PLANT DISEASE DETECTION AND PREVENTION USING DEEP LEARNING

A PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY
IN
ARTIFICIAL INTELLIGENCE

Submitted by

SAURABH SHARMA

2K21/AFI/28

Under the supervision of

Dr. Prashant Giridhar Shambharkar

Assistant Professor Department of Computer Science and Engineering

Department of Computer Science and Engineering
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi 110042

MAY 2023

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, SAURABH SHARMA, Roll No's – 2K21/AFI/28 student of M.Tech (Department of Computer Science and Engineering),hereby declare that the project Dissertation titled "PLANT DISEASE DETECTION AND PREVENTION USING DEEP LEARNING" which is submitted by me to the Department of Computer Science and Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi

Saurabh Sharma

Date: 31.05.2023

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled "PLANT DISEASE DETECTION AND PREVENTION USING DEEP LEARNING" which is submitted by Saurabh Sharma, Roll No's – 2K21/AFI/28, Department of Computer Science and Engineering ,Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Bachelor of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi — Dr. Prashant Giridhar Shambharkar

Date: 31.05.2023 — **SUPERVISOR**

DEPARTMENT OF MECHANICAL ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I wish to express my sincerest gratitude to Dr. Prashant Giridhar Shambharkar and Mr. Nikhil Sharma for their continuous guidance and mentorship that they provided me during the project. They showed me the path to achieve my targets by explaining all the tasks to be done and explained to me the importance of this project as well as its industrial relevance. They was always ready to help me and clear my doubts regarding any hurdles in this project. Without their constant support and motivation, this project would not have been successful.

Place: Delhi
Saurabh Sharma

Date: 31.05.2023

Abstract

Identifying and managing diseases in plants is a critical area of research in agriculture, aiming to ensure optimal crop yield and minimize losses. With India's large population and the increasing demand for food production, it becomes imperative to address crop diseases effectively and efficiently. Detecting diseases in crops allows for timely intervention and preventive measures to protect agricultural resources and meet the growing food requirements. Traditionally, disease identification in plants has relied on the expertise of trained professionals who visually inspect crops or conduct chemical tests. However, these methods can be time-consuming, costly, and dependent on the availability of skilled personnel. To overcome these limitations, an automated system that can accurately detect plant diseases by analyzing plant leaves can revolutionize disease management in crops. By leveraging deep learning models, we can classify leaves based on the specific diseases they exhibit, enabling targeted and timely interventions. In this paper, we focus on detecting diseases in cassava plants, which are widely consumed, including their leaves. The objective of this study is to improve the accuracy and efficiency of disease detection compared to previously proposed models. Deep learning models have shown superior performance in plant disease detection compared to traditional machine learning methods, prompting their adoption in this study. Specifically, we employ the EfficientNet-B0 architecture, which is renowned for its classification capabilities, speed, and scalability in terms of width, depth, and resolution. To ensure robust evaluation and model performance, we utilize k-fold cross-validation, a technique that divides the dataset into k subsets for training and validation. This approach helps to assess the model's generalizability and ensures reliable accuracy measurements. The results obtained from our study indicate a significant advancement in cassava disease detection. Our proposed model achieves an impressive accuracy of 96.68% on the dataset collected from Kaggle, demonstrating the model's effectiveness in identifying diseases in cassava plants. The high accuracy rate achieved by the EfficientNet-B0 model provides promising prospects for real-world applications and the potential to enhance disease management practices in cassava cultivation.

By leveraging the power of deep learning and incorporating automated disease detection systems into agricultural practices, we can significantly improve disease management strategies, reduce crop losses, and enhance food security. The findings of this study contribute to the growing body of research on plant disease detection and provide valuable insights for further advancements in this field.

Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

Cassava is a very important African crops , of which Africa is the single largest producer . cassava is a crop which is successfully cultivated in any climate , non productive soil and drought included . though the crops faces several difficulties during the time of production , i.e. not of good quality , leaf diseases , of which The primary cause of reduced production is leaf diseases , which affects income of the farmers . the identification and treatment of such leaf diseases must be done on priority to regain capacity of production, detect automatically is a step which is very important to detect such diseases in crops on crop leaves . If managed correctly this step helps in improved production of the crops. If the leaf diseases are recognized on time the plant can be saved before the crop is permanently damaged. Researchers are building plants fusion which are resistant to the organisms which damage crops , they also created a system which recognises crop diseases . Smart farming is also a way in which farm owners can increase their profit margins, without doing manual identification of the leaf diseases which is anyway ineffective and inaccurate , instead deep learning methods and image classification techniques can be used to do the same. An automated ML model will help in detection and treatment of these crop diseases , enabling the farmers in their yields. CNN classifiers which are based on deep learning helps detecting the leaf diseases in every possible scenarios where diagnostics based on images are possible with advanced training , portable devices are also used To detect crop diseases after this the datasets are used to enhance accuracy . Researchers then do data pre - processsing which is essential to analysis of image , which has several processes , i.e. reshape , feature extraction , optimization of image , adjustment of color . Hybrid model for leaf diseases detection is discussed which is based on deep learning and image classification . these research have certain challenges which needs to be worked upon .This also encouraged researchers to work on detection disease causes cassava leaf . They are encouraged to develop a more robust system to solve this problem [1]. In India, approx 70% of the population depends on agriculture. So it becomes more necessary to detect the disease in plants and cure the disease before it causes trouble for other plants. In India peoples are leaving farming because they are not getting good profit because their crops are damaged because of some disease. So if disease can be detected before it causes loss then we can help farmers because some crops' leaves show the effect before and we can treat them before they come in the situation where we cannot do anything. Agriculture contributes to the economy in India and also a primary source of revenue. During COVID-19 restrictions on food cause a food crisis in the whole world so for that we also need to save crops from the diseases[2]. The global pandemic has disrupted the food supply chain, highlighting the urgent need for sustainable and resilient agricultural practices. In an effort to meet the increasing demand for food, farmers have increasingly relied on pesticides to boost crop yields. However, this approach often comes at the cost of compromising the quality and safety of the harvested produce. Pesticides have been linked to numerous health risks, and it is estimated that they contribute to approximately 300,000 deaths worldwide each year. To address these challenges and ensure a safer and healthier food production system, researchers have turned to artificial intelligence (AI) algorithms for disease detection in plants. Various machine learning algorithms, such as Support Vector Machines (SVM), k Nearest Neighbors (KNN), logistic regression, and decision trees, have been employed for disease diagnosis in plants. However, recent advancements in deep learning models, specifically Deep Neural Networks (DNN) and Convolutional Neural Networks (CNN), have demonstrated superior performance and efficacy in plant disease detection. DNNs and CNNs have revolutionized the field of image classification and analysis, making them ideal candidates for plant disease detection. These models excel at automatically learning complex patterns and features from large volumes of data, such as digital images of plant leaves. By training DNN and CNN models on extensive datasets of labeled plant images, they can accurately identify and classify diseases, enabling swift and precise intervention. The adoption of AI-based disease detection methods offers several advantages. Firstly, it reduces the reliance on harmful pesticides, promoting sustainable agricultural practices that prioritize environmental and human health. By accurately identifying diseases at an early stage, farmers can implement targeted treatments and management strategies, minimizing the use of chemicals and reducing the risk of crop losses. Moreover, AI algorithms can analyze large-scale datasets more efficiently than manual inspection, allowing for rapid and scalable disease monitoring across vast agricultural areas. The promising results achieved by DNNs and CNNs in plant disease detection have encouraged researchers and practitioners to explore their potential in real-world applications. These advanced AI algorithms have the capacity to transform the way we detect and manage diseases in plants, ultimately leading to a safer, more sustainable, and productive agricultural system. In conclusion, the COVID-19 pandemic has underscored the vulnerabilities in the global food supply chain, necessitating innovative approaches to ensure food security. The excessive use of pesticides poses risks to both human health and the environment. However, the emergence of AI algorithms, particularly DNNs and CNNs, has paved the way for more accurate and efficient plant disease detection. By harnessing the power of these advanced models, we can revolutionize agricultural practices, reducing reliance on harmful chemicals and promoting sustainable food production systems that prioritize quality, safety, and human well-being.[3].So The model which we proposed is using a deep learning model which is good to arrange the type of disease by using a classification algorithm. There are many CNN based models which give very promising results in plants like tomato plants. But for cassava plants we have no model which can give better results. So for that this model is proposed.

Chapter 2

LITERATURE REVIEW

In order to achieve sustainable development in addition to prevent irrelevant waste, including fiscal and other resources, the identification of diseases of plant within time are crucial. There have been many conventional machine learning models which are proposed in the last ten years for detecting and classifying diseases in plant. Al-Hiary et al[4] identified five different types of leaf diseases in plants based on the features derive in the steps of preprocessing, we clustered them using K-Means, split the lesion area, and extracted features based on color and texture to classify these areas using artificial neural networks (ANN). The method proposed by Revathi et al.[5] identifies disease on cotton leaves of six types . In their procedure, they use particle swarm optimization (such as colors, edges, and textures) for selecting features from feature vectors, and diseases are classified using deep forward neural networks with cross-information gain. An algorithm called GMDH logic algorithm was recently proposed by J. Chen et al.[6] to detect plant diseases automatically. Deep learning progresses alot over the last few years, with convolutional neural networks (CNN) making great strides in recognizing targets and classes of images. This success of CNN is attributed to CNN's capability to distinguish learning, which extracts deep features automatically and completes the learning of deep feature. A significant deal of research has been done on the use of CNN for identifying crops diseases, which has shown excellent results. Several deep learning methods are used to diagnose and classify medical imaging diseases. Further, a comprehensive article published in 2019 concentrated on the utilisation of deep learning in monitoring and sorting the disease in plant leaf. Using CNN models (such as AlexNet and GoogleNet) Mohanty et al.[12] classified plant diseases. They achieved 99.35% classification accuracy in their research. According to et al., CNN models like Inception V4, DenseNets 121,ResNet 50,VGG 16

Inception V4 and ResNet 152 were used. According to reports, the DenseNet architecture had less computation time, less parameters, and highest accuracy in experiments i.e. 99.75 attains in research% when compared to other models. Adedoja et al.[7] took Plant village data set nad did migration learning on the it using the NASNet trained network model. The test's accuracy was 93.82%. This strategy failed to create a network topology that was appropriate for data on plant diseases. Chen et al[6] proposed a model named MobileNet-Beta for detection of plant disease by including the classification activation map into the pre-trained model of MobileNet V2. To test the suggested model, they used their own dataset as well as the Plant Village dataset. It is found in the test's that the MobileNet-Beta model on the Plant Village data set gives accuracy up to 99.85% accurate and give accuracy of 99.11% on its own data set . The aforementioned earlier studies show that deep learning frameworks are being utilised more frequently in the published works to identify disease in plant leaf. Still, more research needs to be done on the use of deep learning frameworks which is era-adapted for detection of disease in plant leaf. In particular, it is unavoidable that an effective model with fewer parameters, a quicker training time, and no performance impact will exist.[8]

2.1 EfficientNet

Alina Maryum et al.[9] uses model EfficientNet model B4[9] which gives accuracy of 89.09% on cassava plant dataset[10]. EfficientNet was created by Google in 2019. It uses compound coefficients to scale models in simple and effective ways. It uses depth wise, width wise and resolution scaling. So if the image is of more resolution then more features can be extracted from the images. So the question is how we find the depth and width for scaling. It gives better results than any other model. It is fast so that it can be used in real life.

$$f = \alpha^{\phi}.\beta^{\phi}.\gamma^{\phi} \tag{2.1}$$

Here α is depth of the network, β is width and γ is resolution. In this value of , and are constants. Values of β, α and γ are 1.1, 1.2 and 1.15 respectively. And the value of ϕ is found out by performing grid search which is also called coefficient. It works on 380x380 images.

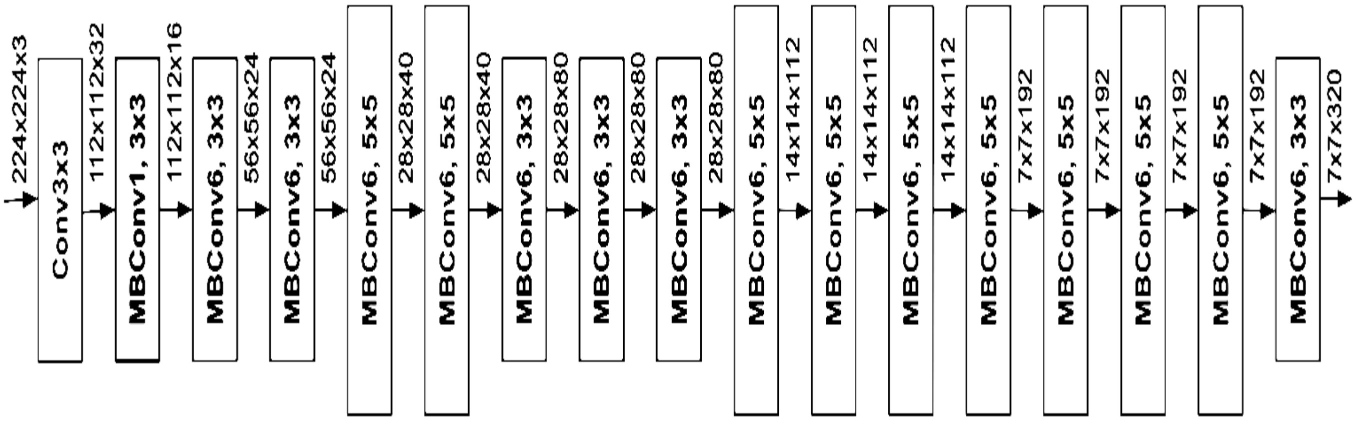


Figure 2.1: Architecture of Efficient Net

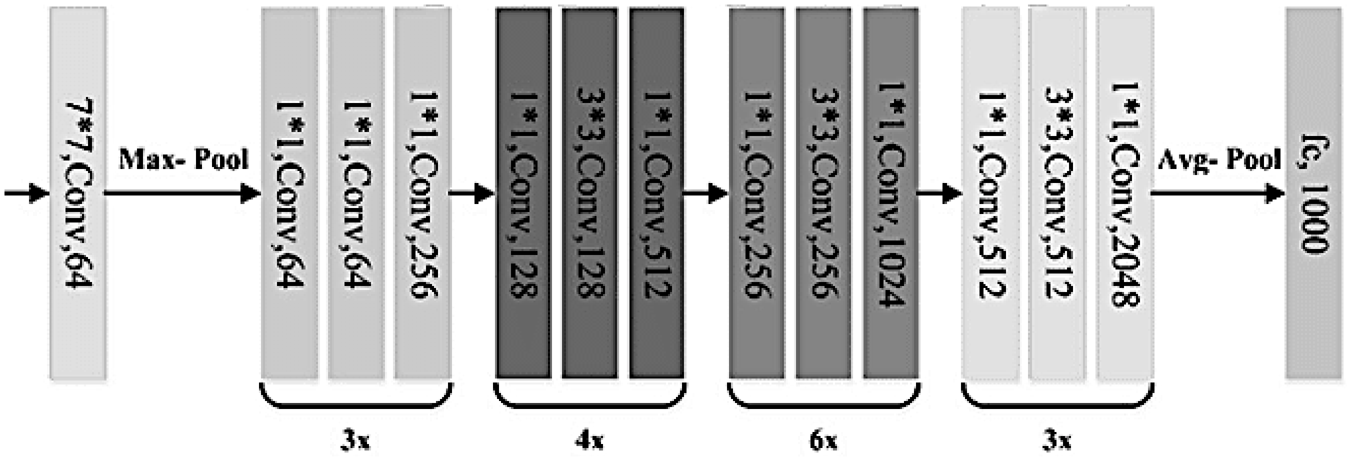


Figure 2.2: Architecture of ResNet

2.2 ResNet50 and SVM Classifier

Sedat METLEK[11] uses the cassava dataset[10] and applies the Resnet50 and SVM Classifier Model which gives 85.4% and 84.4% accuracy respectively[11].

2.2.1 ResNet50

ResNet50 is a 50 layer deep neural network. ResNet stands for Residual Network. It is inspired by the VGG-19 architecture in which shortcut connections are added to overcome the problem of vanishing gradient. It has 48 convolution layers and one layer of max pooling and one layer of average pooling. As shown in fig 2.2 [12]. Disadvantage of ResNet is that is it has vanishing gradient problem. Skip connections are used to resuce this problem in neural networks. In skip connections many connections are skipped directly. The result is not the same after the connections are skipped.

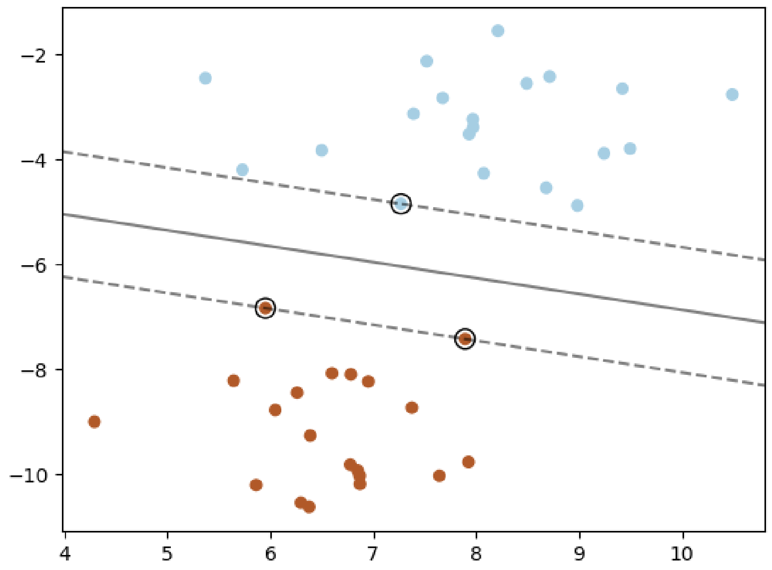


Figure 2.3: SVM[13]

2.2.2 SVM Classifier

SVM is a popular supervised learning algorithm which is used for both regression and classification. Typically, though, this is basically used for problems related to classification in machine learning.

Using SVM, we are able to divide n dimensional space into classes and can put any new data point into the right category easily in the coming time by drawing the best line or decision boundary. A hyperplane represents this best decision boundary.

In SVM, extremum points and vectors are used to create hyperplanes. Support vectors are these extremum cases, so the Support Vector Machine algorithm derives its name from that. A hyperplane is used to classify two distinct categories as shown in fig 2.3[13].

2.3 MobileNet-SSD

Amanda Ramcharan et al.[14] uses the MobileNet-SSD model[14] on the cassava Dataset[10] which gives accuracy of 80.6%. It is a light weighted deep neural network.As we can see that its name suggests it is used to create applications which can be used in mobile. It uses depthwise separable convolution. Single shot detector was developed by google which can work in real time and can run on embedded devices.It is the same as the Efficient

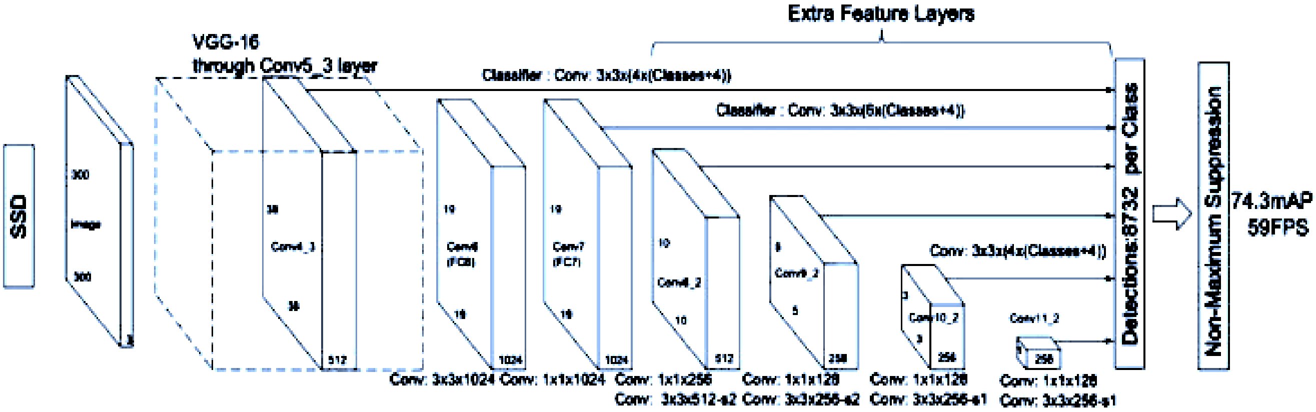


Figure 2.4: Architecture of MobileNet-SSD[14]

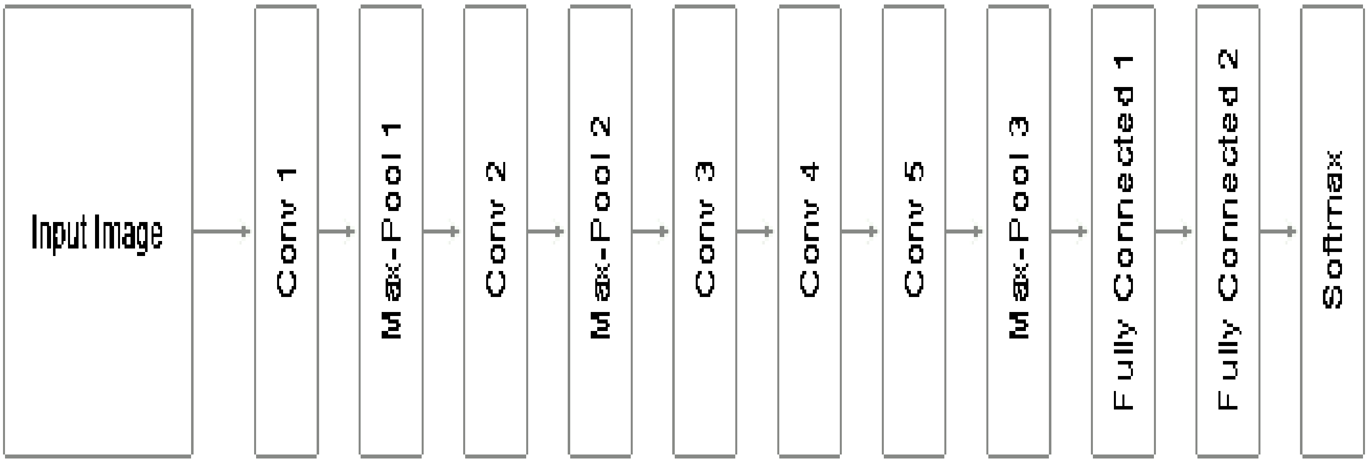


Figure 2.5: Architecture of AlexNet

convolutional neural network which can run on mobile.

2.4 AlexNet

Sharada P. Mohanty et al.[16] use the plant village dataset[15] and apply the AlexNet Model[16] which gives 99.27% accuracy. In this there are 8 weighted layers in this model. As shown in fig 2.5 [17]. There are 5 convolution layers with activation function as ReLU and max pooling operation and there are 3 fully connected layers. First 7 layers of alexnet use the ReLU activation function. After conv5 there is a pooling layer and at last there are two layers which are fully connected and have a dropout layer at last which has a dropout ratio as 0.5.

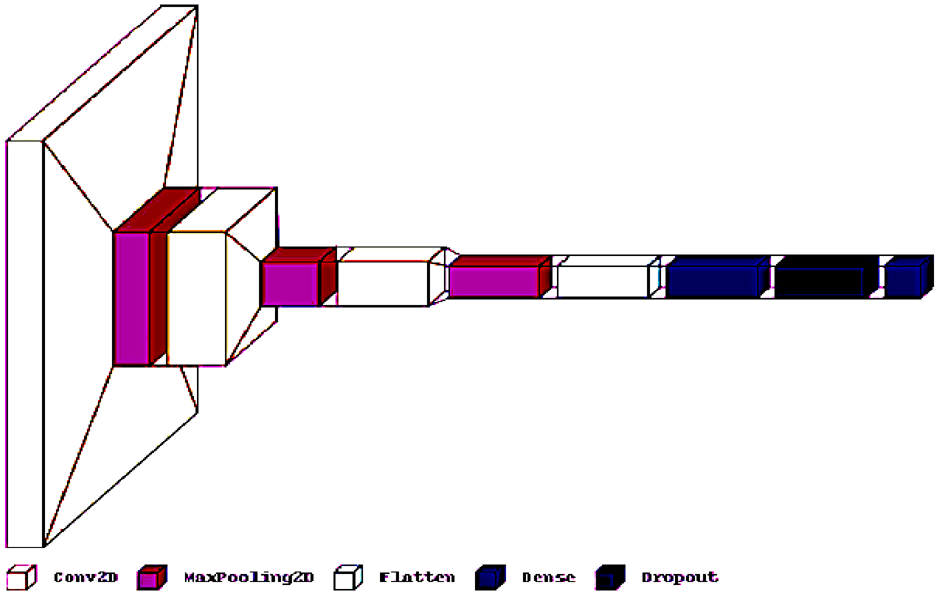


Figure 2.6: Architecture of Deep-CNN[14]

2.5 Deep-CNN

G Geetharamani et al.[18] uses the plant village dataset[15] and applies the Deep CNN Model[18] which gives 96.46% accuracy. In this they use a total 5 weighted layers where 3 layers are convolution layers and each layer is followed by ReLU function as activation function,uses max pooling operation for pooling and two fully connected layers. They trained model for thirty epochs with a 0.5 of learning rate and 64 is the batch size. As shown in fig 2.6.

2.6 DenseNet and VGG-19

Junde Chen et al.[6] uses the Maize plant village dataset[15] and applies the DenseNet[6] and VGG-19[19] Model which gives 98.5% and 92% accuracy respectively.

2.6.1 DenseNet

In DenseNet all the results from the preceding layer is the input for the next layer. In this all the layers are densely connected. After each dense block there is one convolution layer and one pooling layer. In this to increase the accuracy then the number of layers need to increase. But after a limit of layers the problem of vanishing gradient takes place.

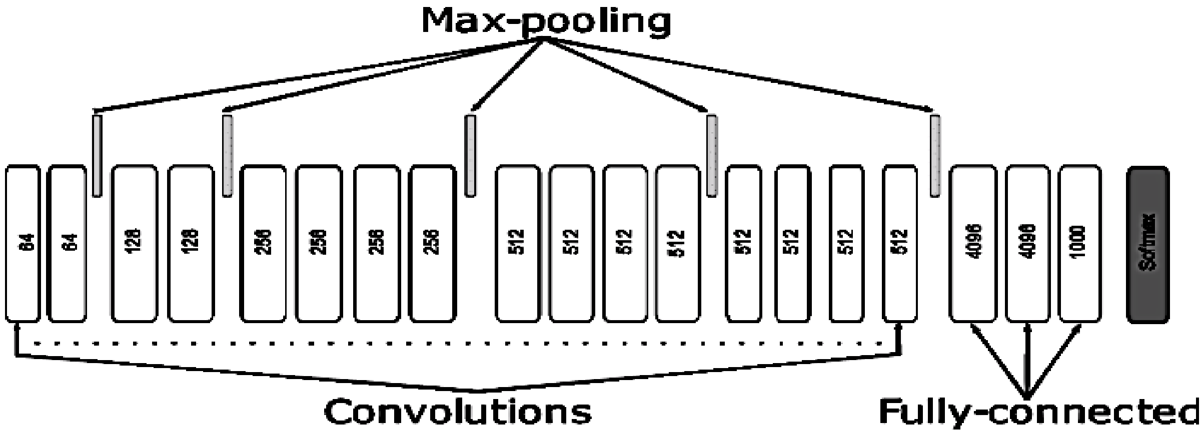


Figure 2.7: VGG-19[19]

Author	**Dataset**	**Method**	**Accuracy**
Alina Maryum[9]	Cassava dataset	EfficientNet model B4	89.09%
Sedat METLEK[11]	Cassava Dataset	ResNet50	85.4%
Sedat METLEK[11]	Cassava Dataset	SVM Classifier	84.4%
A Ramacharan[14]	Cassava dataset	Mobilenet-SSD	80.60%
Mohanty et al.[16]	Plant Village	AlexNet	99.27%
Geetha Ramani et al[18]	Plant Village	CNN	96.46%
chen et al[6]	Maize Plant Village	DenseNet	98.5%
chen et al[6]	Maize Plant Village	VGG-19	92%

Table 2.1: Realated work comparision

2.6.2 VGG-19

The VGG-19 model is a nineteen layer deep model. In this there are sixteen convolution layers and three fully connected layers. There are five Pooling layers. First pooling layer is after two layers, second pooling layer is after four layers, third is after eight layers, fourth id after twelve layers and fifth is after sixteen layers. At last there is a softmax layer and the last three layers before softmax are fully connected layers. As shown in fig 2.7.

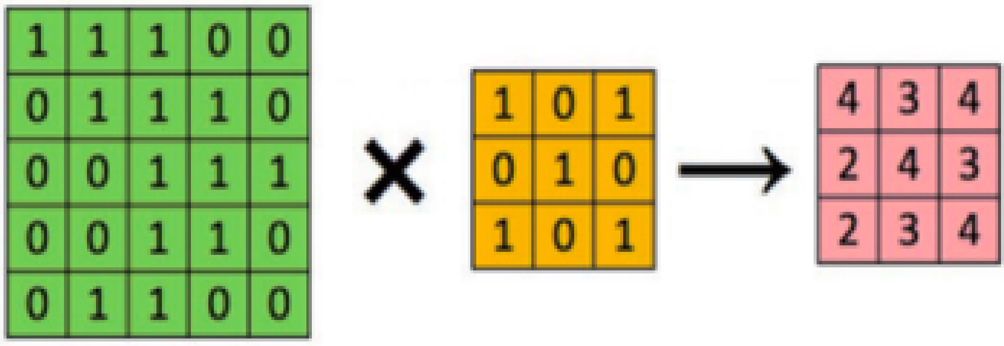
2.7 Overview of Convolution Neural Network

CNNs are frequently utilised in a various of applications, including target recognition, target detection, image recognition, speech recognition, recommendation systems, and image classification. Following their investigation of the cat's visual nerve, Hubel and Wiesel put out the idea of a receptive field and later identified a connection between the processing and categorization of input from external and the visual cortex of the brain. The visual input cells and space that make up the visual cortex as a whole can be duplicated to actualize precise computer vision techniques, they added, which is a key idea in the concept of the sensory field. These notions are then used to generate CNNs, which can be further separated into three categories. Here, first layer is input image, then, for feature extraction, there is convolutional and pooling layer. Then, the third component called joining layer separates and combines the features of the image into a column vector.

2.7.1 Convolutional layer

A convolutional layer has an optimal solution for backpropagation, and together they create a convolution unit. A convolution layer is make up by multiple convolution units. Low-level aspects like edges, lines, and faces of an image, are determined via a convolution operation. A network can bring out more complicated characteristics from low-level information with each additional convolution layer. For example, the first layer uses pixels to extract the information about points and edges, which is then used as the input for the 2nd layer. Squares, circles, and other shapes are identified from the edges and points in the second layer, which serves as the input for the subsequent layers, and so on. This procedure continues until a level of abstraction appropriate for the particular matching task is attained. The total count of convolutions and features to extract can be changed depending on the desired outcome because there are so many characteristics that can be extracted. Figure 1 depicts the convolution of two matrices, with the pink matrix representing the final matrix.

The input image in Fig. 3.1 has a dimension of 5 × 5. The original input space is translated to a 3 x 3 region succeeding convolution along with a convolution core of size 3 x 3. The convolution procedure in this case extracts the input image's features such that



$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 3 & 4 \\ 2 & 4 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

Figure 2.8: Convolution process for CNNs[23]

every pixel can be learned in this manner.

2.7.2 Pooling layer

Through the pre-set convolution layer, which is disposed to overfitting,we acquire abstract features. As a result, it is important to downscale the features dimension in the pooling layer. The average or maximum pooling method reduces the dimension of features by obtaining the average or maximum value in a specific area. This approach can both maintain image details and avoid overfitting. Fig. 3.2 depicts the maximum pooling method. Within convolutional neural networks (CNNs), the pooling layer is applied to downsample the feature maps produced by the preceding convolutional layer. Its purpose is to decrease the number of parameters and computational complexity within the network while also introducing translation invariance to the inputs.[23]

There are two common types of pooling layers Max pooling and average pooling which are utilized in convolutional neural networks (CNNs). These layers are used in reducing the spatial dimensions of feature maps, which aids in decreasing computational complexity and extracting significant features from input data. Max pooling functions by dividing the input feature map into non-overlapping regions, referred to as pooling windows, and selecting the maximum value within each window as the output. By cselecting the maximum value, max pooling retains the most prominent feature within each local neighborhood, thereby preserving crucial information.

In contrast, average pooling calculates the average value within each pooling window

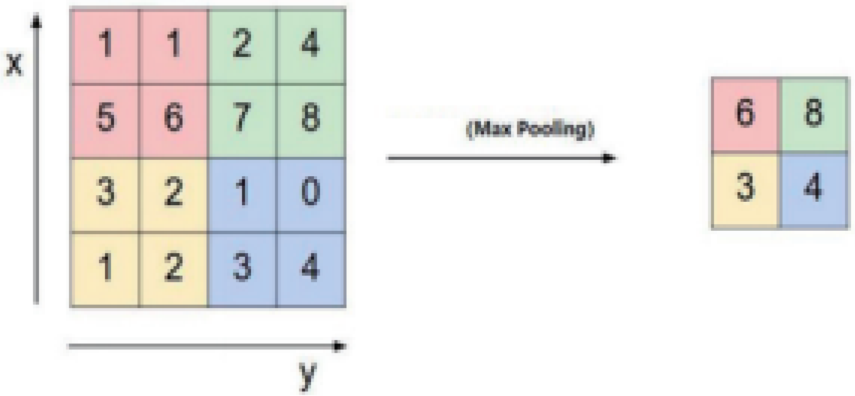


Figure 2.9: Maximum pooling process[23]

and uses it as the output. Instead of selecting the maximum value, average pooling considers the overall intensity of features in a neighborhood. This approach captures a more generalized representation of features and generates a smoother output compared to max pooling. Both the layers are employed to reduce the spatial dimensions of feature maps, downsampling the data and focusing on relevant features. These pooling layers are typically inserted between convolutional layers in CNN architectures, gradually reducing the spatial size of feature maps while retaining vital information. The selection between max pooling and average pooling depends on the specific task and the characteristics of the data being processed.

The pooling layer operates on every feature map independently by using a sliding window approach. The window size and stride are hyperparameters that can be tuned based on the needs of the network architecture. By reducing the resolution of the feature maps, pooling layers help the network learn higher-level features that are more robust to spatial translations and variations in the input data. However, excessive pooling can lead to a loss of information and a decrease in accuracy. Therefore, the design of the pooling layer should be carefully balanced with the other components of the network.

Overall, the pooling layer is a crucial component in CNNs that helps reduce computational costs and increase the network's robustness to spatial variations in the input data.

2.7.3 Fully connected layer

This layer joins every feature in the form of column vector to make two dimensional feature map output by using the first layer. Then, the classifier takes the output values and classifies them into different classes. Fig.3.3 shows the whole process.

Mapping from original data into feature space is done by convolution and pooling layers. It help in increasing the precision of feature matrix and abstraction of original data. Then, we convert feature space into column vector to input it to the classifier. then the next layer help in mapping of pooling layer and convolution output to a linear separable space. This layer can simplify the parameter model, extract high order features from pooling and convolution operations, and decrease the number of training parameters and neurons. The fully connected layer is a critical component in convolutional neural networks (CNNs), often placed at the end of the network. Its purpose is to transform the extracted features from the previous layers into class scores or predictions. Each neuron in the previous layer is connected to every neuron in the output layer, and each connection has its weight parameter.

Before entering the fully connected layer, the input is flattened or reshaped into a vector, which loses the spatial information of the feature maps, and only the learned features are passed through the layer.

The fully connected layer performs a linear transformation on the input features, followed by a non-linear activation function such as ReLU or sigmoid. This helps the network to learn complex and non-linear relationships between the features and the target output.

To optimize the weights of the fully connected layer, the network uses backpropagation during training to reduce maximum error between the predicted output and actual target output.[23] This is basically achieved by using a loss function such as cross-entropy or mean squared error.

The fully connected layer is a crucial component of CNNs that enables the network to transform the learned features into class scores or predictions. Its ability is to learn complex non-linear relationships between the output and the features makes it a valuable tool for a range of applications such as image classification, object detection, and natural language processing.

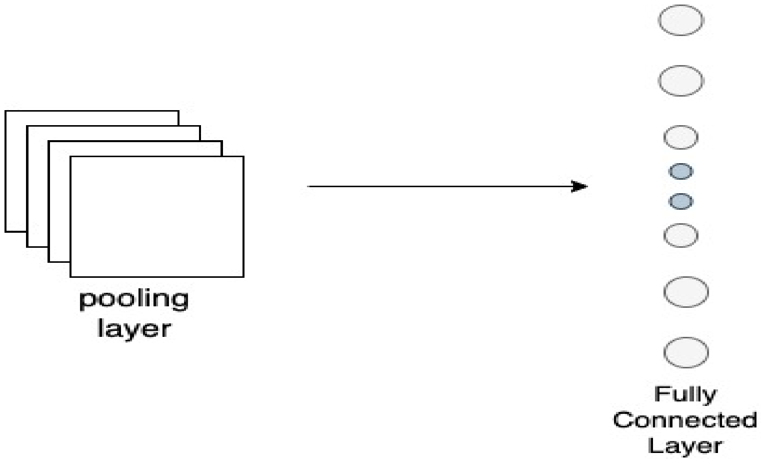


Figure 2.10: Fully connected layer[23]

2.8 Dataset

The dataset which is used in this study is the cassava dataset, collected from Kaggle[10]. It consists images depicting cassava plant leaves exhibiting five different types: healthy leaves, Cassava mosaic disease, Cassava bacterial blight, Cassava brown streak disease, and Cassava green mite. In Figure 9, the bar chart illustrates the distribution of images across these categories: 0 represents the total count of images showing cassava bacterial blight, 1 represents the total count of images depicting Cassava brown streak disease, 2 represents the total count of images displaying Cassava green mite, 3 represents the total count of images representing cassava mosaic disease, and 4 represents the total count of images showing healthy leaves. This information is further depicted in Figure 13.

2.8.1 Healthy leaves

Cassava leaves possess a high nutritional value, being rich in protein, vitamins, and minerals. This is why people in South Africa also consume these leaves. However, it is important to note that cassava is primarily cultivated for its roots, and the leaves are typically regarded as a byproduct. Healthy cassava plant leaves exhibit a green coloration, indicating that the plant is not affected by any diseases and can be safely consumed. This information is illustrated in Figure 5.1.

Figure 2.11: healthy leaf[10]

2.8.2 Cassava Bacterial Blight

This cassava disease was first found in brazil. Then found in different countries where cassava crops are produced. It is the most dangerous disease for cassava plants after the cassava mosaic disease. This causes more damage to crops. Starch in roots are in low quantity because of cassava bacterial blight. The leaves become brownish and yellow as you can see in the fig. It can cause loss of new roots and planting material. Because leaves become dry then leaves start falling due to which they lose leafy vegetables. As shown in fig 5.2.

2.8.3 Cassava Mosaic Disease

It is the too much dangerous and it is most widely spread disease which limits the production of crops in africa. It shows a wide variety of symptoms which are mosaic, mottling, misshapen and twisted leaflets. It also shows that the size of leaves and plants reduces because of cassava mosaic disease.As shown in fig 5.3.

Cassava mosaic begomoviruses cause infection which results in CMD. This virus group is associated with many species and recombinant strains , this form is usually common in East Africa,so this the place for sampling of this plant disease. African cassava mosaic virus in East africa, the species vectored by b.tabaci ,in a strong manner contrasting to CBSIs. A recently infected plant shows its indications from the very top, while plants

Figure 2.12: Cassava Bacterial Blight[10]

which are infected while they were planted cutting show indications in their leafs. CMD indications are a typical mosaic which is a combination of green areas and pale green chlorotic patches. the shape of leaves are distorted unlike CBSD, in severe symptoms , leaves sizes are reduced and the plant is stunted. quantitative declines in yield when stunting and damage chlorophyll in plants occuring from chlorosis .

2.8.4 Cassava Brown Streak Disease

This disease is very dangerous disease for cassava plants and it is widely spread in the east region of africa. Two viruses called ugandan brown streak virus i.e UCBSD and cassava brown streak virus i.e CBSD causes this disease. Both Ugandan brown streak virus i.e UCBSD and cassava brown streak virus i.e CBSD are different genetically but both show the same symptoms on the cassava plants including leaf chlorosis, brown streaks on stems of the plants, and on storage roots brown corky necrosis. As shown in fig 5.4.

Cassava brown streak ipomo viruses causes infection which results in CBSD.(UCBSV) which stands for Ugandan cassava Brown streak virus and (CBSD) which stands for Cassava Brown streak virus are two species which are related to this disease , symptoms are the same . two of these virus species are whiteflies vectored , im semi- persistent manners . a Mottled pattern which is yellow in color is shown by cassava leaves when infected , starting from secondary veins which progresses to tertiary veins in severe infection .

Figure 2.13: Cassava Mosaic Disease[10]

yellow chlorosis spreads along veins making leaves infected very badly. Symptoms of the disease can vary by plant's age and weather conditions . Tolerant varieties and young aged plants may be infected plants and still be asymptomatic . inedible may be render because of the two viruses causes brown streaks on infected plant's stem and necrotic rotting in tuberous roots.

2.8.5 Cassava Green Mite

This is the disease which is caused by the mites. These mites sucks juice from the leaves and resulting which leaves become yellow and dead. Because of which leaves start falling from the plants. This can be detected by checking the color of the plant leaves and if the leaves start falling. This disease spreads from plant to plant. These mites take 12 days from egg to adult and spreads on the nearby plants also. As shown in fig 5.5.

Cassava green mites cause GMD . Cassava green mite is a common pest found in Africa and South America. It causes scratch-like white spots on the leaves where the mites feed. In severe cases, it can result in the entire leaf being covered with patterns. If the infestation leads to a reduction in chlorophyll, the leaves may become stunted, resembling the effects of Cassava Mosaic Disease (CMD). However, the severity of the impact depends on various factors such as environmental conditions and the cassava variety. Infestations by green mites can cause a loss in tuberous root yield of up to 30%.

Figure 2.14: Cassava Brown Streak Disease[10]

Figure 2.15: Cassava Green Mite[10]

2.9 Algorithms Used

2.9.1 EfficientNet-B0

In this we used EfficientNet-B0 which is known as best in terms of classification after the AlexNet and ResNet. As shown in fig 19. So the advantage of using this model is that it is faster than any other model and it can work on high resolution images which is helpful to extract more features from images. It scales on width , depth and resolution which make it different from other models which work on depth only to give good results[23].Then we apply k-fold cross validation on our dataset. Which divides our data into five equal parts and trains model one by one on each. It divides it into 5 parts because we used 5-fold cross validation.Proposed model gives better results than the existing models which are used for cassava dataset. It gives 96.68% accuracy which is better than the other model as you can see in the comparison table 5.1. EfficientNet was created by Google in 2019. It uses compound coefficients to scale models in simple and effective ways. It uses depth wise, width wise and resolution scaling. So if the image is of more resolution then more features can be extracted from the images. So the question is how we find the depth and width for scaling. It gives better results than any other model. It is fast so that it can be used in real life.

$$f = \alpha^{\phi}.\beta^{\phi}.\gamma^{\phi} \tag{2.2}$$

Here α is depth of the network, β is width and γ is resolution. In this value of , and are constants. Values of β, α and γ are 1.1, 1.2 and 1.15 respectively. And the value of ϕ is found out by performing grid search which is also called coefficient. It works on 380x380 images.

2.9.2 k-Fold Cross-Validation

In machine learning, cross-validation is a widely used method for evaluating models by resampling limited data samples. The data samples are divided into groups based on a parameter called 'k,' which determines the number of groups to be formed. This approach is referred to as 'k-fold cross-validation.' For instance, if we choose k=5, we would call it '5-fold cross-validation,' as depicted in Figure 4.2.

In our machine learning model, cross-validation is primarily employed to assess the performance of the model on given data. It leverages small data samples to estimate how well the model will perform when predicting results that were not used during training. This method is preferred due to its simplicity and the fact that it typically yields less biased or more optimistic estimates of the model's skill compared to other methods, such as training and testing.

Moreover, it is crucial to conduct data preparation within the loop on the assigned training dataset rather than on the wider set of data before fitting the model. This principle also applies to tuning hyperparameters. Performing these operations outside the loop can lead to data leakage and an overestimation of the model's skill.

The model skill scores are often summarized by calculating the mean of a k-fold cross-validation run. To provide a measure of variability in the analysis, it is recommended to incorporate additional metrics such as the standard deviation.

Chapter 3

Proposed Model

In our proposed model we used pretrained EfficientNet-b0 with k fold cross validation where k is 5 and the dataset which we used is mentioned in literature Review Why do we choose EfficientNet? The answer for this question is that it is faster and it scale on depth of the model and width and resolution of the images. So if the image resolution is high then we can extract more features from the images and we can classify better. In this we use 5 fold cross validation. So the Error in this is find by the equation.

$$E = \frac{1}{k}\sum_{i=1}^{k} E_i \tag{3.1}$$

In this equation error(E) is equal to the average of the error in each fold. Where E_i is the error in i^{th} fold. EfficientNet was created by Google in 2019. It uses compound coefficients to scale models in simple and effective ways. It uses depth wise, width wise and resolution scaling. So if the image is of more resolution then more features can be extracted from the images. So the question is how we find the depth and width for scaling. It gives better results than any other model. It is fast so that it can be used in real life.

As you can see in equation 1 Here α is depth of the network, β is width and γ is resolution. In this value of , and are constants. Values of β, α and γ are 1.1, 1.2 and 1.15 respectively. And the value of ϕ is found out by performing grid search which is also called coefficient. It works on 380x380 images. EfficientNet are of 8 types from b0-b7. As the number increases then the resolution of the images increases. The EfficientNet B0, EfficientNet B1, EfficientNet B2, EfficientNet B3, EfficientNet B4, EfficientNet B5, EfficientNet B6 and EfficientNet B7 in all these the resolution of the images increases so that it can extract more features and use it to find more feature from the images.[21]

As you can see the values of β, γ and α are in the table 4.1. These values are the stan-

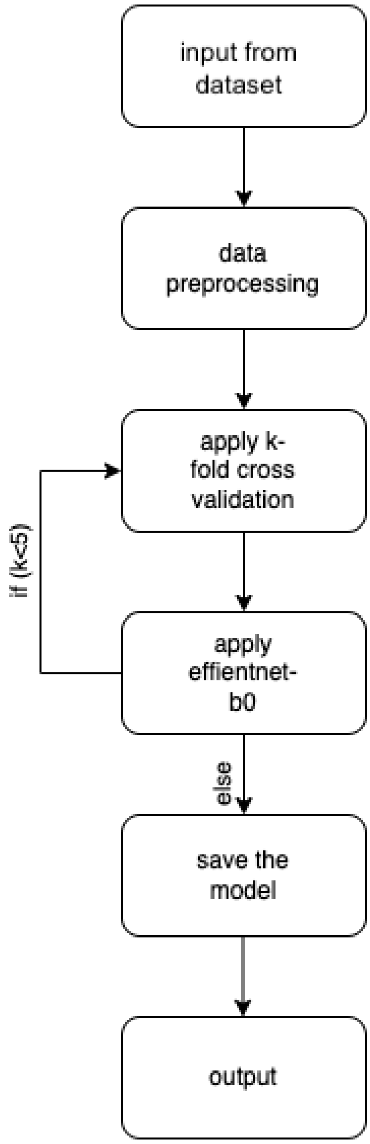


Figure 3.1: Work flow of Proposed Model

Model	Width(β)	Depth(α)	Resolution(γ)
Efficientnet B0	1.0	1.0	224
Efficientnet B1	1.0	1.1	240
Efficientnet B2	1.1	1.2	260
Efficientnet B3	1.2	1.4	300
Efficientnet B4	1.4	1.8	380
Efficientnet B5	1.6	2.2	456
Efficientnet B6	1.8	2.6	528
Efficientnet B7	2.0	3.1	600

Table 3.1: Different efficientnet performance Comparison[19]

dard values but the values are made constant so that it becomes easy to use for everybody. The constant values of β, α and γ are 1.1, 1.2 and 1.15 respectively. In our Model we used the EfficientNet-b0 which works on the images of resolution 224 and uses five fold cross validation in which the data set is splitted into five equal slices which is 4279 images in each part and which gives us the accuracy of 96.68. In machine learning, cross-validation is a method of evaluating models by resampling limited samples of data. Data samples are divided into groups according to a parameter called k indicating the number of groups to be formed. Hence, the procedure is known as k fold cross-validation. Let's say if k=5 is selected, then the reference to the model would be 5 fold cross validation. As shown in fig 4.2. In our machine learning model, cross-validation is used mainly to evaluate skills of machine learning model's on given data. It uses small data sample to estimate how good the model will perform when we predict result which was not used during training[22]. Due to its simplicity and the fact that it usually yields a less biased or more optimistic estimate of the model skill than other methods, such as training and testing, it is widely used.

Steps of K-fold cross validation:

Step 1 : Randomly shuffle the dataset to ensure that the data is mixed up and not ordered in any specific way.

Step 2 : DDivide the shuffled dataset into k segments or parts.

Step 3 : For each segment:

a. Select the segment as the test dataset or hold-out segment, which will be used to evaluate the model's performance.
b. Use all the remaining segments as the training dataset, which will be used to train the model.
c. Train the model using the training dataset and assess its performance by evaluating it on the test dataset.
d. Take note of the score obtained from the performance evaluation and discard the model, as we will repeat this process for each segment.
Step 4 : Analyze the collection of model evaluation scores obtained from all the segments to draw conclusions about the model's skill or performance.

In this procedure, the dataset is first jumbled up to remove any potential biases caused by the order of the data. Then, the dataset is divided into segments, and each segment is used as a test dataset once, while the remaining segments are combined to form the training dataset. This process ensures that the model is evaluated on different subsets of the data. By repeating this process for all segments, we can obtain multiple evaluations of the model's performance. Finally, by analyzing the scores obtained from these evaluations, we can make conclusions about the overall skill or effectiveness of the model.

Furthermore, data preparation must occur on the assigned training dataset within the loop and not on the wider set of data prior to fitting the model. In addition, this applies to tuning hyperparameters as well. If we perform these operation outside the loop then the data leakage and the model skill will be overestimated. Model skill scores are often summarized with the mean of a k-fold cross-validation run. It is also a good practice to including a measure of variance, such as the standard deviation.

Configuration of k : Data samples require careful selection of k values. When k is chosen poorly, the model may have a misrepresentative idea of its skill, like high variance (which can be influenced by the data which is used to fit the model), or highly bias (like overestimation). The following are three common ways to choose k:

- Representative: It is important to choose k such that each train and test group represents the broader dataset statistically.
- k=10: It has been proved experimentally that if we set k as 10 then model will have low bias and modest variance.

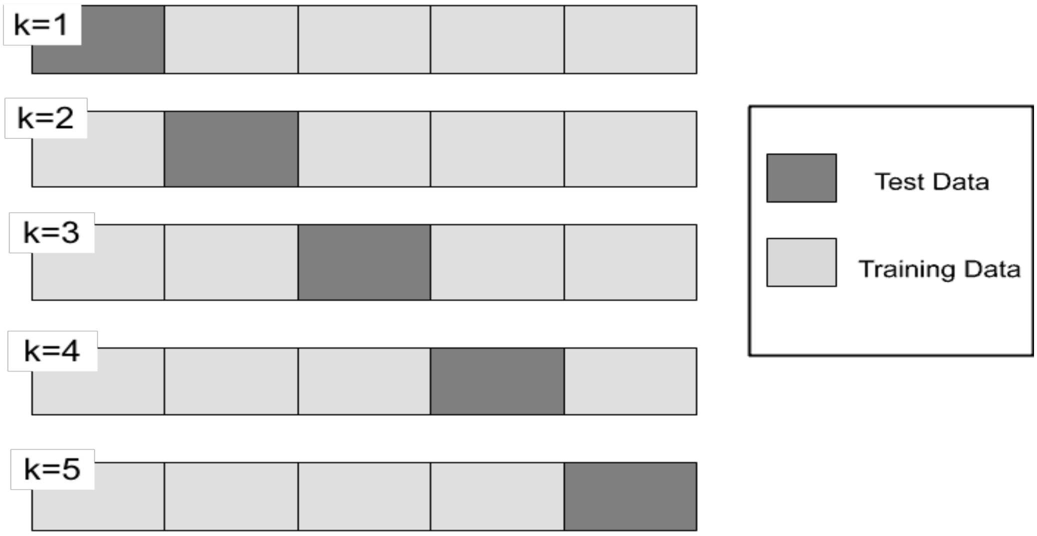


Figure 3.2: k-Fold cross validation

- k=n: For every test sample should be considered in the hold-out dataset, k is permanent at n,where n is define as the number of records in the dataset. It is called cross-validation with one out (leave-one-out).

In the event that k is not chosen at a value that evenly splits the sample, then a remainder of the examples will be included in one group. For model skill scores to be comparable across the k groups, it is preferable to divide the data sample into k equal samples.

Chapter 4

Results and Discussion

In our study, we leveraged the state-of-the-art EfficientNet model in conjunction with K-fold cross-validation to comprehensively evaluate its performance. The results obtained were highly impressive, with an accuracy rate of 96.68%, indicating the model's exceptional ability to classify and make accurate predictions. Beyond accuracy, we also assessed the model's performance using additional metrics such as the F1 score, precision, and recall.

The F1 score, a crucial metric that considers both precision and recall, was measured at an impressive 94.7%. This indicates a strong balance between the precision of the positive predictions and the model's ability to capture the true positive instances. A higher F1 score signifies that the model achieved both high precision in its predictions and effectively identified a significant portion of the true positive instances.

Moreover, the precision achieved by the EfficientNet model was found to be 95.6%. This indicates that the majority of the positive predictions made by the model were indeed relevant, minimizing the occurrence of false positives. A high precision score demonstrates the model's ability to make accurate predictions and minimize incorrect classifications, which is particularly important in tasks where false positives can have significant consequences.

Furthermore, the recall score obtained was 94.34%, highlighting the model's capability to identify a considerable proportion of the true positive instances in the dataset. This indicates that the model successfully captured a high percentage of the actual positive instances, demonstrating its effectiveness in detecting and correctly classifying the target class. A higher recall score is especially important in tasks where false negatives need to be minimized to avoid missing relevant instances.

Author	**Method**	**Accuracy**	**F1 score**	**Precision**	**Recall**
Proposed Model	EfficientNet with k-fold	96.68	94.7	95.06	94.34%
Alina Maryum et al.[9]	EfficientNet	89.09	88.53	90.71	86.45%
Sedat METLEK et al.[11]	ResNet50	85.4	84.3	82.47	86.21%
Sedat METLEK et al.[11]	SVM Classifier	84.4	85.12	88.11	82.32%
A Ramacharan et al.[14]	Mobilenet-SSD	80.60	78.7	76.51	81.01%

Table 4.1: Comparision of Models

The combination of the exceptional accuracy, F1 score, precision, and recall achieved by the EfficientNet model validates its robustness and efficacy in handling the given task. These results underscore the model's ability to extract meaningful features and learn representative patterns from the data. By utilizing K-fold cross-validation, we ensured a thorough evaluation of the model's generalization capabilities, demonstrating its consistent performance across different subsets of the dataset.

In conclusion, the results obtained using EfficientNet with K-fold cross-validation showcase its remarkable performance in accurately classifying and making predictions. The high accuracy, along with the impressive F1 score, precision, and recall, highlights the model's reliability and effectiveness in capturing relevant patterns. These findings solidify EfficientNet as a powerful tool in various machine learning applications, emphasizing its potential to achieve exceptional accuracy and robustness in real-world scenarios.

Proposed model gives better results than the existing models which are used for cassava dataset. It gives 96.68% accuracy which is better than the other model as you can see in the comparison table 5.1.

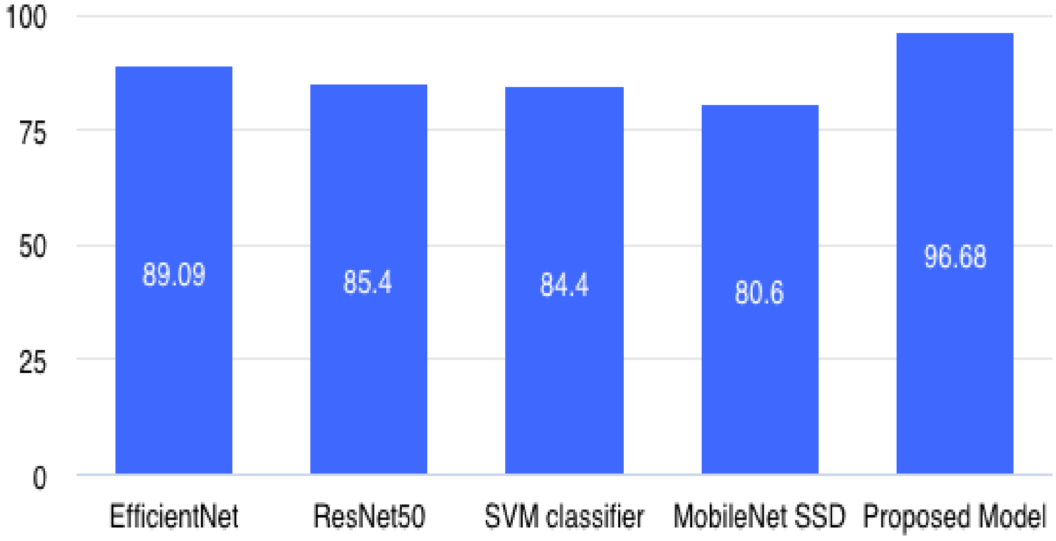


Figure 4.1: Performance Graph

Chapter 5

Conclusion And Future Scope

Plant diseases can have a severe impact on agriculture, leading to reduced crop yields and affecting food security and the economy. Timely detection and diagnosis of plant diseases are crucial for preventing their spread and minimizing crop losses. The use of advanced technologies such as machine learning and computer vision has resulted in the development of automated systems for plant disease detection. These systems can analyze plant images and extract features that indicate the presence of a disease, enabling rapid and accurate identification of infected plants. This allows for targeted interventions such as treatment and removal of diseased plants.

The application of machine learning and computer vision in plant disease detection has significant potential for improving disease management efficiency and accuracy, leading to better crop yields and enhanced food security. Further research and development are necessary to refine and optimize these systems, ensuring their effectiveness across different crops and environments. Plant disease detection is very crucial to prevent the crops from various type of diseases. Chemical testing is not feasible for every plant because it is very expensive and time consuming. So, we detect plant disease by doing plant image classification using deep learning and machine learning approaches. In past years, researchers have made many models such as ResNet, EfficientNet, CNN etc., to detect plant disease. In this paper, proposed model detects Cassava plant disease using EfficientNet with k-fold cross validation with value of k as 5 which outperforms the existing models. In future, we will extend our model to the other crops also and increase the accuracy and we will then integrate our model with the drones. Then with the help of drones we will take the images of plants and find the plants which are affected by disease and apply prevention methods on them by finding which disease causes that plant so that it can become disease free.We will make our model more accurate. So that we can use this model in real life. We will try to increase the accuracy by optimizing the model. This will decrease the time of farmers in finding the plants which are affected by disease and increase the growth of the crop.

References

[1] Lilhore, Umesh Kumar, Agbotiname Lucky Imoize, Cheng-Chi Lee, Sarita Simaiya, Subhendu Kumar Pani, Nitin Goyal, Arun Kumar, and Chun-Ta Li. ”Enhanced convolutional neural network model for cassava leaf disease identification and classification.” Mathematics 10, no. 4 (2022): 580.

[2] Vijayalata, Y., Nikhil Billakanti, Karthik Veeravalli, Ashlin Deepa, and Lokesh Kota. ”Early Detection of Casava Plant Leaf Diseases using EfficientNet-B0.” In 2022 IEEE Delhi Section Conference (DELCON), pp. 1-5. IEEE, 2022.

[3] Falaschetti, Laura, Lorenzo Manoni, Denis Di Leo, Danilo Pau, Valeria Tomaselli, and Claudio Turchetti. ”A CNN-based image detector for plant leaf diseases classification.” HardwareX 12 (2022): e00363.

[4] Al-Hiary, Heba, et al. ”Fast and accurate detection and classification of plant diseases.” International Journal of Computer Applications 17.1 (2011): 31-38.

[5] Sivan, Swathy, K. Arya, M. N. Sheela, B. S. Revathi, and P. V. Abhilash. ”Principal Component Analysis and Clustering of Cassava Germplasm based on N and K Efficiency.” International Journal of Bio-Resource Stress Management 14, no. 1 (2023).

[6] Chen, Junde, Wenhua Wang, Defu Zhang, Adnan Zeb, and Yaser Ahangari Nanehkaran. ”Attention embedded lightweight network for maize disease recognition.” Plant Pathology 70, no. 3 (2021): 630-642.

[7] Adedoja, Adedamola, Pius Adewale Owolawi, and Temitope Mapayi. ”Deep learning based on nasnet for plant disease recognition using leave images.” 2019 international conference on advances in big data, computing and data communication systems (icABCD). IEEE, 2019.

[8] Gao, Fei, Jiming Sa, Zhuoer Wang, and Zhongyu Zhao. ”Cassava disease detection method based on EfficientNet.” In 2021 7th International Conference on Systems and Informatics (ICSAI), pp. 1-6. IEEE, 2021.

[9] Maryum, Alina, Muhammad Usman Akram, and Anum Abdul Salam. ”Cassava leaf disease classification using deep neural networks.” In 2021 IEEE 18th international conference on smart communities: improving quality of life using ICT, IoT and AI (HONET), pp. 32-37. IEEE, 2021.

[10] https://www.kaggle.com/competitions/cassava-disease/data

[11] METLEK, Sedat. ”Disease detection from cassava leaf images with deep learning methods in web environment.” International Journal of 3D Printing Technologies and Digital Industry 5, no. 3 (2021): 625-644.

[12] Ali, Luqman, Fady Alnajjar, Hamad Al Jassmi, Munkhjargal Gocho, Wasif Khan, and M. Adel Serhani. ”Performance evaluation of deep CNN-based crack detection and localization techniques for concrete structures.” Sensors 21, no. 5 (2021): 1688.

[13] Hearst, Marti A., Susan T. Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. ”Support vector machines.” IEEE Intelligent Systems and their applications 13, no. 4 (1998): 18-28.

[14] Ramcharan, Amanda, Peter McCloskey, Kelsee Baranowski, Neema Mbilinyi, Latifa Mrisho, Mathias Ndalahwa, James Legg, and David P. Hughes. ”A mobile-based deep learning model for cassava disease diagnosis.” Frontiers in plant science (2019): 272.

[15] Hughes, David, and Marcel Salathé. ”An open access repository of images on plant health to enable the development of mobile disease diagnostics.” arXiv preprint arXiv:1511.08060 (2015).

[16] Mohanty, Sharada P., David P. Hughes, and Marcel Salathé. ”Using deep learning for image-based plant disease detection.” Frontiers in plant science 7 (2016): 1419.

[17] Alom, Md Zahangir, Tarek M. Taha, Christopher Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Brian C. Van Esesn, Abdul A. S. Awwal, and Vijayan K. Asari. ”The history began from alexnet: A comprehensive survey on deep learning approaches.” arXiv preprint arXiv:1803.01164 (2018).

[18] Geetharamani, G., and Arun Pandian. "Identification of plant leaf diseases using a nine-layer deep convolutional neural network." Computers & Electrical Engineering 76 (2019): 323-338.

[19] Chollet, François. "Xception: Deep learning with depthwise separable convolutions." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1251-1258. 2017.

[20] Chen, Chih-Cheng, Ju Yan Ba, Tie Jun Li, Christopher Chun Ki Chan, Kun Ching Wang, and Zhen Liu. "EfficientNet: A low-bandwidth IoT image sensor framework for cassava leaf disease classification." Sens. Mater 33 (2021): 4031.

[21] Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." In International conference on machine learning, pp. 6105-6114. PMLR, 2019.

[22] Rao, Patike Kiran. "Cassava leaf disease classification using separable convolutions UNet." Turkish Journal of Computer and Mathematics Education (TURCOMAT) 12, no. 7 (2021): 140-145.

[23] Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." In International conference on machine learning, pp. 6105-6114. PMLR, 2019.

[24] Vishakha, B. S., Sharma, N., Bhushan, B., amp; Kaushik, I. (2021). Blockchain-based cultivating ideas for growth: A New AGRONOMICS PERSPECTIVE. Advances in Computing Communications and Informatics, 195–219. https://doi.org/10.2174/9781681088624121010012

[25] Sethi, R., Bhushan, B., Sharma, N., Kumar, R., amp; Kaushik, I. (2020). Applicability of Industrial IoT in Diversified Sectors: Evolution, Applications and Challenges. Studies in Big Data Multimedia Technologies in the Internet of Things Environment, 45-67. doi:10.1007/978-981-15-7965-3_4..

[26] Aggarwal, V. K., Sharma, N., Kaushik, I., Bhushan, B., amp; Himanshu. (2021). Integration of Blockchain and IoT (B-IoT): Architecture, Solutions, amp; Future

Research Direction. IOP Conference Series: Materials Science and Engineering, 1022, 012103.https://doi.org/10.1088/1757-899x/1022/1/012103

List of Publications

P. G. Shambharkar and S. Sharma, ”Plant Disease Detection And Prevention Using Deep Learning,” 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2023, pp. 1832-1838, doi: 10.1109/ICACCS57279.2023.

PAPER NAME

Thesis-5-46.pdf

AUTHOR

Saurabh Sharma

WORD COUNT

8954 Words

CHARACTER COUNT

47927 Characters

PAGE COUNT

42 Pages

FILE SIZE

704.5KB

SUBMISSION DATE

May 30, 2023 3:38 PM GMT+5:30

REPORT DATE

May 30, 2023 3:39 PM GMT+5:30

● 6% Overall Similarity

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

- 4% Internet database
- Crossref database
- 4% Submitted Works database
- 2% Publications database
- Crossref Posted Content database

● Excluded from Similarity Report

- Bibliographic material
- Cited material
- Manually excluded sources
- Quoted material
- Small Matches (Less then 10 words)