human interaction [212]. Due to the underlying qualities, it has recently been discovered that ELM theory and its variants provide significant growth in the sphere of smart agricultural applications like predictive agriculture, identification of diseases, and automated planting.To the best of our ability, in this

paper, a hybrid model (TLMV2-ELM) is developed to take advantage of the benefit of both DL and ML models while also addressing the flaws of both methods to provide more exact and less computationally costly results. The key objective of this paper is to predict leaf disease in tomato crops using the suggested (TLMV2-ELM) framework. In our method, the MobileNetV2 model has been used for feature extraction from the tomato dataset, which is taken from the PlantVillage dataset. The extracted features have been utilized to train the ELM classifier to identify leaf disease. In brief, our contributions are as follows

1. We have proposed a hybrid model (TLMV2-ELM) that incorporates MobileNetV2 and ELM classifier for identifying disease in tomato leaves.
2. The model's performance is measured using a multi-class classification method that splits disease into ten distinct illness classes.
3. The suggested model is assessed and tested using the publicly available tomato dataset which is taken from the PlantVillage dataset and compared with cutting-edge techniques based on different parameters.

3.3.2 RELATED WORK

Plant disease diagnosis techniques based on machine learning technology are routinely used to determine the shape, color, texture, and other properties of disease patches. This technique has a low detection efficiency because it depends on substantial specialist knowledge of agricultural diseases [213]. Numerous academics have performed considerable studies using deep learning methods to increase the accuracy of plant illness detection in the past few years, because of the fast advancement of AI technology. A wide range of available plant disease analysis tools depend on disease categorization.

Govardhan, et al. [214] have proposed a method to detect and recognize plant disease using a random forest machine learning algorithm. The photos are cleaned up, and features are obtained to train the ML algorithm. In addition, the testing results reveal that almost all test photos provide the best classification between image classes. The system's performance is estimated to be 95%. Plant illnesses can be difficult to identify since symptoms resemble several nutritional deficits, such as Mosaic Virus in tomato leaves. The main challenge behind the model is too difficult to generalize on the new or unseen conditions. In 2016, Owomugisha, et al. [214] proposed an ML-based forecasting method for cassava plant disease. The authors used the two most used approaches for extracting features from a picture

collection- the HSV color transformation being one of these, and ORB methodology being the other, which stands for Oriented FAST and Rotated BRIEF and is a robust method for obtaining independent elementary information from photos. The main limitation of this paper is enormous quantities of high-quality data and model complexity which must be balanced to avoid overfitting in training data.Picon, et al. [215]proposed an updated deep neural network-based technique to diagnose several plant ailments in real-world acquisition situations. Numerous improvements have been suggested for quick sickness identification. Every sickness evaluated had an accuracy classification score of more than 0.80. The main limitation of this paper is to ensure that models generalize well to new, previously unknown data. Also, it ensures that models are capable of handling huge datasets while being efficient. Ashok S, et al. [216] proposed a CNN method for the hierarchical extraction of features that maps the initial image pixel values and compares them to the image from the trained dataset. The compared image has been divided into disease-affected and healthy leaf. The limitation of this paper is data imbalance. A skewed distribution of healthy and unhealthy samples can skew the model's forecasts, resulting in higher performance for the larger class and lower results for the minority class. Moussafir M, et al. [217] designed a hybrid model for identifying tomato-related disorders. In the initial phase of this paper, the authors evaluated the efficacy of 7 CNN techniques to identify tomato leaf illnesses. Then, in the second half, two models were chosen to create a weighted average group. Following that, two high-performing models were chosen to construct CNNGA models. Then, using a weighted average ensemble, the author merged 2 CNN-GA methods to create a hybrid technique. In this research, an average-weighted group produced the best outcome, having an accuracy of 98.1%. The main challenge is to enhance the classifier accuracy and improve the error rate. Altalak M, et al. [218]proposed a hybrid approach that is capable of detecting nine distinct tomato illnesses. The proposed system is compact and effective; therefore, the farmer can set it on any electronic device with an image sensor and data analysis capacity. The limitation of this paper is model complexity and less precise results compared to other existing systems.

3.3.3 Proposed Methodology

Our major goal is to develop a hybrid method that can categorize input plant leaf images into different classes. The type of illness is also determined if a sickness is discovered on a plant leaf.

This section delineates the problem statement and presents a comprehensive research method-

ology that comprises various sequential steps adopted in this research work. Additionally, the section outlines the proposed algorithm in a step-by-step manner and presents a complete flowchart of the overall process.

3.3.3.1 Proposed Hybrid Framework

The current study focuses on leaf disease identification and classification using the DL subfield of transfer learning and the ML subfield of ELM. We propose the TLMV2-ELM model, which can categorize and diagnose leaf disease from a range of leaf datasets, to achieve this goal. The major objective of this effort is to increase the accuracy of the model. The concealed nodes simplicity aids in classifying and detecting anomalies.

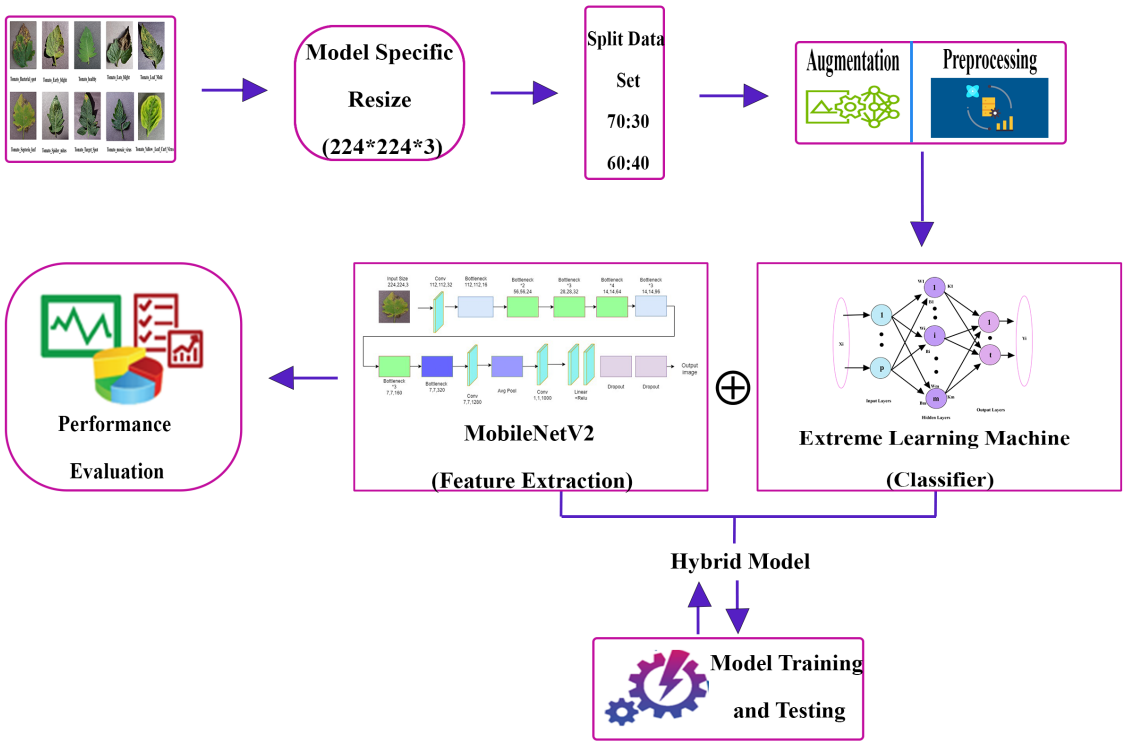


Figure 3.35: Proposed hybrid framework

In Figure 3.35, firstly, the tomato dataset is taken, and then its size is reduced to the size of the specific model. In addition, the dataset is divided between 70 percent training and 30% testing. Then argumentation and several preprocessing techniques such as resizing, cropping, rotation, smoothing, and so on are applied. Image cropping is used to achieve the desired image area, smoothing is used to smooth out images, and image segmentation is used to discover image areas and objects. In the next step, MobileNetV2 is used for feature extraction to determine the object. Features are traits or characteristics that can aid in the differentiation of healthy and sick crops. After feature extraction, TLMV2-ELM categorizes

the processed data image. Training and testing are performed on the hybrid model for the tomato dataset. Finally, the hybrid model's efficacy on the tomato dataset is tested.

In summary, the TLMV2-ELM model enhances performance on image classification tasks by combining the feature extraction capabilities of MobileNetV2 with the quick and effective training of the ELM method. 3.4 is used for training and testing of the TLMV2-ELM model for image categorization. The algorithm begins with two primary inputs, which are labels, a training and testing dataset, and one output, which is capable of identifying images into predetermined classes. Before starting the train and test loops, we construct a hybrid model by performing the following steps: Lines 1–2 show where the training loop starts for each epoch. In Lines 5–11, the forward cycle has been run via a hybrid model to acquire output and calculate avg_training loss and avg_training_acc. In addition, lines 13–15 show that the testing loop starts for every epoch. Then, calculate the loss function using the cross-entropy. Further, the trained DTLVGG16-ELM model is created after all epochs have finished. Furthermore, for each epoch, store the train and testing accuracy and loss data in the model.

The time complexity of the proposed model is $O(m \times n)$, where m denotes the number of layers and n represents the size of the dataset.

3.3.4 Experimental Result

3.3.4.1 Experimental Setup

The proposed hybrid model has been carried out on NVIDIA GTX 1070 GPU. For the implementation of the proposed hybrid model, Colab, Python 3.8, and the Keras library have been used.

3.3.4.2 Dataset Description

The proposed algorithm has been tested against the implementations discussed in research work over an openly available image dataset: Tomato. These datasets have been taken from the public standard dataset, PlantVillage [99], which has over 54000 photos representing 14 different species with healthy and damaged leaves. A brief description of the datasets and their composition is discussed in Chapter 2, Section 2.4.

Algorithm 3.4 Training and Testing Loop for the Proposed TLMV2-ELM Model

Require: Labels, Training dataset, Testing dataset
Ensure: Trained and tested TLMV2-ELM model

Training Phase
for each training epoch **do**
    Set the hybrid model (TLMV2-ELM) to training mode
    Initialize $total\_train\_loss$, $correct_{train}$, $total_{train}$
    for each batch in the training dataset **do**
        Transfer input images and labels to the device
        Perform forward propagation through the hybrid model
        Compute cross-entropy loss
        Perform backward propagation and update model parameters
        Update $total\_train\_loss$, $correct_{train}$, and $total_{train}$
    end for
    Compute average training loss and training accuracy
end for

Testing Phase
for each testing epoch **do**
    Set the hybrid model (TLMV2-ELM) to evaluation mode
    Initialize $total\_test\_loss$, $correct_{test}$, $total_{test}$
    for each batch in the testing dataset **do**
        Transfer input images and labels to the device
        Perform forward propagation through the hybrid model
        Compute cross-entropy testing loss
        Update $total\_test\_loss$, $correct_{test}$, and $total_{test}$
    end for
    Compute average testing loss and testing accuracy
end for

Result:
Store training and testing loss and accuracy for each epoch

3.3.4.3 Training and Testing Result for Tomato Dataset

The outcomes of training and testing for the Tomato Dataset are analysed in this section. The dataset is separated into two parts: 70% is set aside for training, while the remaining 30% is set aside for testing. Then, preprocessing techniques such as normalization, resizing, zoom, etc. are applied to remove the overfitting issues. Further, the training and testing are applied to the hybrid model. The obtained result is shown in Table 3.30.

As illustrated in Figure 3.36 and Figure 3.37, the performance of the TLMV2-ELM suggested hybrid model is examined by demonstrating the connection between training and testing accuracy as well as training and testing loss. The horizontal axis in Figure 3.36 and

Table 3.30: Training and Testing Outcomes of Leaf Disease Detection Using Tomato Dataset

Model	Training Accuracy	Training Loss	Testing Accuracy	Testing Loss
TLMV2-ELM	0.989	0.054	0.990	0.067

Figure 3.37 reflects the number of epochs, whereas the vertical axis represents accuracy and loss, respectively. When the number of epochs is maintained to a minimum, the accuracy is low. However, as the number of epochs increases, so does the accuracy [219]. This usually reflects the model's learning process as it improves at capturing patterns in the training data. Furthermore, the performance of these models can be shown in Figure 3.37, where the loss is initially large with a few numbers of epochs. Yet, as the amount of epochs grows, so does the loss decrease. information is used to assess the model's ability to generalise its learning to novel and previously unseen data.



Figure 3.36: Training and testing accuracy of hybrid model

3.3.4.4 Comparison with Cutting-Edge Techniques

This section compares the outcomes of existing models to our hybrid model (TLMV2-ELM) that employed the same tomato disease datasets, which are taken from the PlantVillage dataset, and analyzed a comparable amount of disease classes (10 diseases). Table 3.31 shows the outcome for these current approaches ([220], [221], [206], [222]) across a range of parameters. For the multiclass classification of the tomato leaf dataset, the suggested hybrid model (TLMV2-ELM) exceeds previous models in terms of final accuracy and final loss. One key advantage of TLMV2-ELM is that it minimizes the challenge of the desired

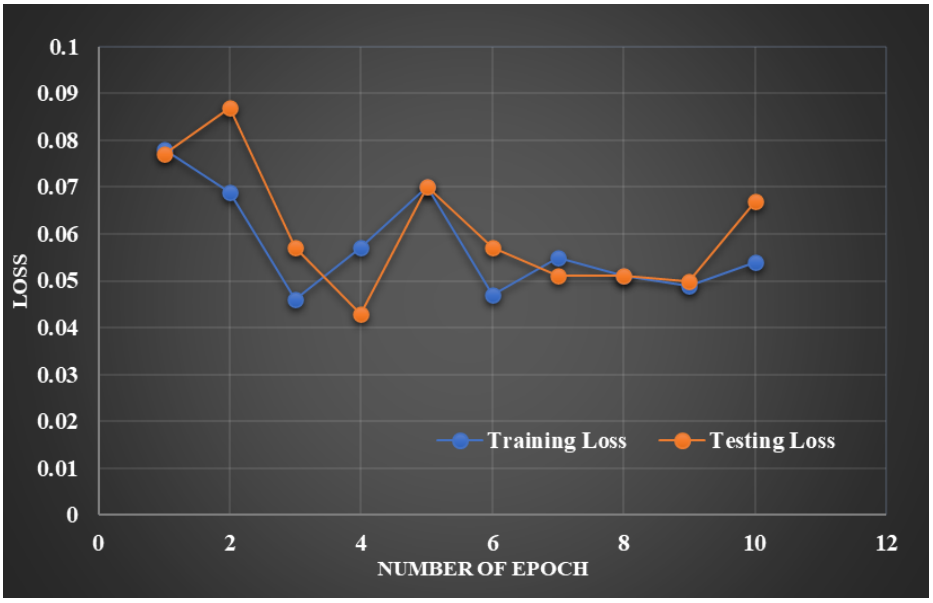


Figure 3.37: Training and testing loss of hybrid model

idea by utilizing concealed node sections to aid in categorization and the identification of anomaly concerns.

Table 3.31: Comparison of The Proposed Model With The Existing Techniques On The Tomato Leaf Dataset

Model	Dataset	Final Accuracy	Final Loss
MobileNetV2 [220]	Tomato Dataset	0.75	0.59
MobileNetV2 [221]	Tomato Dataset	0.90	0.33
MobileNetV2 / NASNetMobile [206]	Tomato Dataset	0.93	0.20
Tan *et al.* [222]	Tomato Dataset	0.91	0.29
ELM	Tomato Dataset	0.97	0.07
Proposed TLMV2-ELM	Tomato Dataset	**0.99**	**0.06**

Figure 3.38shows the bar graph representation of existing techniques versus the proposed hybrid model (TLMV2-ELM) based on accuracy and loss. The result shows that for the tomato leaf dataset, the proposed TLMV2-ELM model has the highest accuracy 0.99% which is 0.24-0.08% better, lowest loss 0.06 better than other cutting-edge techniques. Overall, the proposed hybrid methodology (TLMV2- ELM) outperforms current techniques and can categorize various leaf diseases [223].

Several illnesses can harm tomato plants and reduce production. Deep learning has come to be a viable method for overcoming this problem and avoiding losses in yield. The purpose of this paper is to identify the optimum hybrid model for tomato disease diagnosis. In this paper, we have proposed a hybrid model TLMV2-ELM to detect many diseases in tomato leaf. TLMV2-ELM model combines the advantages of both structures, resulting in

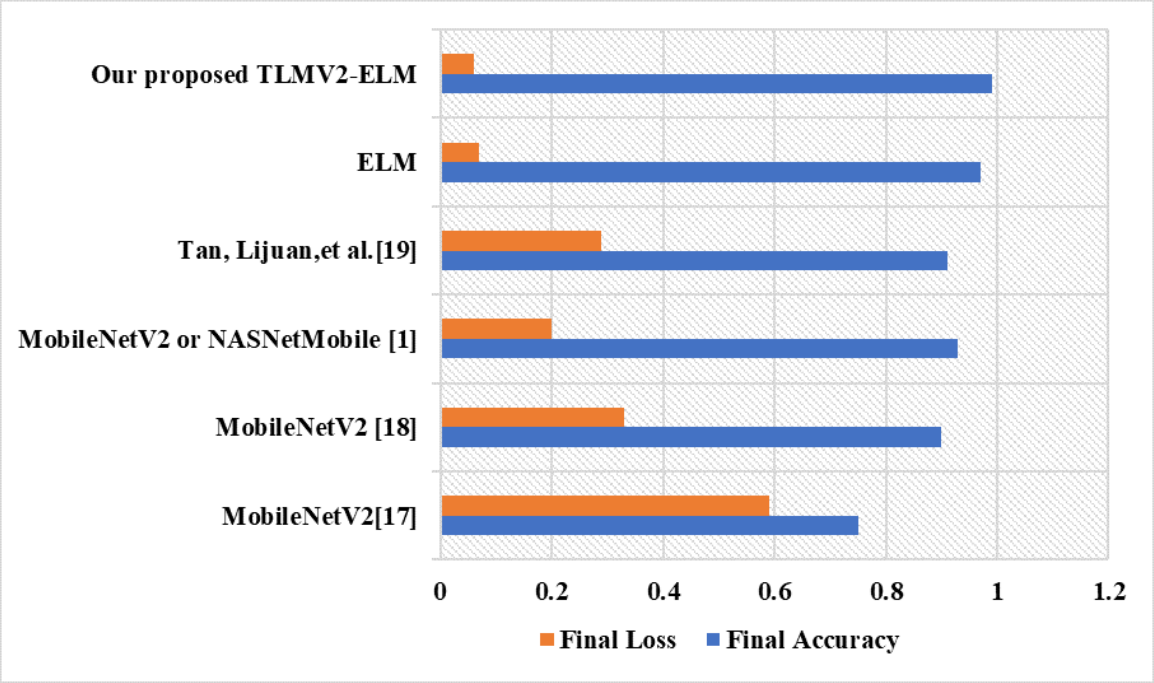


Figure 3.38: Graphical representation of the proposed model with existing state-of-the-art techniques on the different parameters

an effective, rapid, and efficient solution for an extensive variety of ML applications. The outcome indicate that the TLMV2-ELM model has the highest testing accuracy with 0.99% and the lowest loss with 0.06 than the other cutting-edge techniques. This strategy assists farmers in making informed judgments about effectively managing the disease in the field. We hope to modify the model in the future by using a greater number of photos using various plant datasets.

3.4 Summary

In this chapter, we present Transfer Learning Approaches to Plant Leaf Disease Classification, aiming to develop high-accuracy, automated models for detecting agricultural diseases. The chapter includes numerous researchers' results to overcome important gaps in agricultural image analysis, such as overfitting, expensive computation, a scarcity of training data, and poor generalization. Several pre-trained deep convolutional neural network models, including VGG16, MobileNetV2, Xception, InceptionV3, and DenseNet variants, are tested as feature extractors, and their effectiveness is increased by combining them with Extreme Learning Machine (ELM) classifiers to achieve faster training and greater stability.

Extensive experimentation is conducted on diverse plant leaf datasets comprising crops such as peach, strawberry, grapes, apple, cherry, and tomato. The comparative evaluation

demonstrates that transfer learning–based hybrid frameworks achieve superior classification accuracy, often approaching or exceeding 99%, while maintaining reduced training time and balanced model complexity. The findings highlight the effectiveness, robustness, and practical applicability of transfer learning strategies in building reliable plant leaf disease classification systems for real-world agricultural environments.

Chapter 4

Attention-Driven Deep Learning Models for Plant Leaf Disease Classification

*The purpose of this chapter * aims to describe an attention-based transfer learning model for plant leaf disease classification that incorporates Vision Transformer-based architectures and lightweight deep convolutional neural networks. The proposed models, which combine two multi-head self-attention layers and attention-enhanced residual networks, are effective at targeting disease-specific parts of leaf images by leveraging knowledge from large-scale pre-trained datasets. AgriTL-ViT model presents Transformer-ResNet encoder blocks which are more computationally cost-effective to extract features in a hierarchical manner, whereas the ADResNet with Attention model uses asymmetric and dilated convolutions to refine the feature representation as well as classification accuracy. These attention-based techniques enhance generalization of models, decrease overfitting and have a balance between computation efficiency and predictive accuracy. The suggested model is tested on tomato, maize, rice, and CCMT data and contrasted with state-of-the-art methods using several metrics, including accuracy, precision, recall, f1-score, and loss. The findings indicate that attention-based transfer learning methods can offer high-quality, scalable, and resource-efficient methods to automated plant disease diagnosis in contemporary precision agriculture.*

*The research work covered in this chapter has been Communicated/published in Meenakshi Srivastava, Jasraj Meena, "AgriTL-ViT: A Vision Transformer Model with Attention Techniques for Classification of Plant Leaf Disease," *Expert Systems with Applications*, Elsevier, vol. 294, pp.128793, 2025, DOI: https://doi.org/10.1016/j.eswa.2025.128793.
&
Meenakshi Srivastava, Varsha Sisaudia, and Jasraj Meena, "ADResNet with Attention: A Multi-Model Approach for Leaf Disease Detection and Pest Classification," *signal image and video processing*, Springer, 2025. **(SCIE, IF= 2.1)**

4.1 AgriTL-ViT: A Vision Transformer Model with Attention Techniques for Classification of Plant Leaf Disease

4.1.1 Overview

In recent years, Artificial Intelligence (AI) breakthroughs have changed a diversity of industries, particularly farming. One significant use is the categorization of illnesses of leaves using artificial intelligence techniques. Initially, Machine Learning techniques were used to address these challenges, but they proved insufficient in dealing with the intricacies of agricultural data. Many researchers have used various Machine Learning (ML) algorithms to determine plant disease based on symptomatic leaf photos. While ML can recognize plant diseases from digital leaf photos, it has two significant limitations. Firstly, these methods cannot autonomously acquire temporal and geographical picture features for classification purposes. Secondly, they cannot efficiently handle huge image files due to their lack of support for Graphic Processing Units (GPUs). In recent years, advancements in deep learning technology have led to the widespread use of picture identification and categorization techniques for recognizing and categorizing agricultural illnesses. Researchers have employed Deep Learning (DL) technologies, specifically Convolutional Neural Network (CNN), to automate plant disease detection, overcoming traditional methodologies limitations [224].

Vision Transformer (ViT) has recently contributed significantly to computer vision. Global attention-based models benefit medical diagnosis since they understand the interconnection of global features. It consists of numerous encoder blocks linked. The ViT model encoder blocks consist of Multi-Head Attention, Layer Normalisation, and MLP components [225]. Many researchers discovered that it surpassed modern CNN systems. Researchers have utilized the ViT technique to recognize plant illnesses autonomously. In addition, The ViT model performs well but has a significant disadvantage: it includes an MLP unit in its encoder block, which is computationally costly and ineffective for collecting spatial and temporal features from images.

Despite tremendous advances, current tomato, maize and rice disease detection and grading systems still confront numerous hurdles. First, datasets with partial scales and imbalanced classes may result in overfitting throughout model testing and training. Further, labelling

the datasets needs typically laborious and exclusive professional aid. Furthermore, modern methods fail to segment instances precisely, making them pricey for practical use. As an outcome, we developed a less computationally exclusive AgriTL-ViT (Agriculture Transfer Learning Vision transformer) model for recognizing plant illnesses and overcoming these constraints. The primary contribution of this work is given below:

- We have proposed the "AgriTL-ViT" model to increase a computing efficiency of the Vision Transformer (ViT) model for recognizing illness of plant over image analysis. The model is made up of modified encoder blocks known as Transformer ResNet blocks. Transformer ResNet blocks use ResNet modules rather than MLP modules to extract hierarchical characteristics from input images successfully.

- To enhance the model's ability to capture both local and global contextual information, we included a dual multi-head self-attention mechanism. This update improves the model's expressive strength and adaptability, allowing it to focus on complex visual characteristics that are important for reliably recognizing disease symptoms.

- The model performance is assessed with a multiclass categorization technique that examines numerous metrics (confusion metrics, precision, recall and f1 score) to measure its utility in distinguishing between the multiclass.

- To demonstrate its efficacy, the performance of AgriTL-ViT is compared to many existing cutting-edge models. The results show that the proposed model performs competitively or better across all three datasets, demonstrating its resilience and generalizability.

- For model explainability and interpretability, we used visualization approaches such as Grad-CAM, LIME, and SHAP. These methods benefit domain specialists in understanding model behaviour by allowing them to visualize the regions of the leaf images that had the greatest influence on the model's predictions.

- To improve the model's robustness under real-world conditions, we used targeted data augmentation techniques such as ColorJitter (for lighting change) and GaussianBlur (for background noise), resulting in better generalization to various field contexts. In addition, the model was tested on low-resolution photos (128×128 and 200×200) and retained good accuracy, indicating its effectiveness in low-resource deployment circumstances.

4.1.2 Related Work

In farming, AI has been widely used for various identification tasks such as leaf recognition, face recognition, etc. The next part thoroughly assesses the study on plant disease recognition. Most of the research focuses on classification methods that heavily rely on machine learning, deep learning, and transformer models, as stated in the subsection. The shortcomings in present approaches are also addressed in the subsection, which will be used to create an effective method. ML is crucial for identifying and treating plant diseases. It has various advantages over traditional approaches, including the capacity to detect and control diseases more quickly, correctly, and at a lesser cost. Also, DL plays a vital role in plant disease diagnosis by utilizing its image processing and pattern identification abilities. The following is an in-depth review of existing approaches.

4.1.2.1 ML-based approaches

Geetha G et al. [226] used four steps to determine the kind of sickness. The four processes are categorization, leaf segmentation, pre-processing, and feature extraction. Additionally, the author seeks to detect prevalent illnesses on tomato leaves through image processing and ML. Further, the author compares the suggested method to previous systems using appropriate techniques and execution. Furthermore, the suggested method outperforms existing disease identification systems regarding accuracy, precision, and ease of deployment. This study's drawbacks include overfitting, fine-tuning challenges, and conventional methodologies failing to determine disease severity accurately. Liu G et al. [227] suggested a technique for automatic tomato identification in normal colour photos, reducing the impact of illumination and diffraction. In addition, HOG descriptors have been used in this study. The classification problem was implemented using an SVM classifier. A coarse-to-fine framework was utilized during scanning, followed by an FCR approach to remove false positives. Finally, NMS was employed to determine the results. The outcomes indicate that a classifier using only HOG features effectively distinguishes tomatoes from their surroundings. The proper detection rate is 90.67% in bright settings and 89.33% in shaded situations. Yu H, et al. [228] suggested a technique that utilizes K-means clustering and an enhanced deep learning framework to precisely diagnose three prevalent diseases seen in maize leaves: rust, spot leaf, and gray spot. To identify 3 diseases, employ the K-means technique to cluster instances photos and input it into the enhanced DL method. The experimental findings suggest that the approach has a highly significant identification impact on samples with

32 means. Furthermore, the analytic recall rates for leaf spot (89.24%), rust (100%), and gray spot disease (90.95%). The study found that this technology is computationally intensive and expensive. Panigrahi K et al. [229]suggested using supervised machine learning methods, including Random Forests (RF), K-Nearest Neighbours (KNN), Support Vector Machines (SVM), Naive Bayes (NB) and Decision Trees (DT) to identify different illnesses affecting maize leaves. The suggested solution was implemented using labelled image data to train the categorization model. The RF classifier outperformed the other classification techniques when used for illness identification in testing image data. The fundamental limitations of this research are the vast volume of high-quality information and the difficulty of models that must be managed to prevent overfitting in training data. Jena K et al. [230] suggested an ML-based method for identifying rice illness. This study examines various machine learning approaches, including NB, DT, k-NN, RF, NN, LR, SVM, and AB for categorization purposes. This study employs a cross-validation technique with NFs of 2, 3, 5, 10, and 20. The study found that NN outperforms other approaches to CA, including LR, AB, k-NN, DT, RF, NB, and SVM across every scenario.

4.1.2.2 DL-based Approaches

Unlike machine learning, deep learning methods can immediately input data without requiring constructed characteristics. In today's world, computational control provided by high-performance computing (HPC) and graphics processing units (GPU) permits practical training of complex method by concurrently applying concurrency in processing. Several DL methods have been suggested to identify illness in leaf photos.

Wang Q et al. [231] proposed tomato disease identification systems using advanced CNN and recognition of object algorithms. In addition, these strategies employ two models: Faster R-CNN and Mask R-CNN. Faster R-CNN identifies tomato illnesses, while Mask R-CNN detects and segments diseased areas. Further, experiments with internet data demonstrate that the suggested approaches accurately and efficiently identify tomato diseases and segment afflicted parts. Furthermore, ResNet-101 has the highest identification rate while it necessitates the longest length for both training and identification. ResNet-101 outperforms MobileNet in terms of accuracy, although having a faster detection time. The main challenge in the studies is identifying diseases due to poor image resolution. Alatawi et al. [232] suggested a method for recognizing plant diseases by testing leaf photographs. The technique not only determines whether plants are healthy or ill but also classifies the type of disease for each crop category. Our approach uses a VGG- 16 framework to classify 19

diseases of plants using Plant Village dataset. The method obtained 95.2% accuracy with a loss of 0.4418. Abbas A. et al. [233] provided a deep learning method for tomato disease diagnosis that uses the Conditional Generative Adversarial Network (C-GAN) to produce artificial images of tomato plant leaves. After that, the DenseNet121 model was trained on artificial and actual images via transfer learning to categorize tomato leaves into ten illness groups. The suggested model was thoroughly trained and tested using the publicly accessible PlantVillage dataset. Moreover, the suggested method classified tomato leaf images into 5 classes, 7 classes, and 10 classes with accuracy rates of 99.51%, 98.65%, and 97.11%. The proposed strategy outperforms current techniques. Paul S et al. [28]proposed a lightweight custom CNN algorithm and transfer learning-based method to categorize tomato diseases. This paper simulates several tomato leaf diseases using eleven classes, including a healthy one. Also, a study was conducted to determine the appropriate parameters for the suggested model. Further, assessment measures were employed to examine the effectiveness of the suggested and TL-based approaches. Furthermore, using data augmentation approaches, the suggested approach obtained 95.00% accuracy and recall, greater than all other models. Finally, the most effective model was used to create a Web-based system for farmers to recognize tomato leaf illness. Haque M et al. [234] recommended a dl strategy to recognize sick maize crop photos in the field. This research used a baseline training approach to train three models utilizing the 'Inceptionv3' network architecture on unhealthy maize photos. According to the experiment's results, Inception-v3_GAP obtained 95.99% accuracy in a testing. The model proved highly effective in extracting essential features from illness symptoms and accurately recognizing class in new data. This paper suggests that DL method shows excellent potential in accurately recognizing symptoms of crop diseases.

Wang Y et al. [235] introduced the ADSNN-BO model, which is based on the MobileNet framework and has an augmented attention method. Furthermore, the Bayesian optimization technique is utilized to optimize the hyperparameters of the model. Classification tests using cross-validation are performed using a publicly available dataset of rice diseases, which consists of four categories. The results demonstrate that our mobile-compatible ADSNN-BO model achieves a test accuracy of 94.65%, surpassing all the examined cutting-edge methods. This paper limitations include model complexity and less exact outcomes. Wang Y et al. [236] suggested a multi-grained feature extraction technique built on a ViT to recognize tomato leaf illnesses in Beijing. Further, this study uses visual representations to identify and classify tomato disease photos. Furthermore, the outcomes of datasets gathered from various areas demonstrate that using visual transformers and CNN learning for farming

difficulties offers excellent potential. The current research work has been evaluated based on dataset, accuracy, and technique, as shown in Table 4.1. Most research studies on leaf disease detection use datasets such as Tomato, PlantVillage, Cassava leaf, and Wheat Rust.

191

Table 4.1: An Overview of Literature Work on Plant Disease Detection

Author	Techniques	Dataset	Acc.	Prec.	Rec.	F1	Limitations
Geetha G et al. [226]	KNN	Tomato Leaf	✓	✗	✗	✗	Overfitting and difficulty in severity detection.
Liu G et al. [227]	HOG, SVM	Tomato Leaf	✗	✓	✓	✓	Overfitting and inaccurate severity assessment.
Yu H et al. [228]	K-Means	Maize Leaf	✓	✓	✓	✓	Limited real-time experimental capability.
Panigrahi K et al. [229]	SVM, KNN	Maize Leaf	✓	✓	✓	✓	Large data volume causes overfitting issues.
Jena K et al. [230]	RF, KNN, DT, NN	Rice Leaf	✓	✗	✗	✗	Not comparable to prioritized expertise.
Azim M et al. [237]	XGBoost, SVM (RBF)	Rice Leaf	✓	✗	✗	✓	Limited detailed experimental analysis.
Wang Q et al. [231]	Faster R-CNN, Mask R-CNN	Tomato	✓	✗	✗	✗	Low image resolution affects detection.
Alatawi et al. [232]	VGG-16	PlantVillage, Rice	✓	✗	✗	✗	Lack of detailed outcome analysis.
Abbas A et al. [233]	C-GAN	Tomato Leaf	✓	✓	✓	✓	Scalability and computational speed issues.
Paul S et al. [28]	VGG16, VGG19	Tomato Leaf	✓	✓	✓	✓	No real-time implementation study.
Haque M et al. [234]	Inception_V3 + GAP	Maize Leaf	✓	✗	✗	✗	Longer training time required.
Wang Y et al. [235]	LeIAP	Rice Leaf	✓	✓	✓	✓	Model complexity and precision issues.
Wang Y et al. [236]	ViT + CNN	Drone-based Tomato	✗	✓	✓	✓	Extensive training period required.

4.1.2.3 Research Gap and Motivation

Several ways have been proposed to identify illnesses in plant leaves, but substantial limitations exist.

1. Discovering and extracting important characteristics in plant leaves makes it difficult to distinguish between illnesses via traditional image-processing approaches.
2. The performance of ML methods exclusively depends on the manually gathered characteristics. Therefore, automating feature extraction can help select and acquiring the most suitable set of features for categorization purposes.

47

3. Some deep learning models use known frameworks such as VGG16, MobileNet, etc.

Therefore, it uses millions of criteria for categorization. Real-time implementation of these models requires balancing computing cost and accuracy.

4. The DL methods must be trained on a huge sample set to improve feature generalization.

The suggested framework for plant leaf illness classification has been discussed in the next section.

4.1.3 Model Development

4.1.3.1 Proposed Vision Transformer Model (AgriTL-ViT)

48

A vision transformer (ViT) is a transformer-like structure that performs image processing operations. The ViT model consists of layered encoder blocks with three components: Multi Head Attention, Layer Normalisation, and MLP (Multi-Layer Perceptron). The Multi Head Attention component also parallelizes self-attention procedures, allowing the model to identify global relationships between image patches [238].

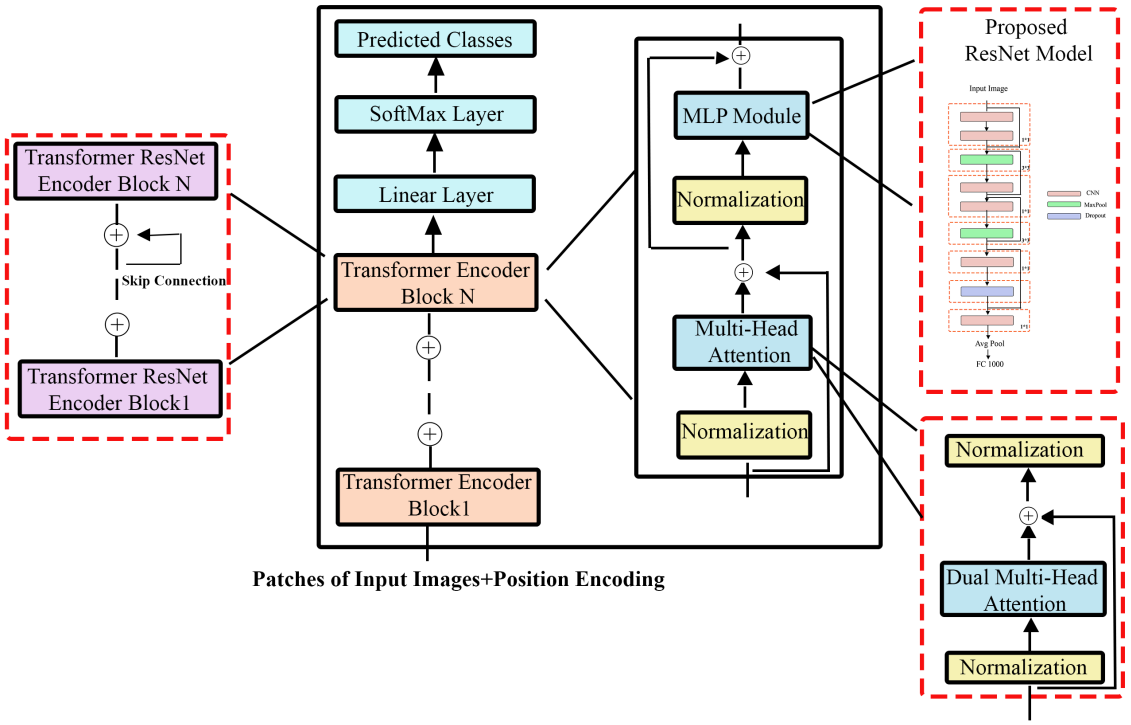


Figure 4.1: Structural representation of the modified pre-existing ViT Transformer

However, this attention component has a significant limitation: It does not improve the model's ability to gather and analyze visual information efficiently. Moreover, The MLP

module uses two closely linked layers to obtain different characteristics from image patches. The MLP module has a severe limitation: every layer is closely coupled. The ViT model's training requires many weight parameters, making it highly computational.

Figure 4.1 shows the structural representation of a pre-existing ViT Transformer and the modification we made in red dotted boxes. To solve the MLP module's limitation, we have replaced the MLP module with the Proposed ResNet model in the transformer encoder block, as depicted in the diagram. In addition, the ResNet model is preferred over the MLP module due to its ability to share parameters and reuse features effectively through skip connections. Further, we have replaced multi-head attention with Dual multi-head attention to increase the model's capability to effectively collect and examine visual data.

Hence, the framework of the suggested vision transformer (AgriTL-ViT) and transformer ResNet block is demonstrated in Figure 4.2. It introduces a AgriTL-ViT method that addresses the limitations of the ViT method. To overcome the problems with the ViT technique, the MLP module in the transformer encoder block has been replaced by the ResNet model. This changed block is known as the "Transformer- ResNet Encoder Block". Additionally, the ResNet model was selected over the MLP module as of its ability to share parameters and reuse features efficiently via skip connections. The skip connections permit the model to achieve identity mapping, which rises gradient flow during training. Moreover, this proposed framework contains dual multi-head self attention to improve the model's ability to gather and analyze visual data successfully.

Also, integrating dual multi-head self-attention in a vision transformer encoder block permit the framework to efficiently capture spatial and channel information, resulting in a more reliable and adaptable depiction of visual input. The suggested AgriTL-ViT encoder blocks include of three modules: Dual Multi-Head Self Attention, Layer Normalization and ResNet Model. One of these three parts is normalization, which is derived from the existing visual transformer encoder block. Additionally, the remaining two are included in the proposed Trans-ResNet encoder block. The components of the Transformer ResNet encoder block is outlined below:

4.1.3.2 Dual Multi Head Self Attention Module

Dual Multi-Head Self Attention is a method that processes input data twice sequentially using multiple heads. This module uses m heads to perform self-attention operations concurrently. In this study, 12-head attention has been used. In addition, self-attention is

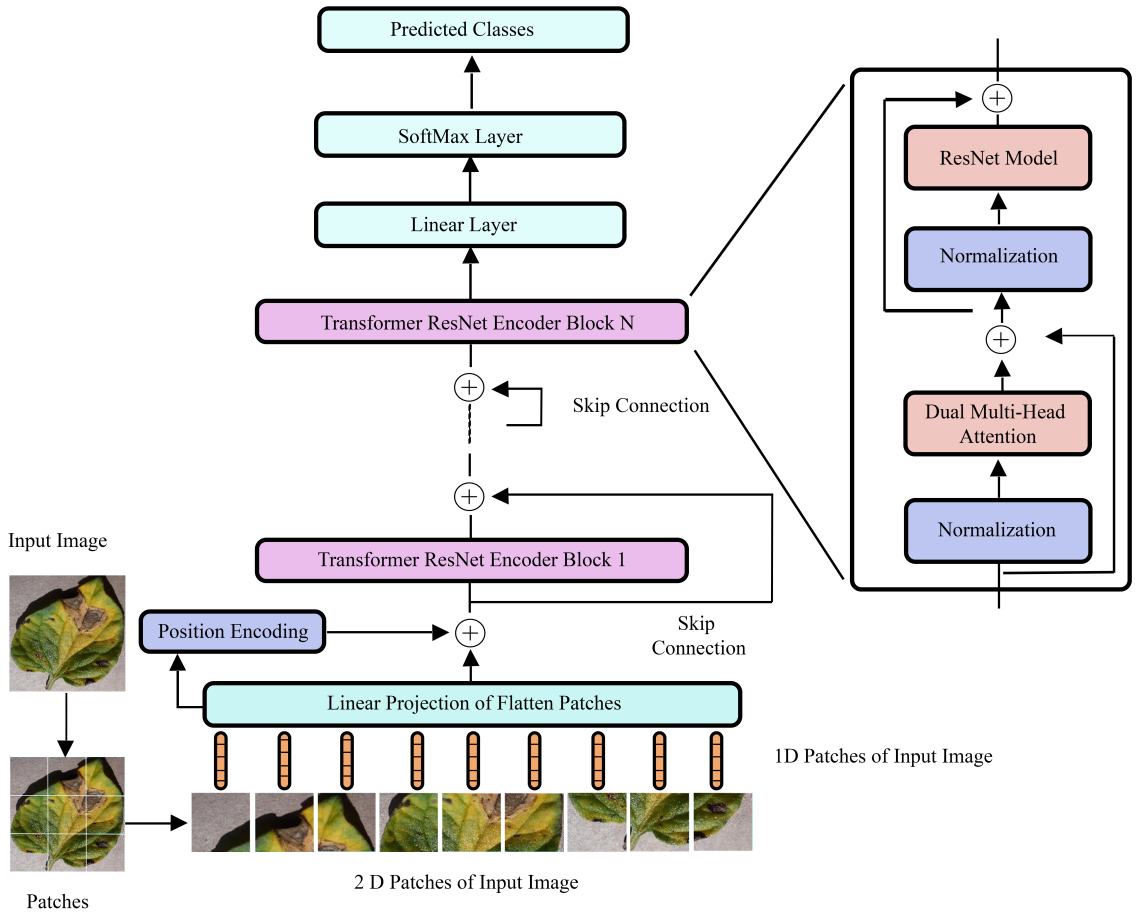


Figure 4.2: AgriTL-ViT model architecture

inspired by the human eye's ability to focus on relevant information while ignoring less important details. This technique captures the correlation between different regions of an image.

Moreover, the dual multi-head self-attention block is divided into two stages. In the first stage, a single multi-head self-attention unit is directed toward capturing spatial relationships between different regions of the input image. This involves splitting the input into multiple heads, computing attention independently for each head, and then concatenating the results. Consider patches represented as $X \in \mathbb{R}^{O_p \times dim}$, where dim denotes the embedding dimension. Furthermore, the backpropagation algorithm is used to train three learnable weight matrices: Query, Key, and Value, where dim_q, dim_k, and dim_v represent the number of columns in the respective weight matrices. The dual multi-head self-attention process involves multiplying the input tensor X with learnable matrices to obtain Q_{fs1}, K_{fs1}, and V_{fs1}, which are represented as follows:

First Phase:

Query for the first phase:

$$Q_{fs1} = X \times W_{Q1} \tag{4.1.1}$$

Key for the first phase:

$$K_{fs1} = X \times W_{K1} \tag{4.1.2}$$

Value for the first phase:

$$V_{fs1} = X \times W_{V1} \tag{4.1.3}$$

Where, $fs1$ denotes the First Phase, X represents the input tensor, and W_{Q1}, W_{K1}, W_{V1} are the learnable weight matrices for the first phase.

After computing Q_{fs1}, K_{fs1}, and V_{fs1}, the attention output of the first phase is:

$$A_{fs1} = \text{SoftMax}\left(\frac{Q_{fs1}K_{fs1}^T}{\sqrt{d}}\right) \tag{4.1.4}$$

3 The outputs of m heads are concatenated and multiplied by an output weight matrix:

$$X'_{fs1} = A_{fs1} \times W_{out1} \tag{4.1.5}$$

Second Phase:

113 The output of the first phase is passed to the second phase multi-head self-attention module.

Query for the second phase:

$$Q_{fs2} = X'_{fs1} \times W_{Q2} \tag{4.1.6}$$

Key for the second phase:

$$K_{fs2} = X'_{fs1} \times W_{K2} \tag{4.1.7}$$

Value for the second phase:

$$V_{fs2} = X'_{fs1} \times W_{V2} \tag{4.1.8}$$

153 Where, $fs2$ denotes the Second Phase, X'_{fs1} is the output of the first phase, and W_{Q2}, W_{K2}, W_{V2} are the learnable weight matrices for the second phase.

The attention output of the second phase is:

$$A_{fs2} = \text{SoftMax}\left(\frac{Q_{fs2}K_{fs2}^T}{\sqrt{d}}\right) \tag{4.1.9}$$

The outputs of m heads are concatenated and multiplied with the output weight matrix:

$$X'_{fs2} = A_{fs2} \times W_{out2} \tag{4.1.10}$$

The tensor X'_{fs2} represents the final output of the Dual Multi-Head Self-Attention block.

4.1.3.3 Normalization Layer

This layer normalizes activations across the feature dimension. Unlike batch normalization, it does not require batching and normalizes features to zero mean and unit variance.

Mean calculation:

$$\mu = \frac{1}{p}\sum_{z=1}^{p} x_z \tag{4.1.11}$$

Variance calculation:

$$\sigma = \sqrt{\frac{1}{p}\sum_{z=1}^{p}(x_z - \mu)^2 + \epsilon} \tag{4.1.12}$$

Where p is the number of features and ϵ is a small constant.

Normalized feature:

$$x_z^{norm} = \frac{x_z - \mu}{\sigma} \tag{4.1.13}$$

Scaling and shifting:

$$y_z = g\, x_z^{norm} + b \tag{4.1.14}$$

4.1.3.4 Proposed ResNet Model

The proposed ResNet module performs six convolutional neural networks (CNN) with 1*1 filter and 3*3 Maxpool operation concurrently after two CNN and one dropout layer shown in Figure 4.3; therefore, it may acquire many temporal and spatial characteristics from images using various receptive fields. The proposed ResNet architecture uses skip connections, allowing efficient sharing of parameters and feature reuse. Additionally, the skip connections enable the framework to acquire and identify mapping, improving the gradient flow throughout training.

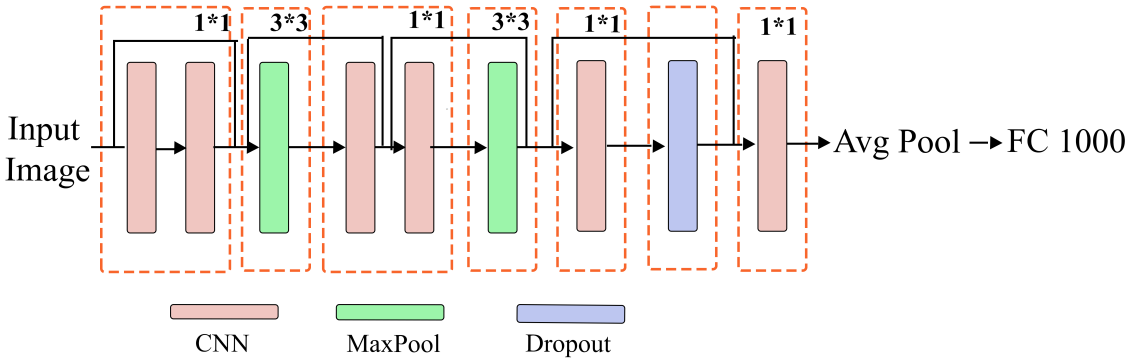


Figure 4.3: Proposed ResNet architecture

Let the input image size 224x224x3 be broken into 196 non-overlapping patches measuring 16×16. These patches are linearly projected onto a 768-dimensional embedding space and augmented with a [CLS] token, achieving:

$$Y \in R^{B_a \times T_e \times D_{im}} \tag{4.1.15}$$

$$Y \in R^{1 \times 197 \times 768} \tag{4.1.16}$$

Where B_a represents the batch size, T_e denotes the temporal or spatial dimension (such as width or sequence length), and D_{im} signifies the embedding or channel dimension.

The module commences with Layer Normalization:

$$\hat{Y} = \text{LayerNorm}(Y) \tag{4.1.17}$$

Layer Normalization normalizes the input according to the feature dimension. For every token (row of shape 768), the mean μ and standard deviation σ are determined and normalized as follows:

$$\hat{y}_i = \frac{y_i - \mu}{\sigma + \epsilon} \tag{4.1.18}$$

For a single vector $y = [x_1, x_2, \ldots, x_{768}]$, compute:

$$\mu = \frac{1}{768}\sum_{i=1}^{768} x_i \tag{4.1.19}$$

$$\sigma = \sqrt{\frac{1}{768}\sum_{i=1}^{768}(x_i - \mu)^2} \tag{4.1.20}$$

This yields an output of shape (1,197,768) – unchanged in size.

To capture fine-grained inter-channel interactions, two 1x1 Conv1D layers are applied successively to the normalized tensor. Input is reshaped to [1, 768, 197] for Conv1D.

$$\hat{Y} \rightarrow \hat{Y}^T \in R^{1\times 768\times 197} \tag{4.1.21}$$

$$F_1 = \text{Conv}_{1\times 1}^{(1)}(\hat{Y}) \tag{4.1.22}$$

$$F_2 = \text{Conv}_{1\times 1}^{(2)}(F_1) \tag{4.1.23}$$

8 Where, F_1 is the output of the first CNN and F_2 is the output of the second CNN applied to F_1.

The convolution has:

Input Channels = 768, Output Channels = 768, Kernel Size = 1, Parameters = $768 \times 768 + 768 = 590,592$ (weights + bias)

Total Params (for both CNNs):

$$2 \times (768^2 + 768) = 2 \times 590,592 = 1,181,184$$

Output Shape: [1, 768, 197] → reshaped back to [1, 197, 768].

Additionally, a 3×3 max pooling operation effectively captures local spatial and temporal context.

$$P_1 = \text{MaxPool}_{3\times 1}(\hat{Y}) \tag{4.1.24}$$

60

Where 1D max pooling operation with kernel size 3, stride 1, and padding 1 is used along the temporal dimension (197).

Output Shape: [1, 197, 768]

Additional parallel 1×1 convolution operations are performed to $\hat{Y}$ and the pooled features.

$$F_3 = \text{Conv}_{1\times1}^{(3)}(P_1) \tag{4.1.25}$$

Same as previous equation number 21: Parameters $= 768 \times 768 + 768 = 590,592$, Output Shape: [1, 197, 768]

A parallel execution of another set of 1×1 convolutions is performed on the initial transformation:

$$F_4 = \text{Conv}_{1\times1}^{(4)}(F_1) \tag{4.1.26}$$

This uses the earlier output F_1 and applies another pointwise convolution.

Parameters: 590,592, Output Shape: [1, 197, 768]

Max pooling is applied once more:

$$P_2 = \text{MaxPool}_{3\times1}(F_2) \tag{4.1.27}$$

Same 3x1 MaxPool applied on F_2, retains shape: [1, 197, 768]

Followed by another 1×1 convolution:

$$F_5 = \text{Conv}_{1\times1}^{(5)}(P_2) \tag{4.1.28}$$

Yet another Conv1D with same parameters: Params: 590,592, Output: [1, 197, 768]

Then, a dropout layer is used to regularize:

$$D = \text{Dropout}(\hat{Y}) \tag{4.1.29}$$

$$F_6 = \text{Conv}_{1\times1}^{(6)}(D) \tag{4.1.30}$$

Dropout adds regularization by randomly setting values to zero with a given probability (e.g., 0.1). A final 1×1 convolution is applied to the dropout output.

Params: 590,592, Output: [1, 197, 768]

All intermediate features are combined with the input Y through element-wise summation.

$$R = Y + F_2 + F_3 + P_1 + P_2 + F_4 + F_5 + D + F_6 \tag{4.1.31}$$

Element-wise addition of all the intermediate features with the original input. This residual fusion aggregates multi-path contextual and spatial features.

Output Shape: [1, 197, 768]

A dual multi-head self-attention block subsequently processes this:

$$B = \text{DualMultiheadSelfAttention}(R) \tag{4.1.32}$$

Number of heads: $h = 12$, Dimension per head: $d = 64$

$$\text{Attention} = \text{SoftMax}\left(\frac{Q \times K^T}{\sqrt{dim}}\right) V \tag{4.1.33}$$

Trainable parameters (Q, K, V, output):

$$4 \times 768 \times 768 = 2,359,296$$

Output shape: [1,197,768]

After feature transformation, global average pooling (GAP) diminishes spatial dimensions.

$$\text{GAP} = \text{AvgPool}(B) \tag{4.1.34}$$

Global Average Pooling computes the mean across the temporal/spatial dimension (i.e., across all 197 tokens). It collapses the [1, 197, 768] tensor into a single vector of shape [1,

768]:

$$\text{GAP}_j = \frac{1}{197} \sum_{i=1}^{197} B_{(i,j)}, \quad \text{for } j = 1, 2, \ldots, 768$$

Finally, the fully connected (FC) layer correlates features to the conclusive class predictions:

$$X = \text{FC}(\text{GAP}) \tag{4.1.35}$$

$$X = \text{GAP} \cdot W + b \tag{4.1.36}$$

Where,

$$W \in R^{768 \times C}, \quad b \in R^C$$

For $C = 4$,

Weight matrix W has $768 \times 4 = 3,072$ parameters, Bias vector b adds 4 parameters

Total FC Parameters: 3,076

Output Shape: [1, 4] — class scores for each category.

The Algorithm builds the AgriTL-ViT model for image categorization. The Algorithm 4.1 start by providing different essential inputs, namely labels, image size, patch embedding size, input channels, number of attention heads, and one result, which can categorize images into specified groups. In Lines 2–8, we build the AgriTL-ViT model by first describing the input parameters and assuming that we can split the picture size evenly by the patch size to make patching work well. The model gathers global and spatial data by setting accessible codes and positioning embeddings. We use dropout regularization on embeddings and work on picture patches in Transformer layers with leftover connections. Finally, we build the classifier using layer normalization and linear layers to enhance the precision of the estimates from the transformer result. Further, Lines 9–18 create the forward function, which takes a batch size from the input and passes the input via patch embedding, then concatenates them with class tokens and patch embedding.

Algorithm 4.1 AgriTL-ViT Model

Require: size_img, input channels, patch_embedding_size, Trans_layer, embed_dim, no._attn_head, no._out_class
Ensure: AgriTL-ViT model
1: Proposed ViT Class
2: Define Input Parameters
3: Assume img_size is divided by patch_embedding_size.
4: Initialise embedding_class as a learnable variable.
5: Set the position_embedding to be a learnable parameter.
6: Set up the embedding_dropout and patch_embedding layers
7: Set the transformer_encoder as a Transformer_ResNet_Block sequential structure
8: Initialize the classifier as a linear container of Layer Norm and Linear layers
9: Create the forward function inside the ModifiedViT class
10: Calculate batch_size from the input Y
11: Increase the class token size to fit the batch size
12: Pass input x via patch_embedding
13: Z = class token + patch embedding across dim = 1
14: Add position embedding (pos_embedd + Z)
15: Set the embedding dropout layer
16: Pass via Transformer_ResNet_Encoder
17: Remove just the class token from the result
18: Pass via the SoftMax classifier
19: Set the learning rate, loss function and batch_size

Algorithm 4.2 outlines the steps for recognizing a train image using a pre-trained AgriTL-ViT model. It begins with three inputs: pre-processed image data, label data, and one output: generating a projected class label for the training image. Line 21 shows where the training loop starts for each epoch. In Lines 6–11, pass the image via the AgriTL-ViT model to extract features. In addition, calculate the train accuracy and average train loss using the formula given in lines 13 and 14.

Algorithm 4.3 is mostly used for evaluating the AgriTL-ViT model and identifying images. It first employs the testing dataset to train the AgriTL-ViT model and subsequently uses the trained model to classify a collection of test photos. It utilizes two pre-existing algorithms for examining and recognising photos. Moreover, line 2 shows where the testing loop starts for each epoch. Additionally, calculate test accuracy and average test loss using the formula given in lines 9 and 10. After finishing all epochs, the trained AgriTL-ViT model is formed. Finally, the approach provides the trained model, which can be utilized to recognize photos in the specified classes.

Time Complexity of the AgriTL-ViT Model

Algorithm 4.2 Model Training

Require: Pre-processed image data, label data, learning rate, loss function
Ensure: Trained AgriTL-ViT model

For training create blank lists to hold Total_Train_Loss, Train_Acc, Test_Loss, Test_Acc
For each epoch do
    Set correct_train = 0, total_train_loss = 0, total_train = 0
    For every image X_i and linked label Y_i in the training dataset do
        For every batch index in train dataset:
            Send information and targets to the device
            Grad_optimizer = 0
            Make a forward loop via the proposed_ViT model
            Calculate Loss
            Execute the reverse process and optimizer process
            Modify correct_train, total_train and total_train_loss
    Calculate Train_Acc and Avg_Train_Loss
        Train_Acc = $(correct\_train)/(total\_train)$
        Avg_Train_Loss = $(Total\_Train\_Loss)/(length(train\ dataset))$
Save training data

Algorithm 4.3 Model Testing and Identification

Require: Pre-processed image data, label data, Trained AgriTL-ViT model
Ensure: Evaluation result

For Testing:
    For every batch index in train dataset:
        Set the Proposed_ViT model in analysis phase using Algorithm I
        Send information and targets to the device
        Make a forward loop via the proposed_ViT model
        Calculate Loss
        Modify Total_test_loss, Total_test and correct_test
Calculate Test_Acc and Avg_Test_Loss
    Test_Acc = $(correct\_test)/(Total\_test)$
    Avg_Test_Loss = $(Total\_Test\_Loss)/(length(test\ dataset))$
Save testing data
Return Metrics

The time complexity of the AgriTL-ViT model is primarily influenced by the patch embedding process, transformer encoder layers, and the final classification head.

The patch embedding layer introduces significant computational overhead with a complexity of

$$O(BHWD),$$

where H and W are the input image dimensions, D is the embedding dimension, and B is the batch size. This arises from the linear projection of flattened patches into a high-

dimensional feature space. The addition of class and positional embeddings contributes a smaller term,

$$O(BND),$$

which depends linearly on the number of patches N. Therefore, the total embedding complexity becomes

$$O(BHWD + BND).$$

The transformer encoder, consisting of multiple self-attention layers, incurs substantial computational cost due to the self-attention mechanism, which introduces a quadratic term,

$$O(LN^2D),$$

where L is the number of transformer layers and N is the number of patches plus one for the class token. The feed-forward networks within the transformer add another significant component,

$$O(LND^2),$$

which scales with the embedding dimension.

Finally, the classification head contributes a comparatively small cost of

$$O(BDK),$$

where K denotes the number of output classes.

Hence, the overall time complexity of the AgriTL-ViT model is given by

$$O(BHWD + BND + LN^2D + LND^2 + BDK).$$

In summary, the computational complexity of the AgriTL-ViT model is dominated by the multi-head self-attention mechanism and feed-forward networks within the transformer layers, particularly for high-resolution images and deeper transformer architectures.

4.1.4 Experimental Setup and Result

The following section covers the dataset experimental setup and a brief overview of the preprocessing technique used. Then, the outcomes are generated using numerous parameters

to identify the AgriTL-ViT model advantages and disadvantages.

4.1.5 Experimental Setup for Dataset

Our tests are done on a Windows 11 Pro version 22H2 computer with an Intel(R) Xeon(R) Silver 4114 CPU @ 2.10GHz and 16GB RAM. The suggested AgriTL-ViT model is implemented on Tomato, Rice and Maize dataset using, Google Colab Pro+, Pytorch, NVIDIA GTX 1070 GPU, and Python 3.8.

4.1.6 Dataset Description

This research article applies the AgriTL-ViT model. The training of this model is carried out on the tomato [102], maize [103], and rice leaf datasets [104], which are publicly available image datasets. The datasets and their structure are discussed in Chapter 2, Section 2.4.

4.1.7 Data Pre-processing

Developing an AgriTL-ViT framework for autonomously identifying plant diseases is essential. This research employed the following data pre-processing methods:

4.1.7.1 Resizing

It can be used to raise or reduce the resolution of images. It aids in making sure that every input image has similar proportions, which many artificial intelligence models require. It also contributes to reducing the complexity of computation. Additionally, image resizing accelerates the training process of deep learning models [239]. The tomato, maize, and rice leaf dataset images were enlarged to 224 $\times$ 224 dimensions using the Pytorch library transform.resize class.

4.1.7.2 Normalization

Normalization reduces an image pixel values to a given range, for example, [0, 1]. It guarantees that pixel values remain identical, making it more straightforward for models to develop and limiting the influence of changing lighting conditions [240]. In addition, Normalization primary purpose is to reduce any biases and errors induced by feature-level differences.

4.1.7.3 Data Augmentation

Deep Learning (DL) method need enormous quantities of generalization data to avoid overfitting. Data augmentation involves image processing methods for example random hori-

zontal flipping, rotation, etc to enhance the amount of the dataset. It expands the number of samples in the initial data set while providing more variation in what the model observes and trains from. In addition, expanding the quantity and variety of a training dataset can be inexpensive, mainly when gathering additional data is costly or time-consuming.

4.1.8 Hyperparameter

The AgriTL-ViT model was trained and evaluated on the tomato, maize, & rice leaf dataset using a hyperparameter. Table 4.2 provides details on the hyperparameters.

Table 4.2: Hyperparameter Values for Executing the AgriTL-ViT model

Hyperparameter	Value
Number of Classes	10
Image Size	224×224
Patch Size	16×16
Number of Attention Heads	12
Optimizer	Adam
Epochs	20
Learning Rate	1×10^{-4}

4.1.9 Performance Evaluation Metrics

Precision, recall, and F1-score were used to assess the AgriTL-ViT model, as previously described in the section above. These measurements are used in this section to assess the model's performance.

4.1.10 Training and Testing Results for Tomato, Maize, and Rice Leaf Dataset

This part analyses our training and testing outcomes for the Tomato, Maize, and Rice Leaf Dataset. Divide this data set into training (70%) and testing (30%) data. Then, data pre-processing methods, like resizing, normalization, etc., have been used to improve their visual quality. The AgriTL-ViT model is also trained and tested using the pre-processed data. Table 4.3 displays the results that have been achieved.

Table 4.3: Training and Testing result of Tomato, Maize, and Rice Leaf Dataset

Dataset	Model	Training Accuracy (%)	Training Loss	Testing Accuracy (%)	Testing Loss
Tomato Leaf	AgriTL-ViT	97.7	0.103	98.5	0.150
Maize Leaf	AgriTL-ViT	96.9	0.020	97.2	0.010
Rice Leaf	AgriTL-ViT	96.8	0.020	97.0	0.100

In addition, we have analyzed the relationship among accuracy of training and testing, as well as loss of training and testing. This analysis allows us to evaluate the outcomes of the AgriTL-ViT model.

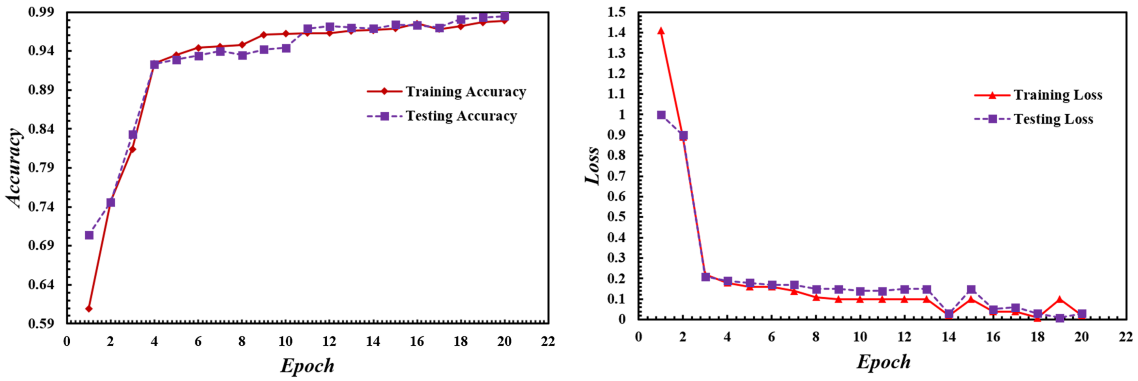


Figure 4.4: Training and testing outcome for tomato leaf dataset

This evaluation is depicted in Figure 4.4 to Fig.9. In Figure 4.4 (a)- Figure 4.6(a), the level of accuracy is shown by the vertical axis and the no. of epochs is signified by the horizontal axis. When the epoch is minimum, the accuracy is zero or low, while increasing epochs leads to higher accuracy.

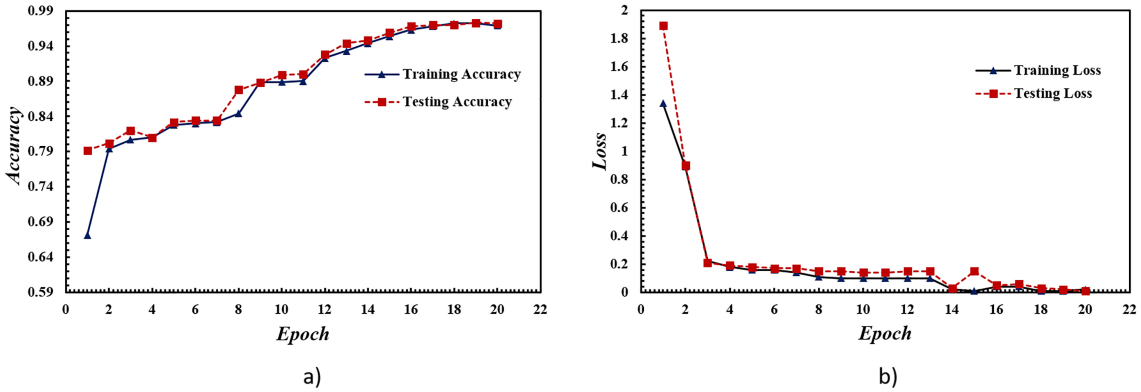
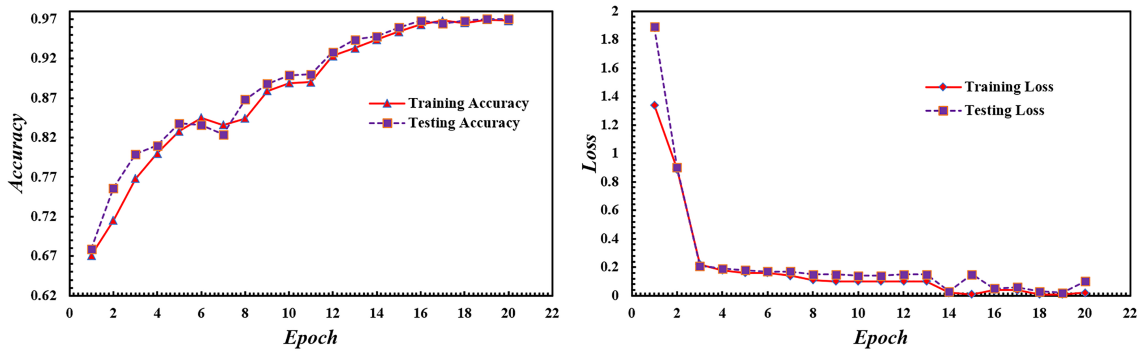


Figure 4.5: Training and testing outcome analysis for maize leaf dataset

The rising pattern shows that the system is gaining knowledge from training data. Further, the results of such models for validation and training loss are demonstrated in Figure 4.4(b)

– Figure 4.6(b). The loss is high when the number of epochs in the beginning stage is near zero. As epochs increase, the loss decreases. It helps determine how well a model generalizes to new, previously unidentified data.



a) b)

Figure 4.6: Training and testing outcome analysis for rice leaf dataset

4.1.11 Visualisation Results

This section discusses visual interpretation strategies that can help you understand how the model detects and concentrates on disease-affected parts of plant leaves. To improve interpretability, three well-known visualization methods Grad-CAM, LIME and SHAP have been used, which are explained in the following sections.

4.1.11.1 Grad-CAM Visualisation

This paper used the Grad-CAM technique to view the model's disease detection focus areas, allowing readers to visually watch the disease feature identification process of the image classification model [241]. As seen in Figure 4.7, the red parts of the heatmaps received the most attention from the model, followed by the yellow regions. By extracting semantic characterisation from the images, the model concentrated on diseased spots and symptomatic areas on the leaves, which are the most important locations for identifying plant diseases.

It was demonstrated that the suggested model could focus on the diseased areas of the tomato, maize, and rice rather than extraneous background information, resulting in accurate disease identification. Grad-CAM display allows for an intuitive understanding of how the AgriTL-ViT classification model works, and it is completely capable of finding disease-specific areas and producing accurate identifications.

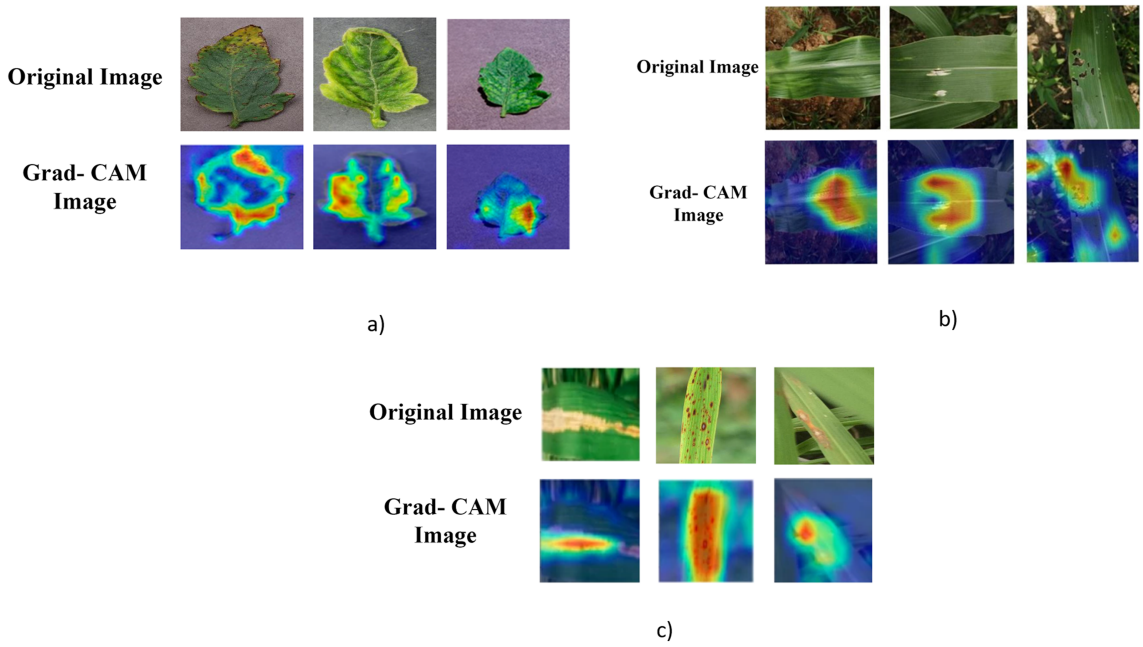


Figure 4.7: Visualisation of grad-cam analysis on (a) tomato (b) maize and (c) rice leaf dataset

4.1.11.2 LIME Visualisation

LIME (Local Interpretable Model-agnostic Explanations) technique to gain a thorough knowledge of the model's decision-making process by determining which characteristics of the input image have the biggest impact on outcomes of classification [242]. As illustrated in Figure 4.8, the yellow-marked areas in the LIME photos represent the super pixels that the model focused on when diagnosing diseases. These regions correspond to apparent symptoms such as lesions and discolouration, indicating that the model is focused on disease-relevant areas rather than background noise. LIME, when applied to tomato, maize, and rice leaves, gives an intuitive and interpretable image that aids domain specialists in understanding and validating the model's predictions. This increases trust in the AgriTL-ViT model's capacity to reliably detect plant diseases.

4.1.11.3 SHAP Visualisation

SHAP technique to show the model's focus regions during illness classification, allowing for a clear understanding of how various areas of the image contributed to the model's prediction [242]. As illustrated in Figure 4.9, the red-highlighted locations in the SHAP heatmaps represent places with a larger positive impact on the model's choice, whereas the blue areas had a negative influence. In all three datasets tomato (a), maize (b), and rice (c), SHAP correctly recognised unhealthy regions such as yellowing, lesions, and discolouration,

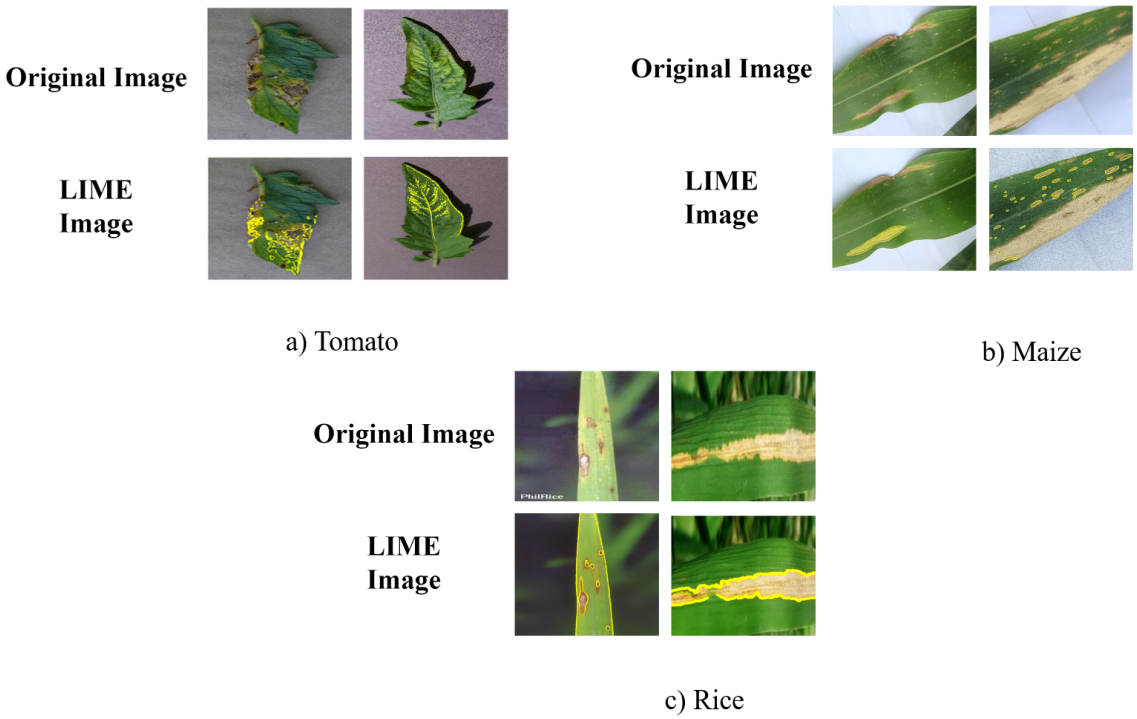


Figure 4.8: Visualisation of lime analysis on (a) tomato (b) maize and (c) rice leaf dataset

which are critical for disease detection.By extracting fine-grained attribution values from the photos, the model was able to prioritize symptomatic areas above the background, demonstrating its capacity to recognize disease-relevant features. The SHAP visualization provides a clear and complete explanation of how the AgriTL-ViT classification algorithm interprets input photos, enhancing the transparency and dependability of its predictions in agricultural disease diagnosis.

4.1.12 Evaluation Results for Tomato, Maize, and Rice Dataset

This section gives performance measures for every different category in the Tomato, Maize, and Rice Leaf Dataset. The dataset presents a significant barrier in identifying and categorizing labels, necessitating a thorough analysis of classifier quality. Table 4.4-Table 4.6. shows the quality assessments and the number of tables examined at each level. Further, Table 4.4- Table 4.6 provides the f1 score, support, recall and precision values for all different classes for the AgriTL-ViT models. Table 4.4 shows that the AgriTL-ViT model for Tomato_Tomato_Yellow_Leaf_Curl_Virus has good precision but low recall for quality measures across ten classes. This indicates no false positives and that photos from various classes were mistakenly classified as Tomato_Tomato_Yellow_Leaf_Curl_Virus.

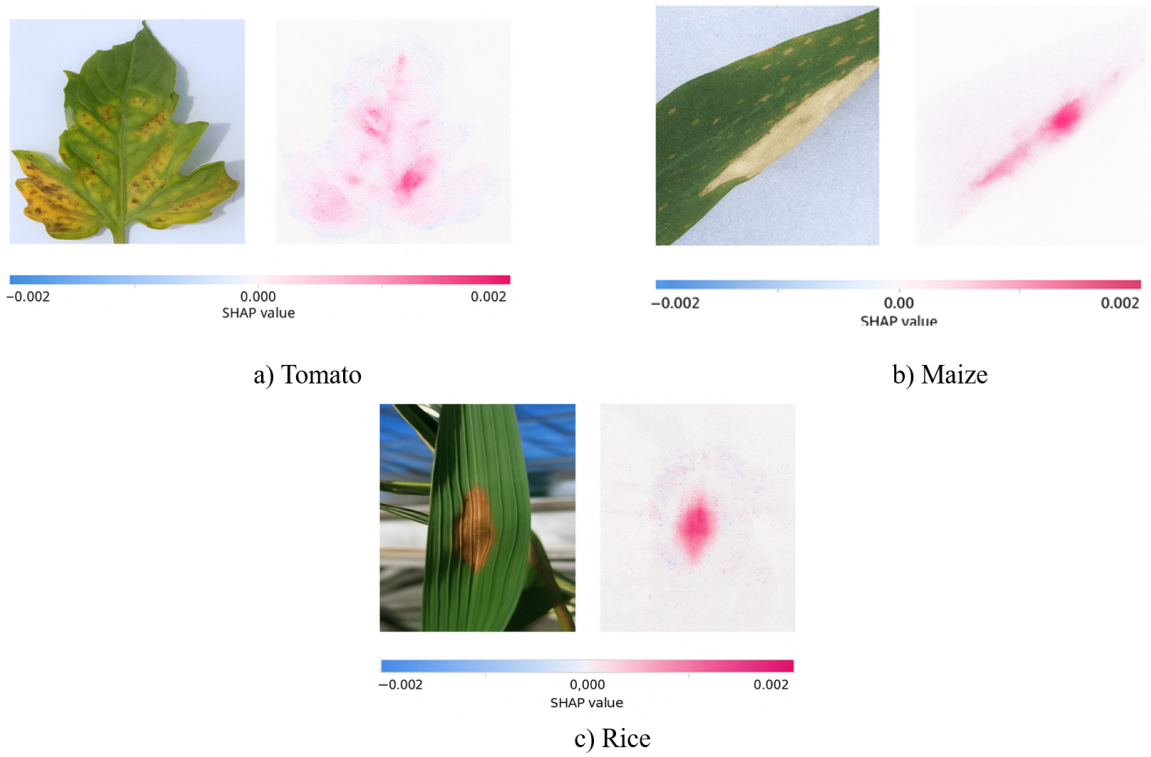


Figure 4.9: Visualisation of shap analysis on (a) tomato (b) maize and (c) rice leaf dataset

Table 4.4: AgriTL-ViT Model Image Quality for Ten Classes of Tomato Dataset

Class	Precision	Recall	F1-Score	Support
Tomato_Bacterial_spot	0.958	0.959	0.969	511
Tomato_Early_blight	0.977	0.958	0.857	240
Tomato_healthy	0.969	0.959	0.935	459
Tomato_Late_blight	0.952	0.942	0.932	229
Tomato_Leaf_Mold	0.966	0.959	0.975	426
Tomato_Septoria_leaf_spot	0.944	0.967	0.952	403
Tomato_Spider_mites	0.966	0.971	0.966	337
Tomato_Target_Spot	0.972	0.981	0.976	1286
Tomato_Tomato_mosaic_virus	0.952	0.989	0.952	90
Tomato_Tomato_Yellow_Leaf_Curl_Virus	0.979	0.964	0.982	382

As demonstrated in Table 4.5, the Gray_Leaf_Spot in the AgriTL-ViT model reacts similarly to maize blight. Furthermore, the AgriTL-ViT model has high precision but low recall. This indicates that the model is better at accurately predicting positive cases but may miss some actual positive cases. It is vital to consider both recall and precision when evaluating the use of a model for classification.

Table 4.5: AgriTL-ViT Model Image Quality for Four Classes of Maize Dataset

Class	Precision	Recall	F1-Score	Support
Blight	0.958	0.920	0.974	344
Common_Rust	0.966	0.979	0.915	392
Gray_Leaf_Spot	0.952	0.958	0.935	173
Healthy	0.962	0.970	0.972	359

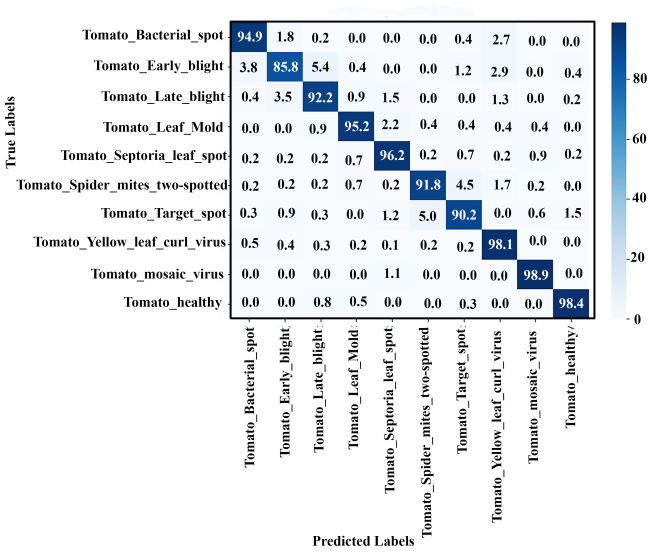
However, a low precision and high recall value has been for the Neck_Blast and Narrow Brown Leaf Spot classes for the AgriTL-ViT model as shown in Table 4.6. Moreover, a low F1-Score value has been seen for Brown Spot class for the AgriTL-ViT model.

Table 4.6: AgriTL-ViT Model Image Quality for Nine Classes of Rice Dataset

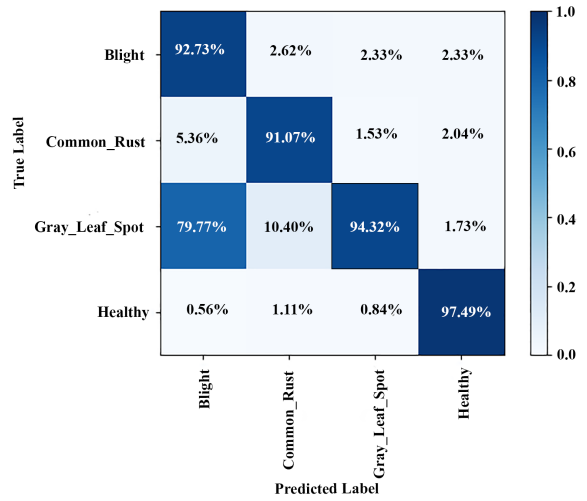
Class	Precision	Recall	F1-Score	Support
Bacterial Leaf Blight	0.958	0.949	0.929	511
Brown Spot	0.966	0.858	0.857	240
Healthy Rice Leaf	0.969	0.943	0.935	459
Leaf Blast	0.952	0.940	0.930	229
Leaf Scald	0.966	0.959	0.975	426
Narrow Brown Leaf Spot	0.952	0.967	0.952	403
Neck_Blast	0.966	0.934	0.906	337
Rice Hispa	0.972	0.981	0.976	1286
Sheath Blight	0.918	0.970	0.952	90

The assessment results show that AgriTL-ViT models accurately recognize and categorize different classes in the Tomato, Maize and Rice Leaf dataset. The evaluation results are given via the confusion matrix (cm) for the AgriTL-ViT technique.

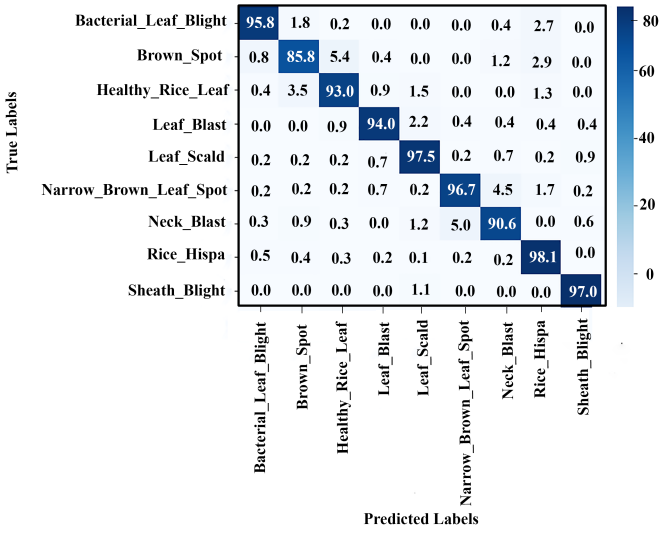
The Confusion Matrix (CM) graph (Figure 4.10) helps identify the various classes that cause mistakes in the trained method. The CM visualizes the recall and accuracy values for each disease and technique by evaluating the number of true or false positives for each class of images [243]. Figure 4.10 depicts the true label on the y-axis and the predicted class on the x-axis. The graph displays the number of occasions where the AgriTL-ViT model accurately recognized one class (x-axis), whereas the y-axis represented another. Optimal values for top-left to bottom-right diagonal areas indicate correct predictions, whereas lower numbers for off-diagonal areas indicate inaccurate estimates. In Table 4.7, we also calculated the weighted average and macro average to evaluate the model's overall performance across three datasets. The macro value helps to evaluate the classifier's overall performance by considering a more familiar category label, whereas the weighted average accounts for class



(a) Tomato leaf dataset



(b) Maize leaf dataset



(c) Rice leaf dataset

Figure 4.10: Confusion matrices for tomato, maize, and rice leaf datasets using the AgriTL-ViT model

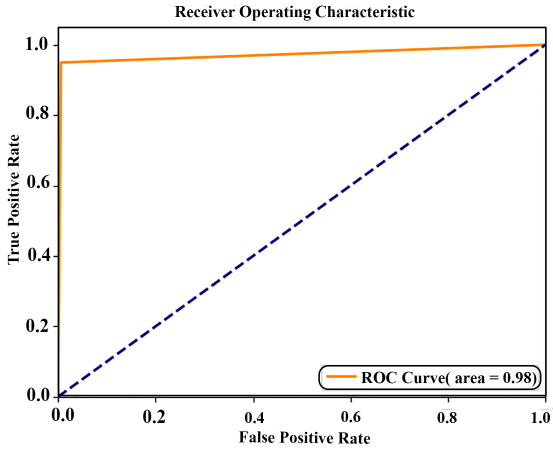
imbalance. The macro average score of the Tomato, Maize, and Rice datasets is 0.97, 0.962, and 0.962, respectively. This suggests that the proposed approach performs well across all categories, guaranteeing that efficiency in smaller categories is not overshadowed by efficiency in bigger classes.

Table 4.7: Classification Results for Detecting Peach Leaf Disease.

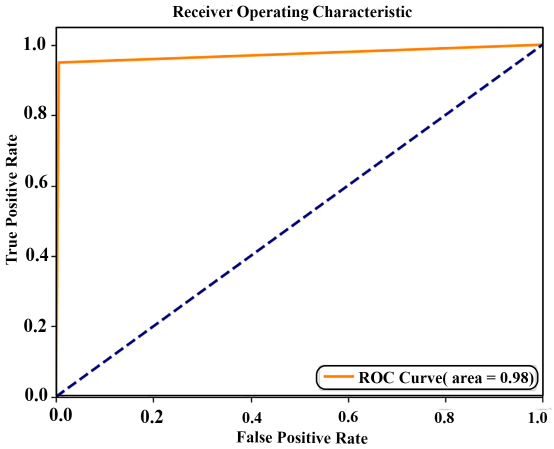
	Precision	Recall	F1-Score
Tomato Leaf Dataset			
Macro Avg	0.962	0.976	0.976
Weighted Avg	0.971	0.975	0.975
Maize Leaf Dataset			
Macro Avg	0.959	0.968	0.968
Weighted Avg	0.961	0.969	0.970
Rice Leaf Dataset			
Macro Avg	0.960	0.969	0.968
Weighted Avg	0.959	0.969	0.970

The Receiver Operating Characteristic (ROC) graph in Figure 4.11 demonstrates the relationship among precision and recall at many levels for identifying positive values in our technique. The FPR and TPR graph, with an X and Y axis, also illustrates the relationship between sensitivity and specificity.

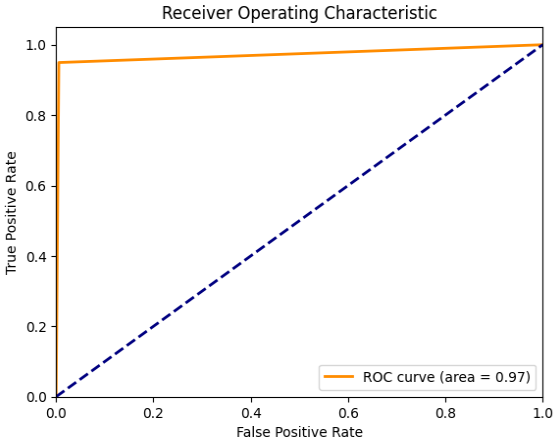
In Figure 4.11(a), the ROC line is located in the top-left region of the coordinate (0,1), indicating a 0.98 no false negative rate and a 0.98 no false positive rate, respectively. Similarly, in Figure 4.11(b), the ROC line reveals 0.98 no false negative rate and 0.98 no false positive rate. In Figure 4.11(c), the ROC line shows 0.97 with no false negative rate and 0.97 with no false positive. So, the Y-axis is referred to as a better classification. Therefore, the classification performance improves as the ROC line moves closer to the top-left corner. This suggests a reduced rate of false positives and a greater actual positive rate. In our studies, we choose to examine the recursive network results using additional evaluation metrics Like False Negative Rate (FNR), False Omission Rate (FOR), False Discovery Rate (FDR), and False Positive Rate (FPR). These performance metrics evaluate our recursive network's stability under different scenarios. Figure 4.12 shows a bar chart depicting the overall outcome of the proposed ViT model (AgriTL-ViT) FPR, FNR, FDR, and FOR. In Figure 4.12 (a), FDR outperforms the other approaches (FPR, FNR, and FOR) with a score of 0.04%. Further, FNR outperforms FPR and FOR with a score of 0.047%. It shows how often the model mistakenly classifies infected leaves as healthy. Furthermore, compared



(a)



(b)



(c)

Figure 4.11: ROC curve for (a) tomato, (b) maize and (c) rice Dataset

to the other two approaches (FDR, FNR), FOR and FPR have the lowest performance at 0.018%. The model's performance is assessed when errors are more significant than positive, and the cost of inaccurate outcomes is low. Overall, FDR outperformed the other three approaches for illness classification. Figure 4.12 (b) shows that FNR outperforms the other techniques (FDR, FOR, and FPR) with a score of 0.47%. Compared to the other two methods, FPR and FDR, FOR has a lower score of 0.017%.

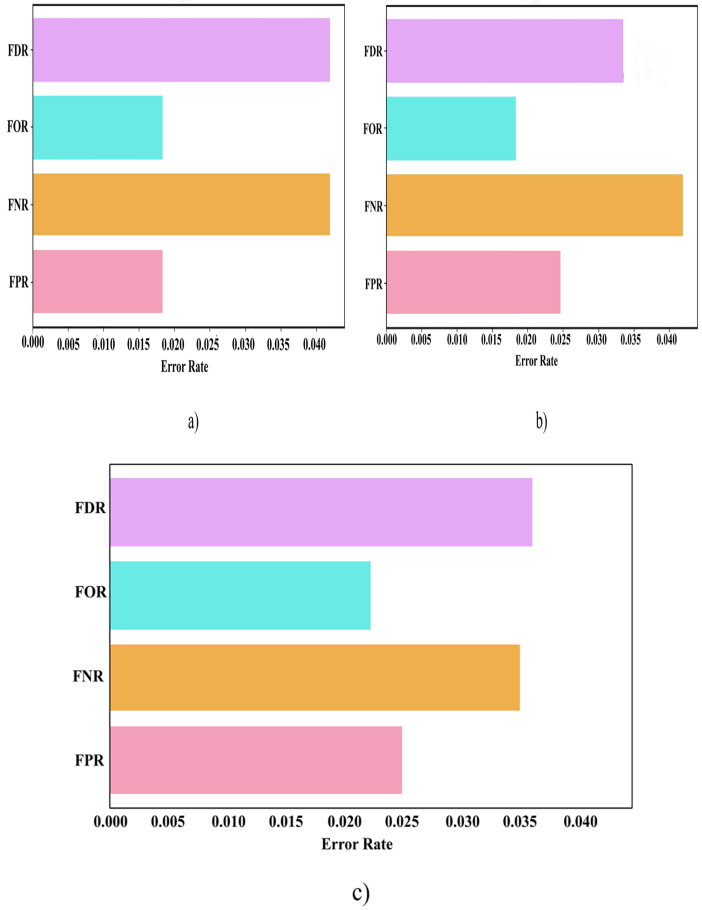


Figure 4.12: Bar graph representation of the AgriTL-ViT model based on FPR, FNR, FOR and FDR for the (a) tomato, (b) maize and (c) rice Datasets

In the rice dataset, FDR outperforms three techniques (FNR, FOR, and FPR), as illustrated in Figure 4.12(c). Among the four techniques, FPR, FDR, FNR, and FOR, the FOR method has the lowest performance.

4.1.13 Generalizability of the proposed model

To ensure that the proposed model functions consistently in real-world agricultural contexts, its generalizability has been tested across unseen datasets and different environmental circumstances. The following subsections assess its resilience to data variances caused by

unknown crop varieties, meteorological circumstances, and geographic variety.

4.1.13.1 Generalizability across unseen data

To show the proposed model's generalization capabilities, we ran additional tests on the grape [100] and apple leaf datasets [244] not included in the initial training set. The AgriTL-ViT models were tested on these unseen datasets, and the results show that the models obtained good classification accuracy, demonstrating their potential to identify leaf diseases in completely new crop kinds. This confirms that the model is not overfitted and may effectively generalise to many plant species, as shown in Table 4.8.

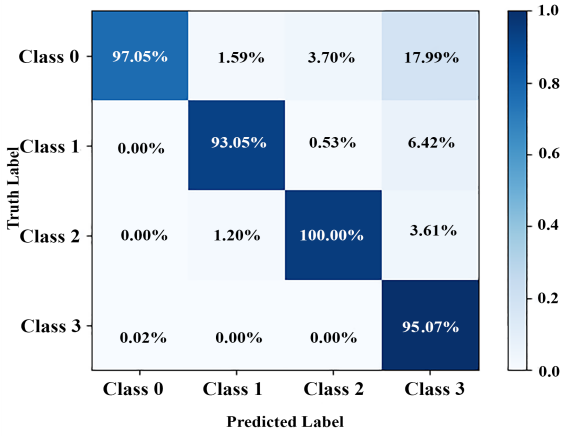
Table 4.8: Accuracy of the Proposed AgriTL-ViT Model on Unseen Datasets

Model	Apple Leaf Dataset Testing Accuracy	Grape Leaf Dataset Testing Accuracy
AgriTL-ViT	0.962	0.971

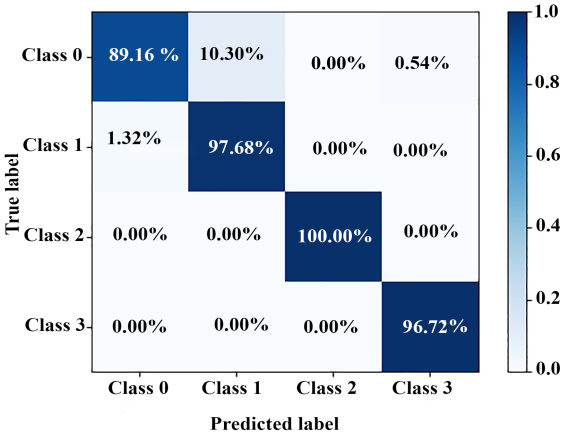
As demonstrated in Figure 4.13 (a), the confusion matrix for the apple dataset performs well across most classes, with particularly accurate predictions for later-stage disease categories. However, there are a few misclassifications between early-stage classes, indicating that certain disease signs may have visual similarities that impair the model's discriminative capacity. Similarly, Figure 4.13 (b) depicts the confusion matrix for the grape dataset, demonstrating that the model performs well overall but exhibits substantial confusion between individual disease classifications, especially when visual characteristics are modest or overlapping. These analyses demonstrate the model's durability while also pointing out opportunities for development, such as increasing the dataset or using finer-grained feature extraction techniques to better distinguish between visually identical illness symptoms.

4.1.13.2 Generalizability across varied climates and geographic regions

To evaluate our proposed model's generalisation capacity and robustness across varied climates and geographic regions, we tested the model using potato and strawberry crop datasets from the PlantVillage repository [99]. The PlantVillage collections are mostly made up of images taken in the United States, particularly at Pennsylvania State University and different land-grant educational institutions, with regulated lighting and background circumstances. These depict temperate habitats and consistent imaging settings, which help to build a solid baseline before testing in more diverse, real-world scenarios. To supplement this, we tested the model on datasets from Indonesia (potato) [245] and South Korea (straw-



(a) Apple Dataset



(b) Grapes Dataset

Figure 4.13: Confusion matrix chart with grapes and apple leaf dataset for model AgriTL-ViT

berry) [246], which reflect tropical and greenhouse (humid temperate) settings, respectively. The Table 4.9 shows the comparable performance on an unseen crop dataset:

Table 4.9: Unseen Crop Testing Accuracy Across Geographic and Climate Zones

Crop	Dataset Source	Region	Accuracy (%)
Potato	PlantVillage	USA	0.979
Potato	Field Dataset (Mendeley)	Indonesia	0.920
Strawberry	PlantVillage	USA	0.981
Strawberry	South Korea Dataset	South Korea	0.963

These findings show that the model works extraordinarily well on controlled datasets and its accuracy stays consistently high on datasets from complicated fields and greenhouse circumstances, demonstrating its robustness and applicability to a wide range of real-world agricultural settings.

4.1.14 Computational Cost Analysis

In this study, we investigated the computing cost of three alternative models: the Vision Transformer (ViT), Modified ResNet, and the AgriTL-ViT (Agriculture Transfer Learning Vision Transformer), which is proposed specifically for agricultural image classification tasks. The AgriTL-ViT model offers a more efficient solution without losing performance. Below is a comparison of the time spent by each model for training on various crops:

Table 4.10: Training Time Comparison of Different Models on Leaf Datasets

Model	Tomato	Maize	Rice
With ViT	42m 40s	12m 20s	34m 40s
With Proposed ResNet	28m	8m 40s	24m
With AgriTL-ViT (Proposed Model)	38m 20s	11m 30s	30m 20s

As Table 4.10. indicates, ViT requires the highest computing time of all datasets, particularly Tomato and Rice. On the other hand, the Modified ResNet model is more efficient, taking much less time for all three crops, particularly maize. Our proposed model, AgriTL-ViT, balances the other two models. It requires significantly less computational time than ViT, giving it a more realistic solution for large-scale agricultural picture classification. For example, Tomato takes 38 minutes and 40 seconds, significantly faster than ViT while still providing competitive performance. Similarly, the model demonstrates efficient processing

times on the Maize and Rice datasets, completing them in 11 minutes and 30 seconds and 30 minutes and 20 seconds, respectively. In conclusion, the AgriTL-ViT model is a more computationally efficient solution that performs well across various crops. Our technique provides a better balance between speed and accuracy than existing models, making it ideal for real-time farming applications.

4.1.15 Ablation Study

This part includes ablation research to comprehensively evaluate the components of the approach and the contributions they make. The evaluation encompasses the subsequent cases:

Case 1: Train the Dataset utilizing a 1D Convolutional Neural Network (CNN)

Case 2: Train the Dataset using the Vision Transformer Model (ViT)

Case 3: Train the Dataset employing the ResNet Model

Case 4: Train the dataset utilizing the Proposed ResNet

Case 5: Train the Dataset using AgriTL-ViT (our suggested method)

This thorough assessment of specific instances seeks to thoroughly analyze the impact and effectiveness included in each version of the approach. The goal is to get a thorough understanding of their responsibilities and the implications for the system's overall effectiveness.

Table 4.11: Score of Ablation Study for the AgriTL-ViT Model for Tomato Dataset

Ablation study cases	Cases	Accuracy	Precision	Recall	F1-Score
Train the dataset with 1D CNN	1	0.912	0.891	0.902	0.862
Train the dataset with ViT	2	0.901	0.882	0.912	0.853
Train the dataset with ResNet	3	0.900	0.853	0.874	0.842
Train the dataset with the Proposed ResNet Model	4	0.941	0.937	0.942	0.939
Train the dataset with AgriTL-ViT (our proposed method)	5	0.985	0.977	0.982	0.979

Table 4.11-Table 4.13 depicts ablation research conducted on the AgriTL-ViT models for Tomato, Maize and Rice datasets, thoroughly analysing many training conditions. The instances labelled 1 through 5 illustrate various neural network structures used in training. The examined metrics include critical indicators of performance, including f1-score, precision, recall and accuracy. From Table 4.11, Case 1 uses a 1D CNN to obtain an accuracy of 0.912, with precision (0.891), recall (0.902) and f1 score values of 0.862.

Case 2 includes training using a ViT model, which results in an accuracy of 0.901 and 0.882 for precision, 0.912 for recall, and 0.853 for f1 score. In Case 3, training with a

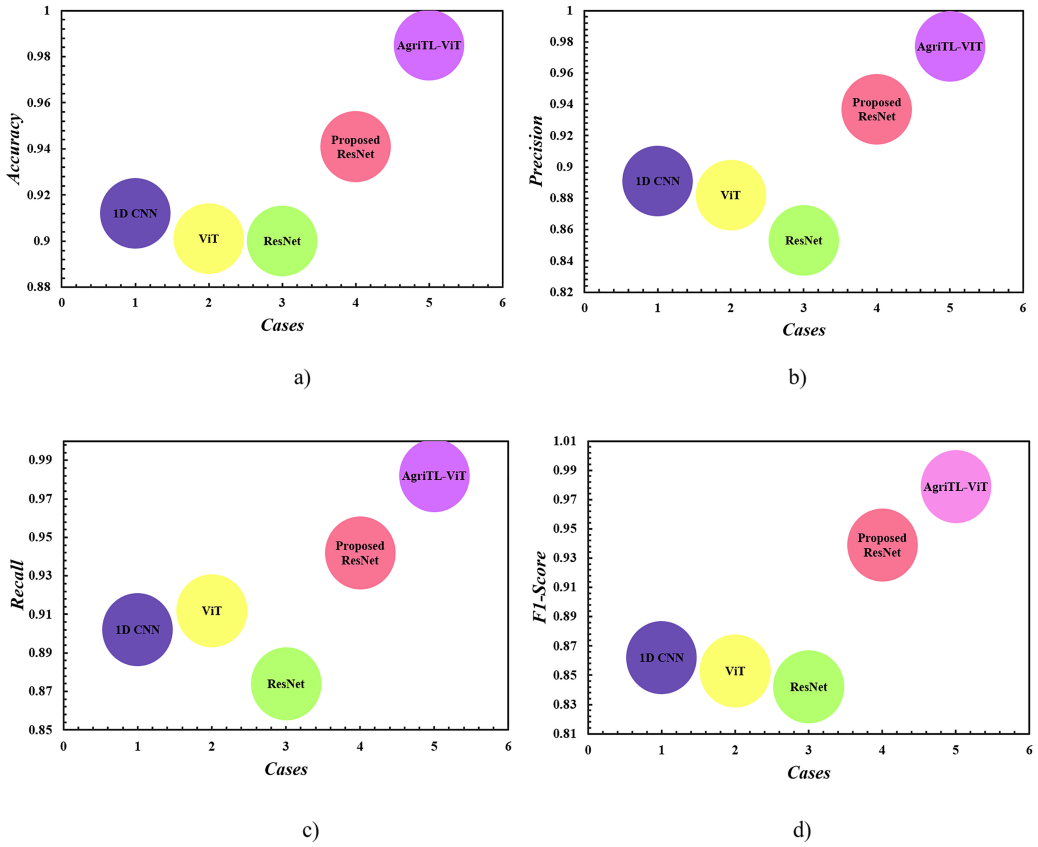


Figure 4.14: Results of the Ablation Study for Tomato Leaf Dataset

ResNet, an accuracy of 0.900, as well as precision (0.853), f1 score (0.842) and recall (0.874), correspondingly. The AgriTL-ViT model demonstrates significant performance, illustrated by instances 4 and 5. Case 4 obtains an accuracy of 0.941, with intense precision (0.937), recall (0.942) and f1 score (0.939). Case 5 has competitive accuracies of 0.985, 0.977 for precision, 0.982 for recall, and 0.979 for f1 score values, as demonstrated in Figure 4.14.

This detailed ablation research gives an in-depth view of how different neural network designs contribute to the overall efficacy of the AgriTL-ViT. The findings highlight the usefulness of the proposed models, particularly AgriTL-ViT, in obtaining high accuracy and a solid ability to predict.

Table 4.12: Ablation Study Score for the AgriTL-ViT Model for Maize Dataset

Ablation study cases	Cases	Accuracy	Precision	Recall	F1-Score
Train the dataset with 1D CNN	1	0.820	0.795	0.820	0.784
Train the dataset with ViT	2	0.923	0.878	0.861	0.901
Train the dataset with ResNet	3	0.941	0.902	0.941	0.890
Train the dataset with the Proposed ResNet Model	4	0.952	0.899	0.921	0.920
Train the dataset with AgriTL-ViT (our proposed method)	5	0.972	0.962	0.970	0.972

Table 4.12 presents a comprehensive analysis of the ablation study. In this table, a 1D CNN

is utilized to achieve an accuracy of 0.820. The value for precision 0.795, 0.820 for recall, and 0.784 for f1 score.

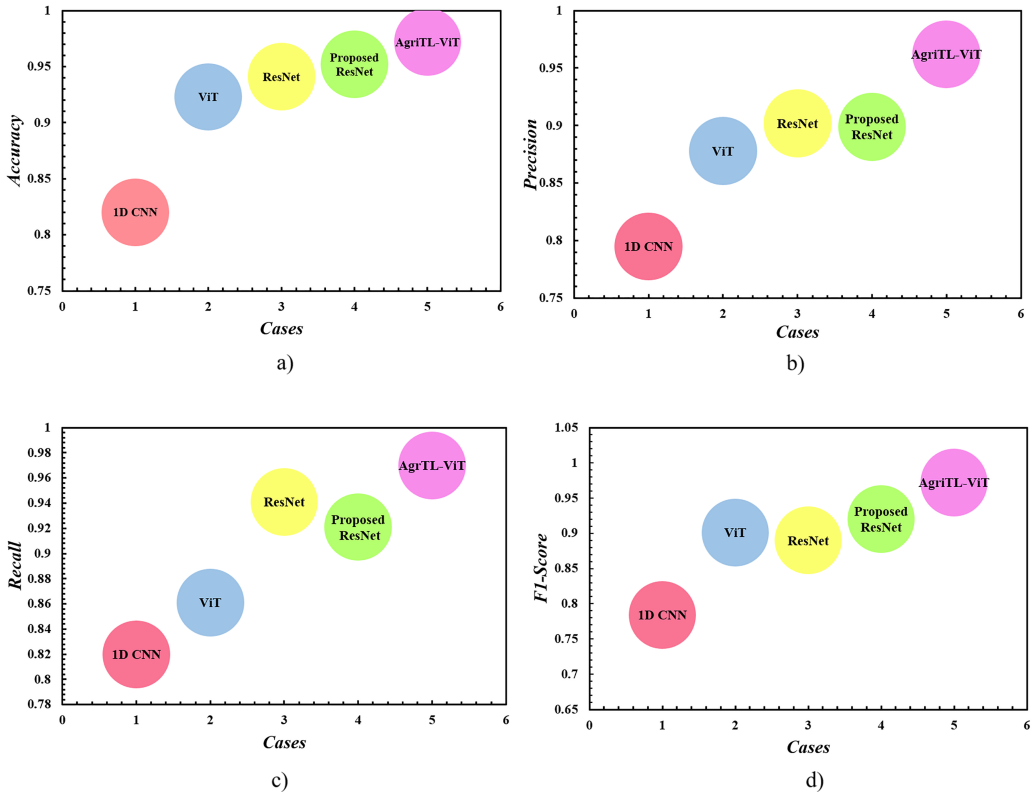


Figure 4.15: Results of the Ablation Study for Maize Leaf Dataset

In Case 2, a ViT model is used for training, leading to an accuracy score of 0.923 and 0.878 for precision, 0.961 for recall, and 0.901 for f1 score. An accuracy of 0.941, as well as score of precision (0.902), a score of recall (0.941), and a score of f1 score (0.890) were achieved in Case 3. Cases 4 and 5 show that the AgriTL-ViT model performs very well. Case 4 gets an accuracy score of 0.952, with a high level of precision (0.899), recall (0.921), and f1 score (0.920). As shown in Figure 4.15, Case 5 has close accuracies of 0.972, 0.962, 0.970, and 0.972, with similar precision, recall, and f1 score values.

Table 4.13: Ablation Study Score for the AgriTL-ViT Model for Rice Dataset

Ablation study cases	Cases	Accuracy	Precision	Recall	F1-Score
Train the dataset with 1D CNN	1	0.922	0.902	0.902	0.901
Train the dataset with ViT	2	0.932	0.910	0.898	0.932
Train the dataset with ResNet	3	0.901	0.911	0.901	0.921
Train the dataset with the Proposed ResNet Model	4	0.942	0.914	0.899	0.927
Train the dataset with AgriTL-ViT (our proposed method)	5	0.970	0.918	0.902	0.952

Table 4.13 indicates that Case 1 uses a 1D CNN to obtain an accuracy of 0.922, while score 0.902 for precision, 0.902 for recall, and 0.901 for f1 score. Case 2 involves training with

a ViT model, which produces 0.932 for accuracy along with 0.910 for precision, 0.898 for recall, and 0.932 for f1 score.

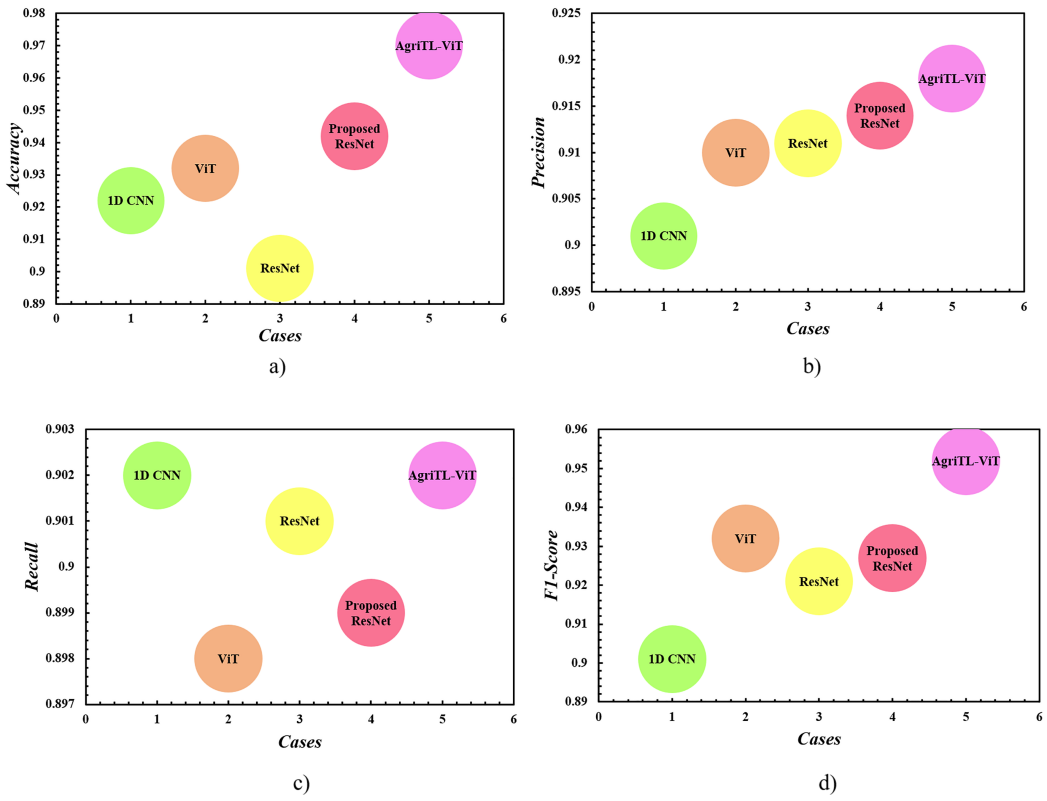


Figure 4.16: Results of the ablation study for rice leaf dataset

In Case 3, training with a ResNet produced an accuracy of 0.901, 0.911 for precision, 0.901 for recall, and 0.921 for f1 scores. The AgriTL-ViT model functions admirably, as seen in instances 4 and 5. Case 4 has 0.942 for an accuracy, with good precision of 0.914, 0.899 for recall, and 0.927 for f1 score. Case 5 demonstrates competitive accuracies of 0.970, 0.918 for precision ,0.902 for recall and 952 with f1 score values, as demonstrated in Figure 4.16. Case 5 exhibits the model's robust performance across a variety of settings. Overall, the AgriTL-ViT model display reliable and impressive results across different cases, demonstrating its dependability and effectiveness in many scenarios. Its high recall, precision, accuracy, f1 score values make it a highly helpful for data study and decision-making in farming applications.

4.1.16 Comparison with the State-of-the-Art Techniques

The following section compares AgriTL-ViT to other cutting-edge methods for recognizing and categorizing leaf disease using multiclass tomato leaf datasets. Table 4.14 shows the performance outcomes for previous methods ([206], [207], [247], [28], [248]) across many

criteria. Regarding precision, recall, f1-score, and accuracy, the AgriTL-Vit model surpasses other approaches for multiclass categorizing tomato leaf datasets. Our significant benefit of the AgriTL-ViT model is that it can successfully identify spatial and temporal characteristics from leaf pictures.

The outcomes demonstrate that our AgriTL-ViT model outperforms all baseline techniques in the tomato leaf dataset, with the highest precision (0.977%) which is 0.024-0.032% better, highest recall (0.982%) which is 0.019-0.041% better, highest f1 score (0.979%) which is 0.014-0.035% better, and highest accuracy (0.985%) which is 0.014-0.044 % higher. Because it solves the accuracy and model complexity problems connected with detecting sickness in leaves, this study compares the output of the AgriTL-ViT model to the most accurate approach for diagnosing leaf diseases.

Table 4.14: Comparison of AgriTL-ViT Technique with State-of-The-Art Techniques on Tomato leaf Dataset

References	Model	Accuracy	Precision	Recall	F1-Score
Al-gaashani M et al. [206]	MLR	0.971	0.945	0.963	0.965
Tian X et al. [207]	Deep neuro fuzzy network	0.941	0.942	0.941	0.944
Hossain S et al. [247]	MaxViT	0.970	0.950	0.950	0.952
Paul S et al. [28]	Proposed CNN	0.950	0.952	0.954	0.957
Haseeb Wajid A et al. [248]	CNN	0.960	0.953	0.955	0.958
Our proposed method	AgriTL-ViT	0.985	0.977	0.982	0.979

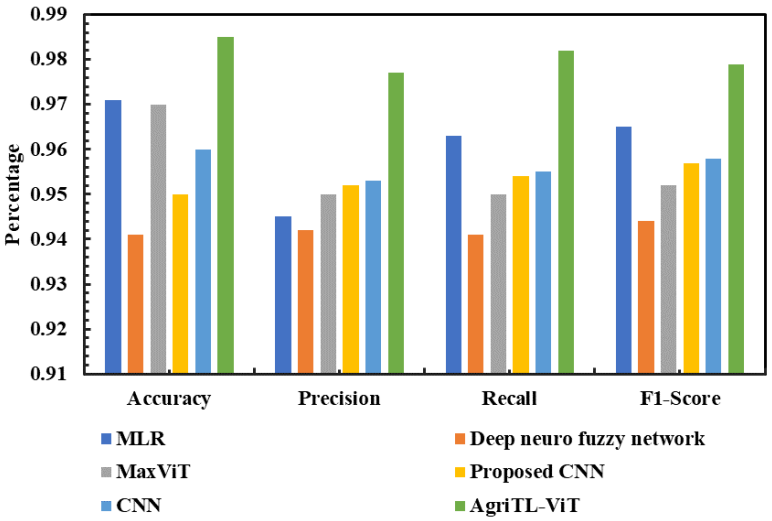


Figure 4.17: Graphical presentation of the AgriTL-ViT model with state-of-the-art techniques

Figure 4.17 compare the AgriTL-ViT recall, f1-score, accuracy and precision to the cutting-edge work using a bar graph. The proposed method (AgriTL-ViT) outperforms current techniques in classifying various leaf diseases.

Table 4.15: Comparison of AgriTL-ViT Model with Existing Techniques on Maize Leaf Dataset

References	Models	Accuracy	Precision	Recall	F1-Score
Yu H et al. [228]	CNN	0.934	0.934	0.934	0.932
Panigrahi K et al. [229]	RF	0.792	0.821	0.799	0.799
Rachmad A et al. [249]	LBP with KNN	0.941	0.818	0.811	0.809
Our proposed method	AgriTL-ViT	0.972	0.962	0.970	0.972

Table 4.15 and Table 4.16 present the performance outcomes of the existing approaches ([228], [229], [249], [250], [235], [251]) across several criteria. The proposed AgriTL-ViT method demonstrates better results than existing methods regarding precision, recall, and overall accuracy for the multiclass classification of the Maize and Rice datasets. The results indicate that our proposed AgriTL-ViT model achieves the highest precision of 0.962%, which is 0.14-0.02% better than other approaches. It also achieves the highest recall of 0.970%, which is 0.17-0.03% better and highest F1 score of 0.972%, which is 0.17-0.04% better. Additionally, our model achieves the highest accuracy of 0.972%, which is 0.18-0.03% higher than any other state-of-the-art method.

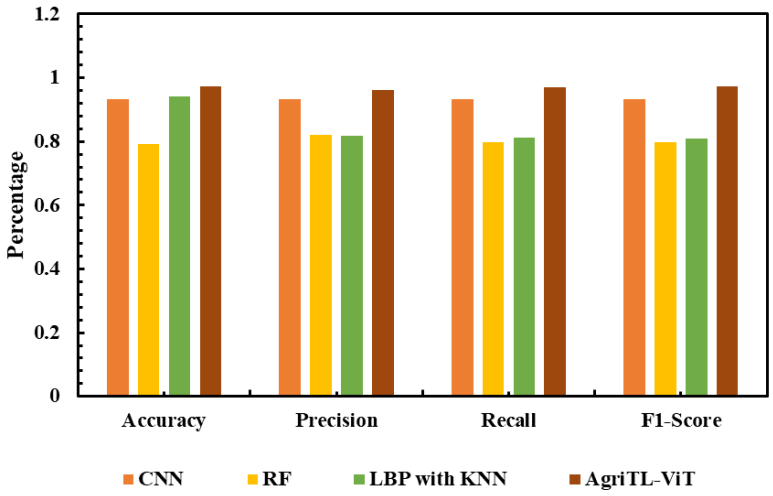


Figure 4.18: Graphical presentation of the AgriTL-ViT model with existing base line techniques1

Regarding the Rice Dataset, our updated method achieves the highest precision of 97.8%, which is an improvement of 0.04-0.007% compared to other approaches. Additionally, it achieves the highest recall of 97.5%, which is 0.01-0.04% better, and the highest accuracy of 97.7%, which is 0.04-0.008% higher than other existing approaches.

Figure 4.18 and Figure 4.19 display a bar graph comparing the accuracy, precision, and recall

Table 4.16: Comparison of AgriTL-ViT Model with Existing Techniques on Rice Leaf Dataset

References	Models	Accuracy	Precision	Recall	F1-Score
Senthil Pandi S et al. [250]	DCNN+GAP	0.965	0.934	0.955	0.943
Wang Y et al. [235]	ADSNN-BO	0.946	0.926	0.874	0.896
Zeng N et al. [251]	ICAI-V4	0.955	0.954	0.955	0.958
Our proposed method	AgriTL-ViT	0.972	0.962	0.970	0.972

of the suggested model to the baseline research effort. In general, the proposed AgriTL-ViT model outperforms the already employed methodologies and is capable of accurately classifying various leaf diseases.

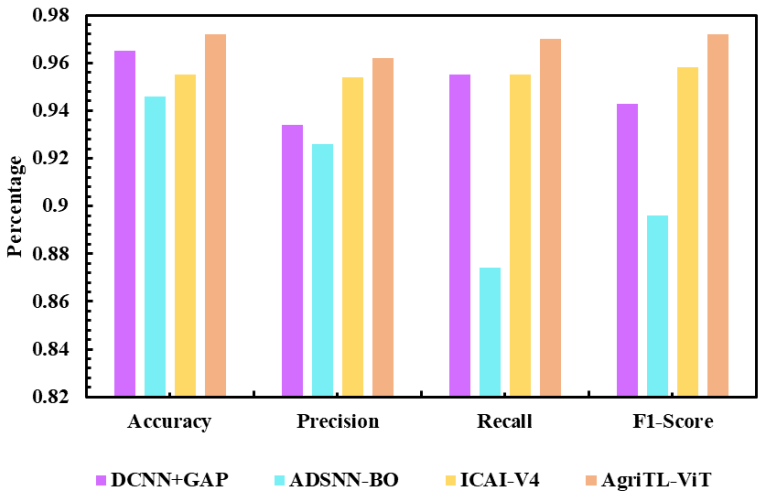


Figure 4.19: Graphical presentation of the AgriTL-ViT model with existing base line techniques

4.1.17 Discussion on the Proposed AgriTL-ViT Model

The proposed AgriTL-ViT model has been developed to effectively classify plant diseases across various crops, addressing the limitations found in existing state-of-the-art models. Traditional models face high processing costs, sluggish inference times, and limited adaptability across varied agricultural datasets. These drawbacks make them unsuitable for real-time or resource-constrained agricultural environments.

To address these challenges, we developed the AgriTL-ViT model, a transfer learning-based Vision Transformer optimized for agricultural picture classification. This approach successfully decreases computational complexity while retaining excellent classification performance. It exhibits a balanced trade-off between accuracy and processing speed, making it

ideal for large-scale deployment. Through studies on Tomato, Maize, and Rice leaf disease datasets, AgriTL-ViT achieved much quicker processing speeds than ViT while maintaining equivalent or better classification accuracy.

The AgriTL-ViT model has demonstrated superior performance across multiple crop datasets, including Tomato, Maize, and Rice, compared to existing state-of-the-art methods. However, certain limits must be recognized. One key issue is the model's generalizability. Although the model works well on this study's tomato, maize, and rice leaf datasets, its application to other plant species or illnesses is undetermined. Given the diversity of plant illnesses, future work will include validation of additional crops and disease types beyond the scope of this study to thoroughly assess the model's robustness across diverse agricultural settings.

Another limitation is that it depends on the dataset's quality and diversity. The current work focuses on three crops: tomato, maize, and rice; however, the performance of the AgriTL-ViT model may differ when applied to other plant species or illnesses that were not included in the training dataset. This highlights the significance of expanding the dataset to cover a broader range of plant diseases to further assess the model's generalizability.

In addition, environmental factors such as lighting, weather, and camera quality may impact the model's effectiveness in real-world farming applications. The dataset utilized in this study may not adequately account for these characteristics, thereby affecting the model's accuracy in field situations. Future studies should investigate how environmental heterogeneity influences model performance, particularly in real-time disease detection in agricultural settings.

Overall, The AgriTL-ViT model performs better than existing methods in identifying leaf diseases in tomatoes, maize, and rice. Its potential application to other crops and diseases and its robustness across different environmental conditions require further exploration. Expanding the dataset to include more plant species and disease variations and considering real-world environmental factors will enhance the model's effectiveness and reliability for practical agricultural disease detection systems

4.1.17.1 Environmental Constraint

To overcome the restriction of relying simply on curated datasets, which may not fully capture the variety found in real-world agricultural situations, a comprehensive data augmentation technique was implemented. This augmentation process was created to model

common field-level issues like lighting variations, backdrop complexity, and partial occlusions. ColorJitter, GaussianBlur, and RandomErasing are some of the approaches used to imitate environmental fluctuations and increase the model's generalisation capacity (Figure 4.20).

To understand the model's performance in such scenarios, we conducted a limited-scale evaluation using real-world field images. The results are in Table 4.17 showed promising accuracy scores of 74.9% on the Tomato Leaf dataset, 61.4% on the Maize Leaf dataset, and 80.3% on the Rice Leaf dataset despite 3 distinct distortions applied simultaneously. The individual attack results are also appended in the Table 4.17. This data indicates that, while there is room for development, our model AgriTL-ViT has significant promise for successful application in field situations. These steps will enhance the robustness and reliability of our approach in real-world deployment scenarios.

Furthermore, we observe that real-world field testing is crucial for validating the practical applicability of our system. As part of our future work, we plan to collect and evaluate our model on images captured directly from agricultural fields using mobile devices under diverse environmental conditions.

Table 4.17: Accuracy of the Proposed Model in Real-World Conditions with Various Augmentation Techniques

Augmentation	Tomato Leaf Dataset Accuracy	Maize Leaf Dataset Accuracy	Rice Leaf Dataset Accuracy
Only ColorJitter	0.712	0.574	0.769
Only GaussianBlur	0.698	0.562	0.758
Only RandomErasing	0.723	0.591	0.781
Combine	0.749	0.614	0.803

This analysis demonstrates that each augmentation approach adds uniquely to increasing the model's resilience in practical situations. Across all three datasets, each technique produced measurable performance gains over models trained without augmentation. This demonstrates their specific utility in addressing various environmental restrictions commonly seen in agricultural field scenarios. Furthermore, we observe that real-world field testing is crucial for validating the practical applicability of our system. As part of our future work, we plan to collect and evaluate our model on images captured directly from agricultural fields using mobile devices under diverse environmental conditions.

4.1.17.2 Hardware Constraint

To represent realistic farmer use scenarios, we tested the AgriTL-ViT model on low-resolution images created by downsizing the existing Tomato, Maize, and Rice leaf datasets. This sim-



Figure 4.20: Original leaf image and its augmented versions using color jitter, gaussian blur, and random erasing

ulates conditions comparable to those observed in mobile-acquired photos from outdoor contexts. We downsampled the Tomato, Maize, and Rice leaf datasets to 128×128 pixels and 200×200 pixels, the resolution frequently captured by mobile devices in agricultural situations.

Table 4.18: Accuracy of AgriTL-ViT on Low-Resolution (128×128) and (200×200) Images Across Different Plant Disease Datasets

Model	Dataset	Accuracy (200×200)	Accuracy (128×128)
AgriTL-ViT	Tomato Leaf	0.976	0.941
AgriTL-ViT	Maize Leaf	0.961	0.916
AgriTL-ViT	Rice Leaf	0.958	0.923

These results show that AgriTL-ViT maintains good classification accuracy even when evaluated on lower-resolution images, demonstrating its reliability and usability in field deployments. However, when resolution drops, accuracy declines across all datasets. This underlines an important problem in real-world applications: ensuring consistent performance even under suboptimal imaging conditions.

4.1.17.3 Model Parameters

To evaluate the impact of important hyperparameters on model performance, we ran a sensitivity analysis of the AgriTL-ViT model on the Tomato dataset, changing the number of attention heads and embedding dimensions. As demonstrated in Table 4.19, the default configuration with 12 attention heads and 768-dimensional embeddings had the maximum accuracy of 98.5%. However, decreasing these parameters to 8 heads and 512 dimensions led to a minor loss in accuracy (96.8%), demonstrating a good balance between performance and computational cost. A further decrease to 4 heads and 384 dimensions resulted in 94.5% accuracy, showing that the model maintains reasonable performance even in lightweight setups. These findings are critical for directing future model compression, especially for deployment in resource-constrained situations like edge devices and mobile phones.

Table 4.19: Sensitivity Analysis on the Tomato Dataset Showing the Impact of Attention Heads and Embedding Dimensions on Model Accuracy.

Configuration	Attention Heads	Embedding Dimensions	Test Accuracy (%)
Config-1	4	384	0.945
Config-2	8	512	0.958
Config-3	12	768	0.985

Identifying diseases of leaves at an early stage is a critical research endeavour in farming, potentially enhancing yield output and profitability. To solve this problem, researchers used machine learning, CNN-based approaches, and a combination of both to diagnose plant illnesses automatically. However, in this study, we proposed an AgriTL-ViT model with a stacked Trans ResNet Encoder. The proposed AgriTL-ViT model uses ResNet modules instead of MLP modules to extract hierarchical features from the input image. In the Vision Transformer model, we have also proposed dual multi-head self-attention to provide greater expressive power and versatility in collecting local and global elements, which is essential for understanding complicated visual landscapes. The AgriTL-ViT model's performance has been compared to five cutting-edge techniques using the tomato, maize, and rice leaf dataset. Over the experimental analysis, the AgriTL-ViT model outperformed cutting-edge approaches in the tomato, maize, and rice leaf dataset, with an accuracy of 0.985%, 0.972% and 0.970% respectively. In future, we will use various plant disease datasets to improve the value of our proposed method. Using real-time data, we will also construct a real-world environment to evaluate the suggested model. Furthermore, we intend to expand the model to incorporate multi-label classification, which will allow it to detect many co-occurring diseases inside a single leaf image. This modification will make the model more suitable to complicated agricultural settings where mixed illnesses are widespread.

4.2 ADResNet with Attention: A Multi-Model Approach for Leaf Disease Detection and Pest Classification

4.2.1 Overview

Agriculture is a pillar of national prosperity and economic stability, driven by the rising requirement for food production to feed a continually expanding population. However, cli-

mate change, pests, illnesses, and pathogens significantly impact agricultural output and crop quality. Environmental factors influence plant development, but unfavourable climates can foster the spread of illnesses and pests, resulting in decreased yields and crop quality. Preventing these risks is critical to minimize economic losses, making it an increasing concern for scholars. Traditionally, professionals diagnose plant leaf pests and diseases through visual inspection, which is time-consuming and heavily reliant on their knowledge. Furthermore, incorrect pesticide application to control pests might impair the plant's developing environment, aggravating agricultural issues [252]. Technological advancements in agriculture, such as artificial intelligence, communication technologies, and machine learning techniques, like BPNN, SVM, and Kmeans clustering, have been utilised to recognise crop leaf pests and diseases [253]. However, previous feature selection and extraction approaches rely substantially on the researcher's skill, making image pre-processing complex and challenging. In recent years, convolutional neural networks (CNNs) have demonstrated outstanding results in feature extraction methods and classifiers for plant leaf illness and pest identification tasks [253]. Because CNNs' learning process does not need manual selection of features, this technology has been widely used in agriculture. However, most CNN-based models include deep layers, complicated parameters, and computationally intensive operations that necessitate sophisticated hardware, making them difficult to deploy on resource-constrained devices for real-time identification. Recently, lightweight models have increasingly been employed for plant leaf disease identification [253]. However, existing lightweight models are inadequate for accurately, quickly, and efficiently recognising disease spots in plant leaves, which might vary in size and colour. Therefore, early detection of pests and leaf diseases on limited-resource terminal devices is challenging due to restricted memory and computational capabilities. This work presents a novel ADResNet model with attention to addressing the abovementioned issue. The primary contributions of the study are outlined as follows:

- We proposed a lightweight and computationally efficient architecture incorporating Asymmetric Dilated Convolution (ADCNN) to decrease computational complexity and parameter count while preserving high accuracy.
- We proposed combining InceptionResNetV1 with a self-attention strategy to enable robust feature extraction and better focus on disease-specific regions, improving classification performance.
- We tested the proposed model on the CCMT dataset and found that it outperformed state-of-the-art techniques in plant illness classification, scoring 97.7% accuracy, 97.5%

6

Table 4.20: Comparison of Different Methods for Plant Disease Classification

Author	Purpose	Methodology	Dataset	Performance		Parameters		Advantages	Disadvantages
				Acc.	Prec.	Recall	F1		
[254]	A reliable method for identifying rice plant diseases by analyzing the size, shape, and color of lesions in leaf images.	CNN	Rice	✓	×	×	×	Combines feature extraction and categorization to improve performance.	Relies on Otsu's thresholding, which may not generalize to diverse datasets.
[255]	To efficiently identify plant diseases.	CNN	Rice, Plant, Cassava	✓	✓	✓	✓	High accuracy on multiple datasets.	Decreased precision for imbalanced datasets, such as Cassava.
[256]	Automated recognition of plant leaf disease.	GRU+CBAM	CCMT	✓	✓	✓	✓	Reduces need for expensive labeled data.	Limited performance evaluation on complex datasets.
[257]	To identify and classify tomato leaf disease.	VGG19, ResNet101, MobileNetV2	PlantVillage, CCMT	✓	✓	✓	✓	Uses pre-trained models to save training time.	May require fine-tuning for different datasets.
[258]	To decrease computational cost, memory utilization, and overfitting in plant disease identification.	DCNN	Rice Leaf	✓	✓	✓	✓	Reduces computational cost, memory use, and overfitting.	Requires additional validation on larger, more diverse datasets.
[259]	Enhancing feature extraction and classification performance, particularly with minimal training data.	EfficientNetB0	DiaMOS Plant	✓	✓	✓	✓	Improves classification accuracy by focusing on key regions.	Needs fine-tuning for various datasets.
[260]	To recognize and classify plant diseases and pests.	LSTM-CNN	Turkey	✓	✓	✓	✓	Classifies plant pests and diseases with high accuracy (99.2%).	LSTM layers require significant computational resources.
[261]	To improve feature extraction and classification performance for rice leaf disease identification.	ResNet34 with self-attention	Rice	✓	✓	✓	✓	Enhances feature extraction by focusing on key regions of an image.	Limited evaluation on real-world diverse datasets.
[262]	To create a lightweight object-detection model using YOLOv8n for vegetable disease and pest identification.	YOLO-FMDI	Tomato	✓	×	×	×	Lightweight and efficient for edge technologies.	Evaluations focused mainly on tomatoes, limiting generalization.
[263]	To improve plant disease identification.	MWOA-CNN	PlantVillage	✓	✓	✓	✓	Optimizes CNN hyperparameters effectively.	Implementation may be complex for inexperienced users.

29

precision, 97.2% recall, and 96.7% F1 score.

4.2.2 Related Work

Artificial Intelligence (AI) advancements have led to breakthroughs in computer vision, with extensive research accessible on detecting crop illnesses and pests. Table 4.20. summarises numerous research approaches used to identify plant diseases, including their aims, datasets, performance measures, benefits, and limitations.

4.2.3 ADResNet with Attention: A Lightweight Model

This section describes the proposed model, ADResNet with Attention, in detail.

4.2.3.1 Proposed ADResNet with Attention

The proposed ADResNet with Attention model is a deep learning architecture that classifies plant illnesses using convolutional layers, AD convolutions, InceptionResNetV1, and a Self-attention mechanism. This model uses CNN layers to systematically extract low- and high-level features for discriminating between plant illnesses. The Asymmetric Dilated

131

Convolution improves the model's capacity to capture local and global information, which is particularly important when illnesses damage large areas of plant leaves. Following that, the InceptionResNetV1 module improves network effectiveness by retrieving multi-scale fea-

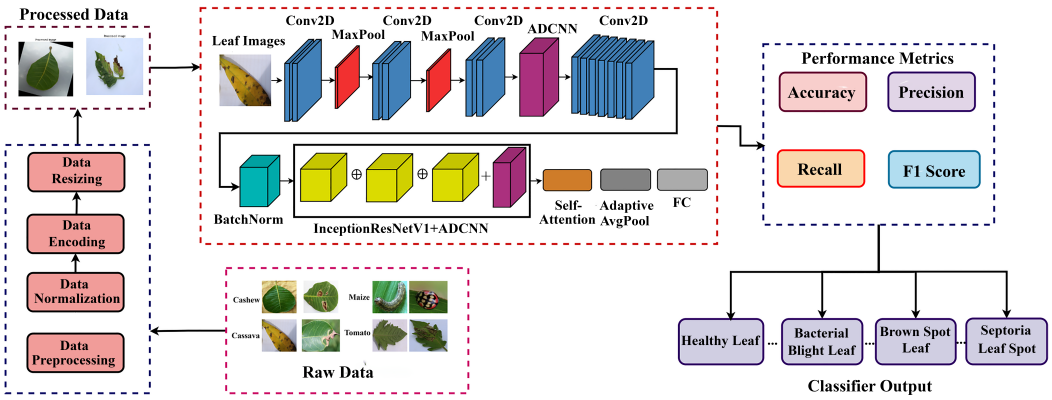


Figure 4.21: Proposed hybrid model architecture

tures, while the residual connections aid model training by addressing the vanishing gradient problem. To better emphasise regions affected by illness, the model includes a Self-Attention mechanism that improves accuracy by focusing on subtle disease signs. Finally, using adaptive average pooling with a fully connected layer improves network performance independent of input image size. This makes it highly flexible and effective at spotting plant diseases, as seen in Fig. 4.21. Here is an explanation of each model component and its significance:

1. The model starts with two CNN layers that collect fundamental information from the input image. The initial CNN layer (Conv1) uses a 3x3 filter with 64 channels, followed by a ReLU activation. This is necessary to capture low-level details, such as edges and textures. Subsequently, a MaxPooling layer downsamples the feature map to reduce spatial dimensions and improve computational efficiency. The second CNN layer (Conv2) has 128 filters, allowing the model to recognise more complicated characteristics. Another MaxPooling process follows, lowering the size of the feature maps while keeping the necessary spatial organisation. Conv1:

$$X_1 = \text{ReLU}(F_{\text{tr1}} \cdot Y + \text{bias}_1) \tag{4.2.1}$$

Where Y represents the input image, F_{tr1} is a 3×3 filter with 64 channels, bias_1 denotes the bias term, and $*$ indicates the convolution operation. The output X_1 is the feature map obtained after applying the convolution operation followed by the ReLU activation function. **MaxPooling (Conv1):**

$$X_1^{\text{pooled}} = \text{MaxPooling}(X_1) \tag{4.2.2}$$

This operation decreases the spatial dimensions of X_1 while preserving the most important features. Another CNN and Maxpooling process follows the same process as ConV1.

2. To improve feature extraction, we use an Asymmetric Dilated Convolution layer (ADCNN). This layer enhances the receptive field without raising the parameters, allowing the network to extract additional contextual information from a larger spatial region in the image.

$$X_{\text{ADCNN}} = \text{ReLU}(F_{\text{ADCNN}} * X_2^{\text{pooled}} + b_{\text{ADCNN}}) \tag{4.2.3}$$

where F_{ADCNN} represents the dilated filter, X_2^{pooled} denotes the output from the second pooling layer, and X_{ADCNN} is the resulting output feature map from the ADCNN operation.

3. Subsequently, Batch Normalization facilitates training stabilisation and normalizes feature distributions before their transmission to subsequent layers.

$$\hat{X}_{BN} = \frac{X_{ADCNN} - \mu}{\sigma + \epsilon} \tag{4.2.4}$$

$$X_{BN} = \gamma \hat{X}_{BN} + \beta \tag{4.2.5}$$

Where, μ, σ represent Batch mean and variance, ϵ represent a small constant, and γ, β represent learnable scale and shift parameters.

4. After that, the InceptionResNetV1 block contains three parallel branches that process the input: The first branch uses a 1×1 CNN to minimize dimensionality while capturing fine-grained features:

$$X_1^{\text{inception}} = \text{ReLU}(F_{1\times1} * X_{\text{BN}} + b_{1\times1}). \tag{4.2.6}$$

The second branch employs a 1×1 CNN followed by a 3×3 CNN, which focuses on intermediate-level spatial data:

$$X_2^{\text{inception}} = \text{ReLU}(F_{3\times3} * \text{ReLU}(F_{1\times1} * X_{\text{BN}} + b_{1\times1}) + b_{3\times3}). \tag{4.2.7}$$

The third branch uses a 1×1 CNN followed by two 3×3 CNNs, allowing the network to capture more complex features and broaden its receptive field:

$$X_3^{\text{inception}} = \text{ReLU}\big(F_{3\times3} * \text{ReLU}(F_{3\times3} * \text{ReLU}(F_{1\times1} * X_{\text{BN}} + b_{1\times1}) + b_{3\times3}) + b_{3\times3}\big). \tag{4.2.8}$$

The output of the InceptionResNetV1 block consists of a concatenation of the branches, followed by a residual connection:

$$X_{\text{inception}} = X_1^{\text{inception}} \oplus X_2^{\text{inception}} \oplus X_3^{\text{inception}} + X_{\text{BN}}, \tag{4.2.9}$$

where $\oplus$ represents the concatenation operation.

59

5. We add a Self-Attention mechanism to help the model focus on the most essential traits. This technique dynamically balances various parts of the feature map, allowing the model to prioritise disease-affected areas.

$$X_{\text{attention}} = \text{Attention}(X_{\text{inception}}). \tag{4.2.10}$$

32

6. Following the extraction of features and attention steps, an Adaptive Average Pooling layer is used to reduce the feature map to a fixed-size vector. This step ensures that the model can withstand fluctuations in input image size.

$$X_{\text{pooled}} = \text{AdaptiveAvgPool}(X_{\text{attention}}). \tag{4.2.11}$$

41

7. Lastly, a Fully Connected (FC) layer maps the pooled characteristics to the output space and predicts the plant disease's class label.

$$\hat{X} = \text{FC}(X_{\text{pooled}}) = W_{\text{fc}} \cdot X_{\text{pooled}} + b_{\text{fc}}. \tag{4.2.12}$$

134

where W_{fc} is the weight matrix, and $\hat{X}$ is the predicted output.

Overall, the proposed ADResNet with Attention model provides effective and precise plant

Table 4.21: ADResNet with Attention Model Architecture Details

Layers	Output Shape	Parameters
Conv2d-1	(32,64,224,224)	1,792
MaxPool2d-2	(32,64,112,112)	0
Conv2d-3	(32,128,112,112)	73,856
MaxPool2d-4	(32,128,56,56)	0
Conv2d-5	(32,256,56,56)	98,560
Conv2d-6	(32,256,56,56)	196,864
ADCNN-7	(32,256,56,56)	0
Conv2d-8	(32,512,56,56)	131,584
Conv2d-9	(32,512,56,56)	131,584
Conv2D-10	(32,512,56,56)	2,359,808
Conv2D-11	(32,512,56,56)	131,584
Conv2D-12	(32,512,56,56)	2,359,808
Conv2d-13	(32,512,56,56)	2,359,808
Conv2d-14	(32,512,56,56)	786,944
BatchNorm2d-15	(32,512,56,56)	1,024
InceptionResNetV1-16	(32,512,56,56)	0
Conv2d-17	(32,64,56,56)	32,832
Conv2d-18	(32,64,56,56)	32,832
Conv2d-19	(32,512,56,56)	262,656
Attention-20	(32,512,56,56)	0
AdaptiveAvgPool2d-21	(32,512,1,1)	0
Linear-22	(32,22)	11,286
Total Parameters		**8,972,822**

disease categorization, making it appropriate for real-world farming uses.

4.2.3.2 Layer details of ADResNet with Attention Model

This section describes the configuration in layers of the ADResNet suggested by the Attention framework. Table 4.21 depicts the layers in the suggested paradigm. A batch of 32 photos requires an input size of 224 × 224 pixels. Each layer is specified by its output form and the number of parameters. The new "ADResNet with Attention Model " has 8,972,822 parameters.

4.2.4 Experimental Setup

This section discusses the experimental steps conducted to validate the suggested model.

Table 4.22: Hyperparameter Settings for the Model

Hyperparameter	Values
Classes	22
Optimizer	Adam
Epoch	30
Learning Rate	Constant, Decay
Batch Size	32

4.2.4.1 Dataset Description :

This research paper uses the ADResNet with Attention model. This model is trained on a publicly available CCMT dataset for crop pest and disease detection [105], which is discussed in Chapter 2, Section 2.4.

4.2.4.2 Hardware Specification and Hyperparameter Table:

Our tests are done on a Windows 11 Pro version 22H2 computer with an Intel(R) Xeon(R) Silver 4114 CPU @ 2.10GHz and 16GB RAM. The suggested ADResNet with Attention model is implemented on the CCMT dataset using Google Colab Pro+, Pytorch, NVIDIA GTX 1070 GPU, and Python 3.8. Hyperparameter Table: Each experiment is conducted for 30 epochs. The batch size is configured to 32. The Adam optimizer is employed to adjust the model weights. Two distinct learning rates are employed: a constant learning rate for robust training and a decaying learning rate that progressively reduces across epochs to improve convergence and mitigate overfitting. The hyperparameter specifications are presented in Table 4.22.

4.2.5 Result Analysis

This section assesses training, validation, and testing outcomes for the CCMT datasets. Divide the dataset into three categories: training (70%), validation (30%), and testing (30%). To improve the visual quality, the data was pre-processed with scaling, normalization, and other techniques. The ADResNet with Attention model is also trained and validated on pre-processed data. Table 4.23. shows the outcomes that were achieved.

4.2.5.1 Evaluation Result for CCMT Dataset

This section presents performance metrics for all 23 classes in the CCMT dataset. The dataset poses a significant challenge for detecting and classifying labels, necessitating a thorough analysis of classifier performance. Table 4.24. lists the quality evaluations and

Table 4.23: Quantitative Results of Leaf Disease Identification Using CCMT Dataset

Dataset	Model	Set	Accuracy (%)	Loss
CCMT	ADResNet with Attention	Train Set	96.5	0.020
		Validation Set	97.8	0.010
		Testing Set	97.7	0.011

quantity of tested tables for each level. The table displays the recall, f1 score, and precision values for all 22 classes across ADResNet with Attention model.

Table Table 4.24 shows that the ADResNet with Attention model has higher precision and lower recall for *Tomato verticillium wilt* diseased compared to the 22 other classes. This indicates that the response had no false positives. The model outperforms in classifications such as *Cashew Anthracnose3102*, *Cassava brown spot*, and *Tomato verticillium wilt*, with high Precision, Recall, and F1-Score. However, it is adversely affected in certain situations, like *Maize fall armyworm* and *Maize leaf blight*. The precision for *Maize fall armyworm* is lower than 0.678, signifying that the model generates more false positives in this category. Similarly, *Maize leaf blight* has a poorer F1-Score (0.774) due to missing cases, as evidenced by its Recall of 0.920. The model performs consistently in most classes, particularly for tomato plants, cassava, and cashew illnesses. However, some variation in metrics within the same kind of crop indicates that specific categories are more difficult for the model to classify effectively.

4.2.5.2 Ablation Study

This section discusses the ablation study that was conducted to extensively analyze each of the different sections in the proposed framework. The study includes six cases that examine the performance and efficacy of various models. The cases are outlined as follows:

Case 1: Training the dataset with a 2D Convolutional Neural Network (CNN)

Case 2: Training the dataset with a 2DCNN+Asymmetric Dilated Convolutional Neural Network (ADCNN)

Case 3: Training the dataset employing the 2DCNN+ADCNN+modified Inception ResNetV1 model.

Case 4: Training the dataset using the proposed hybrid model (ADResNet with Attention)

Table 4.25. illustrates ablation research completed on the ADResNet with Attention models

Table 4.24: Image Quality Metrics of ADResNet with Attention Model for 23 Classes of CCMT Dataset

Class	Precision	Recall	F1-Score
Cashew Anthracnose	0.958	0.959	0.969
Cashew Gumosis	0.855	0.858	0.857
Cashew Healthy	0.969	0.959	0.935
Cashew Leaf Miner	0.952	0.942	0.932
Cashew Red Rust	0.966	0.959	0.975
Cassava Bacterial Blight	0.944	0.967	0.952
Cassava Bacterial Blight	0.910	0.902	0.906
Cassava Brown Spot	0.972	0.981	0.976
Cassava Green Mite	0.918	0.989	0.952
Cassava Healthy	0.908	0.914	0.982
Cassava Mosaic	0.952	0.940	0.930
Maize Fall Armyworm	0.678	0.934	0.906
Maize Grasshopper	0.876	0.967	0.952
Maize Healthy	0.962	0.970	0.972
Maize Leaf Beetle	0.852	0.858	0.857
Maize Leaf Blight	0.958	0.920	0.774
Maize Leaf Spot	0.945	0.878	0.935
Maize Streak Virus	0.920	0.911	0.915
Tomato Healthy	0.969	0.959	0.935
Tomato Leaf Blight	0.952	0.942	0.932
Tomato Leaf Curl	0.966	0.959	0.975
Tomato Septoria Leaf Spot	0.966	0.959	0.975
Tomato Verticillium Wilt	0.979	0.914	0.982

Table 4.25: Ablation Study Score for ADResNet with Attention Model for CCMT Datasets

Ablation Study Cases	Cases	Accuracy	Precision	Recall	F1 Score
with 2D CNN	1	0.875	0.870	0.865	0.867
with 2D CNN + ADCNN	2	0.903	0.890	0.880	0.885
with 2D CNN + ADCNN + InceptionResNetV1	3	0.942	0.930	0.920	0.925
with ADResNet with Attention (our proposed model)	4	0.977	0.975	0.972	0.967

for the CCMT datasets, which involved thoroughly analysing different training situations. The instances labelled 1–4 show different neural network architectures used in training. The metrics under consideration include crucial performance measures such as f1-score, precision, recall, and accuracy. Table 4.25. shows that Case 1 employs a 2D CNN to achieve an accuracy of 0.875, with precision (0.870), recall (0.865), and f1 score values of 0.867.

Our studies revealed that different deep learning architectures produce varying performance levels on the CCMT dataset. Initially, we trained the 2DCNN+ADCNN model (Case 2), which yielded an accuracy of 0.903, precision of 0.890, recall of 0.880, and F1-score of 0.885. To boost performance further, we tested the 2D CNN +ADCNN+ InceptionResNetV1 model (Case 3), which enhanced accuracy by 0.942, precision by 0.930, recall by 0.920, and F1-score of 0.925. To address feature extraction and learning limitations, we included a self-attention mechanism in Case 4, which increased accuracy to 0.977, precision of 0.975, recall of 0.972, and F1-score of 0.967. These results show how architectural changes and attention processes lead to a model that improves classification accuracy while addressing issues like overfitting and generalization.

4.2.5.3 Comparison with the State of Art Techniques

The following part compares ADResNet with Attention to various cutting-edge techniques for detecting and classifying leaf disease on multi-class CCMT datasets. Table displays the performance results for earlier techniques [258], [264] and [265]) over a variety of criteria. Regarding precision, recall, f1-score, and accuracy, the ADResNet with Attention model outperforms existing multi-class categorisation methods for CCMT datasets. Table 4.26. provides a comparative analysis of the proposed ADResNet with Attention model in relation to various baseline methods, evaluated under constant and decay learning rate conditions. The findings indicate that employing a decay learning rate enhances the performance of all models relative to a constant learning rate, facilitating improved weight adjustments and mitigating convergence issues. VGG19 demonstrates a notable enhance-

Table 4.26: Comparison of Models with Constant and Decay Learning Rate

Reference	Model	Constant Learning Rate				Decay Learning Rate			
		Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
[258]	VGG19	0.809	0.801	0.789	0.799	0.927	0.927	0.959	0.908
[264]	GRU+CBAM	0.916	0.924	0.924	0.925	0.956	0.944	0.964	0.955
[265]	CNN + Swin Transformer	0.934	0.956	0.956	0.956	0.956	0.956	0.956	0.956
Our	**ADResNet + Attention**	**0.959**	**0.944**	**0.952**	**0.956**	**0.977**	**0.975**	**0.972**	**0.967**

ment in performance, increasing from an accuracy of 0.809 with a constant learning rate to 0.927 with a decay learning rate. GRU+CBAM shows an improvement in accuracy from 0.916 to 0.956, highlighting the effectiveness of attention mechanisms in disease classification. The CNN utilizing the Swin Transformer demonstrates strong performance in both scenarios, showing a marginal increase in accuracy from 0.934 to 0.956, which suggests its robustness.

The proposed ADResNet with Attention model demonstrates superior performance compared to baseline methods, attaining an accuracy of 0.959, precision of 0.944, recall of 0.952, and an F1-score of 0.956, all while maintaining a constant learning rate. Implementing a decay learning rate enhances performance, achieving an accuracy of 0.977, precision of 0.975, recall of 0.972, and an F1-score of 0.967. This indicates that integrating Asymmetric and Dilated Convolution with a self-attention mechanism improves feature extraction and classification accuracy.

This study presents the ADResNet with Attention model, utilizing advancements in artificial intelligence for precise and efficient identification of plant diseases, particularly on resource-limited devices. Integrating AD Convolution and self-attention mechanisms enables the model to concentrate on disease-specific areas, thereby improving detection accuracy. The proposed model demonstrates superior performance compared to traditional and state-of-the-art methods on the CCMT dataset, attaining an accuracy of 0.977, precision of 0.975, recall of 0.972, and an F1-score of 0.967 with a decay learning rate. The results validate its efficacy in classifying plant diseases. Nonetheless, its relevance to various agricultural contexts and intricate disease variations necessitates additional research. Future research must prioritize expanding datasets, enhancing environmental robustness, and integrating real-time deployment to optimize the model's effectiveness in practical agricultural applications.

4.2.6 Summary

This chapter describes an attention-based transfer learning model that classifies plant leaf diseases using AgriTL-ViT and Attention-based ADResNet. The proposed approach combines crowdsourced Vision Transformer systems with attention-enhanced residual networks to target disease-specific sections and improve feature extraction. It reduces computing costs while maintaining high classification quality for tomatoes, maize, rice, and CCMT. A comparison analysis shows higher accuracy, precision, recall, and F1-score than previous approaches. Overall, the chapter demonstrates that attention-guided deep learning models are useful and reliable methods for automating plant disease diagnosis.

Chapter 5

Federated Learning based Model for Plant Leaf Disease Classification

*The purpose of this chapter * is to propose a federated learning-based framework for plant leaf disease classification in distributed agricultural environments. The framework combines CNN-based feature extraction, Transformer encoders, and federated optimization techniques such as FedAvg and FedProx to ensure accurate, privacy-preserving, and efficient collaborative training across distributed clients.*

5.1 Towards Distributed Method for Plant Disease Diagnosis: A Hybrid CNN-Transformer Model with Federated Learning

5.1.1 Overview

This chapter, which is based on the approach to methodologies presented in Chapters 3 and 4, proposes a federated learning based framework for plant leaf disease classification suitable to distributed agricultural settings. In contrast to the other chapters where a centralized model training process was performed, this chapter pays attention to the collaborative model training process of the distributed clients without exchanging raw data.

Federated Learning (FL) is a distributed training procedure that enables multiple clients, such as edge devices or agricultural centres, to collaboratively train a shared model without

*The research work covered in this chapter has been communicated in: Meenakshi Srivastava, Varsha Sisaudia and Jasraj Meena, "Towards Distributed Method for Plant Disease Diagnosis: A Hybrid CNN-Transformer Model with Federated Learning," *Engineering Applications of Artificial Intelligence*, April 2026

sharing raw data. Only local model updates are aggregated on a central server. This design assurances data privacy, minimises transmission expenses, and allows the model to acquire from various real-world, non-IID data distributions. In farming, FL is mainly adept at making strong and scalable artificial intelligence keys that maintain farmers' data autonomy while enabling continuous education across many fields.

We present an Federated Learning (FL)-based architecture designed to recognise plant leaf illness, guaranteeing a healthy and privacy-preserving result suited for real-world farming contexts. This architecture is intended for distributed contexts, where each client, such as a local agricultural centre, trains a model on its non-IID leaf image dataset. The suggested model integrates the spatial feature extraction capabilities of Convolutional Neural Networks (CNNs) using a ResNet34 backbone with the global context modelling strengths of Transformer encoders. A central server coordinates the training method, employing FedAvg to combine locally trained model weights from four clients. To mitigate divergence caused by data heterogeneity among consumers, we utilise FedProx regularisation, enhancing model generalisation and training stability across diverse data distributions.

Each client uses the AdamW optimiser to achieve sufficient local training while balancing the model's convergence rate and efficacy. Moreover, Focal Loss addresses class imbalances in agricultural datasets, whilst attention entropy regularisation enhances the model's concentration on illness-affected areas. To improve model transparency, we utilise SHAP and attention-based visualisations, which show stakeholders the areas of the leaf image that exerted the most substantial impact on the model's conclusions. This federated learning centred hybrid approach offers scalability, privacy, and adaptability, while mitigating the substantial disadvantages of centralised training in agricultural artificial intelligence.

Our contributions are summarised below:

- We present a CNN + Transformer hybrid model optimised with AdamW and FedProx for decentralised FL environments.
- We test the model on two datasets, Mango Leaf Disease and Rice Leaf Disease, with real-world non-IID data distributions and four simulated clients.
- We use Focal Loss and Attention Regularisation to address data imbalance and improve performance in rice illness classification.
- We offer interpretability with SHAP and attention heatmaps, which allow users to see which areas of the leaf images are driving predictions.

- In low-resource and heterogeneous data conditions, our model outperforms CNN-only, ViT-only, and baseline FL techniques by achieving up to 97.4% accuracy on mango and 94.10% on rice.

5.1.2 Related Work

Federated Learning (FL) has emerged as a viable decentralised learning approach that overcomes the limitations of traditional centralised training, particularly in data-intensive industries such as healthcare, banking, and agriculture. In FL, data is still distributed across client devices or nodes, with only local model updates relayed to a central server for aggregation, privacy preservation, communication overhead reduction, and scalability improvement. FL holds great promise in agriculture for developing AI-based diagnostic solutions that don't need centralised data collection. This is especially valuable when crop image data is collected on remote farms or agricultural research institutes, where internet access is frequently limited and data ownership is a big concern. FL delivers localised learning on these edge devices, enabling collaborative plant disease categorisation across geographically scattered locations while safeguarding data privacy and institutional autonomy. Given the rising implication of privacy-preserving and scalable AI in precision farming, FL is being calculated for applications like crop monitoring, illness diagnosis, and yield forecasting. Yet, problems like as non-IID data transmission, limited computing power at edge clients, and interpretability of models continue to stop its widespread adoption in real-world situations. This section looks at numerous AI and FL-based methodologies suggested in the research for recognizing plant illnesses. To better appreciate the growth and limitations of this topic, we will review the available literature on FL in farming and plant disease classification, as shown in Table 5.1.

Aggarwal M et al. [266] introduce a lightweight, privacy-preserving FL technique for properly recognizing rice leaf illnesses in an Internet of Agriculture Things (IoAT) environment. The authors investigate two methods: Federated Transfer Learning, in which pre-trained models, such as EfficientNetB3, are fine-tuned at edge devices, and Federated Feature Extraction, in which only local characteristics are extracted and transmitted to a central server for global model training. The results show that the federated feature extraction method significantly reduces GPU memory, CPU/GPU usage, and training time while maintaining high accuracy, precision, recall, and AUC. Hari P et al. [267] suggested FDL-IWT (Federated Deep Learning with Intelligent Weight Transferring), an innovative and resource-effective

method for dealing with non-IID and imbalanced plant illness data in federated learning environments. The authors also grow a lightweight model, PMACNN (Parallel Multi-Scale Attention CNN), which has only 0.13 million parameters and can run efficiently on edge devices. Unlike traditional federated systems that aggregate local models equally, FDL-IWT assigns adaptive weights to every client's model based on regional performance, resulting in more accurate and balanced global updates. The system was tested using public plant illness datasets in simulated federated settings and achieved a high-test accuracy of 97.5%, outperforming existing methods like FedAvg, FedAdam, and FedAdagrad.

Kaur K et al. [268] suggested a federated learning framework for cucurbit leaf illness diagnosis based on CNN, permitting for decentralized training across many farms without sharing raw data. A lightweight CNN is implemented on local devices to evaluate leaf images and train models, while a central server accumulates gradient changes, protecting data privacy and lowering bandwidth. Evaluations on a multi-site dataset show that this collaborative method is as accurate as centralized models while knowingly improving privacy and scalability in real-world farming situations. The results show that federated CNNs hold promise for illness categorization in numerous distributed situations. Tripathy R et al. [269] suggested a federated learning framework customised for rice leaf illness categorisation, which was supplemented with an optimisation-based technique to increase model performance across distant clients. Unlike typical federated averaging, this strategy addresses data heterogeneity and resource restrictions by optimizing hyperparameters and network structure at local and global levels. Using a variety of non-IID rice leaf illness datasets, the technique incorporates third-party optimization algorithms, like Bayesian or Swarm-based tuning, to improve feature extraction and model updates, resulting in significant improvements in accuracy and communication efficiency over typical FL models. Experimental results show that the optimized federated architecture recovers classification accuracy while significantly decreasing convergence time and costs, making it a promising method for scalable, privacy-preserving rice illness identification.

Mamba Kabala D et al. [270] Provides a privacy-preserving strategy for identifying crop diseases based on a federated learning (FL) architecture. Instead of uploading sensitive agricultural picture to a centralised machine, the model is trained locally on several client devices (such as mobile phones or farm edge devices), and only the acquired model parameters are communicated with a central server for aggregate purposes. The authors use deep convolutional neural networks (CNNs) to accurately diagnose crop disease from photos across distributed, diverse datasets. The study shows that the federated learning strategy

can perform similarly to typical centralized training approaches, with only a minor reduction in accuracy.

Table 5.1: Analysis of the Literature Survey on Different Parameters

Author	Dataset	Method Used	Performance Indicators	Limitation
Aggarwal M et al. [266]	Rice Leaf	TL + FL	Accuracy, Loss	The paper's key weakness is its reliance on stable network access for federated updates, which may be difficult in rural agricultural locations with inadequate infrastructure.
Hari P et al. [267]	PlantVillage	FDL-IWT	MSE, RMSE, MAE, Accuracy	The limitation of this paper is that it focuses on a curated dataset and a refined ResNet50 architecture but does not address aspects like generalizability, real-world deployment, and interdisciplinary integration.
Kaur K et al. [268]	Cucurbit Leaves	CNN + FL	Accuracy, F1-Score, Precision, Recall	The paper's key disadvantage is the possibility of performance degradation caused by data heterogeneity and changing image quality among remote client devices.
Tripathy R et al. [269]	Rice Leaf	SHAO_Lenet + FL	Precision, Recall, F1-Score, Accuracy, MSE, RMSE	To increase computing complexity and tuning cost caused by the optimisation process, which may limit implementation on low-power edge devices.
Mamba Kabala D et al. [270]	Grape, Apple, Corn, Tomato	ViT-B16, ViT-B32	F1-Score, Accuracy	The model's performance slightly degrades in highly non-IID data scenarios, affecting accuracy and convergence stability.
Aggarwal M et al. [65]	Rice Dataset	DL + FL	Accuracy	The proposed lightweight FL model struggles to maintain high accuracy when clients have unbalanced class distributions, particularly with underrepresented disease classes, leading to biased global model performance.

5.1.3 Proposed Federated Learning with Transformer-augmented CNN (FL-TransCNN) Model

This section addresses the proposed hybrid model's design, which combines Convolutional Neural Networks (CNN) and Transformer encoders in a Federated Learning (FL) framework

for plant disease categorization. The model intends to learn detailed spatial and contextual information while ensuring data privacy across distributed clients.

5.1.3.1 Overview

The proposed model uses a hybrid deep learning architecture that combines the strengths of Convolutional Neural Networks (CNN) and Transformer encoders within a Federated Learning (FL) framework to classify plant leaf diseases while maintaining data privacy reliably which is shown in Figure 5.1. At its foundation, each distributed client has a lightweight model that includes a ResNet34-based CNN backbone for extracting rich spatial data from leaf images and Transformer encoder layers that capture global contextual relationships between those features. These features are combined and fed into a classification head to forecast the disease class. The model is trained locally on each client's data rather than sharing raw photos, thereby addressing real-world challenges such as data ownership, privacy, and geographical distribution. Each client periodically sends only the learnt model parameters to a central server. The server uses Federated Averaging (FedAvg) to aggregate the weights and update the global model. To reduce data heterogeneity across clients (non-IID distributions), we use FedProx regularisation during local training to stabilise convergence. This federated hybrid technique works notably well for agricultural information collected from places with considerable diversity in illumination, leaf texture, and disease signs. The result is a scalable, privacy-aware, and interpretable system capable of accurate plant disease categorization in real-world distributed settings.

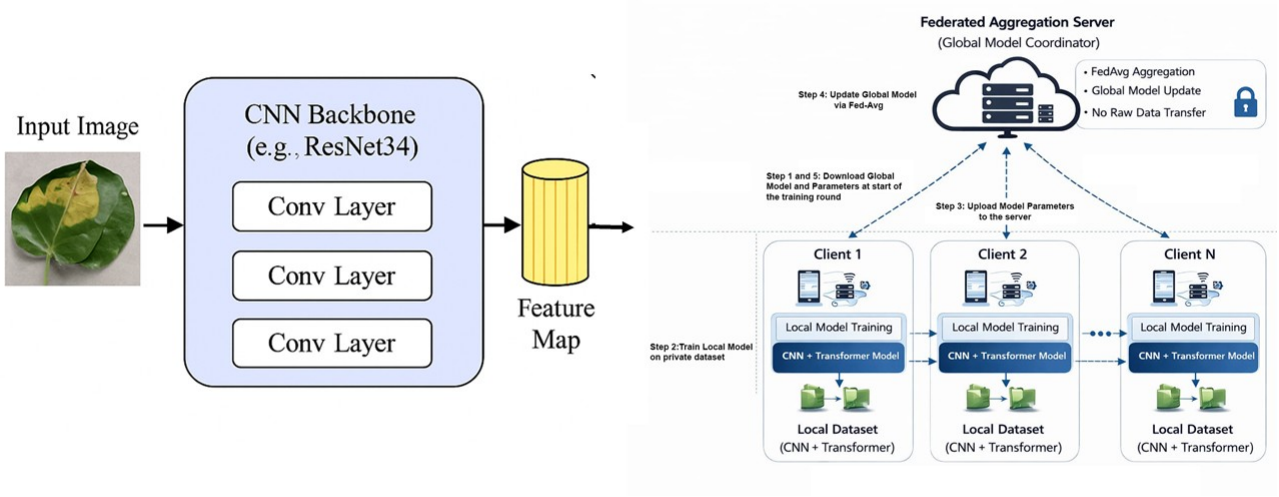
5.1.3.2 Architecture Components

The proposed model comprises three main architectural components: a CNN-based feature extractor (ResNet34), Transformer encoder layers to describe global dependencies, and a classification head which is shown in Figure 5.2. These parts are carefully combined to maintain spatial features while capturing long-range contextual linkages, which are critical for detecting subtle disease trends in leaf images.

Input and Preprocessing

Let $I \in \mathbb{R}^{H_e \times W_i \times 3}$ be an RGB input image of size $224 \times 224 \times 3$. It passes through a preprocessing pipeline that includes:

- Random horizontal flipping
- Color jitter (brightness, contrast, saturation)



39

Figure 5.1: Overview of the proposed model

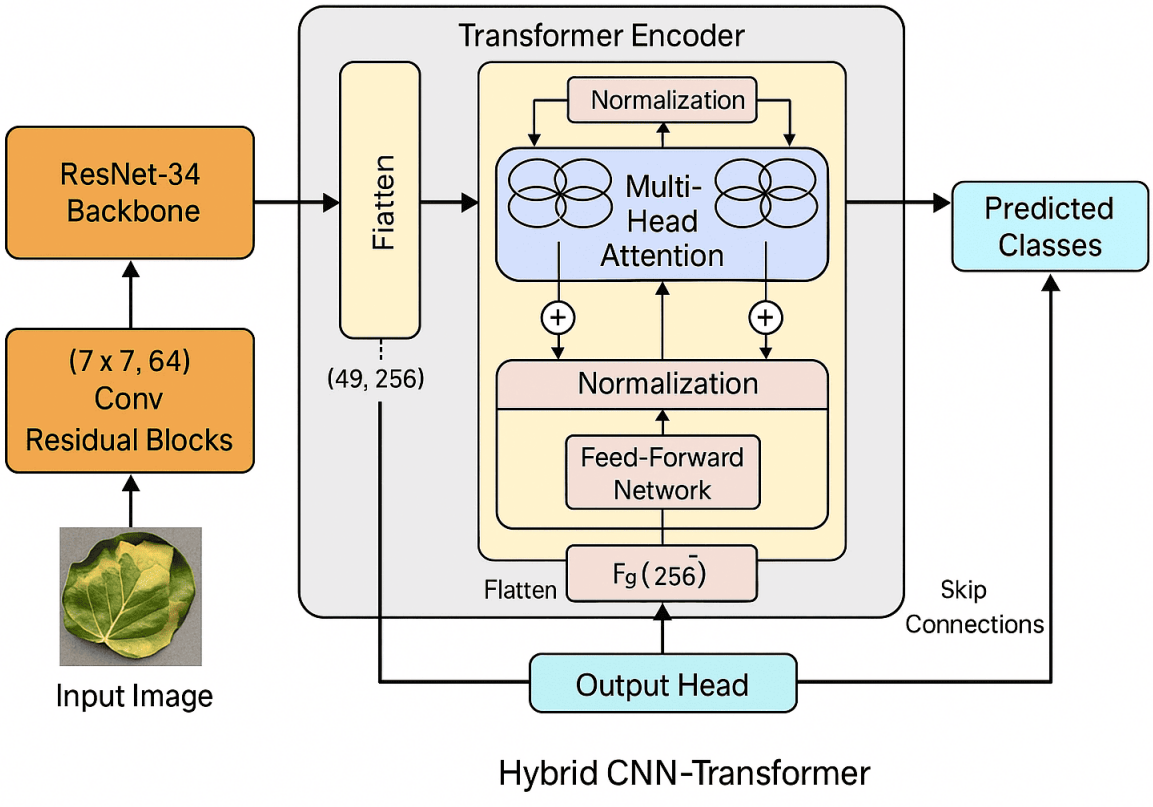


Figure 5.2: Architecture component of proposed model

- Gaussian blur (to simulate noise)

This improves generalization across diverse real-world agricultural conditions.

CNN Feature Extractor (ResNet34)

Hierarchical spatial features are extracted using a pretrained ResNet34 backbone. The final CNN block outputs a tensor:

$$F_{\text{CNN}} = f_{\text{CNN}}(I) \in \mathbb{R}^{B\times C\times \bar{H}_e\times \bar{W}_i} = \mathbb{R}^{B\times 512\times 7\times 7} \tag{5.1.1}$$

9 where B is batch size, C is number of channels, and $\bar{H}_e = \bar{W}_i = 7$ after down-sampling.

Flattening and Projection

The CNN feature map is flattened into tokens for Transformer input:

$$F_{\text{flat}} = \text{Reshape}(F_{\text{CNN}}) \in \mathbb{R}^{B\times 49\times 512} \tag{5.1.2}$$

These tokens are projected to embedding dimension $d = 256$:

$$Y = F_{\text{flat}} W_e + b_e \in \mathbb{R}^{B\times 49\times 256} \tag{5.1.3}$$

where $W_e \in \mathbb{R}^{512\times 256}$ is the learnable projection matrix.

Positional Encoding

To retain spatial order, learnable positional embeddings are added:

$$X_{\text{pos}} = Y + P \tag{5.1.4}$$

where $P \in \mathbb{R}^{49\times 256}$.

Transformer Encoder Block

112 Two Transformer encoder layers are used. Each layer contains:

- Multi-Head Self-Attention (MHSA)
- Feedforward Network (FFN)

- Residual connections and LayerNorm

Multi-Head Attention

Given $X \in \mathbb{R}^{B\times N\times d}$:

$$\text{Attention}(Q,K,V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{5.1.5}$$

where

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V,$$

and $W_Q, W_K, W_V \in \mathbb{R}^{d\times d_k}$.

Using four heads:

$$\text{MHSA}(X) = \text{Concat}(\text{head}_1, \ldots, \text{head}_4)W_O \tag{5.1.6}$$

Feedforward Network

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{5.1.7}$$

where $W_1 \in \mathbb{R}^{256\times 512}$ and $W_2 \in \mathbb{R}^{512\times 256}$.

Global Pooling and Classification Head

After Transformer processing, tokens $A \in \mathbb{R}^{B\times 49\times 256}$ are aggregated using mean pooling:

$$A_{\text{global}} = \frac{1}{49}\sum_{i=1}^{49} A_i \in \mathbb{R}^{B\times 256} \tag{5.1.8}$$

Final prediction:

$$\hat{y} = \text{Softmax}(\text{LayerNorm}(A_{\text{global}})W_c + b_c) \in \mathbb{R}^{B\times K} \tag{5.1.9}$$

where K is number of classes.

Loss and Optimization

Focal Loss is used for class imbalance:

$$L_{\text{focal}} = -\alpha_t (1 - \hat{y}_t)^{\gamma} \log(\hat{y}_t) \tag{5.1.10}$$

Optimization is performed using AdamW with weight decay regularization.

5.1.3.3 Federated Learning Integration

To simulate real-world scenarios in which data is scattered across different sources (e.g., farms, regional labs), we employ Federated Learning (FL) to train the model jointly without centralizing data. Our configuration consists of four clients, each with a local copy of the hybrid CNN-Transformer model, as shown in Figure 5.3.

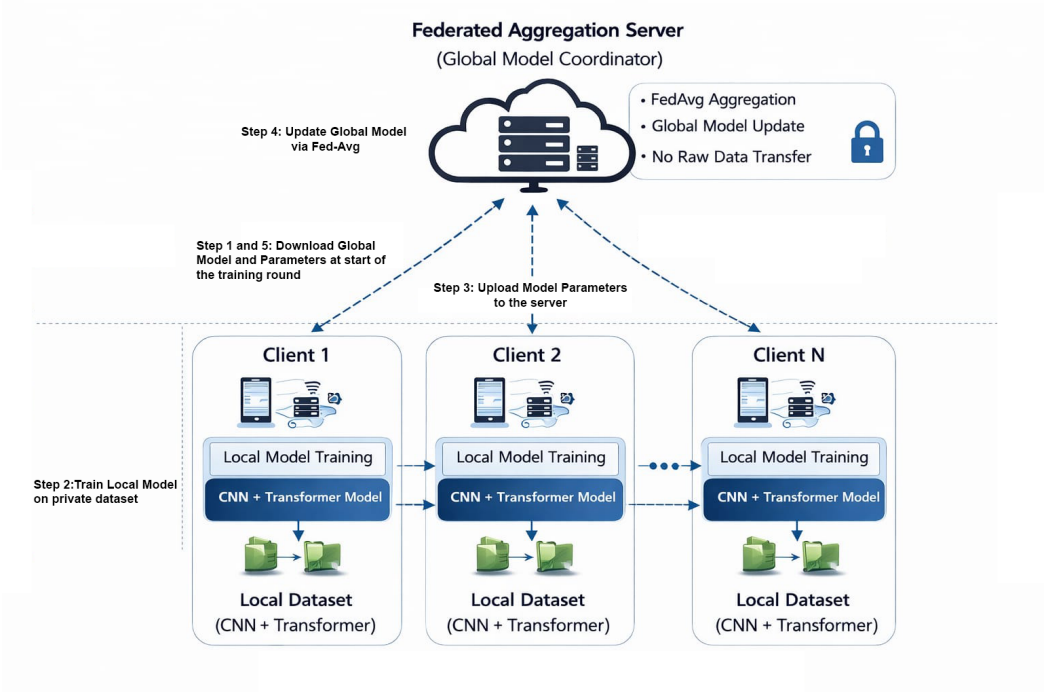


Figure 5.3: Part of the FL work

Local Training on Clients

Each client C_i trains the model with its local dataset D_i. The FedProx objective updates the local model parameters θ_i to maintain them close to the global model.

$$L^i = L^{(i)}_{\text{local}} + \frac{\mu}{2} \left\| \theta_i - \theta^G \right\|^2 \tag{5.1.11}$$

Where, θ^G is a global parameter from the server, μ : regularization term controlling local deviation, $L^{(i)}_{\text{local}}$: focal loss on local client.

Model Upload to Server

After local training for a set number of epochs, each client sends updated weights (not data) to a central server.

Global Aggregation

The server aggregates local updates from all N clients using the FedAvg algorithm:

$$\theta_{t+1}^{G} = \sum_{i=1}^{N} \frac{n_i}{n} \theta_i \tag{5.1.12}$$

Where, n_i is the number of samples on client i, and $n = \sum n_i$ is the total number of samples across all clients.

This produces a new global model θ_{t+1}^{G}, capturing collective knowledge.

Broadcasting the Global Model Back to Clients

Once aggregated, the server sends the new global model to all clients. Each client replaces its local model weights with θ_{t+1}^{G} to ensure synchronization before the following cycle.

This concludes one FL communication round, which is repeated until convergence.

5.1.4 Experimental Setup and Results

The following section describes the experimental dataset setup and briefly explains the pre-processing technique. The results are then created using a variety of parameters to determine the FL-TransCNN model's advantages and shortcomings.

5.1.5 Dataset Description

This research article uses the FL-TransCNN model. This model is trained using publicly available image datasets of Mango and rice leaves. The datasets and their structure are briefly discussed in Chapter 2, Section 2.4.

5.1.5.1 Optimizer Configuration

In the proposed FL-TransCNN model, we use a hybrid optimizer technique that includes:

1. AdamW (Adaptive Moment Estimation with Weight Decay) is used at each client for local training.
2. FedProx at the server level allows for global aggregation.

AdamW is selected because of its adaptable learning rate, decoupled weight decay regularisation, and enhanced convergence behaviour in training deep hybrid models. It helps to maintain the learning of both CNN-based spatial features and Transformer-based contextual dependencies.

FedProx is a server-side optimizer to solve client heterogeneity and instability, standard in non-IID (non-identically distributed) data sets. The objective function includes a proximal term to keep local client updates from deviating too far from the global model, ensuring stable and efficient convergence in federated environments.

5.1.6 Hyperparameter Table

Table 5.2 below summarizes the important hyperparameters used during training across all experiments:

Table 5.2: Hyperparameter Table

Hyperparameter	**Value**
Optimizer (Client)	AdamW
Optimizer (Server)	FedProx
Learning Rate	0.0003
Weight Decay	1e-4
Local Epochs per Round	5
Number of Communication Rounds	50
Batch Size	32
Image Input Size	224×224
Transformer Layers	2
Transformer Heads	4
Feedforward Dim (ViT)	512
CNN Backbone	ResNet-34
Dropout (Transformer)	0.1
FL Clients	4
Aggregation Strategy	Weighted Averaging (FedAvg)

This configuration enables training stability and efficient convergence, particularly across numerous clients with varied data distributions. AdamW inhibits overfitting, and FedProx ensures robustness in federated training environments. These choices help the model perform better on both the Mango and Rice Leaf Disease datasets.

5.1.7 Performance Evaluation Criteria

We employed performance indicators, such as f1-score, accuracy, recall, precision, and resource utilisation, to assess the usefulness of our plant disease identification approach. Precision demonstrates the model's capacity to avoid false positives, ensuring that healthy plants are not misidentified as ill. Recall assesses how successfully the model identifies diseased plants, reducing the likelihood of undiagnosed infections. The F1 score balances precision and recall, making it especially effective for imbalanced datasets [271] . Accuracy measures overall correctness, but the confusion matrix provides specific information on true positives, false positives, and false negatives, allowing us to understand the model's strengths and limitations thoroughly [272].

5.1.8 Result Analysis

This section assesses the effectiveness of the suggested FL-TransCNN model on two plant illness identification datasets: the Mango Leaf Disease Dataset and the Rice Leaf Disease Dataset. To tackle data imbalance, the model was trained across four federated clients using the FedProx + AdamW optimization approach, which included Focal Loss. Precision, Recall, F1-score, and Accuracy are reported for each class and client.

5.1.8.1 Client-Wise Classification Report (Mango & Rice Datasets)

The suggested hybrid CNN + Transformer model, developed inside the Federated Learning (FL) framework, was trained and evaluated separately on two datasets: mango leaf disease and rice leaf disease. Each dataset was divided among four clients to simulate distributed training. We measured per-class performance in both datasets using Precision, Recall, F1-score, and Accuracy. In Table 5.3, Client 1 performs well in terms of per-class metrics on the Mango dataset, with Class 1 achieving a Precision of 95.51%, a Recall of 92.73%, and an F1-score of 94.10%. Clients 2-4 showed similar stability, with F1 scores averaging more than 94%. These findings show that the hybrid model can accurately capture local and global disease patterns from mango leaf pictures, even in non-IID distributed environments.

In Table 5.4, each customer identifies six disease categories (Classes 1–6), and their performance is assessed using Precision, Recall, F1-Score, and Accuracy. The results demonstrate constant performance increases from Client 1 to Client 4, demonstrating that the FedProx + AdamW optimization technique efficiently improves training even in the presence of non-IID data. Client 4 obtains the highest precision and recall values across most classes, with

Table 5.3: Client-wise Classification Report on the Mango Dataset

Client	Class	Precision	Recall	F1-Score	Accuracy
Client 1	Anthracnose	95.51	92.73	94.10	98
	Bacterial Canker	93.90	93.08	93.49	97
	Cutting Weevil	93.80	93.52	93.66	97
	Die Back	93.20	95.65	94.41	96
	Gall Midge	92.85	91.00	91.91	96
	Healthy	96.35	97.10	96.72	98
	Powdery Mildew	94.00	94.89	94.44	97
	Sooty Mould	95.15	96.20	95.67	98
Client 2	Anthracnose	95.28	92.37	93.80	98
	Bacterial Canker	93.95	93.13	93.54	97
	Cutting Weevil	93.89	93.62	93.75	97
	Die Back	93.95	96.14	95.03	96
	Gall Midge	94.62	95.00	94.81	97
	Healthy	96.44	96.89	96.66	98
	Powdery Mildew	93.92	94.15	94.03	97
	Sooty Mould	95.05	96.50	95.77	98
Client 3	Anthracnose	95.91	93.37	94.62	98
	Bacterial Canker	94.95	94.26	94.61	98
	Cutting Weevil	95.23	95.01	95.12	97
	Die Back	94.06	96.21	95.12	97
	Gall Midge	95.05	95.90	95.70	98
	Healthy	96.00	97.20	96.59	98
	Powdery Mildew	95.33	94.45	94.89	97
	Sooty Mould	96.28	96.90	96.59	98
Client 4	Anthracnose	97.72	95.40	96.55	99
	Bacterial Canker	96.21	96.15	96.18	98
	Cutting Weevil	95.85	95.66	95.76	98
	Die Back	95.05	96.85	95.94	98
	Gall Midge	96.35	96.20	96.27	99
	Healthy	97.00	97.55	97.27	99
	Powdery Mildew	96.02	96.66	96.34	98
	Sooty Mould	97.41	97.90	97.65	99

F1-scores above 95% in most illness classes. This consistency among clients verifies the model's durability and generalizability within federated limitations.

Table 5.4: Client-wise Classification Report on the Rice Dataset

Client	Class	Precision	Recall	F1-Score	Accuracy
Client 1	bacterial_leaf_blight	91.75	90.20	90.97	92.6
	brown_spot	90.60	89.70	90.15	92.6
	healthy	92.85	90.45	91.64	92.6
	leaf_blast	93.20	91.80	92.49	92.6
	leaf_scald	90.50	89.90	90.19	92.6
	narrow_brown_spot	91.60	90.50	91.04	92.6
Client 2	bacterial_leaf_blight	92.10	91.00	91.54	93.1
	brown_spot	91.85	90.70	91.27	93.1
	healthy	93.45	91.60	92.52	93.1
	leaf_blast	94.00	92.10	93.04	93.1
	leaf_scald	91.40	90.80	91.09	93.1
	narrow_brown_spot	92.30	91.10	91.69	93.1
Client 3	bacterial_leaf_blight	93.80	92.30	93.04	93.6
	brown_spot	92.55	91.85	92.20	93.6
	healthy	94.60	92.90	93.74	93.6
	leaf_blast	94.10	93.15	93.62	93.6
	leaf_scald	93.10	92.30	92.69	93.6
	narrow_brown_spot	93.70	92.50	93.09	93.6
Client 4	bacterial_leaf_blight	94.30	92.95	93.62	94.2
	brown_spot	93.85	92.80	93.32	94.2
	healthy	95.20	93.75	94.47	94.2
	leaf_blast	95.65	94.50	95.07	94.2
	leaf_scald	94.80	93.70	94.24	94.2
	narrow_brown_spot	94.50	94.00	94.24	94.2

5.1.8.2 Average Metrics Across Clients

To assess overall performance at the client level, we calculated Macro F1, Weighted F1, and Micro Accuracy for both datasets.

In Table 5.5. Client 4 performed the best in the Mango dataset, with a Macro F1 of 96.11%, Weighted F1 of 96.09%, and Micro Accuracy of 96.08%, closely followed by the other clients. This high level of consistency demonstrates the model's good categorization performance, even with distributed learning.

In Table 5.6, for the rice dataset, the Macro Average F1-score, which provides equal weight to all classes regardless of sample size, consistently climbs from 91.78% (Client 1) to 93.63%

Table 5.5: Macro, Weighted, and Micro Averages per Client for Mango Leaf Dataset

Client	Macro F1 (%)	Weighted F1 (%)	Micro Accuracy (%)
Client 1	93.92	93.96	93.96
Client 2	94.04	94.18	94.17
Client 3	94.87	94.92	94.91
Client 4	96.11	96.09	96.08

(Client 4), indicating that the model maintains a balanced result across all six classes. The Weighted Average F1-score, which compensates for class imbalance, likewise rises with better client training (Client 4: 93.84%). Micro Accuracy, which measures global prediction correctness, follows a similar rising trend, reaching 94.2% for Client 4. These patterns demonstrate that the proposed federated architecture achieves training convergence and cross-client learning stability.

Table 5.6: Macro / Weighted / Micro Averages per Client for Rice Leaf Dataset

Client	Macro Avg (%)	Weighted Avg (%)	Micro Accuracy (%)
Client 1	91.78	91.92	92.60
Client 2	92.12	92.30	93.10
Client 3	93.00	93.09	93.60
Client 4	93.63	93.84	94.20

5.1.8.3 Global Aggregated Model Metrics

Following several FL rounds, client models were combined into a global model using the FedProx optimization approach. The aggregated model was then validated using centralized test data for each dataset.

Table 5.7: Global Aggregated Model Metrics for Rice and Mago Leaf Dataset

Dataset	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
Mango Leaf	94.65	94.32	94.48	97.40
Rice Leaf	93.23	92.88	93.05	94.10

In Table 5.7 on the Mango dataset, the aggregated model achieves a precision of 94.65%, a recall of 94.32%, an F1-score of 94.48%, and an accuracy of 97.40%. This demonstrates great generalization and how FedProx-based aggregation successfully incorporates knowledge from distributed networks.

For the rice dataset, the global model achieves 93.23% precision, 92.88% recall, and 93.05% F1-score, along with a high global classification accuracy of 94.10%. These findings demonstrate that, despite data decentralisation and class imbalance, the hybrid model effectively integrates learnt features from individual clients and generalises to new data. It also highlights the role of the Transformer encoder's global attention mechanism in enhancing the detection of subtle illness patterns across multiple classes.

5.1.8.4 Resource Utilization

Efficiency and scalability are crucial for federated learning (FL) systems designed for agricultural applications, where computational resources are often limited. To determine the feasibility of the proposed FL-TransCNN model, we measured GPU memory usage, CPU utilisation, and virtual memory consumption on the Mango and Rice Leaf Disease datasets, as shown in Table 5.8.

For both datasets, GPU memory consumption was consistent at 71.9 MB across all clients, indicating that the model's architecture is lightweight and optimized. This persistent use illustrates that, despite including CNN and Transformer modules, the model does not require high-end hardware and can run efficiently on edge devices.

Table 5.8: Resource Utilization on Mango and Rice Leaf Datasets

Dataset	Client	GPU Memory (MB)	CPU Usage	Virtual Memory (MB)
Mango Leaf	Client 1	71.9	54.1	54.2
Mango Leaf	Client 2	71.9	45.4	55.7
Mango Leaf	Client 3	71.9	35.2	53.9
Mango Leaf	Client 4	71.9	57.8	60.5
Rice Leaf	Client 1	71.9	54.1	54.2
Rice Leaf	Client 2	71.9	45.4	55.7
Rice Leaf	Client 3	71.9	35.2	53.9
Rice Leaf	Client 4	71.9	57.8	60.5

In terms of CPU consumption, Rice dataset clients reported usage ranging from 35.2% (Client 3) to 57.8% (Client 4), whereas Mango dataset customers revealed similar CPU trends, with all devices remaining below 60%. This suggests that the model's computing requirements are manageable and do not overwhelm conventional processors.

Virtual memory use was likewise comparable, ranging from 53.9 MB to 60.5 MB in the Rice sample and remaining within a similar range in Mango. These minimal memory needs reinforce the model's appropriateness for resource-constrained contexts.

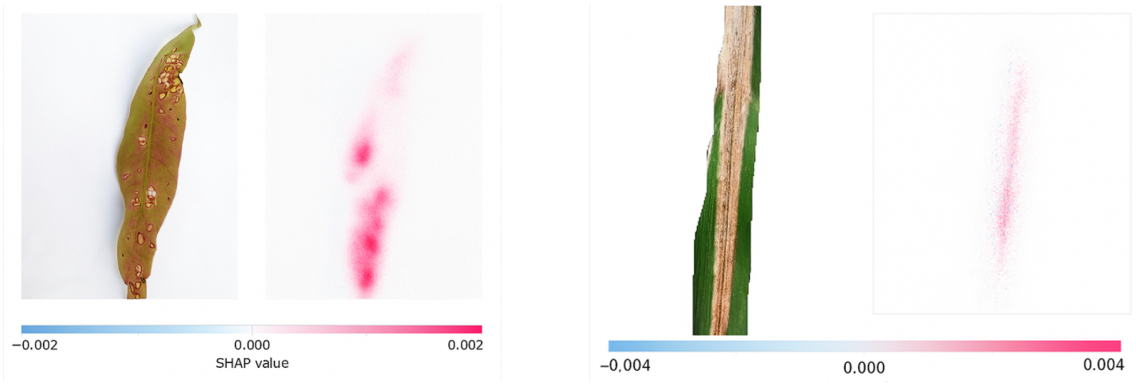
In conclusion, the model's resource profile across both datasets demonstrates that it is correct, interpretable, efficient and hardware-friendly. This makes it a formidable candidate for implementation in new agricultural systems, particularly in developing countries where access to computational infrastructure is limited.

5.1.8.5 Visualization Result

To improve the interpretability of the suggested Hybrid CNN + Transformer model within the Federated Learning framework, we used explainable AI (XAI) approaches. We used SHAP (Shapley Additive Explanations) and LIME (Local Interpretable Model-Agnostic Explanations) algorithms to show illness-affected areas on plant leaves in both the Mango and Rice Leaf Disease datasets. These strategies promote confidence and comprehension among users, particularly domain specialists such as agronomists and farmers, by indicating which parts of the leaf image significantly influenced the model's predictions.

SHAP Visualization

SHAP values are calculated using cooperative game theory to determine the contribution of each feature (in this example, picture region) to the model's prediction. The resulting SHAP heatmaps vividly show how different areas of the leaf affect the classification output.



a) Mango Leaf b) Rice Leaf

Figure 5.4: Visualization of SHAP analysis on (a) mango and (b) rice leaf dataset

As illustrated in the left panel of Figure 5.4, SHAP indicates the sick parts of the leaf image (in red hues) that contributed the most to the predicted class. Blue regions represent places that had a negative impact on the categorization score. The SHAP overlay demonstrates that the model successfully learns to focus on pathological textures and colorations found in infected leaf zones. This interpretation indicates that the model's choice is consistent with the visual signals from disease specialists.

LIME Visualization

LIME approximates the model locally with an interpretable linear model before perturbing the input image to examine changes in output. It discovers superpixels (pixel clusters) that impact the forecast most.



a) Mango Leaf a) Rice Leaf

Figure 5.5: Visualization of LIME analysis on (a) mango and (b) rice leaf dataset

As shown in the right panel of Figure 5.5, the highlighted yellow region corresponds to the super pixels that LIME determined were most relevant for the model's categorization. These areas correspond to disease lesions on the leaf, such as spots or discolorations, indicating that the model accurately identified these patterns as disease-specific classes. This localization lends weight to the premise that the hybrid CNN+Transformer framework is both practical and interpretable.

5.1.8.6 Ablation Study

To test the performance of each component in our proposed architecture, we do an ablation study with the following configurations:

1. Case1: With ResNet34
2. Case2: With ResNet34+Transformer Encoder
3. Case3: With ResNet34+Transformer Encoder + FL (our proposed model)

We evaluate the models on both the Mango Leaf and Rice Leaf datasets based on their accuracy, precision, recall, F1-score, and training time (in minutes).

The ablation study for the Mango Leaf Disease dataset clearly shows the progressive benefits of combining Transformer encoders and Federated Learning. In Table 5.9, the baseline model, which only uses ResNet34, obtains an accuracy of 90.45% and an F1-score of 88.92%,

Table 5.9: Score of Ablation Study for FL-TransCNN Model for Mango Leaf Dataset

Ablation Study Case	Case	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Training Time
With ResNet34	1	90.45	89.20	88.65	88.92	21 min 3 sec
With ResNet34 + Transformer Encoder	2	93.65	92.45	91.80	92.12	26 min 7 sec
With ResNet34 + Transformer Encoder + FL (Proposed)	3	95.90	94.80	94.25	94.52	30 min 1 sec

indicating a strong CNN-only benchmark. When a Transformer encoder is added, performance improves dramatically, with accuracy increasing to 93.65% and F1-score reaching 92.12%, indicating that self-attention processes improve feature learning, particularly for capturing contextual dependencies in leaf images.

The proposed model, which includes ResNet34, Transformer encoders, and Federated Learning, achieves the greatest results. This setup has a high accuracy of 97.40%, precision of 94.65%, recall of 94.32%, and an F1-score of 94.48%. The improvement indicates that federated learning allows for effective knowledge aggregation from distributed data sources while maintaining privacy. Although the training duration increases somewhat from 21.3 to 30.1 minutes, the trade-off is reasonable given the approximately 7% improvement in accuracy over the baseline. These findings support the efficacy of our suggested FL-TransCNN model in plant disease categorization.

A comparable outcome trend is visible in the Rice Leaf Disease dataset. In Table 5.10, the baseline ResNet34 model obtains 89.60% accuracy and 88.09% F1-score, suggesting adequate generalization considering the complexity of the 6-class problem. Adding the Transformer encoder increases accuracy to 92.85% and F1-score to 91.17%, supporting the notion that Transformers can learn more complex spatial correlations in leaf textures and disease patterns.

The proposed FL-TransCNN model achieves 94.10% accuracy and 93.05% F1-score, with enhanced precision and recall. This enhancement highlights the efficacy of the FL method, which improves model generalization by learning from different client data while avoid-

Table 5.10: Score of Ablation Study for FL-TransCNN Model for Rice Leaf Dataset

Ablation Study Case	Case	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Training Time
With ResNet34	1	89.60	88.30	87.90	88.09	19 min 8 sec
With ResNet34 + Transformer Encoder	2	92.85	91.60	90.75	91.17	24 min 2 sec
With ResNet34 + Transformer Encoder + FL (Proposed Model)	3	94.10	93.23	92.88	93.05	27 min 9 sec

ing centralization. The training duration increases moderately to 27.9 minutes, which is reasonable given the performance benefits.

Across both datasets, the ablation study emphasizes the additional significance of each module CNN, Transformer, and FL. While CNNs alone provide effective local feature extraction, the addition of Transformers adds global context awareness. When combined with Federated Learning, the model obtains exposure to broader, distributed data variability, which improves its generalization power while maintaining privacy. This layered improvement supports the design motivation for the proposed FL-TransCNN architecture.

5.1.8.7 Comparison with the State of the Art

The following section compares our proposed FL-TransCNN model with existing cutting-edge methods for multiclass classification of mango and rice leaf diseases. Table 5.11 compares the results of earlier published models ([273], [274], [275], [276]) terms of accuracy, precision, recall, F1-score, and model type.

Among the centralized models, WaveVisionNet and MDCN attained comparatively high accuracy values of 95.21% and 93.01%, respectively, with precision and recall of 91-94%. However, our suggested FL-TransCNN, a decentralized technique, beats all centralized models with accuracy (97.40%), precision (94.65%), recall (94.32%), and F1-score (94.48%).

This performance boost spans from +2.19% to +7.04% in accuracy, +0.71% to +3.31% in precision, +0.30% to +4.08% in recall, and +1.56% to +3.36% in F1-score when compared to existing approaches. The FL-TransCNN model's main advantage is its ability to accu-

Table 5.11: Comparison of the FL-TransCNN Model with State-of-The-Art Techniques on the Mango Leaf Dataset

References	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Model Type
Sharma A et al. [273]	CNN	90.36	–	–	–	Centralized
Fahim-Ul-Islam M et al. [274]	WaveVisionNet	95.21	92.25	94.02	91.12	Centralized
Swapno S et al. [275]	EfficientNetvL	94.99	–	–	–	Centralized
Chandrashekar C et al. [276]	MDCN	93.01	91.34	90.24	92.92	Centralized
Our Proposed Method	FL-TransCNN	97.40	94.65	94.32	94.48	Decentralized

rately extract spatial and temporal characteristics from leaf pictures while functioning in a distributed setting.

In addition to the previous comparison, Table 5.12 compares the proposed FL-TransCNN model to existing centralized and decentralized methods for leaf disease classification. Among the centralized techniques, the CNN model obtained 91.06% accuracy, 90.03% precision, 91.30% recall, and a 92.20% F1-score. While the YOLO V5- Efficient model demonstrated high precision (92.22%), it had lower accuracy (89.20%) and F1-score (87.08%), showing an imbalance between precision and overall effectiveness.

In contrast, the SHAO_LeNet + FL model, a decentralized framework, performed marginally better, with accuracy (91.30%), precision (91.30%), recall (92.20%), and F1-score (91.70%), demonstrating the potential of federated learning in improving model resilience while maintaining data privacy.

However, our suggested FL-TransCNN model exceeds all prior techniques, with the best accuracy (94.10%), precision (93.23%), recall (92.88%), and F1-score (93.05%). These increases vary from +1.90% to +4.90% in accuracy and from +0.93% to +5.97% in F1-score, demonstrating the model's capacity to extract relevant spatial-temporal characteristics while

Table 5.12: Comparison of the FL-TransCNN Model with State-of-The-Art Techniques on the Rice Leaf Dataset

References	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Model Type
Haikal A et al. [277]	CNN	91.06	90.03	91.30	92.20	Centralized
Gao W et al. [278]	YOLO V5-EFFICIENT	89.20	92.22	90.03	87.08	Centralized
Tripathy R et al. [269]	SHAO_LeNet + FL	91.30	91.30	92.20	91.70	Decentralized
Our Proposed Method	FL-TransCNN	94.10	93.23	92.88	93.05	Decentralized

functioning in a federated (decentralized) environment. This validates FL-TransCNN's success in real-world agricultural applications requiring data decentralization, privacy, and high performance.

Plant leaf diseases continue to pose a significant threat to global food security, particularly in developing countries where timely detection is challenging. This study developed a unique Federated Learning (FL)-based hybrid deep learning model for disease classification that combines ResNet34 with a Transformer Encoder while maintaining privacy and communication efficiency. The model was trained and tested on two different real-world datasets—Mango Leaf Disease and Rice Leaf Disease, and achieved high classification accuracy while maintaining data locality and privacy.

The proposed FL framework allows several decentralized clients to train models using non-IID local data, while global updates are aggregated via the FedAvg algorithm. This configuration substantially eliminates the need for data centralization, making it ideal for agricultural contexts with low bandwidth and strict privacy requirements. Experimental results show that using the Transformer encoder improves global context modeling, whereas Federated Learning preserves model performance and adds robustness to heterogeneous data distributions. Efficiency research confirmed the model's lightweight nature, making it suitable for deployment on edge devices such as mobile phones or farm-level IoT systems.

In the future, the current technique can be extended to cross-device federated learning situations involving a variety of clients, including drones, smartphones, and edge gateways. Furthermore, using self-supervised learning approaches could improve generalization across previously unknown situations, and using explainable AI (XAI) methods would improve model transparency and decision interpretability both of which are critical for establishing trust in real-world agriculture applications.

5.1.9 Summary

In this chapter, the attention-based federated learning system is proposed to achieve correct classification of mango and rice leaf diseases in a decentralized and privacy-aware agricultural scenario. The hybrid model proposed in the current study combines the CNN spatial feature extraction with Transformer encoders to obtain long-range contextual patterns and the FedAvg with the FedProx regulation to provide stable training in non-IID clients. Other strategies like focus loss and attention regularization are useful to improve performance in case of class imbalance. The effectiveness of the experiments is high (97.4% and 94.10% accuracy on Mango Leaf and Rice Leaf datasets, respectively), which proves the appropriateness of federated attention-based methods in the context of real-life smart agriculture systems.

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

Artificial intelligence has grown significantly in recent years, affecting almost every element of human life, including healthcare, communication networks, and agriculture. Artificial intelligence (AI) has revolutionized agriculture by automating disease detection, precision farming, and crop monitoring.

This thesis focuses on the AI-based plant leaf disease techniques for the agricultural system. Chapter one discusses plant leaf diseases, their economic impact on agriculture, and the use of automated detection methods. It emphasizes the need for AI-powered solutions to address issues such as delayed diagnosis, a lack of expert access in rural areas, and rising crop losses.

In Chapter Two, we provide a thorough research analysis on AI-based plant disease detection systems, datasets, tools, and implementation strategies. A thorough classification of AI-IoT plant disease detection systems is proposed, which includes supporting technologies such as Machine Learning, Deep Learning, IoT sensors, and edge computing. The report finds critical research gaps in scalability, data security, and latency, and proposes using blockchain, Federated Learning, and next-generation networks to overcome these challenges.

Chapter three introduces transfer learning-based hybrid frameworks for plant leaf disease classification that make use of pre-trained deep learning architectures, including VGG16, MobileNetV2, InceptionV3, Xception, and DenseNet. These models are used with Extreme Learning Machine classifiers to obtain high accuracy, short computation time, and better generalization. Experimental results from diverse crop datasets show classification accuracy reaching 99%, validating the efficacy of transfer learning in real-world agricultural contexts.

In Chapter Four, attention-based transfer learning models, such as AgriTL-ViT and attention-based ADResNet, are introduced to improve feature extraction and classification accuracy. The proposed models improve crop performance in tomatoes, maize, and rice while reducing computational costs by combining Vision Transformer architectures with attention-guided residual networks. The findings show that attention-based deep learning is practicable for automated disease diagnosis.

Chapter 5 presents a privacy-aware attention-based Federated Learning paradigm for decentralized plant leaf disease categorization. The approach offers consistent learning across heterogeneous clients while maintaining data privacy by combining CNN-based feature extraction with Transformer encoders and FedAvg with FedProx optimization. Experimental evaluations on the mango and rice leaf datasets yield high classification accuracy, demonstrating the utility of federated attention-based learning in smart agriculture systems. However, using centralized AI training presents privacy, security, and scalability problems, especially for sensitive agricultural and farmer data. Federated Learning (FL) enables collaborative model training without sharing raw data, potentially addressing these concerns.

Overall, this study demonstrates that combining AI, transfer learning, attention mechanisms, and Federated Learning can dramatically improve plant leaf disease identification systems while maintaining security, privacy, and scalability. The proposed frameworks offer dependable, efficient, and privacy-preserving solutions suitable for real world smart agriculture applications.

6.2 Future Scope

The future work may include:

1. **Real-World Field Deployment and Validation:** Future research initiatives include applying the presented models on a large-scale real-world scale to a variety of geographic areas, crop varieties, and meteorological conditions to test their strength in situations other than controlled datasets.

2. **Lightweight Edge-Optimized Models:** The development of ultra-lightweight deep learning models that can run on low-power IoT and mobile devices deployed by farmers will increase their utility in rural areas.

3. **Integration with Blockchain for Secure Data Sharing:** Federated Learning,

along with blockchain technology, can increase secure and tamper-proof agricultural data sharing across stakeholders.

4. **Early Disease Forecasting Using Time-Series Analysis:** The use of temporal modelling tools (e.g., LSTM/Transformer-based forecasting) can assist in predicting disease outbreaks.

9

5. **Multimodal Data Integration:** Integrating image data with environmental sensor data (temperature, humidity, soil moisture, pH, etc.) can enhance disease prediction accuracy and enable early-stage detection.

List of Publications

Journals

[1] **Meenakshi Srivastava**, Varsha Sisaudia, and Jasraj Meena, "A Review on Internet of Things and Artificial Intelligence based Plant Disease Detection: Taxonomy, Issues and Challenges," *Journal of Plant Diseases and Protection*, Springer, 132(6), 181. 2025. DOI:https://doi.org/10.1007/s41348-025-01184-0. **(SCIE, IF=2.2)**.

[2] **Meenakshi Srivastava** and Jasraj Meena, "Plant leaf disease detection and classification using modified transfer learning models," *Multimedia Tools and Applications*, Springer, vol. 83(13), pp.38411–38441, 2024., DOI: https://doi.org/10.1007/s11042-023-16929-y. **(SCIE, IF=3.0)**.

[3] **Meenakshi Srivastava**, Varsha Sisaudia, and Jasraj Meena, "AgriTL-ViT: A Vision Transformer model with attention techniques for classification of plant leaf disease," *Expert Systems with Applications*, Elsevier, vol. 294 p. 128793, 2025. DOI: https://doi.org/10.1016/j.eswa.2025.128793. **(SCIE, IF=7.5)**.

[4] **Meenakshi Srivastava**, Varsha Sisaudia, and Jasraj Meena, "A Hybrid VESTACK Model for Plant Disease Classification Using Deep Transfer Learning Approach," *Computers and Electrical Engineering*, Elsevier. *-Communicated*

[5] **Meenakshi Srivastava**, Varsha Sisaudia, and Jasraj Meena, "ADResNet with Attention: A Multi-Model Approach for Leaf Disease Detection and Pest Classification," *Signal, Image and Video Processing*, Springer. *-Communicated*

[6] **Meenakshi Srivastava**, Varsha Sisaudia, and Jasraj Meena, "Towards Decentralized Plant Disease Diagnosis: A Hybrid CNN-Transformer Model with Federated Learning," *Engineering Applications of Artificial Intelligence*, Elsevier. *-Communicated*

Conferences

[1] **Meenakshi Srivastava**, Varsha Sisaudia, and Jasraj Meena, "Tomato Leaf Disease Detection and Classification using MobileNetV2 and Extreme Learning Method:

A Hybrid Approach", *IEEE 10th International Conference on Soft Computing & Machine Intelligence(ISCMI 2023)*, Mexico, pp. 108-112.10.1109/ISCMI59957.2023.10458596

[2] **Meenakshi Srivastava**, Varsha Sisaudia, and Jasraj Meena, "Study and Analysis of Machine and Deep Learning based Existing Techniques for Plant Disease Classification", *5th International Conference on Intelligent Vision and Computing (ICIVC 2025)*, Dehradun, pp. 108-112. IEEE, 2023,10.1109/ISCMI59957.2023.10458596

Bibliography

[1] S. Savary, L. Willocquet, S. J. Pethybridge, P. Esker, N. McRoberts, and A. Nelson, "The global burden of crop loss due to diseases," *Phytopathology*, vol. 109, no. 4, pp. 635–645, 2019.

[2] J. Bruinsma, "The resource outlook to 2050," in *How to Feed the World in 2050*, pp. 24–26, FAO, 2009. Expert meeting on how to feed the world in 2050.

[3] A. Abbas, S. Jain, M. Gour, and S. Vankudothu, "Tomato plant disease detection using transfer learning with c-gan synthetic images," *Computers and Electronics in Agriculture*, vol. 187, p. 106279, 2021.

[4] D. S. Joseph, P. M. Pawar, and R. Pramanik, "Intelligent plant disease diagnosis using convolutional neural network: A review," *Multimedia Tools and Applications*, vol. 82, no. 14, pp. 21415–21481, 2023.

[5] A. P. Kale and S. P. Sonavane, "Iot based smart farming: Feature subset selection for optimized high-dimensional data using improved ga based approach for elm," *Computers and Electronics in Agriculture*, vol. 161, pp. 225–232, 2019.

[6] S. De Alwis, Z. Hou, Y. Zhang, M. H. Na, B. Ofoghi, and A. Sajjanhar, "A survey on smart farming data, applications and techniques," *Computers in Industry*, vol. 138, p. 103624, 2022.

[7] A. Nayak, S. Mohammad, R. Raja, B. Lal, G. Priyanka, P. Bhattacharyya, K. Anjani, R. Tripathi, M. Sangita, B. Panda, *et al.*, "Identification and man-agement of nutrient disorders and diseases in rice: A visual diag-nostic tool," *Central Rice Research Institute, Cuttack, India*, 2013.

[8] N. Leena and K. Saju, "Classification of macronutrient deficiencies in maize plants using optimized multi class support vector machines," *Engineering in Agriculture, Environment and Food*, vol. 12, no. 1, pp. 126–139, 2019.

[9] C. A. Reddy, S. Oraon, S. D. Bharti, A. K. Yadav, and S. Hazarika, "Advancing disease management in agriculture: A review of plant pathology techniques," *Plant Science Archives*, vol. 9, pp. 16–18, 2024.

[10] M. Shafay, T. Hassan, M. Owais, I. Hussain, S. G. Khawaja, L. Seneviratne, and N. Werghi, "Recent advances in plant disease detection: challenges and opportunities," *Plant Methods*, vol. 21, no. 1, p. 140, 2025.

[11] M. Putnam, "Evaluation of selected methods of plant disease diagnosis," *Crop protection*, vol. 14, no. 6, pp. 517–525, 1995.

[12] S. Sankaran, A. Mishra, R. Ehsani, and C. Davis, "A review of advanced techniques for detecting plant diseases," *Computers and electronics in agriculture*, vol. 72, no. 1, pp. 1–13, 2010.

[13] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "Identification of plant leaf diseases using a 9-layer deep convolutional neural network," *Computers and Electronics in Agriculture*, vol. 161, pp. 366–373, 2019.

[14] X. E. D. M. Pantazi and A. A. Tamouridou, "Automated leaf disease detection in different crop species through image feature analysis and one-class classifiers," *Computers and Electronics in Agriculture*, pp. 96–104, 2019.

[15] P. K. Sethy, N. K. Barpanda, A. K. Rath, and S. K. Behera, "Plant disease detection using deep learning and convolutional neural networks: A review," *Artificial Intelligence in Agriculture*, vol. 4, pp. 1–10, 2020.

[16] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning for smart agriculture: Concepts, tools, applications, and opportunities," *International Journal of Agricultural and Biological Engineering*, vol. 11, no. 4, pp. 1–14, 2018.

[17] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, p. 1419, 2016.

[18] M. Cordeiro *et al.*, "Towards smart farming: Fog-enabled intelligent irrigation system using deep neural networks," *Future Generation Computer Systems*, vol. 129, pp. 115–124, 2022.

[19] Y. Kaya and E. Gürsoy, "A review of deep learning architectures for plant disease detection," *Turkish Journal of Biology*, vol. 49, no. 5, pp. 459–497, 2025.

[20] T. D. Salka, M. B. Hanafi, S. M. S. A. A. Rahman, D. B. M. Zulperi, and Z. Omar, "Plant leaf disease detection and classification using convolution neural networks model: a review," *Artificial Intelligence Review*, vol. 58, no. 10, p. 322, 2025.

[21] T. Nyawose, R. C. Maswanganyi, and P. Khumalo, "A review on the detection of plant disease using machine learning and deep learning approaches," *Journal of Imaging*, vol. 11, no. 10, p. 326, 2025.

[22] S. M. Hassan, K. Amitab, M. Jasinski, Z. Leonowicz, E. Jasinska, T. Novak, and A. K. Maji, "A survey on different plant diseases detection using machine learning techniques," *Electronics*, vol. 11, no. 17, p. 2641, 2022.

[23] V. K. Shrivastava and M. K. Pradhan, "Rice plant disease classification using color features: a machine learning paradigm," *Journal of Plant Pathology*, vol. 103, no. 1, pp. 17–26, 2021.

[24] R. Saleem, J. H. Shah, M. Sharif, M. Yasmin, H.-S. Yong, and J. Cha, "Mango leaf disease recognition and classification using novel segmentation and vein pattern technique," *Applied Sciences*, vol. 11, no. 24, p. 11901, 2021.

[25] R. Dwivedi, T. Dutta, and Y.-C. Hu, "A leaf disease detection mechanism based on l1-norm minimization extreme learning machine," *IEEE geoscience and remote sensing letters*, vol. 19, pp. 1–5, 2021.

[26] J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanehkaran, "Using deep transfer learning for image-based plant disease identification," *Computers and electronics in agriculture*, vol. 173, p. 105393, 2020.

[27] A. Beikmohammadi, K. Faez, and A. Motallebi, "Swp-leafnet: A novel multistage approach for plant leaf identification based on deep cnn," *Expert Systems with Applications*, vol. 202, p. 117470, 2022.

[28] S. G. Paul, A. A. Biswas, A. Saha, M. S. Zulfiker, N. A. Ritu, I. Zahan, M. Rahman, and M. A. Islam, "A real-time application-based convolutional neural network approach for tomato leaf disease classification," *Array*, vol. 19, p. 100313, 2023.

[29] P. Kaur, S. Harnal, V. Gautam, M. P. Singh, and S. P. Singh, "A novel transfer deep learning method for detection and classification of plant leaf disease," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 9, pp. 12407–12424, 2023.

[30] I. Khan, S. S. Sohail, D. Ø. Madsen, and B. K. Khare, "Deep transfer learning for fine-grained maize leaf disease classification," *Journal of Agriculture and Food Research*, vol. 16, p. 101148, 2024.

[31] M. Nawaz, T. Nazir, A. Javed, S. T. Amin, F. Jeribi, and A. Tahir, "Coffeenet: A deep learning approach for coffee plant leaves diseases recognition," *Expert Systems with Applications*, vol. 237, p. 121481, 2024.

[32] M. Shoaib, A. Sadeghi-Niaraki, F. Ali, I. Hussain, and S. Khalid, "Leveraging deep learning for plant disease and pest detection: A comprehensive review and future directions," *Frontiers in Plant Science*, vol. 16, p. 1538163, 2025.

[33] E. Yilmaz, S. C. Bocekci, C. Safak, and K. Yildiz, "Advancements in smart agriculture: A systematic literature review on state-of-the-art plant disease detection with computer vision," *IET Computer Vision*, vol. 19, no. 1, p. e70004, 2025.

[34] Z. Iqbal, M. A. Khan, M. Sharif, J. H. Shah, M. H. ur Rehman, and K. Javed, "An automated detection and classification of citrus plant diseases using image processing techniques: A review," *Computers and Electronics in Agriculture*, vol. 153, pp. 12–32, 2018.

[35] S. Kaur, S. Pandey, and S. Goel, "Plants disease identification and classification through leaf images: A survey," *Archives of Computational Methods in Engineering*, vol. 26, no. 2, pp. 507–530, 2019.

[36] V. P. Kour and S. Arora, "Recent developments of the internet of things in agriculture: A survey," *IEEE Access*, vol. 8, pp. 129924–129957, 2020.

[37] R. I. Hasan, S. M. Yusuf, and L. Alzubaidi, "Review of the state of the art of deep learning for plant diseases: A broad analysis and discussion," *Plants*, vol. 9, no. 10, p. 1302, 2020.

[38] H. Orchi, M. Sadik, and M. Khaldoun, "On using artificial intelligence and the internet of things for crop disease detection: A contemporary survey," *Agriculture*, vol. 12, no. 1, p. 9, 2021.

[39] P. S. Thakur, P. Khanna, T. Sheorey, and A. Ojha, "Trends in vision-based machine learning techniques for plant disease identification: A systematic review," *Expert Systems with Applications*, vol. 208, p. 118117, 2022.

[40] S. R. Prasad and G. S. Thyagaraju, "Leaf analysis based early plant disease detection using internet of things, machine learning and deep learning: A comprehensive review," *Journal of Integrated Science and Technology*, vol. 12, no. 2, pp. 734–734, 2024.

[41] M. Dang, H. Wang, Y. Li, T.-H. Nguyen, L. Tightiz, N. Xuan-Mung, and T. N. Nguyen, "Computer vision for plant disease recognition: A comprehensive review," *The Botanical Review*, vol. 90, no. 3, pp. 251–311, 2024.

[42] W. Ding, M. Abdel-Basset, I. Alrashdi, and H. Hawash, "Next generation of computer vision for plant disease monitoring in precision agriculture: A contemporary survey, taxonomy, experiments, and future direction," *Information Sciences*, vol. 665, p. 120338, 2024.

[43] A. Ahmad, D. Saraswat, and A. El Gamal, "A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools," *Smart Agricultural Technology*, vol. 3, p. 100083, 2023.

[44] J. Zhang, Y. Huang, R. Pu, P. Gonzalez-Moreno, L. Yuan, K. Wu, and W. Huang, "Monitoring plant diseases and pests through remote sensing technology: A review," *Computers and Electronics in Agriculture*, vol. 165, p. 104943, 2019.

[45] M. Sophocleous and J. Georgiou, "Precision agriculture: Challenges in sensors and electronics for real-time soil and plant monitoring," in *2017 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pp. 1–4, IEEE, 2017.

[46] O. Debnath and H. N. Saha, "An iot-based intelligent farming using cnn for early disease detection in rice paddy," *Microprocessors and Microsystems*, vol. 94, p. 104631, 2022.

[47] G. Nagasubramanian, R. K. Sakthivel, R. Patan, M. Sankayya, M. Daneshmand, and A. H. Gandomi, "Ensemble classification and iot-based pattern recognition for crop disease monitoring system," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12847–12854, 2021.

[48] V. VaniB and C. D. Guruprakash, "A review on smart agricultural applications: Crop yield and plant disease prediction," in *Proceedings of the International Conference on Applied Artificial Intelligence and Computing (ICAAIC 2022)*, pp. 885–892, IEEE, 2022.

[49] R. Manavalan, "Automatic identification of diseases in grains crops through computational approaches: A review," *Computers and Electronics in Agriculture*, p. 105802, 2020.

[50] D. C. Tsouros, A. Triantafyllou, S. Bibi, and P. G. Sarigannidis, "Data acquisition and analysis methods in uav-based applications for precision agriculture," in *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, IEEE, 2019.

[51] D. Brunelli, A. Albanese, D. d'Acunto, and M. Nardello, "Energy neutral machine learning based iot device for pest detection in precision agriculture," *IEEE Internet of Things Magazine*, vol. 2, no. 4, pp. 10–13, 2020.

[52] A. Escamilla-García, G. M. Soto-Zarazúa, M. Toledano-Ayala, E. Rivas-Araiza, and A. Gastélum-Barrios, "Applications of artificial neural networks in greenhouse technology and overview for smart agriculture development," *Applied Sciences*, vol. 10, no. 11, p. 3835, 2020.

[53] R. Akhter and S. A. Sofi, "Precision agriculture using iot data analytics and machine learning," *Journal of King Saud University – Computer and Information Sciences*, vol. 34, no. 8, pp. 5602–5618, 2022.

[54] K. N. Bhanu, H. J. Jasmine, and H. S. Mahadevaswamy, "Machine learning implementation in iot based intelligent system for agriculture," in *2020 International Conference for Emerging Technology (INCET)*, IEEE, 2020.

[55] L. Nóbrega, P. Gonçalves, P. Pedreiras, and J. Pereira, "An iot-based solution for intelligent farming," *Sensors*, vol. 19, no. 3, p. 603, 2019.

[56] F. K. Syed, A. Paul, A. Kumar, and J. Cherukuri, "Low-cost iot+ ml design for smart farming with multiple applications," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–5, IEEE, 2019.

[57] N. Kundu *et al.*, "Iot and interpretable machine learning based framework for disease prediction in pearl millet," *Sensors*, vol. 21, no. 16, p. 5386, 2021.

[58] A. Xenakis, V. C. Gerogiannis, A. Xenakis, G. Papastergiou, and G. Stamoulis, "Applying a convolutional neural network in an iot robotic system for plant disease diagnosis," 2020. Available online.

[59] A. A. Ahmed and G. H. Reddy, "A mobile-based system for detecting plant leaf diseases using deep learning," *AgriEngineering*, vol. 3, no. 3, pp. 478–493, 2021.

[60] P. K. Kashyap, S. Kumar, A. Jaiswal, M. Prasad, and A. H. Gandomi, "Towards precision agriculture: Iot-enabled intelligent irrigation systems using deep learning neural network," *IEEE Sensors Journal*, vol. 21, no. 16, pp. 17479–17491, 2021.

[61] R. A. Khalil, N. Saeed, M. Masood, Y. M. Fard, M. S. Alouini, and T. Y. Al-Naffouri, "Deep learning in the industrial internet of things: Potentials, challenges, and emerging applications," *IEEE Internet of Things Journal*, 2021.

[62] S. Qazi, B. A. Khawaja, and Q. U. Farooq, "Iot-equipped and ai-enabled next generation smart agriculture: A critical review, current challenges and future trends," *Ieee Access*, vol. 10, pp. 21219–21235, 2022.

[63] A. Triantafyllou, P. Sarigiannidis, and S. Bibi, "Precision agriculture: A remote sensing monitoring system architecture," *Information*, vol. 10, no. 11, p. 348, 2019.

[64] R. Morais, N. Silva, J. Mendes, T. Adão, L. Pádua, J. A. López-Riquelme, N. Pavón-Pulido, J. J. Sousa, and E. Peres, "mysense: A comprehensive data management environment to improve precision agriculture practices," *Computers and Electronics in Agriculture*, vol. 162, pp. 882–894, 2019.

[65] M. Aggarwal, V. Khullar, N. Goyal, A. Alammari, M. A. Albahar, and A. Singh, "Lightweight federated learning for rice leaf disease classification using non independent and identically distributed images," *Sustainability*, vol. 15, no. 16, p. 12149, 2023.

[66] A. N. Harun, N. Mohamed, R. Ahmad, A. R. A. Rahim, and N. N. Ani, "Improved internet of things (iot) monitoring system for growth optimization of brassica chinensis," *Computers and Electronics in Agriculture*, vol. 164, p. 104836, 2019.

[67] C. Hirsch, E. Bartocci, and R. Grosu, "Capacitive soil moisture sensor node for iot in agriculture and home," in *2019 IEEE 23rd International Symposium on Consumer Technologies (ISCT)*, pp. 97–102, IEEE, 2019.

[68] D. Popescu, F. Stoican, G. Stamatescu, L. Ichim, and C. Dragana, "Advanced uav–wsn system for intelligent monitoring in precision agriculture," *Sensors*, vol. 20, no. 3, p. 817, 2020.

[69] C. Cambra Baseca, S. Sendra, J. Lloret, and J. Tomas, "A smart decision system for digital farming," *Agronomy*, vol. 9, no. 5, p. 216, 2019.

[70] A. Muminov, D. Na, C. Lee, H. K. Kang, and H. S. Jeon, "Retracted: Modern virtual fencing application: Monitoring and controlling behavior of goats using gps collars and warning signals," *Sensors*, vol. 19, no. 7, p. 1598, 2019.

[71] I. Potamitis, I. Rigakis, N.-A. Tatlas, and S. Potirakis, "In-vivo vibroacoustic surveillance of trees in the context of the iot," *Sensors*, vol. 19, no. 6, p. 1366, 2019.

[72] S. Kim, M. Lee, and C. Shin, "Iot-based strawberry disease prediction system for smart farming," *Sensors*, vol. 18, no. 11, p. 4051, 2018.

[73] Y. Zhao, L. Liu, C. Xie, R. Wang, F. Wang, Y. Bu, and S. Zhang, "An effective automatic system deployed in agricultural internet of things using multi-context fusion network towards crop disease recognition in the wild," *Applied Soft Computing*, vol. 89, p. 106128, 2020.

[74] D.-H. Park and J.-W. Park, "Wireless sensor network-based greenhouse environment monitoring and automatic control system for dew condensation prevention," *Sensors*, vol. 11, no. 4, pp. 3640–3651, 2011.

[75] K. A. Fatih, "Identification of acidosis disease in cattle using iot," in *International Conference on Computer Science and Engineering*, 2019.

[76] R. Ding, H. Jin, D. Xiang, X. Wang, Y. Zhang, D. Shen, and C. Zhou, "Soil moisture sensing with uav-mounted ir-uwb radar and deep learning," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 7, no. 1, pp. 1–25, 2023.

[77] S. Sundaresan, S. D. Johnson, V. M. Bharathy, P. M. P. Kumar, and M. Surendar, "Machine learning and iot-based smart farming for enhancing the crop yield," in *Journal of Physics: Conference Series*, vol. 2466, p. 012028, IOP Publishing, 2023.

[78] R. K. Jain, "Experimental performance of smart iot-enabled drip irrigation system using and controlled through web-based applications," *Smart Agricultural Technology*, vol. 4, p. 100215, 2023.

[79] G. Patrizi, A. Bartolini, L. Ciani, V. Gallo, P. Sommella, and M. Carratù, "A virtual soil moisture sensor for smart farming using deep learning," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–11, 2022.

[80] D. Chakraborty, H. Başağaoğlu, S. Alian, A. Mirchi, D. N. Moriasi, P. J. Starks, and J. A. Verser, "Multiscale extrapolative learning algorithm for predictive soil moisture modeling & applications," *Expert Systems with Applications*, vol. 213, p. 119056, 2023.

[81] M. Muchlasin, M. Hasbi, and S. Siswanti, "Decision tree method for automation of plant sprinklers and monitoring based on soil moisture," *Jurnal Nasional Pendidikan Teknik Informatika: JANAPATI*, vol. 12, no. 1, pp. 25–32, 2023.

[82] S.-H. Park, B.-Y. Lee, M.-J. Kim, W. Sang, M. C. Seo, J.-K. Baek, J. E. Yang, and C. Mo, "Development of a soil moisture prediction model based on recurrent neural network long short-term memory (rnn-lstm) in soybean cultivation," *Sensors*, vol. 23, no. 4, p. 1976, 2023.

[83] S.-H. Yu, Y. Kang, and C.-G. Lee, "Comparison of the spray effects of air induction nozzles and flat fan nozzles installed on agricultural drones," *Applied Sciences*, vol. 13, no. 20, p. 11552, 2023.

[84] S. A. Shah, G. M. Lakho, H. A. Keerio, M. N. Sattar, G. Hussain, M. Mehdi, R. B. Vistro, E. A. Mahmoud, and H. O. Elansary, "Application of drone surveillance for advance agriculture monitoring by android application using convolution neural network," *Agronomy*, vol. 13, no. 7, p. 1764, 2023.

[85] P. Kumar and N. Kumar, "Drone-based apple detection: Finding the depth of apples using yolov7 architecture with multi-head attention mechanism," *Smart Agricultural Technology*, vol. 5, p. 100311, 2023.

[86] M. K. Rajagopal and B. M. MS, "Artificial intelligence based drone for early disease detection and precision pesticide management in cashew farming," *arXiv preprint arXiv:2303.08556*, 2023.

[87] T. Hiraguri, H. Shimizu, T. Kimura, T. Matsuda, K. Maruta, Y. Takemura, T. Ohya, and T. Takanashi, "Autonomous drone-based pollination system using ai classifier to replace bees for greenhouse tomato cultivation," *IEEE Access*, vol. 11, pp. 99352–99364, 2023.

[88] L. N. Thalluri, S. D. Adapa, A. V. N. Sarma, S. N. Venkat, *et al.*, "Drone technology enabled leaf disease detection and analysis system for agriculture applications," in *2021 2nd International Conference on Smart Electronics and Communication (ICOSEC)*, pp. 1079–1085, IEEE, 2021.

[89] M. H. M. Ghazali and W. Rahiman, "Vibration-based fault detection in drone using artificial intelligence," *IEEE Sensors Journal*, vol. 22, no. 9, pp. 8439–8448, 2022.

[90] C.-J. Chen, Y.-Y. Huang, Y.-S. Li, Y.-C. Chen, C.-Y. Chang, and Y.-M. Huang, "Identification of fruit tree pests with deep learning on embedded drone to achieve accurate pesticide spraying," *Ieee Access*, vol. 9, pp. 21986–21997, 2021.

[91] G. B. Loganathan, Q. S. Mahdi, I. H. Saleh, and M. M. Othman, "Agribot: Energetic agricultural field monitoring robot based on iot enabled artificial intelligence logic," in *International Conference on Emerging Technology Trends in Internet of Things and Computing*, pp. 16–30, Springer, 2021.

[92] J. Yang, J. Ni, Y. Li, J. Wen, and D. Chen, "The intelligent path planning system of agricultural robot via reinforcement learning," *Sensors*, vol. 22, no. 12, p. 4316, 2022.

[93] T. M. Shah, D. P. B. Nasika, and R. Otterpohl, "Plant and weed identifier robot as an agroecological tool using artificial neural networks for image identification," *Agriculture*, vol. 11, no. 3, p. 222, 2021.

[94] M. Skoczeń, M. Ochman, K. Spyra, M. Nikodem, D. Krata, M. Panek, and A. Pawłowski, "Obstacle detection system for agricultural mobile robot application using rgb-d cameras," *Sensors*, vol. 21, no. 16, p. 5292, 2021.

[95] M. S. Alam, M. Alam, M. Tufail, M. U. Khan, A. Güneş, B. Salah, F. E. Nasir, W. Saleem, and M. T. Khan, "Tobset: A new tobacco crop and weeds image dataset and its utilization for vision-based spraying by agricultural robots," *Applied Sciences*, vol. 12, no. 3, p. 1308, 2022.

[96] S. K. Apat, J. Mishra, K. S. Raju, and N. Padhy, "The robust and efficient machine learning model for smart farming decisions and allied intelligent agriculture decisions," *Journal of Integrated Science and Technology*, vol. 10, no. 2, pp. 139–155, 2022.

[97] N. R. N. Zadeh, H. Hashimoto, D. Raper, E. Tanuyan, and M. S. Bruca, "Autonomous smart farming system using flann-based feature matcher with robotic arm," in *AIP Conference Proceedings*, vol. 2502, p. 040004, AIP Publishing LLC, 2022.

[98] S. S. Chouhan, U. P. Singh, A. Kaul, and S. Jain, "A data repository of leaf images: Practice towards plant conservation with plant pathology," in *2019 4th International Conference on Information Systems and Computer Networks (ISCON)*, pp. 700–707, IEEE, 2019.

[99] D. Hughes, M. Salathé, *et al.*, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," *arXiv preprint arXiv:1511.08060*, 2015.

[100] J. AP and G. Gopal, "Data for: Identification of plant leaf diseases using a 9-layer deep convolutional neural network," *Mendeley Data*, vol. 1, p. 2019, 2019.

[101] Kaggle, "Apple leaves disease prediction." `https://www.kaggle.com/code/shamikrana/apple-leaves-disease-prediction/input`, 2020. [Accessed: 10-Feb-2026].

[102] Kaggle, "Plantvillage tomato leaf dataset," 2024. Accessed: Mar. 20, 2024.

[103] Kaggle, "Corn or maize leaf disease dataset," 2024. Accessed: Apr. 16, 2024.

[104] Kaggle, "Rice leaf diseases dataset," 2024. Accessed: Apr. 16, 2024.

[105] P. K. Mensah, V. Akoto-Adjepong, K. Adu, M. A. Ayidzoe, E. A. Bediako, O. Nyarko-Boateng, S. Boateng, E. F. Donkor, F. U. Bawah, N. S. Awarayi, *et al.*, "Ccmt: Dataset for crop pest and disease detection," *Data in Brief*, vol. 49, p. 109306, 2023.

[106] S. I. Ahmed, M. Ibrahim, M. Nadim, M. M. Rahman, M. M. Shejunti, T. Jabid, and M. S. Ali, "Mangoleafbd: A comprehensive image dataset to classify diseased and healthy mango leaves," *Data in Brief*, vol. 47, p. 108941, 2023.

[107] J. P. Shah, "Rice leaf disease dataset." Online, n.d. `https://archive.ics.uci.edu/ml/datasets/Rice%2BLeaf%2BDiseases`.

[108] T. Wiesner-Hanks, E. L. Stewart, N. Kaczmar, C. DeChant, H. Wu, R. J. Nelson, H. Lipson, and M. A. Gore, "Image set for deep learning: field images of maize annotated with disease symptoms," *BMC research notes*, vol. 11, no. 1, p. 440, 2018.

[109] H. T. Rauf, B. A. Saleem, M. I. U. Lali, M. A. Khan, M. Sharif, and S. A. C. Bukhari, "A citrus fruits and leaves dataset for detection and classification of citrus diseases through machine learning," *Data in brief*, vol. 26, p. 104340, 2019.

[110] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, "Plantdoc: A dataset for visual plant disease detection," in *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, pp. 249–253, ACM, 2020.

[111] M.-L. Huang and Y.-H. Chang, "Dataset of tomato leaves," 2020.

[112] H. Getachew, "Wheat leaf dataset," 2021.

[113] A. Ayaz, "Cotton crop plant leaves dataset for disease classification," 2022.

[114] g. salma, "Bell pepper disease detection." Kaggle Dataset, 2023. Accessed: Feb. 07, 2023.

[115] R. Mukherjee, A. Ghosh, C. Chakraborty, J. N. De, and D. P. Mishra, "Rice leaf disease identification and classification using machine learning techniques: A comprehensive review," *Engineering Applications of Artificial Intelligence*, vol. 139, p. 109639, 2025.

[116] M. Srivastava and J. Meena, "Plant leaf disease detection and classification using modified transfer learning models," *Multimedia Tools and Applications*, vol. 83, no. 13, pp. 38411–38441, 2024.

[117] S. M. Javidan, A. Banakar, K. A. Vakilian, and Y. Ampatzidis, "Diagnosis of grape leaf diseases using automatic k-means clustering and machine learning," *Smart agricultural technology*, vol. 3, p. 100081, 2023.

[118] P. Sharma, P. Hans, and S. C. Gupta, "Classification of plant leaf diseases using machine learning and image preprocessing techniques," in *2020 10th international conference on cloud computing, data science & engineering (Confluence)*, pp. 480–484, IEEE, 2020.

[119] N. Krishnamoorthy, L. N. Prasad, C. P. Kumar, B. Subedi, H. B. Abraha, and V. Sathishkumar, "Rice leaf diseases prediction using deep neural networks with transfer learning," *Environmental Research*, vol. 198, p. 111275, 2021.

[120] A. S. Paymode and V. B. Malode, "Transfer learning for multi-crop leaf disease image classification using convolutional neural network vgg," *Artificial Intelligence in Agriculture*, vol. 6, pp. 23–33, 2022.

[121] R. Mahum, H. Munir, Z.-U.-N. Mughal, M. Awais, F. Sher Khan, M. Saqlain, S. Mahamad, and I. Tlili, "A novel framework for potato leaf disease detection using an efficient deep learning model," *Human and Ecological Risk Assessment: An International Journal*, vol. 29, no. 2, pp. 303–326, 2023.

[122] G. Singh and K. K. Yogi, "Performance evaluation of plant leaf disease detection using deep learning models," *Archives of Phytopathology and Plant Protection*, vol. 56, no. 3, pp. 209–233, 2023.

[123] A. Kumar and V. K. Patel, "Classification and identification of disease in potato leaf using hierarchical based deep learning convolutional neural network," *Multimedia Tools and Applications*, vol. 82, no. 20, pp. 31101–31127, 2023.

[124] R. Sujatha, J. M. Chatterjee, N. Jhanjhi, and S. N. Brohi, "Performance of deep learning vs machine learning in plant leaf disease detection," *Microprocessors and Microsystems*, vol. 80, p. 103615, 2021.

[125] F. Samie, L. Bauer, and J. Henkel, "From cloud down to things: An overview of machine learning in internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4921–4934, 2019.

[126] B. B. Sinha and R. Dhanalakshmi, "Recent advancements and challenges of internet of things in smart agriculture: A survey," *Future Generation Computer Systems*, vol. 126, pp. 169–184, 2022.

[127] G. Garg, S. Gupta, P. Mishra, A. Vidyarthi, A. Singh, and A. Ali, "Cropcare: an intelligent real-time sustainable iot system for crop disease detection using mobile vision," *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 2840–2851, 2021.

[128] M. Ouhami, A. Hafiane, Y. Es-Saady, M. El Hajji, and R. Canals, "Computer vision, iot and data fusion for crop disease detection using machine learning: A survey and ongoing research," *Remote Sensing*, vol. 13, no. 13, p. 2486, 2021.

[129] T. N. Pham, L. Van Tran, and S. V. T. Dao, "Early disease classification of mango leaves using feed-forward neural network and hybrid metaheuristic feature selection," *IEEE access*, vol. 8, pp. 189960–189973, 2020.

[130] J. D. Stevens and T. Shaikh, "Microcea: developing a personal urban smart farming device," in *2018 2nd International Conference on Smart Grid and Smart Cities (ICSGSC)*, pp. 49–56, IEEE, 2018.

[131] G. Nagasubramanian, R. K. Sakthivel, R. Patan, M. Sankayya, M. Daneshmand, and A. H. Gandomi, "Ensemble classification and iot-based pattern recognition for crop disease monitoring system," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12847–12854, 2021.

[132] V. K. Shrivastava and M. K. Pradhan, "Rice plant disease classification using color features: a machine learning paradigm," *Journal of Plant Pathology*, vol. 103, no. 1, pp. 17–26, 2021.

[133] M. Kumar, A. Kumar, and V. S. Palaparthy, "Soil sensors-based prediction system for plant diseases using exploratory data analysis and machine learning," *IEEE Sensors Journal*, vol. 21, no. 16, pp. 17455–17468, 2020.

[134] J. Wei Tan, S.-W. Chang, S. Abdul-Kareem, H. J. Yap, and K.-T. Yong, "Deep learning for plant species classification using leaf vein morphometric," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 17, no. 1, pp. 82–90, 2018.

[135] N. Kundu, G. Rani, V. S. Dhaka, K. Gupta, S. C. Nayak, S. Verma, M. F. Ijaz, and M. Woźniak, "Iot and interpretable machine learning based framework for disease prediction in pearl millet," *Sensors*, vol. 21, no. 16, p. 5386, 2021.

[136] V. Gonzalez-Huitron, J. A. León-Borges, A. Rodriguez-Mata, L. E. Amabilis-Sosa, B. Ramírez-Pereda, and H. Rodriguez, "Disease detection in tomato leaves via cnn with lightweight architectures implemented in raspberry pi 4," *Computers and Electronics in Agriculture*, vol. 181, p. 105951, 2021.

[137] M. Mishra, P. Choudhury, and B. Pati, "Modified ride-nn optimizer for the iot based plant disease detection," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, pp. 691–703, 2021.

[138] K. Zhang, L. Zhang, and Q. Wu, "Identification of cherry leaf disease infected by podosphaera pannosa via convolutional neural network," *International Journal of Agricultural and Environmental Information Systems (IJAEIS)*, vol. 10, no. 2, pp. 98–110, 2019.

[139] C. Shorten, T. M. Khoshgoftaar, and B. Furht, "Text data augmentation for deep learning," *Journal of big Data*, vol. 8, no. 1, p. 101, 2021.

[140] S. Roy, R. Ray, S. R. Dash, and M. K. Giri, "Plant disease detection using machine learning tools with an overview on dimensionality reduction," *Data analytics in bioinformatics: a machine learning perspective*, pp. 109–144, 2021.

[141] R. Karthickmanoj, T. Sasilatha, and J. Padmapriya, "Automated machine learning based plant stress detection system," *Materials Today: Proceedings*, vol. 47, pp. 1887–1891, 2021.

[142] M. Ahmad, M. Abdullah, H. Moon, and D. Han, "Plant disease detection in imbalanced datasets using efficient convolutional neural networks with stepwise transfer learning," *IEEE Access*, vol. 9, pp. 140565–140580, 2021.

[143] A. Khattak, M. U. Asghar, U. Batool, M. Z. Asghar, H. Ullah, M. Al-Rakhami, and A. Gumaei, "Automatic detection of citrus fruit and leaves diseases using deep neural network model," *IEEE access*, vol. 9, pp. 112942–112954, 2021.

[144] V. Udutalapally, S. P. Mohanty, V. Pallagani, and V. Khandelwal, "scrop: A novel device for sustainable automatic disease prediction, crop selection, and irrigation in internet-of-agro-things for smart agriculture," *IEEE Sensors Journal*, vol. 21, no. 16, pp. 17525–17538, 2020.

[145] S. Nalini, N. Krishnaraj, T. Jayasankar, K. Vinothkumar, A. S. F. Britto, K. Subramaniam, and C. Bharatiraja, "Paddy leaf disease detection using an optimized deep neural network," *Computers, Materials & Continua*, vol. 68, no. 1, pp. 1117–1128, 2021.

[146] A. Sharma, M. Georgi, M. Tregubenko, A. Tselykh, and A. Tselykh, "Enabling smart agriculture by implementing artificial intelligence and embedded sensing," *Computers & Industrial Engineering*, vol. 165, p. 107936, 2022.

[147] U. P. Singh, S. S. Chouhan, S. Jain, and S. Jain, "Multilayer convolution neural network for the classification of mango leaves infected by anthracnose disease," *IEEE access*, vol. 7, pp. 43721–43729, 2019.

[148] C. Bisogni, A. Castiglione, S. Hossain, F. Narducci, and S. Umer, "Impact of deep learning approaches on facial expression recognition in healthcare industries," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5619–5627, 2022.

[149] D. S. Joseph, P. M. Pawar, and R. Pramanik, "Intelligent plant disease diagnosis using convolutional neural network: a review," *Multimedia Tools and Applications*, vol. 82, no. 14, pp. 21415–21481, 2023.

[150] M. Shaha and M. Pawar, "Transfer learning for image classification," in *2018 second international conference on electronics, communication and aerospace technology (ICECA)*, pp. 656–660, IEEE, 2018.

[151] S. Liu, P. Chen, and M. Woźniak, "Image enhancement-based detection with small infrared targets," *Remote Sensing*, vol. 14, no. 13, p. 3232, 2022.

[152] P. Enkvetchakul and O. Surinta, "Effective data augmentation and training techniques for improving deep learning in plant leaf disease recognition," *Applied Science and Engineering Progress*, vol. 15, no. 3, pp. 3810–3810, 2022.

[153] X. Chao, X. Hu, J. Feng, Z. Zhang, M. Wang, and D. He, "Construction of apple leaf diseases identification networks based on xception fused by se module," *Applied Sciences*, vol. 11, no. 10, p. 4614, 2021.

[154] I. Ahmed, M. Anisetti, A. Ahmad, and G. Jeon, "A multilayer deep learning approach for malware classification in 5g-enabled iiot," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1495–1503, 2022.

[155] H. Orchi, M. Sadik, and M. Khaldoun, "On using artificial intelligence and the internet of things for crop disease detection: A contemporary survey," *Agriculture*, vol. 12, no. 1, p. 9, 2021.

[156] P. S. Chatterjee, N. K. Ray, and S. P. Mohanty, "Livecare: An iot-based healthcare framework for livestock in smart agriculture," *IEEE Transactions on Consumer Electronics*, vol. 67, no. 4, pp. 257–265, 2021.

[157] A. D. Boursianis, M. S. Papadopoulou, P. Diamantoulakis, A. Liopa-Tsakalidi, P. Barouchas, G. Salahas, G. Karagiannidis, S. Wan, and S. K. Goudos, "Internet of things (iot) and agricultural unmanned aerial vehicles (uavs) in smart farming: A comprehensive review," *Internet of Things*, vol. 18, p. 100187, 2022.

[158] S. K. Roy, S. Misra, N. S. Raghuwanshi, and S. K. Das, "Agrisens: Iot-based dynamic irrigation scheduling system for water management of irrigated crops," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 5023–5030, 2020.

[159] R. Saha, S. Misra, and P. K. Deb, "Fogfl: Fog-assisted federated learning for resource-constrained iot devices," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 8456–8463, 2020.

[160] S. Liu, S. Wang, X. Liu, C.-T. Lin, and Z. Lv, "Fuzzy detection aided real-time and robust visual tracking under complex environments," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 1, pp. 90–102, 2020.

[161] V. VaniB and C. Guruprakash, "A review on smart agricultural applications: crop yield and plant disease prediction," in *2022 International conference on applied artificial intelligence and computing (ICAAIC)*, pp. 885–892, IEEE, 2022.

[162] N. K. Jha and P. K. Shukla, "Classification and health prediction in plants using deep convolutional neural networks," in *2023 3rd International Conference on Innovative Practices in Technology and Management (ICIPTM)*, pp. 1–5, IEEE, 2023.

[163] N. Sengar, M. K. Dutta, and C. M. Travieso, "Computer vision based technique for identification and quantification of powdery mildew disease in cherry leaves," *Computing*, vol. 100, no. 11, pp. 1189–1201, 2018.

[164] N. S. Russel and A. Selvaraj, "Leaf species and disease classification using multiscale parallel deep cnn architecture," *Neural Computing and Applications*, vol. 34, no. 21, pp. 19217–19237, 2022.

[165] K. Zhang, L. Zhang, and Q. Wu, "Identification of cherry leaf disease infected by podosphaera pannosa via convolutional neural network," *International Journal of Agricultural and Environmental Information Systems (IJAEIS)*, vol. 10, no. 2, pp. 98–110, 2019.

[166] S. Barburiceanu, S. Meza, B. Orza, R. Malutan, and R. Terebes, "Convolutional neural networks for texture feature extraction. applications to leaf disease classification in precision agriculture," *IEEE Access*, vol. 9, pp. 160085–160103, 2021.

[167] J. G. A. Barbedo, "Expert systems applied to plant disease diagnosis: survey and critical view," *IEEE Latin America Transactions*, vol. 14, no. 4, pp. 1910–1922, 2016.

[168] A. K. Singh, S. Sreenivasu, U. Mahalaxmi, H. Sharma, D. D. Patil, and E. Asenso, "[retracted] hybrid feature-based disease detection in plant leaf using convolutional neural network, bayesian optimized svm, and random forest classifier," *Journal of Food Quality*, vol. 2022, no. 1, p. 2845320, 2022.

[169] A. Pavithra, G. Kalpana, and T. Vigneswaran, "Retracted article: Deep learning-based automated disease detection and classification model for precision agriculture," *Soft Computing*, vol. 28, no. Suppl 2, pp. 463–463, 2024.

[170] R. Dwivedi, T. Dutta, and Y.-C. Hu, "A leaf disease detection mechanism based on l1-norm minimization extreme learning machine," *IEEE geoscience and remote sensing letters*, vol. 19, pp. 1–5, 2021.

[171] J. Kotwal, R. Kashyap, and S. Pathan, "Agricultural plant diseases identification: From traditional approach to deep learning," *Materials Today: Proceedings*, vol. 80, pp. 344–356, 2023.

[172] S. Yu, L. Xie, and Q. Huang, "Inception convolutional vision transformers for plant disease identification," *Internet of Things*, vol. 21, p. 100650, 2023.

[173] V. N. T. Le, S. Ahderom, B. Apopei, and K. Alameh, "A novel method for detecting morphologically similar crops and weeds based on the combination of contour masks and filtered local binary pattern operators," *GigaScience*, vol. 9, no. 3, p. giaa017, 2020.

[174] A. Bhatia, A. Chug, and A. Prakash Singh, "Application of extreme learning machine in plant disease prediction for highly imbalanced dataset," *Journal of Statistics and Management Systems*, vol. 23, no. 6, pp. 1059–1068, 2020.

[175] A. Abbas, S. Jain, M. Gour, and S. Vankudothu, "Tomato plant disease detection using transfer learning with c-gan synthetic images," *Computers and electronics in agriculture*, vol. 187, p. 106279, 2021.

[176] S. Ashwinkumar, S. Rajagopal, V. Manimaran, and B. Jegajothi, "Automated plant leaf disease detection and classification using optimal mobilenet based convolutional neural networks," *Materials Today: Proceedings*, vol. 51, pp. 480–487, 2022.

[177] S. Sood and H. Singh, "A comparative study of grape crop disease classification using various transfer learning techniques," *Multimedia Tools and Applications*, vol. 83, no. 2, pp. 4359–4382, 2024.

[178] A. S. Kini, K. Prema, and S. N. Pai, "Early stage black pepper leaf disease prediction based on transfer learning using convnets," *Scientific Reports*, vol. 14, no. 1, p. 1404, 2024.

[179] I. Khan, S. S. Sohail, D. Ø. Madsen, and B. K. Khare, "Deep transfer learning for fine-grained maize leaf disease classification," *Journal of Agriculture and Food Research*, vol. 16, p. 101148, 2024.

[180] M. Rezaei, D. Diepeveen, H. Laga, and F. Sohel, "Automatic pixel-level annotation for plant disease severity estimation," *Computers and electronics in agriculture*, vol. 241, p. 111316, 2026.

[181] A. Chug, A. Bhatia, A. P. Singh, and D. Singh, "A novel framework for image-based plant disease detection using hybrid deep learning approach," *Soft Computing*, vol. 27, no. 18, pp. 13613–13638, 2023.

[182] A. Beikmohammadi, K. Faez, and A. Motallebi, "Swp-leafnet: A novel multistage approach for plant leaf identification based on deep cnn," *Expert Systems with Applications*, vol. 202, p. 117470, 2022.

[183] D. Aqel, S. Al-Zubi, A. Mughaid, and Y. Jararweh, "Extreme learning machine for plant diseases classification: a sustainable approach for smart agriculture," *Cluster Computing*, vol. 25, no. 3, pp. 2007–2020, 2022.

[184] L. Liu, C. Xie, R. Wang, P. Yang, S. Sudirman, J. Zhang, R. Li, and F. Wang, "Deep learning based automatic multiclass wild pest monitoring approach using hybrid global and local activated features," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7589–7598, 2020.

[185] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[186] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[187] G.-B. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: a survey," *International journal of machine learning and cybernetics*, vol. 2, no. 2, pp. 107–122, 2011.

[188] P. Kaur, S. Harnal, R. Tiwari, S. Upadhyay, S. Bhatia, A. Mashat, and A. M. Alabdali, "Recognition of leaf disease using hybrid convolutional neural network by applying feature reduction," *Sensors*, vol. 22, no. 2, p. 575, 2022.

[189] J. S. Prashanth, N. R. Moparthi, G. B. Krishna, A. K. Prasad, B. Sravankumar, and P. R. Rao, "Mpcsar-ahh: A hybrid deep learning model for real-time detection of cassava leaf diseases and fertilizer recommendation," *Computers and Electrical Engineering*, vol. 119, p. 109628, 2024.

[190] R. Dogra, S. Rani, A. Singh, M. A. Albahar, A. E. Barrera, and A. Alkhayyat, "Deep learning model for detection of brown spot rice leaf disease with smart agriculture," *Computers and Electrical Engineering*, vol. 109, p. 108659, 2023.

[191] A. Nag, P. R. Chanda, and S. Nandi, "Mobile app-based tomato disease identification with fine-tuned convolutional neural networks," *Computers and Electrical Engineering*, vol. 112, p. 108995, 2023.

[192] M. H. Junos, A. S. Mohd Khairuddin, S. Thannirmalai, and M. Dahari, "Automatic detection of oil palm fruits from uav images using an improved yolo model," *The visual computer*, vol. 38, no. 7, pp. 2341–2355, 2022.

[193] A. Dolatabadian, T. X. Neik, M. F. Danilevicz, S. R. Upadhyaya, J. Batley, and D. Edwards, "Image-based crop disease detection using machine learning," *Plant Pathology*, vol. 74, no. 1, pp. 18–38, 2025.

[194] X. E. Pantazi, D. Moshou, and A. A. Tamouridou, "Automated leaf disease detection in different crop species through image features analysis and one class classifiers," *Computers and electronics in agriculture*, vol. 156, pp. 96–104, 2019.

[195] E. C. Tetila, B. B. Machado, G. K. Menezes, A. d. S. Oliveira, M. Alvarez, W. P. Amorim, N. A. D. S. Belete, G. G. Da Silva, and H. Pistori, "Automatic recognition of soybean leaf diseases using uav images and deep convolutional neural networks," *IEEE geoscience and remote sensing letters*, vol. 17, no. 5, pp. 903–907, 2019.

[196] S. Kaur, S. Pandey, and S. Goel, "Semi-automatic leaf disease detection and classification system for soybean culture," *IET Image processing*, vol. 12, no. 6, pp. 1038–1048, 2018.

[197] Y. Sun, Z. Jiang, L. Zhang, W. Dong, and Y. Rao, "Slic_svm based leaf diseases saliency map extraction of tea plant," *Computers and electronics in agriculture*, vol. 157, pp. 102–109, 2019.

[198] A. S. Paymode and V. B. Malode, "Transfer learning for multi-crop leaf disease image classification using convolutional neural network vgg," *Artificial Intelligence in Agriculture*, vol. 6, pp. 23–33, 2022.

[199] Z. Zhang, Y. Qiao, Y. Guo, and D. He, "Deep learning based automatic grape downy mildew detection," *Frontiers in Plant Science*, vol. 13, p. 872107, 2022.

[200] S. P. Praveen, R. Nakka, A. Chokka, V. N. Thatha, S. S. Vellela, and U. Sirisha, "A novel classification approach for grape leaf disease detection based on different attention deep learning techniques," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 14, no. 6, p. 2023, 2023.

[201] J. Lin, X. Chen, R. Pan, T. Cao, J. Cai, Y. Chen, X. Peng, T. Cernava, and X. Zhang, "Grapenet: A lightweight convolutional neural network model for identification of grape leaf diseases," *Agriculture*, vol. 12, no. 6, p. 887, 2022.

[202] P. S. Thakur, T. Sheorey, and A. Ojha, "Vgg-icnn: A lightweight cnn model for crop disease identification," *Multimedia Tools and Applications*, vol. 82, no. 1, pp. 497–520, 2023.

[203] P. S. Thakur, P. Khanna, T. Sheorey, and A. Ojha, "Explainable vision transformer enabled convolutional neural network for plant disease identification: Plantxvit," *arXiv preprint arXiv:2207.07919*, 2022.

[204] S. Liu, H. BAI, F. LI, D. WANG, Y. ZHENG, Q. JIANG, and F. SUN, "An apple leaf disease identification model for safeguarding apple food safety," *Food Science and Technology*, vol. 43, p. e104322, 2023.

[205] R. Suguna, S. Deepa, and A. MS, "Apple and tomato leaves disease detection using emperor penguins optimizer based cnn," in *2023 9th International conference on advanced computing and communication systems (ICACCS)*, vol. 1, pp. 1803–1808, IEEE, 2023.

[206] M. S. Al-gaashani, F. Shang, M. S. Muthanna, M. Khayyat, and A. A. Abd El-Latif, "Tomato leaf disease classification by exploiting transfer learning and feature concatenation," *IET Image Processing*, vol. 16, no. 3, pp. 913–925, 2022.

[207] X. Tian, X. Meng, Q. Wu, Y. Chen, and J. Pan, "Identification of tomato leaf diseases based on a deep neuro-fuzzy network," *Journal of The Institution of Engineers (India): Series A*, vol. 103, no. 2, pp. 695–706, 2022.

[208] M. Agarwal, A. Singh, S. Arjaria, A. Sinha, and S. Gupta, "Toled: Tomato leaf disease detection using convolution neural network," *Procedia Computer Science*, vol. 167, pp. 293–301, 2020.

[209] S. Parez, N. Dilshad, T. M. Alanazi, and J.-W. Lee, "Towards sustainable agricultural systems: A lightweight deep learning model for plant disease detection.," *Comput. Syst. Sci. Eng.*, vol. 47, no. 1, pp. 515–536, 2023.

[210] S. M. Hassan, A. K. Maji, M. Jasiński, Z. Leonowicz, and E. Jasińska, "Identification of plant-leaf diseases using cnn and transfer-learning approach," *Electronics*, vol. 10, no. 12, p. 1388, 2021.

[211] J. Thomkaew and S. Intakosum, "Improvement classification approach in tomato leaf disease using modified visual geometry group (vgg)-inceptionv3," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 12, 2022.

[212] G.-B. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: a survey," *International journal of machine learning and cybernetics*, vol. 2, no. 2, pp. 107–122, 2011.

[213] S. Ahmed, M. B. Hasan, T. Ahmed, M. R. K. Sony, and M. H. Kabir, "Less is more: Lighter and faster deep neural architecture for tomato leaf disease classification," *IEEE Access*, vol. 10, pp. 68868–68884, 2022.

[214] G. Owomugisha and E. Mwebaze, "Machine learning for plant disease incidence and severity measurements from leaf images," in *2016 15th IEEE international conference on machine learning and applications (ICMLA)*, pp. 158–163, IEEE, 2016.

[215] A. Picon, A. Alvarez-Gila, M. Seitz, A. Ortiz-Barredo, J. Echazarra, and A. Johannes, "Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild," *Computers and Electronics in Agriculture*, vol. 161, pp. 280–290, 2019.

[216] S. Ashok, G. Kishore, V. Rajesh, S. Suchitra, S. G. Sophia, and B. Pavithra, "Tomato leaf disease detection using deep learning techniques," in *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, pp. 979–983, IEEE, 2020.

[217] M. Moussafir, H. Chaibi, R. Saadane, A. Chehri, A. E. Rharras, and G. Jeon, "Design of efficient techniques for tomato leaf disease detection using genetic algorithm-based and deep neural networks," *Plant and Soil*, vol. 479, no. 1, pp. 251–266, 2022.

[218] M. Altalak, M. A. Uddin, A. Alajmi, and A. Rizg, "A hybrid approach for the detection and classification of tomato leaf diseases," *Applied Sciences*, vol. 12, no. 16, p. 8182, 2022.

[219] A. S. Paymode and V. B. Malode, "Transfer learning for multi-crop leaf disease image classification using convolutional neural network vgg," *Artificial Intelligence in Agriculture*, vol. 6, pp. 23–33, 2022.

[220] V. Gonzalez-Huitron, J. A. León-Borges, A. Rodriguez-Mata, L. E. Amabilis-Sosa, B. Ramírez-Pereda, and H. Rodriguez, "Disease detection in tomato leaves via cnn with lightweight architectures implemented in raspberry pi 4," *Computers and Electronics in Agriculture*, vol. 181, p. 105951, 2021.

[221] S. Z. M. Zaki, M. A. Zulkifley, M. M. Stofa, N. A. M. Kamari, and N. A. Mohamed, "Classification of tomato leaf diseases using mobilenet v2," *IAES International Journal of Artificial Intelligence*, vol. 9, no. 2, p. 290, 2020.

[222] L. Tan, J. Lu, and H. Jiang, "Tomato leaf diseases classification based on leaf images: a comparison between classical machine learning and deep learning methods," *AgriEngineering*, vol. 3, no. 3, pp. 542–558, 2021.

[223] D. Aqel, S. Al-Zubi, A. Mughaid, and Y. Jararweh, "Extreme learning machine for plant diseases classification: a sustainable approach for smart agriculture," *Cluster Computing*, vol. 25, no. 3, pp. 2007–2020, 2022.

[224] M. S. Alzahrani and F. W. Alsaade, "Transform and deep learning algorithms for the early detection and recognition of tomato leaf disease," *Agronomy*, vol. 13, no. 5, p. 1184, 2023.

[225] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[226] G. Geetha, S. Samundeswari, G. Saranya, K. Meenakshi, and M. Nithya, "Plant leaf disease classification and detection system using machine learning," in *Journal of Physics: Conference Series*, vol. 1712, p. 012012, IOP Publishing, 2020.

[227] G. Liu, S. Mao, and J. H. Kim, "A mature-tomato detection algorithm using machine learning and color analysis," *Sensors*, vol. 19, no. 9, p. 2023, 2019.

[228] H. Yu, J. Liu, C. Chen, A. A. Heidari, Q. Zhang, H. Chen, M. Mafarja, and H. Turabieh, "Corn leaf diseases diagnosis based on k-means clustering and deep learning," *Ieee Access*, vol. 9, pp. 143824–143835, 2021.

[229] K. P. Panigrahi, H. Das, A. K. Sahoo, and S. C. Moharana, "Maize leaf disease detection and classification using machine learning algorithms," in *Progress in computing, analytics and networking: proceedings of ICCAN 2019*, pp. 659–669, Springer, 2020.

[230] K. K. Jena, S. K. Bhoi, D. Mohapatra, C. Mallick, and P. Swain, "Rice disease classification using supervised machine learning approach," in *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, pp. 328–333, IEEE, 2021.

[231] Q. Wang, F. Qi, M. Sun, J. Qu, and J. Xue, "Identification of tomato disease types and detection of infected areas based on deep convolutional neural networks and object detection techniques," *Computational intelligence and neuroscience*, vol. 2019, no. 1, p. 9142753, 2019.

[232] A. A. Alatawi, S. M. Alomani, N. I. Alhawiti, and M. Ayaz, "Plant disease detection using ai based vgg-16 model," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 4, 2022.

[233] A. Abbas, S. Jain, M. Gour, and S. Vankudothu, "Tomato plant disease detection using transfer learning with c-gan synthetic images," *Computers and electronics in agriculture*, vol. 187, p. 106279, 2021.

[234] M. A. Haque, S. Marwaha, C. K. Deb, S. Nigam, A. Arora, K. S. Hooda, P. L. Soujanya, S. K. Aggarwal, B. Lall, M. Kumar, *et al.*, "Deep learning-based approach for identification of diseases of maize crop," *Scientific reports*, vol. 12, no. 1, p. 6334, 2022.

[235] Y. Wang, H. Wang, and Z. Peng, "Rice diseases detection and classification using attention based neural network and bayesian optimization," *Expert Systems with Applications*, vol. 178, p. 114770, 2021.

[236] Y. Wang, Y. Chen, and D. Wang, "Convolution network enlightened transformer for regional crop disease classification," *Electronics*, vol. 11, no. 19, p. 3174, 2022.

[237] M. A. Azim, M. K. Islam, M. M. Rahman, and F. Jahan, "An effective feature extraction method for rice leaf disease classification," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 19, no. 2, pp. 463–470, 2021.

[238] G. Li, Y. Wang, Q. Zhao, P. Yuan, and B. Chang, "Pmvt: a lightweight vision transformer for plant disease identification on mobile devices," *Frontiers in Plant Science*, vol. 14, p. 1256773, 2023.

[239] M. Altalak, M. A. Uddin, A. Alajmi, and A. Rizg, "A hybrid approach for the detection and classification of tomato leaf diseases," *Applied Sciences*, vol. 12, no. 16, p. 8182, 2022.

[240] M. Gehlot, R. K. Saxena, and G. C. Gandhi, ""tomato-village": a dataset for end-to-end tomato disease detection in a real-world environment," *Multimedia Systems*, vol. 29, no. 6, pp. 3305–3328, 2023.

[241] T. Kim, H. Kim, K. Baik, and Y. Choi, "Instance-aware plant disease detection by utilizing saliency map and self-supervised pre-training," *Agriculture*, vol. 12, no. 8, p. 1084, 2022.

[242] T. Abekoon, H. Sajindra, N. Rathnayake, I. U. Ekanayake, A. Jayakody, and U. Rathnayake, "A novel application with explainable machine learning (shap and lime) to predict soil n, p, and k nutrient content in cabbage cultivation," *Smart Agricultural Technology*, vol. 11, p. 100879, 2025.

[243] A. Umamageswari, S. Deepa, and K. Raja, "An enhanced approach for leaf disease identification and classification using deep learning techniques," *Measurement: Sensors*, vol. 24, p. 100568, 2022.

[244] H. Sharma, D. Padha, and N. Bashir, "D-kap: A deep learning-based kashmiri apple plant disease prediction framework," in *2022 seventh international conference on parallel, distributed and grid computing (PDGC)*, pp. 576–581, IEEE, 2022.

[245] N. H. Shabrina, S. Indarti, R. Maharani, D. A. Kristiyanti, *et al.*, "Potato leaf disease dataset in uncontrolled environment," *Mendeley Data*, vol. 1, 2023.

[246] U. Afzaal, B. Bhattarai, Y. R. Pandeya, and J. Lee, "An instance segmentation model for strawberry diseases based on mask r-cnn," *Sensors*, vol. 21, no. 19, p. 6565, 2021.

[247] S. Hossain, M. Tanzim Reza, A. Chakrabarty, and Y. J. Jung, "Aggregating different scales of attention on feature variants for tomato leaf disease diagnosis from image data: a transformer driven study," *Sensors*, vol. 23, no. 7, p. 3751, 2023.

[248] A. H. Wajid, N. Saher, S. A. Nawaz, M. Arshad, and M. Nasir, "Tomato leaf disease detection and classification using convolutional neural network and machine learning," *Journal of Computing & Biomedical Informatics*, 2024.

[249] A. Rachmad, M. Syarief, S. Rifka, F. Sonata, W. Setiawan, and E. M. S. Rochman, "Corn leaf disease classification using local binary patterns (lbp) feature extraction," in *Journal of Physics: Conference Series*, vol. 2406, p. 012020, IOP Publishing, 2022.

[250] S. S. Pandi, A. Senthilselvi, J. Gitanjali, K. ArivuSelvan, J. Gopal, and J. Vellingiri, "Rice plant disease classification using dilated convolutional neural network with global average pooling," *Ecological Modelling*, vol. 474, p. 110166, 2022.

[251] N. Zeng, G. Gong, G. Zhou, and C. Hu, "An accurate classification of rice diseases based on icai-v4," *Plants*, vol. 12, no. 11, p. 2225, 2023.

[252] G. Dai, Z. Tian, J. Fan, C. Sunil, and C. Dewi, "Dfn-psan: Multi-level deep information feature fusion extraction network for interpretable plant disease classification," *Computers and Electronics in Agriculture*, vol. 216, p. 108481, 2024.

[253] D. Agarwal, M. Chawla, and N. Tiwari, "Plant leaf disease classification using deep learning: A survey," in *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 643–650, IEEE, 2021.

[254] S. K. Upadhyay and A. Kumar, "A novel approach for rice plant diseases classification with deep convolutional neural network," *International Journal of Information Technology*, vol. 14, no. 1, pp. 185–199, 2022.

[255] S. M. Hassan and A. K. Maji, "Plant disease identification using a novel convolutional neural network," *IEEE Access*, vol. 10, pp. 5390–5401, 2022.

[256] Y. Wang, Y. Yin, Y. Li, T. Qu, Z. Guo, M. Peng, S. Jia, Q. Wang, W. Zhang, and F. Li, "Classification of plant leaf disease recognition based on self-supervised learning," *Agronomy*, vol. 14, no. 3, p. 500, 2024.

[257] M. Bahrami, A. Pourhatami, and M. Maboodi, "Tomato leaf disease detection using transfer learning: A comparative study," in *2024 13th Iranian/3rd International Machine Vision and Image Processing Conference (MVIP)*, pp. 1–5, IEEE, 2024.

[258] M. Bahrami, A. Pourhatami, and M. Maboodi, "Tomato leaf disease detection using transfer learning: A comparative study," in *2024 13th Iranian/3rd International Machine Vision and Image Processing Conference (MVIP)*, pp. 1–5, IEEE, 2024.

[259] P. Alirezazadeh, M. Schirrmann, and F. Stolzenburg, "Improving deep learning-based plant disease classification with attention mechanism," *Gesunde Pflanzen*, vol. 75, no. 1, pp. 49–59, 2023.

[260] W. Shafik, A. Tufail, C. D. S. Liyanage, and R. A. A. H. M. Apong, "Using a novel convolutional neural network for plant pests detection and disease classification," *Journal of the Science of Food and Agriculture*, vol. 103, no. 12, pp. 5849–5861, 2023.

[261] A. Stephen, A. Punitha, and A. Chandrasekar, "Designing self attention-based resnet architecture for rice leaf disease classification," *Neural Computing and Applications*, vol. 35, no. 9, pp. 6737–6751, 2023.

[262] H. Sun, I. T. Nicholaus, R. Fu, and D.-K. Kang, "Yolo-fmdi: A lightweight yolov8 focusing on a multi-scale feature diffusion interaction neck for tomato pest and disease detection," *Electronics*, vol. 13, no. 15, p. 2974, 2024.

[263] A. Ahila, V. Prema, S. Ayyasamy, and M. Sivasubramanian, "An enhanced deep learning model for high-speed classification of plant diseases with bioinspired algorithm," *The Journal of Supercomputing*, vol. 80, no. 3, pp. 3713–3737, 2024.

[264] Y. Wang, Y. Yin, Y. Li, T. Qu, Z. Guo, M. Peng, S. Jia, Q. Wang, W. Zhang, and F. Li, "Classification of plant leaf disease recognition based on self-supervised learning," *Agronomy*, vol. 14, no. 3, p. 500, 2024.

[265] R. Karthik, A. Ajay, A. S. Bisht, T. Illakiya, and K. Suganthi, "A deep learning approach for crop disease and pest classification using swin transformer and dual-attention multi-scale fusion network," *IEEE Access*, 2024.

[266] M. Aggarwal, V. Khullar, N. Goyal, and T. A. Prola, "Resource-efficient federated learning over ioat for rice leaf disease classification," *Computers and Electronics in Agriculture*, vol. 221, p. 109001, 2024.

[267] P. Hari and M. P. Singh, "Adaptive knowledge transfer using federated deep learning for plant disease detection," *Computers and Electronics in Agriculture*, vol. 229, p. 109720, 2025.

[268] K. Kaur, S. Chattopadhyay, A. Verma, V. Kukreja, and S. Mehta, "A synergistic approach to cucurbit leaf disease classification with federated learning cnns," in *2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*, vol. 2, pp. 1–6, IEEE, 2024.

[269] R. Tripathy, J. Mandala, S. R. Pappu, and G. K. D. Gopisetty, "Optimization based rice leaf disease classification in federated learning," *Multimedia Tools and Applications*, vol. 83, no. 29, pp. 72491–72517, 2024.

[270] D. Mamba Kabala, A. Hafiane, L. Bobelin, and R. Canals, "Image-based crop disease detection with federated learning," *Scientific Reports*, vol. 13, no. 1, p. 19220, 2023.

[271] Z. Liu, G. Zhou, W. Zhu, Y. Chai, L. Li, Y. Wang, Y. Hu, W. Dai, R. Liu, and L. Sun, "Identification of rice disease under complex background based on psoc-drcnet," *Expert Systems with Applications*, vol. 249, p. 123643, 2024.

[272] Y. Wang, S. Wang, W. Ni, and Q. Zeng, "Past-net: a swin transformer and path aggregation model for anthracnose instance segmentation," *Multimedia Systems*, vol. 29, no. 3, pp. 1011–1023, 2023.

[273] A. Sharma, R. K. Bijral, J. Manhas, and V. Sharma, "Mango leaf diseases detection using deep learning," *International Journal of Knowledge Based Computer Systems*, vol. 10, no. 1, pp. 40–44, 2022.

[274] M. Fahim-Ul-Islam, A. Chakrabarty, R. Rahman, H. Moon, and M. J. Piran, "Advancing mango leaf variant identification with a robust multi-layer perceptron model," *Scientific Reports*, vol. 14, no. 1, p. 27406, 2024.

[275] S. M. R. Swapno, S. N. Nobel, M. B. Islam, R. Haque, V. Meena, and F. Benedetto, "A novel machine learning approach for fast and efficient detection of mango leaf diseases," in *2024 IEEE 3rd international conference on computing and machine intelligence (ICMI)*, pp. 1–7, IEEE, 2024.

[276] C. Chandrashekar, K. Vijayakumar, K. Pradeep, and A. Balasundaram, "Mdcn: Modified dense convolution network based disease classification in mango leaves," *Computers, Materials, & Continua*, vol. 78, no. 2, p. 2511, 2024.

[277] A. L. A. Haikal, N. Yudistira, and A. Ridok, "Comprehensive mixed-based data augmentation for detection of rice leaf disease in the wild," *Crop Protection*, vol. 184, p. 106816, 2024.

[278] W. Gao, C. Zong, M. Wang, H. Zhang, and Y. Fang, "Intelligent identification of rice leaf disease based on yolo v5-efficient," *Crop Protection*, vol. 183, p. 106758, 2024.