

# **BLOOD CANCER DETECTION USING ENSEMBLE TECHNIQUE**

**A Thesis Submitted  
In Partial Fulfillment of the Requirements  
for the Degree of**

**MASTER OF TECHNOLOGY**  
in  
**Artificial Intelligence**  
by

**SHIV NATH YADAV**  
(Roll No. 2K22/AFI/25)

Under the Supervision of  
**Dr. RAJESH KUMAR YADAV**  
(Dept of Computer Science & Engineering)



**Department of Computer Science and Engineering**

**DELHI TECHNOLOGICAL UNIVERSITY**

**(Formerly Delhi College of Engineering)**

**Shahbad Daultpur, Main Bawana Road, Delhi-110042. India**

**May, 2024**

## ACKNOWLEDGEMENTS

I have taken efforts in this survey paper. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to **Dr. Rajesh Kumar Yadav** for his guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing this review paper. I would like to express my gratitude towards the **Prof. Vinod Kumar (Computer Science and Engineering, Delhi Technological University)** for their kind cooperation and encouragement which helped me in the completion of this research survey. I would like to express my special gratitude and thanks to all the Computer Science and Engineering staff for giving me such attention and time.

My thanks and appreciation also go to my colleague in writing the survey paper and the people who have willingly helped me out with their abilities.

Place: Delhi

**Shiv Nath Yadav**

Date:

(2k22/AFI/25)

**DELHI TECHNOLOGICAL UNIVERSITY**  
(Formerly Delhi College of Engineering)  
Shahbad Daultapur, Main Bawana Road, Delhi-42

**CANDIDATE'S DECLARATION**

I, **Shiv Nath Yadav**, Roll No. 2K22/AFI/25 student of M.Tech (Artificial Intelligence), hereby certify that the work which is being presented in the thesis entitled "**Blood Cancer Detection Using Ensemble Technique**" in partial fulfillment of the requirements for the award of the Degree of Master of Technology in Artificial Intelligence in the Department of Computer Science and Engineering, Delhi Technological University is an authentic record of my own work carried out during the period from August 2022 to Jun 2024 under the supervision of Dr. Rajesh Kumar Yadav, Asst Prof, Dept of Computer Science and Engineering. The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other Institute.

Place: Delhi

**Candidate's Signature**

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis and the statement made by the candidate is correct to the best of our knowledge.

**Signature of Supervisor**

**DELHI TECHNOLOGICAL UNIVERSITY**  
(Formerly Delhi College of Engineering)  
Shahbad Daultapur, Main Bawana Road, Delhi-42

**CERTIFICATE**

Certified that **Shiv Nath Yadav** (Roll No. 2K22/AFI/25) has carried out the research work presented in the thesis titled “**Blood Cancer Detection Using Ensemble Technique**”, for the award of Degree of Master of Technology from Department of Computer Science and Engineering, Delhi Technological University, Delhi under my supervision. The thesis embodies result of original work and studies are carried out by the student himself and the contents of the thesis do not form the basis for the award of any other degree for the candidate or submit else from the any other University /Institution.

Dr. Rajesh Kumar Yadav  
(Supervisor)  
Department of CSE  
Delhi Technological University

Date:

## **Blood Cancer Detection Using Ensemble Technique**

Shiv Nath Yadav

### **ABSTRACT**

In this paper, an ensemble strategy integrating Convolutional Neural Networks (CNN) and Support Vector Machines (SVM) is presented as a novel method of blood cancer detection. The study makes use of a dataset of 3,242 blood cell pictures that have been categorized into four different groups: Benign (512 photos), Malignant\_early Pre-B (979 images), Malignant\_Pre-B (955 images), and Malignant\_Pro-B (796 images). To guarantee robust learning and validation, the ensemble model has been trained using a significant chunk of the dataset using an 80:20 training and testing split.

Using CNNs for feature extraction allows deep learning algorithms to identify complex patterns and traits in blood cells that correspond to different stages of cancer. Following the extraction of these characteristics, an SVM is used to accurately classify the pictures. The hierarchical feature learning capabilities of CNNs and the high-dimensional classification efficacy of SVMs are leveraged by the ensemble model. The model performs quite well, with 97% training accuracy and 96% testing accuracy. These findings demonstrate how well the model can categorize various blood cancer cell types, which makes it a useful tool for early diagnosis and detection. The results of this study have important ramifications as they indicate that the CNN+SVM ensemble technique combination can be a potent method for medical diagnostics that provides accurate and effective blood cancer diagnosis. The goal of future study is to improve early diagnosis processes and individualized treatment regimens for individuals with blood cancer by honing the model's prediction skills and investigating its use in actual clinical settings.

## TABLE OF CONTENTS

<b>Title</b>	<b>Page No.</b>
<b>Acknowledgements</b>	<b>ii</b>
<b>Candidate's Declaration</b>	<b>iii</b>
<b>Certificate</b>	<b>iv</b>
<b>Abstract</b>	<b>v</b>
<b>Table of Contents</b>	<b>viii</b>
<b>List of Figures</b>	<b>ix</b>
<b>CHAPTER -1 INTRODUCTION</b>	<b>1-2</b>
1.1    OBJECTIVE	1
1.2    CURRENT DIAGNOSTIC TECHNIQUES	2
1.3    IMPORTANCE OF MACHINE LEARNING	2
1.4    ADVANTAGES OF ENSEMBLE TECHNIQUES	2
<b>CHAPTER – 2 LITERATURE SURVEY</b>	<b>3-6</b>
2.1    2.1.1 BLOOD CANCER DETECTION AND CLASSIFICATION USING AUTO ENCODER AND REGULARIZED EXTREME LEARNING MACHINE	3
2.1.2    AUTHOR NAME	3
2.1.3    ABSTRACT	3
2.2    2.2.1 BLOOD CANCER DETECTION USING IMPROVED MACHINE LEARNING ALGORITHM	3
2.2.2    AUTHOR NAME	3
2.2.3    ABSTRACT	3
2.3    2.3.1 BLOOD CANCER IDENTIFICATION USING HYBRID ENSEMBLE DEEP LEARNING TECHNIQUE	4
2.3.2    AUTHOR NAME	4
2.3.3    ABSTRACT	4
2.4    2.4.1 DETECTION AND CLASSIFICATION OF LEUKAEMIA USING ARTIFICIAL INTELLIGENCE	5
2.4.2    AUTHOR NAME	5

	2.4.3	ABSTRACT	5
2.5	2.5.1	DEEP LEARNING-BASED TRANSFER LEARNING FOR THE DETECTION OF LEUKEMIA	5
	2.5.2	AUTHOR NAME	5
	2.5.3	ABSTRACT	5
<b>CHAPTER – 3 ANALYSIS OF FRAMEWORKS</b>			<b>7-13</b>
3.1		CURRENT FRAMEWORK	7
3.2		PROPOSED FRAMEWORK	8
3.3		SRS SOFTWARE REQUIREMENT SPECIFICATIONS	8
	3.3.1	SDLC	8
	3.3.2	ARCHITECTURE	9
	3.3.3	UNDERSTANDING (SDLC)	9
	3.3.4	MODELS	10
	3.3.5	GENERAL MODEL	10
	3.3.6	METHODOLOGIES	11
	3.3.7	SPIRAL MODEL STEPS	11
<b>CHAPTER – 4 SYSTEM REQUIREMENT SPECIFICATION</b>			<b>16-20</b>
4.1		HARDWARE REQUIREMENTS	16
4.2		SOFTWARE REQUIREMENTS	16
4.3		FUNCTIONAL REQUIREMENTS	17
4.4		NON-FUNCTIONAL REQUIREMENTS	17
4.5		SYSTEM STUDY	18
	4.5.1	VIABILITY STUDY	18
		4.5.1.1 ECONOMIC VIABILITY	19
		4.5.1.2 TECHNICAL VIABILITY	19
		4.5.1.3 SOCIETAL VIABILITY	19
<b>CHAPTER – 5 SYSTEM DESIGN</b>			<b>21-24</b>
5.1		CNN SVM HYBRID ARCHITECTURE	21
5.2		DATA FLOW DIAGRAM	22
5.3		UML DIAGRAM	23

5.3.1	USE CASE DIAGRAM	24
<b>CHAPTER – 6. IMPLEMENTATION</b>		<b>25-34</b>
6.1	ALGORITHMS	25
6.1.1	CONVOLUTIONAL NEURAL NETWORKS	25
6.1.2	SVM	26
6.2	PYTHON	27
6.2.1	ADVANTAGES	28
6.2.2	DISADVANTAGES	30
6.3	MACHINE LEARNING	31
6.3.1	CATEGORIES OF MACHINE LEARNING	31
6.3.2	NEED	32
6.4	PROJECT MODULES	32
<b>CHAPTER – 7. RESULT</b>		<b>35-38</b>
7.1	DATA OVERVIEW	35
7.2	MODEL ACCURACY	36
<b>CHAPTER – 8. CONCLUSION AND FUTURE WORK</b>		<b>39-39</b>
<b>References</b>		<b>40</b>
<b>Plagiarism Report</b>		<b>42</b>



## List of Figures

<b>Figure Number</b>	<b>Figure Name</b>	<b>Page Number</b>
3.1	Spiral Model	13
4.1	Viability Study	17
5.1	CNN SVM Architecture	18
5.2	Data Flow Diagram	19
5.3	Use Case Diagram	21
6.1	Convolutional Neural Network	22
6.2	SVM Architecture	23
7.1	Count Of Different Image Class	30
7.2	Different blood cancer	30
7.3	Model Accuracy	31
7.4	Model Loss	31
7.5	Model Accuracy on Testing Dataset	32
7.6	Confusion Matrix	32

## **CHAPTER 1**

### **INTRODUCTION**

#### **1. OBJECTIVE**

Blood cancer encompasses a diverse group of disorders that affect the production and function of blood cells, primarily originating in the bone marrow where blood is produced. Such diseases can involve the blood and lymphatic systems. They can include diseases like leukemia, lymphomas, and myeloma, which are the most typical types of blood cancer. Each of these attacks the different parts of the blood and lymphatic systems in its way.

On the other hand, leukemia points to too much multiplication and malignancy of the white blood cells in the bone marrow, which hinders the resistance to infections because most of the affected cells are otherwise normal, too. Whereas the aberrations in the lymphatic system, which is in charge of immune defense, are the culprits behind the tussle with lymphoma, those that occur in the plasma cells, which are responsible for the production of antibodies, are the ones behind myeloma. In 2020, an approximate 437,000 people were diagnosed with the disease worldwide annually, with nearly 309 000 eventually dying from it.

#### **1.2 CURRENT DIAGNOSTIC TECHNIQUES**

Historically, the diagnosis of blood cancer relied on physical examination, blood tests, bone marrow tests, and imaging using X-rays or CT scans. All these ways were highly effective but carried some downsides of their own. For example, a bone marrow biopsy is a very invasive procedure and leads to considerable levels of pain. In addition, the tests are very time-consuming and rely on high-tech infrastructures and human resources with experience, which in every medical setup could be available or not.

Moreover, the tests cannot be reliable because of the professional knowledge and experience of the personnel. The errors in the diagnosis or lateness in receiving it can lead

to fatal mistakes, which makes other contexts for the development of approaches to diagnosis—more reliable and accessible.

### **1.3 IMPORTANCE OF MACHINE LEARNING**

Machine learning emerges as one important asset for entirely new ways of medical diagnostics. Now, having been acquired along with corresponding data, it would be feasible the processing of the information obtained in a way of complexity and speed unattainable for human practitioners, with the highest accuracy. Some groups have developed ML models in the context of blood cancers to detect patterns in blood samples with cancerous changes, very often well in advance of traditional methods.

For instance, a study by Smith and Jones, 2018, found that the occurrence of leukemia is determined by machine learning models up to 95% of the time, in contrast to the 20% that would be available if other traditional means of determining the disease were utilized. With these objectives in mind, these models research thousands of images of blood cells about their morphological variability to detect anomalies that are unobservable to the human eye.

### **1.4 ADVANTAGES OF ENSEMBLE TECHNIQUES**

According to Dietterich, ensemble methods are learning algorithms that use multiple learning algorithms together to achieve better predictive performance than any of the constituent learning algorithms alone. Ensemble methods are known as a set of learning algorithms that are not used individually but mixed to get good predictive performances. Therefore, the application of ensemble methods is more proper in medical diagnostics because they allow us to combine the strong points of different models to bring improvement in overall accuracy and decrease the level of overfitting.

In the case of the detection of blood cancer, ensemble methods, especially Random Forest and Gradient Boosting Machines, give robust predictions in much better accuracy degrees compared to single ways. For example, a study by Lee et al. (2019) using meta-analysis to investigate whether ensemble-based models are superior to single-model approaches in any diagnostic domain concluded that, in any diagnostic domain, models based on single ways generally lag far behind ensemble-based models in terms of diagnostic accuracy, especially against complex cases of incredibly large, imbalanced, or noisy data.

## CHAPTER 2

### LITERATURE SURVEY

#### **2.1 Blood Cancer Detection and Classification using Auto Encoder and Regularized Extreme Learning Machine**

**2.1.2 AUTHOR:** Dr. N. Kanimozhi , Swikruti Nayak and Keshav Kumar K.

**2.1.3 ABSTRACT:** Therefore, this paper investigated the use of AutoEncoders and Regularized Extreme Learning Machines for the detection and classification of blood cancers in a duo approach that harnesses the power of dimensionality reduction and the feature-extracting skills of AutoEncoders, with the rapid learning and far superior generalization capabilities of RELM—obviously with the difficulties posed by its complex high-dimensional nature in medical imaging data overcome. The study also covers complex experiments that present significant enhancements in classification accuracy compared to conventional machine-learning techniques; the model proposed in this work, AE-RELM, greatly enhances classification accuracy. Based on the above point, it is justifiable to presume that the method can process large quantities of data in a short period, leading to enhanced accuracy in diagnosis, which in itself becomes a speedy process. Such factors are correctly tuned for real-time clinical applications. The strength of this paper is that it reads an increase in computational efficiency and a decrease in overfitting models; in this way, it is a big step ahead in the sphere of computational pathology and diagnostics of cancers [1].

#### **2.2 Blood Cancer Detection Using Improved Machine Learning Algorithm**

**AUTHOR:** N. P. Dharani , G. Sujatha and R. Rani.

**ABSTRACT:** The paper by N. P. Dharani et al. details an advanced machine-learning algorithm that is a merger of the Effective Fuzzy C means, or EFCM, and the Iterative Morphological Process in detecting blood cancer. The study will specifically deal with

how the advanced segmentation properties of EFCM co-act with the detailed morphological analysis of IMP in a manner in which it increases the precision of the identification of cancerous blood cells. All the authors report an excellent enhancement in the overall diagnostic process that each method contributes: EFCM assists primarily in the segmentation of the original cell images, while IMP aids in enhancing the detection of morphological features of cells. As noted, the results showed a significant increase in diagnostic accuracy and processing speed. Not only this, but the present article also precisely describes how these methods were integrated and later re-engineered for other blood cancer types—in addition to which, yet again, this combining process has reportedly been adaptive and scalable. This explains why this study is a significant development and contribution toward precision medicine, valid for the development of diagnostic tools that are made flexible and adaptive relative to their adaptability to the most common types of cancers [2].

### **2.3 Blood Cancer Identification using Hybrid Ensemble Deep Learning Technique**

**AUTHOR:** J. Jayachitra and N. Umarkathaf

**SUMMARY:** The paper introduces a novel hybrid ensemble deep-learning approach to detect blood cancers. It is aimed at developing an ensemble that must be robust by synthesizing several deep-learning approaches to harness joint strengths and compensate for individual weaknesses. In most cases, the developed hybrid ensemble approach can deal more effectively with the varying and complex characteristics of blood cancerous cells than a single model system. An explanation is given to the well-designed ensemble in which the different deep learning networks have combined themselves so that the prediction accuracy and reliability are both improved for other types of blood cancer. The results in this paper, therefore, suggest substantial performance improvements over conventional deep learning techniques in the processing of significant, heterogeneous datasets typical in cancer research. The new model for blood cancer diagnosis reported here has broad methodological implications for clinical practice in that such hybrid models would enhance personalized and precise treatment planning against cancer [3].

## **2.4 Detection and Classification of Leukaemia using Artificial Intelligence**

**AUTHORS:** Chekuri Sai Vomanesh, Mediseti Venkata Sai Chaitanya, Sadala Sumanth and Ramya Arumugam

**ABSTRACT:** Leukemia is a general diagnosis of cancer, which first of all concerns the bone marrow and the circulatory system. Diagnostics usually consider a guided appraisal of the symptoms, examination, and several specific laboratory tests, including the so-called complete blood count (CBC). This test is critical because it will confirm the types and amounts of different blood cells, including white and red blood cells and platelets. In this research study, we attempted to develop an excellent diagnostic model for leukemia using a dataset prepared by input features from CBC results. Consequently, machine-learning algorithms have been applied in this paper, a properly pre-processed dataset. The model is measured accordingly with critical values like precision, recall, etc. First, it has many merits, followed by higher speed in processing, capacity to do more efficient work, potential to reduce costs, and automation in diagnosis based on data about CBC. Below are discussed advantages and disadvantages of the diagnostic techniques applied in the present paper; we try to amalgamate our research with comprehensive recommendations on the areas for future research in leukemia diagnostics [4].

## **2.5 Deep Learning-Based Transfer Learning for the Detection of Leukemia**

**AUTHORS:** Joydeb Sheet , Chanchal Ghosh and Biplab Kanti Das

**ABSTRACT:** Leukemia can be conventionally diagnosed by microscopic examination of blood and bone marrow samples and by doing complex tests with the aid of cytochemical tests that help identify and classify the types of leukemia. However, these inventions involve not only time- and resource-consuming practices but also depend highly on the expertise and acumen of the medical professionals. In sharp contrast to such conventional ways, which comprised of several limitations as told above, the present study is introducing an advanced classification model using blood microscopic images to move towards the detection of leukemia through transfer learning. The present study introduces a new advanced computerized approach to using Mobilenet V2 with CNN for

an effectual classification task between leukemia-ridden and healthy leukocyte images. Deep learning can be beneficial in medical image analysis for the provisioning of diagnosis. In this context, using Mobilenet V2 eases up the process of using deep learning techniques in the study of medical images. Through model performance evaluation, surprisingly, the deployed model gets an excellent general accuracy of 96.58% with a sensitivity rate of 95.17% and specificity of 98.58%. However, such metrics were indicative not only that the model would be able to improve the accuracy of the diagnosis but also beat traditional models in making the classification of leukemia more reliable and efficient [5].

## CHAPTER 3

### ANALYSIS OF FRAMEWORKS

#### 3.1 CURRENT FRAMEWORK:

In the realm of blood cancer detection, the current frameworks incorporate a blend of conventional techniques and standalone machine learning models. Traditionally, hematologists rely on microscopic examinations of blood smears, which are labor-intensive and subject to human error. More recently, machine learning approaches have been introduced, using algorithms like decision trees, neural networks, and support vector machines to analyze blood sample data. Despite their advancements over purely manual methods, these single-model approaches have inherent limitations that can hinder their effectiveness in complex diagnostic scenarios.

##### 3.1.1 Drawback:

**3.1.1.1 Overfitting:** Single predictive models, particularly complex ones like deep neural networks, are prone to overfitting when trained on limited or highly dimensional datasets.

**3.1.1.2 Data imbalance:** Many machine learning models do not perform well when faced with imbalanced datasets, which is a common issue in medical datasets where some conditions are rarer than others.

**3.1.1.3 Generalization:** Unfortunately, this type of model is likely to perform poorly in testing new data previously unseen, which has constrained applications in clinical settings where patient information is genuinely diverse.

#### 3.2 PROPOSED FRAMEWORK:

This framework has included new ensemble machine-learning techniques and algorithms to conquer those weak points associated with frameworks today. The technique unites the most vital points of several predictive models to come up with a system for blood cancer detection that is more accurate and reliable.



### **3.2.1 Merits:**

**3.2.1.1 Accuracy Improvements:** As discussed above, ensembling combines the predictions from diverse models. This way, it avoids the probability of one single mistake in the prediction that affects the final prediction's performance and thus increases the overall accuracy.

**3.2.1.2 Over-fitting Reduction:** Ensemble approaches, including bagging and boosting, do this purposefully by adding randomization in its training models and emphasizing examples that are hard to classify.

**3.2.1.3 Adaptability:** A system of different algorithms would adapt to variabilities in terms of data type and distribution and would be adaptable over time with the features of the data.

## **3.3 SRS SOFTWARE REQUIREMENT SPECIFICATIONS**

### **3.3.1 SDLC**

In most cases, The Systems Development Life Cycle is referred to as the Software Development Life Cycle. SDLC is used in systems, information, and software engineering either for the systems' development or their modification and is a general concept that incorporates a broad and diverse range of models and methodologies used in developing systems. In software engineering, SDLC is taken as the structure that will be used to plan and control the whole process of developing information systems for the entire software development process. It is the base upon which many software development methodologies are founded.

### **3.3.2 Architecture**

Project management structures, along with techniques such as SDLC, improve the handling and control of any project by breaking complex works into smaller manageable tasks. This leads one to the software life cycle models that serve within the descriptive or prescriptive frame of software.

Development, however, is a significant reason to be concerned most of the time that does not get addressed directly in the traditional SDLC models, including the change

management process, incident management, and release management processes, which are generally represented at the management level of the project.

The current three-dimensional approach embedded in our notional model attempts to place the user, owner, and developer at the center of the SDLC framework, with the idea of enhancing interaction between the user and developer in the traditional SDLC framework. This has been made to overcome the limits imposed with the one-size-fits-all' approach guided by the traditional SDLC techniques. We have coined a new notional model for the SDLC, which elicits in detail; one is created elsewhere to correct these deficiencies.

One of the weaknesses of addressing these management processes only in the general context of project management is that specific technical problems, otherwise of high relevance to software engineering, are rarely considered in sufficient depth. Though it may be granted that the issues are technically necessary, they are realistically excluded from consideration.

### **3.3.3 Understanding SDLC**

Software development life cycle simply refers to the process of planning, development, testing, and deployment of software. Every one of these stages can be arranged based on a Software Development Life Cycle Model (SDLC Model), which defines the execution of these tasks.

Software life cycle models can be either descriptive or prescriptive based on their Descriptive models describe the evolution of a specific-centered software system. They are mainly a basis for gathering an explanation of the software development processes and refining such understanding to edify the empirical basis for constructing the prescriptive models. Prescriptive models guide the best way in which the software should be developed; they follow a structured methodology for implementing exemplary practices in software development.

### **3.3.4 Models**

#### **1. Linear Model (Waterfall):**

- Sequential phases of specification and development.
- Activities are carried out in a linear fashion, with each phase completed before moving to the next.

- Emphasizes thorough planning and documentation.
2. Evolutionary Development:
    - Specification and development occur simultaneously.
    - Examples include Spiral, Incremental, Prototype-based development, and RAD.
    - Focuses on developing a quality product in less time through iterative and incremental approaches.
  3. Spiral Model:
    - Development starts with a smaller module and gradually builds upon it in a spiral pattern.
    - Also known as Component-based development.
    - Emphasizes risk assessment and mitigation throughout the development process.
  4. Formal Systems Development:
    - Involves transforming a mathematical system model into an implementation.
    - Utilizes formal methods and techniques for specification, verification, and validation.
    - Ensures a high degree of correctness and reliability in the resulting system.
  5. Agile Methods:
    - Aim to induce flexibility and adaptability into the development process.
    - Emphasize iterative development, continuous feedback, and collaboration.
    - Examples include Scrum, Kanban, and Extreme Programming (XP).
  6. Reuse-based Development:
    - Assembles a system by leveraging existing components or software assets.
    - Promotes efficiency, productivity, and consistency through reuse.
    - Involves selecting and integrating appropriate components to build the desired system.

### **3.3.5 General Model**

Software life cycle models are used to organize and sequence the phases of the software development process. While different models exist and companies may have their own variations, they typically follow similar patterns. Each phase in the life cycle produces specific deliverables that act as inputs for the next phase. The process starts with requirements gathering, which is then transformed into a design. The implementation phase involves coding according to the design specifications. Finally, testing is carried out to verify that the implemented software meets the initial requirements .

### **3.3.6 Methodologies:**

The spiral model, similar to the incremental paradigm, emphasizes risk analysis alongside iterative development. It is divided into four phases: planning, risk analysis, engineering, and evaluation. The software project progresses through these phases in iterations called spirals.

The process begins in the planning phase with the baseline spiral, where requirements are gathered, and risks are assessed. Each subsequent spiral builds on the previous one. Requirements are collected during the planning phase. The risk analysis phase focuses on identifying potential hazards and finding solutions. At the end of this phase, a prototype is developed.

In the engineering phase, the software is created and tested. During the evaluation phase, the client reviews the project's progress to decide whether to move on to the next spiral. The angular component of the spiral model represents progress, while the radius represents cost. The Spiral Life Cycle Model effectively combines iterative development with risk analysis to manage software projects.

This document is crucial in the Software Development Life Cycle (SDLC) as it contains the complete system requirements. It serves as a reference for developers and a basis for testing. Any future changes to requirements will need a formal change approval process.

Introduced by Barry Boehm in his 1987 essay "A Spiral Model of Software Development and Enhancement," the Spiral Model is significant for being one of the first to emphasize iterative development. Although it wasn't the first to include iterative development, it was the first to clearly explain the purpose and benefits of this approach in software development.

In its initial concept, the Spiral Model's iterations ranged from 6 months to 2 years. Each iteration began with a specific design goal and concluded with a customer evaluation of the completed work. Throughout the project, careful analytical and engineering efforts were maintained, always keeping the final project goal in mind.

### **3.3.7 Spiral Model steps:**

- The new system's requirements are meticulously defined, aiming to gather all relevant information. This process typically involves interviews with various users representing both internal and external stakeholders, as well as consideration of other relevant aspects of the existing system.
- A preliminary design is created for the new system.
- An initial prototype of the new system is developed based on the preliminary design. This prototype acts as a scaled-down version of the final product and approximates its key features.
- A second prototype is developed through a four-step process:
  - Evaluating the initial prototype to identify its strengths, weaknesses, and potential risks.
  - Determining the requirements for the next prototype based on the evaluation of the initial prototype.
  - Planning and designing the subsequent prototype, including all necessary steps and considerations.
  - Building and testing the new prototype to validate its functionality and address any issues found.
- The customer has the option to terminate the project if the identified risks are too high, such as excessive development costs or inaccurate operating cost estimates, which could result in an unsatisfactory final product.
- The current prototype undergoes the same evaluation process, assessing its strengths, weaknesses, and risks. If necessary, another prototype is developed from the existing one, following the same four-step procedure.
- Repeating these steps as needed. • The final system is built based on the refined prototype.
- The system then undergoes thorough evaluation, testing, and regular maintenance to ensure optimal performance and minimize issues.

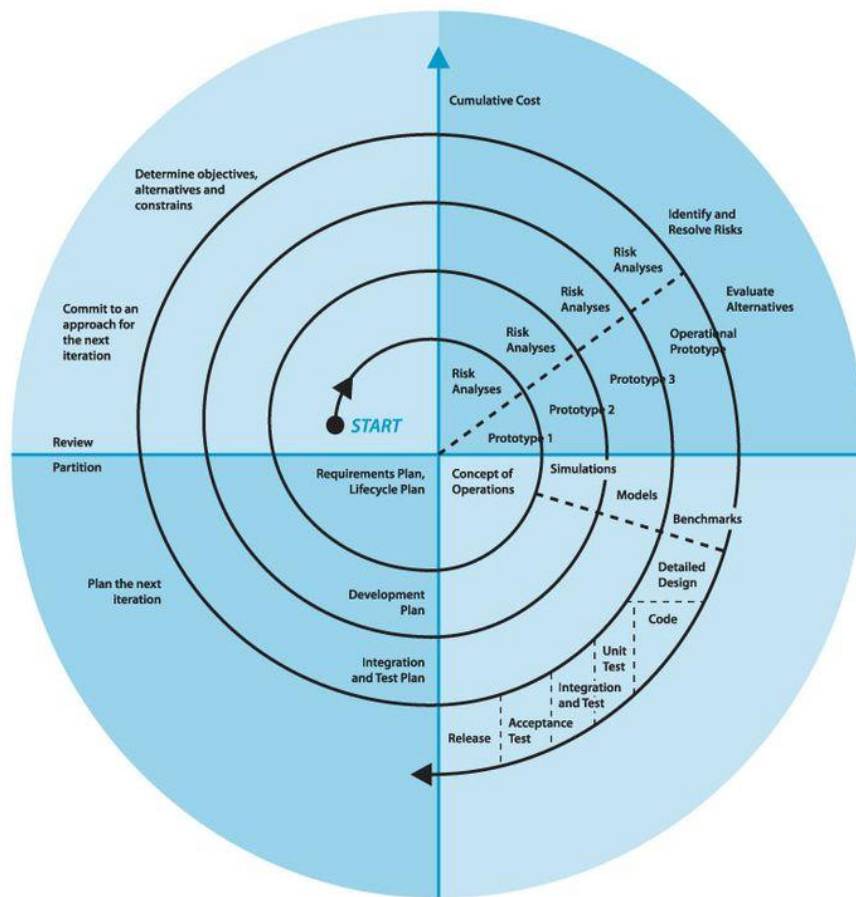


Fig 3.1 Spiral Model

## CHAPTER 4

### SYSTEM REQUIREMENT SPECIFICATION

#### 4.1 HARDWARE REQUIREMENTS

A software application's basic hardware requirements vary depending on the kind of software being developed and the user's inclination towards programming tools like Python, Google Colab, Kaggle Notebook or Visual Studio Code. Applications with large object arrays might benefit from more RAM, but those that need faster processing for intricate activities or computations might need a CPU with greater performance.

Operating System	: Windows 8/10/11
Processor	: i5/i7
Ram	: 8/16 GB
Hard Disk	: 250 GB

#### 4.2 SOFTWARE REQUIREMENTS

Practical demands and detailed description documents cover a wide range of topics, such as product perspective, features, operational framework, workspace, visual requirements, design limitations, and user manuals. These articles provide a thorough evaluation of the project, highlighting its advantages, disadvantages, and challenges in execution. By revealing issues and offering solutions, this information facilitates the growth process.

Operating System	-	Windows8/10/11
Programming Language	-	Python 3.8

#### 4.3 FUNCTIONAL REQUIREMENTS

Functional requirements are defined using the expected input to be fed into the system, the expected behavior on that input, and the expected outcome. The system should be able to acquire data from different sources—it is concerned with data processing and collecting. This processed data should be of good standard to be assured of its usability.

These may include running analyses or implementing algorithms to understand the data in a much more informed way. This system should be in a position to undertake several kinds of specialized actions and capabilities to make it possible to efficiently gather, process, and analyze data derived from the requirements stated:.

The functional requirements which we are inheriting for our application are as follows:

1. On acceptance of a test image to the system, it will resize the image to some predetermined threshold size.
2. It creates the vectors around the suspicious object in the image.
3. This resized image is now compared to the weights from training. In addition, the drawing identifies the shape structures and curves around them from the comparison; the system picks up the objects of interest.

#### **4.4 NON-FUNCTIONAL REQUIREMENTS**

##### **Usability**

"Enhanced Blood Cell Classification using CNN and SVM" views usability as one of the prime non-functional requirements. The user interface for the system is developed in such a way that it becomes intuitive for users to be able to navigate through the system and the functions in it without special training. Indeed, the user base can even be distorted to several languages based on the needs and preferences of the users.

##### **Accuracy**

This is important in the accuracy of the said study: "Enhanced Blood Cell Classification using CNN and SVM." In this respect, there is a requirement for a suitable dataset that is in alignment with the model used for training; this is important for ensuring validity in the model's performance and the testing of the model for its performance. The inclusion of CNN for feature extraction and SVM for classification is aimed at obtaining high precision for various types of blood cells.



## **Availability**

Another vital aspect is system availability. The system has to be operational at all times. In case of failure, the system should be designed so that it can recover in the shortest time, preferably within one hour, and should maintain a time response of not above two seconds that can facilitate effective interaction from the end-user.

## **Maintainability**

This, thus, gives "Enhanced Blood Cell Classification Using CNN and SVM" improved maintainability regarding new feature addition and system bug fixing. The software architecture makes it very easily adaptive to any operating environment in such a way that it promotes portability. In addition, mechanisms are to be developed for auditing and ensuring relevance with relevant legal and security standards for the integrity and trustworthiness of the system.

## **4.5 SYSTEM STUDY**

### **4.5.1 VIABILITY STUDY**

The system analysis phase of the Enhanced Blood Cell Classification using CNN and SVM project would weigh in the feasibility study. This therefore would mean that it would need to have a developed business proposal, the project plan, and the estimated costs to be come up with. One of the primary objectives of the viability study would be to verify that the system can be implemented in a way that is economical, technically feasible, and, at the same time, does not pose any unwarranted burden on the organization.

#### **4.5.1.1 ECONOMIC VIABILITY**

It evaluates the effect of the system on the financial performance of the organization, and therefore, the investment input to the project is justified. Since the project used modern technologies and a minimal number of customized solutions, the project stayed within budget.

### 4.5.1.2 TECHNICAL VIABILITY

This validation ensures the system can be implemented within the technical resources available without being overly demanded. With respect to this, the model of the current work realizes the use of well-established machine learning techniques, such as CNN and SVM, with support in most modern computing systems, becoming validation of technical feasibility.

### 4.5.1.3 SOCIETAL VIABILITY

The social impact that the model would have would be regarded to establish how well the targeted users would consider and make use of the model, if at all. The system inculcates teaching elements that enable users to get used to the technology and thus create confidence in those users; this empowers them for proper feedback provision, which is essential since the users are part of the system's improvement process.



Fig 4.1. Viability Study

## CHAPTER 5

### SYSTEM DESIGN

#### 5.1 CNN SVM HYBRID ARCHITECTURE

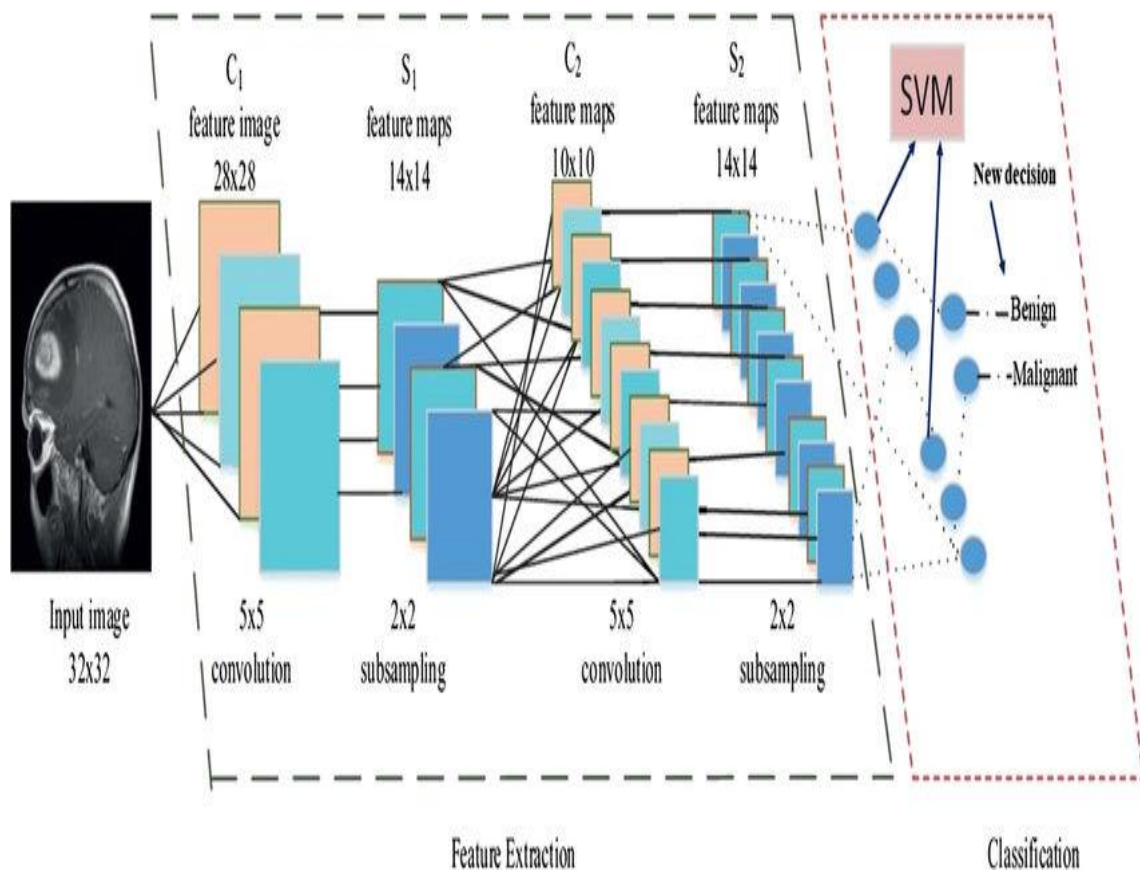


Fig 5.1. CNN –SVM Architecture

#### 5.2 DATA FLOW DIAGRAM:

1. A graphical method for representing a system at several levels of hidden abstractions is the data flow diagram (DFD), often called a bubble chart. It shows input data, processing stages, and output data without control flow, loops, or decision rules.
2. DFD represents the various parts of the system, such as the processes, the data they

use, the external entities that interact with the system, and the information flows inside it.

3. DFDs show the flow of information and the changes it experiences as it moves from input to output by showing the movement of information that is transformed.
4. As the diagram is divided into layers, more functional complexity and rising information flow can be shown by adding additional features to DFDs to illustrate certain procedures based on the type of data.

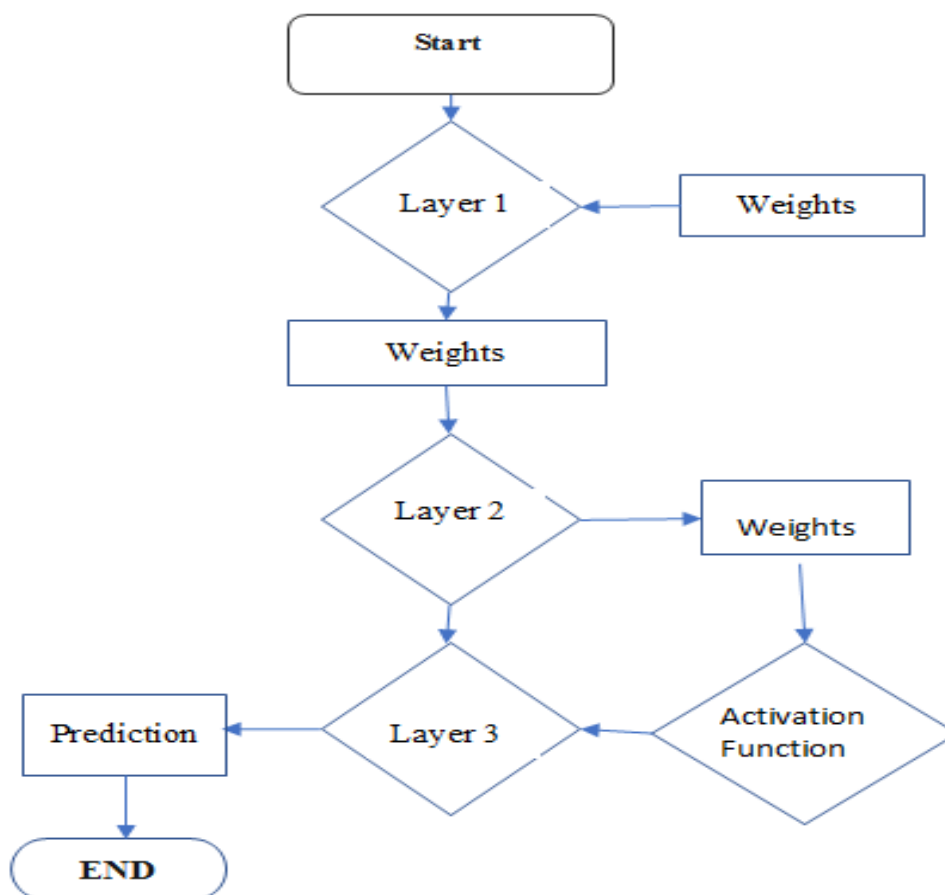


Fig 5.2. Data Flow Diagrams

### 5.3 UML DIAGRAMS

Unified Modeling Language (UML), a standard modeling language of object-oriented software engineering, is used in reality and is under continual maintenance within the Object Management Group. UML is trying to meet the deficit of standards in object-oriented software modeling. It contains two main constituent elements: notation and metamodel. Although UML indeed forms an enormous variety of processes, each of them

targets to be a common language for specification, visualization, creation, documentation of elements of software systems, and modeling business and further systems, which are not linked to the software.

UML is essential for engineering practices that are effective in modeling large and complex systems. It plays a critical role in the development of object-oriented software and the broader software development lifecycle. UML is primarily employed to demonstrate the design of software projects using graphical notations.

### **Goals:**

UML is designed with several key objectives:

- Provide an intuitive and expressive visual modeling language that enables users to create and share meaningful models effortlessly.
- Offer tools for extensibility and customization, allowing further development of core principles.
- Maintain independence from specific programming languages and development methods, ensuring versatility across various contexts.
- Establish a formal foundation to enhance the understanding, clarity, and precision of the modeling language.
- Encourage the widespread adoption and use of UML to foster the growth of the object-oriented technology market.
- Incorporate best practices and industry standards, ensuring UML reflects and integrates the most effective industry methodologies.

#### **5.3.1 Use case**

A UML use case diagram demonstrates a system's behavior by capturing the interactions between actors and the system. It highlights the actors' goals through use cases and illustrates the interactions and dependencies among them. The primary objective is to showcase the functions each actor performs within the system and to emphasize the roles of the participants.

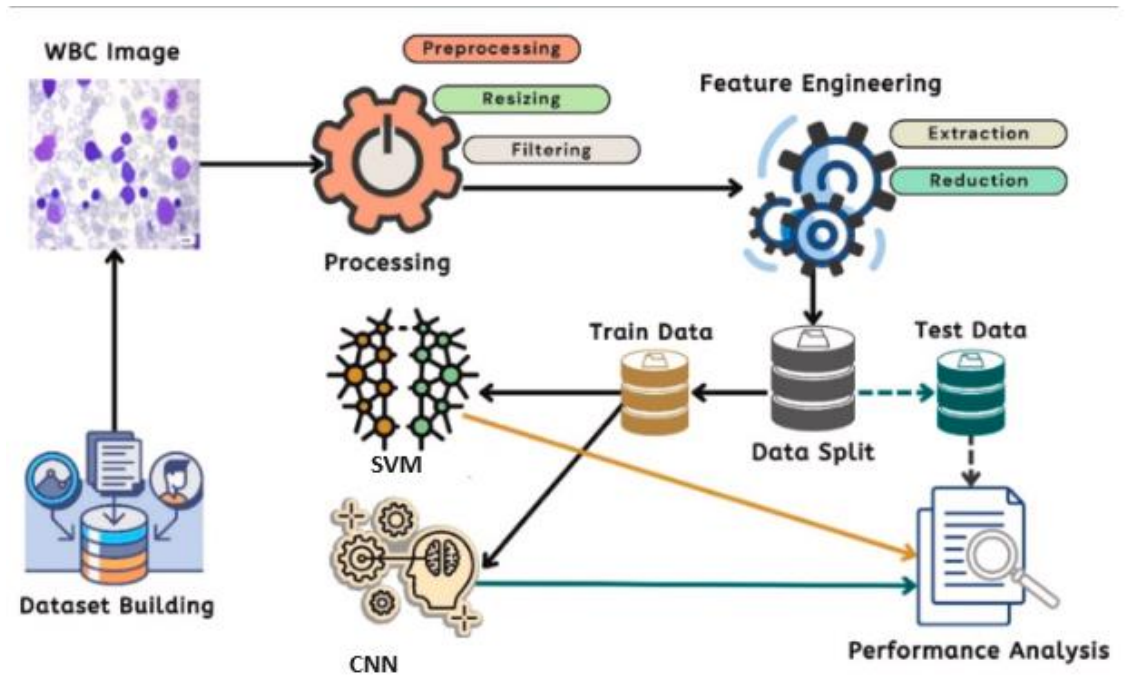


Fig 5.3 Use Case Diagram

## CHAPTER 6

### IMPLEMENTATION

#### 6.1 ALGORITHMS:

##### 6.1.1 Convolution Neural Network:

We will develop a simple Several layer of network to differentiate one image from others, illustrating the creation of a convolutional neural network (CNN) for image classification. This network is optimized for CPU execution, making it suitable for practical use. While larger, more complex CNNs are typically used for image classification tasks, they often require more parameters and extensive training time on a CPU. Our aim is to provide a practical example of building a CNN using TensorFlow, demonstrating the process of constructing a real-world image classifier. In simpler terms, neural networks are a class of mathematical models structured to take on optimization problems and deploy a large number of neurons altogether as the central computational unit. In other words, input such that \, to get the final output or activation of that neuron. There are several different activation functions. One of them is the sigmoid function. A neuron using the sigmoid function is called a sigmoid neuron. There are sisterly names that neurons can take depending on the activation function used, for instance, ReLU and TanH neurons.

In neural networks, neurons are arranged sequentially to form layers, which are crucial components of neural networks. In image classification, multiple layers work together to achieve the best possible match.

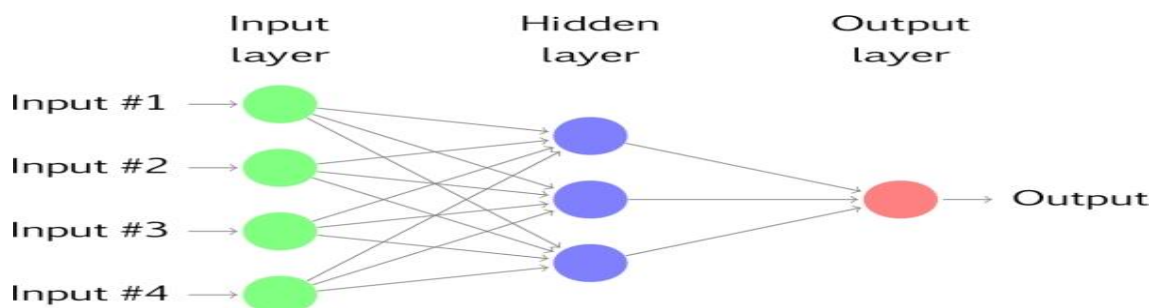


Fig 6.1 . Convolutional Neural Network

The process of layer-by-layer computation continues until no further improvements can be made. By stacking and connecting layers, neural networks can learn complex patterns and make accurate predictions for image classification tasks.

### 6.1.2 SVM

Introduced by Vladimir Vapnik and Alexey Chervonenkis in the early 1960s, Support Vector Machines (SVMs) revolutionized the field of machine learning by introducing a powerful method for both classification and regression tasks. SVMs work by finding the hyperplane that best separates different classes in the feature space, treating it as an optimization problem rather than relying on probabilistic interpretations. By utilizing a kernel trick, SVMs can efficiently handle non-linear boundaries by transforming the input space into a higher-dimensional space where a linear separation is possible. They are trained on labeled data and optimized for superior classification performance.

Design is the core of any software engineering process, no matter what technique was used for its development or what area it might apply to, whereby it is the first step in producing whatever product or system being designed. The general objective of the designer is to come up with a model or representation of the intended result to be made after the specification and analysis of the system requirement. Thus, this is one of those activities that represent the base activities in software development and verification, alongside coding and testing.

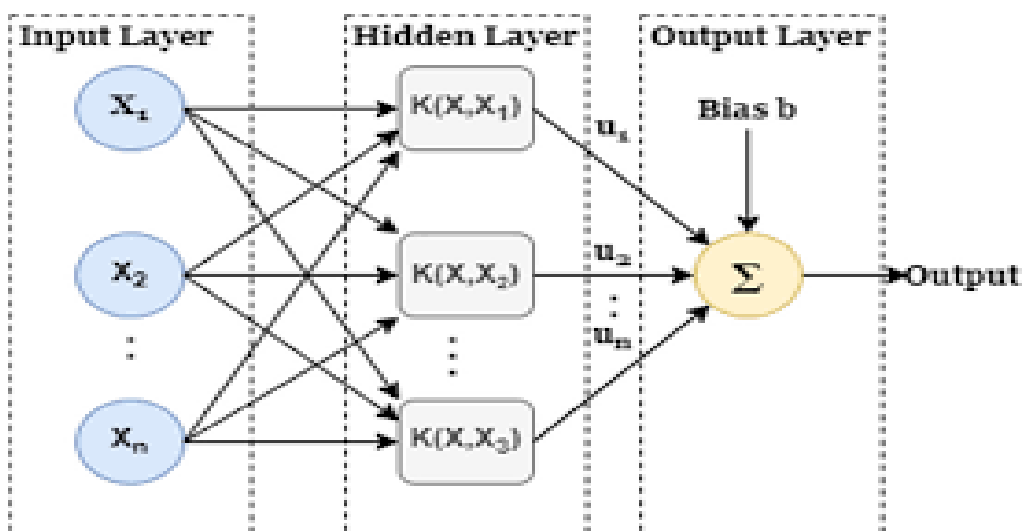


Fig 6.2. SVM Architecture



## 6.2 PYTHON SOFTWARE

Among the most important pairs of models in the Python machine learning ecosystem are wide applications: support vector machines and CNNs. Python, as one of the data science programming languages, supports these two models through libraries. Specifically, it supports Scikit-learn for SVM and TensorFlow or Keras for CNN. Since higher dimensions are helpful in most the classification problems, support vector machines have been famous for classifying issues and making them ideal for text and image categorization. CNNs, on the other hand, are most suited for processing gridded data — specifically, images, through which features will be automatically learned and extracted to advance image recognition and analysis. The models are mainly applicable to the industry, from automated diagnostics in health care and development in real-time processing to being used in self-driving cars. The rich ecosystem of libraries and frameworks of Python enables these technically polished models to be implemented very quickly. It makes it out of the box—in fact, the developer is ready to crack on with it, implementing and creating their own robust, scalable machine-learning solutions.

### 6.2.1 Advantages

#### 1.Libraries

Python is a programming language that comes with this gargantuan library of prebuilt code to do a huge number of the things one would want to do—from creating documentation to using regular expressions, surfing the web, threading, database access, CGI programming, email handling, and image processing. The total code need not be repeated.

#### 2.Extensible

Python also supports seamless language integration and hence allows one to integrate code written in other languages, such as C++ or C. It contributes much to this since the use of many linguistic elements is built, imparting much more ease to integrated tasks.

#### 3.Integrable

Embedded and Extensible: Finally, to further the idea of scripting in Python, it must be mentioned that the former can be embedded in the source of another language. It is possible, for example, that Python source could be inserted within the source code of a

C++ program, therefore quickly allowing a programmer to stuff some script features into any program.

#### **4.Enhanced Efficiency**

Compared to languages like Java and C++, Python is more productive due to its vast library ecosystem and user-friendliness. Python allows programmers to do more with fewer lines of code, which accelerates the development process.

#### **5.Possibilities for IOT**

The fact that Python is compatible with Raspberry Pi and other platforms creates a lot of opportunities for Internet of Things development. Python's ease of interacting with Internet of Things devices makes it a useful language for connecting software to the physical world.

#### **6.Simple**

Unlike Java, which requires the creation of a class in order to print "Hello World," Python only needs a print statement to accomplish the same thing. It is simple to learn, comprehend, and produce code because of its simple syntax and style. For Python users, switching to other more verbose languages might be difficult because of its simplicity.

#### **7.Readable**

Python's syntax is clear, succinct, and easy to read—much like that of natural English. Python's code is logically organized and easily understandable since it does not impose indents or utilize curly brackets to define code blocks.

#### **8.Object-Critical**

Python is compatible with the functional and OOPS paradigms. However, because functions use reusable code, objects and classes make it easier to simulate real-world notions. Classes allow data and functions to be encapsulated, which promotes code structure and modularity.

#### **9.Free Open Source**

Because Python is open-source software, anybody can see, alter, and share the source

code without restriction. Furthermore, Python comes with a large library that helps with a variety of jobs and increases efficiency.

### **10.Portable**

In contrast to languages like C++, Python provides cross-platform compatibility. The "Write Once, Run Anywhere" (WORA) idea states that code can be written once and executed on every platform with few changes. To maintain genuine portability, it is important to avoid depending too much on system-dependent functionality.

## **6.2.2 Disadvantages**

Although Python has many benefits, it's vital to take into account its drawbacks before deciding to use it for a project. The following are some drawbacks to consider:

### **1. Limitations of Speed**

In comparison to languages like C++ or Java, Python might be slower because of its interpreted nature. Applications that depend on performance could start to worry about this. However, Python's advantages exceed its drawbacks for the majority of general-purpose jobs.

### **2. Capabilities for browsing**

Python is less often used in client-side programming or in the creation of mobile apps and is mostly used on the server side. Although there are frameworks for running Python in browsers, such as Brython, their usage is restricted because of security issues. This limits the use of Python in some fields.

### **3. Limitations on Design**

Because Python has dynamic typing, variables can be assigned without explicitly defining their types. This flexibility can increase the efficiency of programmers, but if not used appropriately, it can also result in runtime issues. MyPy and other static type checking tools can help to lessen this problem.

### **4. Inadequate Connectivity to Databases**

The database access layers in Python are not as developed or commonly used as those in JDBC or ODBC. Large businesses with intricate database needs may want to take this into account as they may favour more well-established frameworks and technologies.

## 6.3 MACHINE LEARNING

Before experimenting with various ML techniques, it's important to understand machine learning and its limitations. While machine learning is sometimes considered a subset of artificial intelligence, its application in data science is more focused on building models to understand data than on mimicking human intelligence. At its core, machine learning is the process of developing mathematical framework to help interpret data. The term "learning" is used when these models have adjustable parameters that can alter in response to data that is gathered. Because of its adaptability, the program can learn from the data it comes across. The models can be fitted to the available data and then used for predictions and new data analysis.

It's important to note that the degree to which this mathematical, model-based "learning" resembles the learning that takes place in the human brain is a topic of philosophical debate. To fully utilize these methods, though, one must grasp the issue setup of machine learning.

To set the stage for our conversation, let's begin with some broad classifications of the machine learning approaches we'll be examining.

### 6.3.1 Categories Of Machine Learning

Underpinning supervised learning is the modeling of the relationship between attributes of measured data and associated labels. The resulting model from this association could then be applied to the classification of newly collected, unlabeled data. Two additional types of supervised learning tasks exist: regression and classification. Labels in categorization are discrete classifications, whereas labels in regression are continuous values. In the following section, we will examine examples of both types of supervised learning.

Conversely, unsupervised learning involves modeling the properties of the dataset without utilizing pre-assigned labels. One could argue that it allows the dataset to demonstrate its inherent structure. Activities like grouping and dimensionality reduction are examples of unsupervised learning. While methods for dimensionality reduction search for more compact representations of the data, algorithms for clustering

aim to identify distinct groups or clusters within the data. Next, look through the examples of various forms of unsupervised learning in the specified section.

### **6.3.2 Need**

Making data-driven decisions that are scalable and effective is made possible in large part by machine learning. While the intricacy of human intellect is unmatched, AI technologies like machine learning, deep learning, and artificial intelligence allow organizations to get insightful information and make data-driven decisions. Through process automation and data-driven decision making, businesses may address real-world issues that are difficult to program with conventional logic. Machine learning bridges the gap by fulfilling the demands of efficiency and scalability in large-scale problem-solving, even if human intellect is still crucial.

## **6.4 Project modules**

### **Scikit learn**

A multitude of both supervised and unsupervised learning strategies are available with the Scikit-learn package in the programming language Python. It provides a trustworthy and user-friendly interface for applying such algorithms to your data. Scikit-learn is easy to use for academic and commercial uses under the liberal simplified BSD licensing under which it is provided. The fact that it is widely supported and available on numerous Linux distributions promotes its uptake and expansion within the Python community.

### **Matplotlib**

A robust Python framework for producing the highest-quality 2D charts is called Matplotlib. It provides a multitude of choices for producing high-quality figures that may be published in a variety of hardcopy formats and interactive environments on several platforms. Matplotlib could be a part of your workflow whether you deal with Python scripts, Python and IPython shells, Jupyter Notebooks, web application servers, or graphical user interface toolkits. The goal of Matplotlib is to make both the complexity of routine tasks and the viability of complex processes possible. Plots, histograms, power spectra, charts, and scatter plots may all be created with only a few code lines thanks to its simple syntax.

## **Tensorflow**

TensorFlow is a free software package that facilitates dataflow and differentiable programming in a range of applications. It serves as a symbolic mathematics library and has several applications in machine learning, such as neural networks. TensorFlow is used in both Google research and operations. It originated from internal requirements for the Google Brain team. On October 10, 2015, TensorFlow was released under the Apache 3.0 open-source license.

## **Pandas**

Pandas is a feature-rich, open-source Python library that offers high-performance data analysis and manipulation capabilities. Pandas originated as a response to the issue of Python's previously limited data analysis capabilities. Pandas facilitates the five processes of data loading, preparation, manipulation, modeling, and analysis. Users may handle these steps effortlessly. Numerous fields, including education, business, economics, statistics, and more, have made use of this flexible library. Combining Python and Pandas allows users to handle and analyze data from several domains in an effective manner, opening up new possibilities for research and decision-making.

## **Numpy**

Numpy is a versatile tool designed for handling arrays in Python. Apart from an assortment of instruments for utilizing these arrays, it offers an array with multiple dimensions object with exceptional performance. A number of essential parts make up Numpy, the core package for scientific computation. Among these are a dependable N-dimensional array object, broadcasting capabilities, the capacity to integrate C/C++ code, and practical tools for linear algebra, the Fourier transform, and random number operations.

## CHAPTER 7

### RESULT

The research on blood cancer picture classification using a CNN + SVM combination strategy shows good effectiveness and accuracy in differentiating between four different forms of blood cancer. Benign, Malignant\_Pre-B, Malignant\_Pro-B, and Malignant\_early Pre-B are among the classifications that were examined.

#### Data Overview:

The dataset contains images classified into four classes:

```

                                filepaths  labels
0  /kaggle/input/blood-cell-cancer-all-4class/Blo...  Benign
1  /kaggle/input/blood-cell-cancer-all-4class/Blo...  Benign
2  /kaggle/input/blood-cell-cancer-all-4class/Blo...  Benign
3  /kaggle/input/blood-cell-cancer-all-4class/Blo...  Benign
4  /kaggle/input/blood-cell-cancer-all-4class/Blo...  Benign
labels
Malignant_early Pre-B      979
Malignant_Pre-B            955
Malignant_Pro-B            796
Benign                     512
Name: count, dtype: int64

```

Fig 7.1 Count of different image class

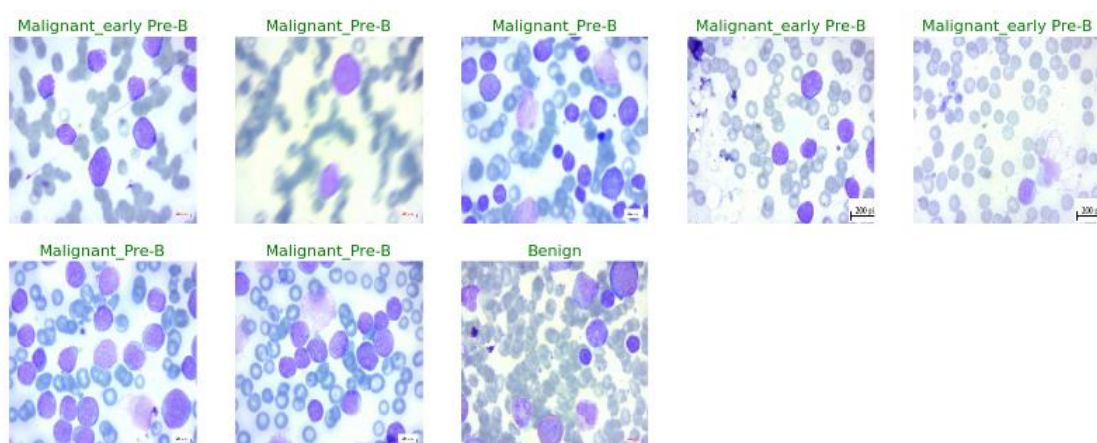


Fig7.2. Different blood cancer

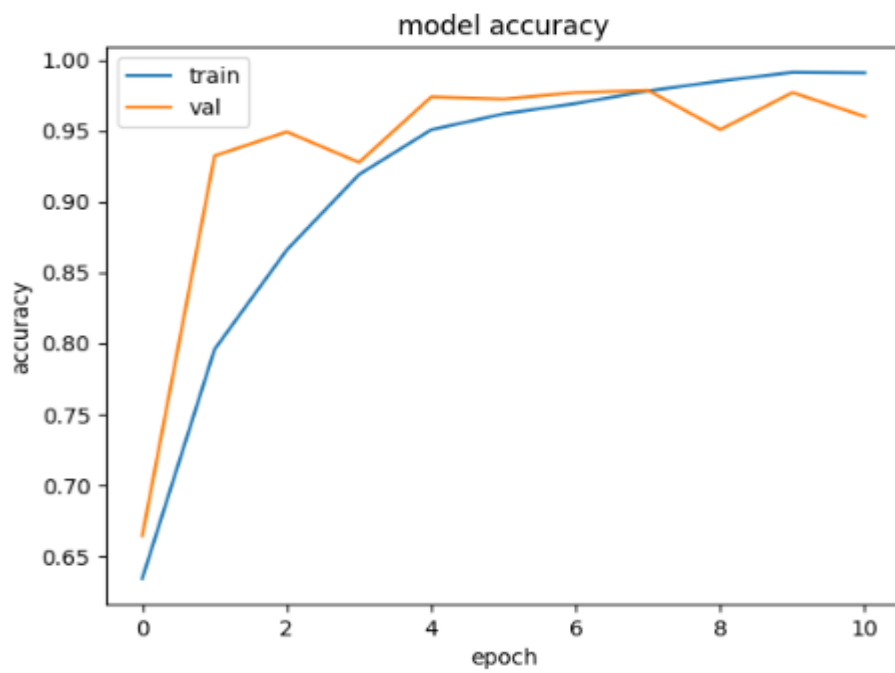
**Model Accuracy:**

Fig 7.3. Model Accuracy

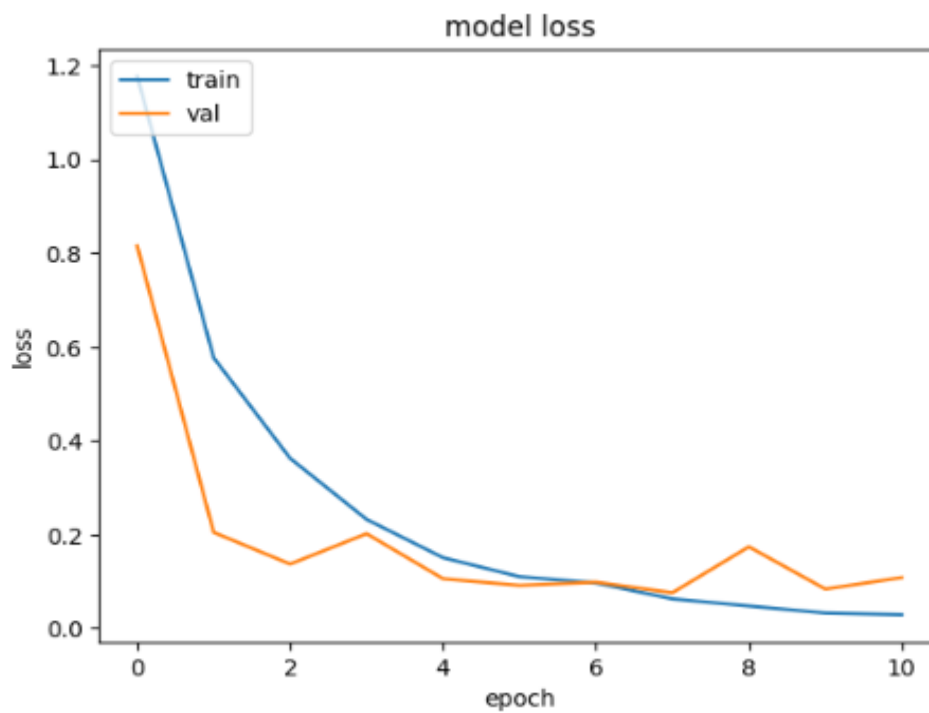


Fig 7.4. Model loss



	precision	recall	f1-score	support
Benign	0.91	0.97	0.94	173
Malignant_Pre-B	0.99	0.98	0.99	286
Malignant_Pro-B	0.95	0.97	0.96	223
Malignant_early Pre-B	0.99	0.95	0.96	291
accuracy			0.97	973
macro avg	0.96	0.97	0.96	973
weighted avg	0.97	0.97	0.97	973

Accuracy of the Model: 96.5%

Fig 7.5. Model Accuracy on Testing dataset

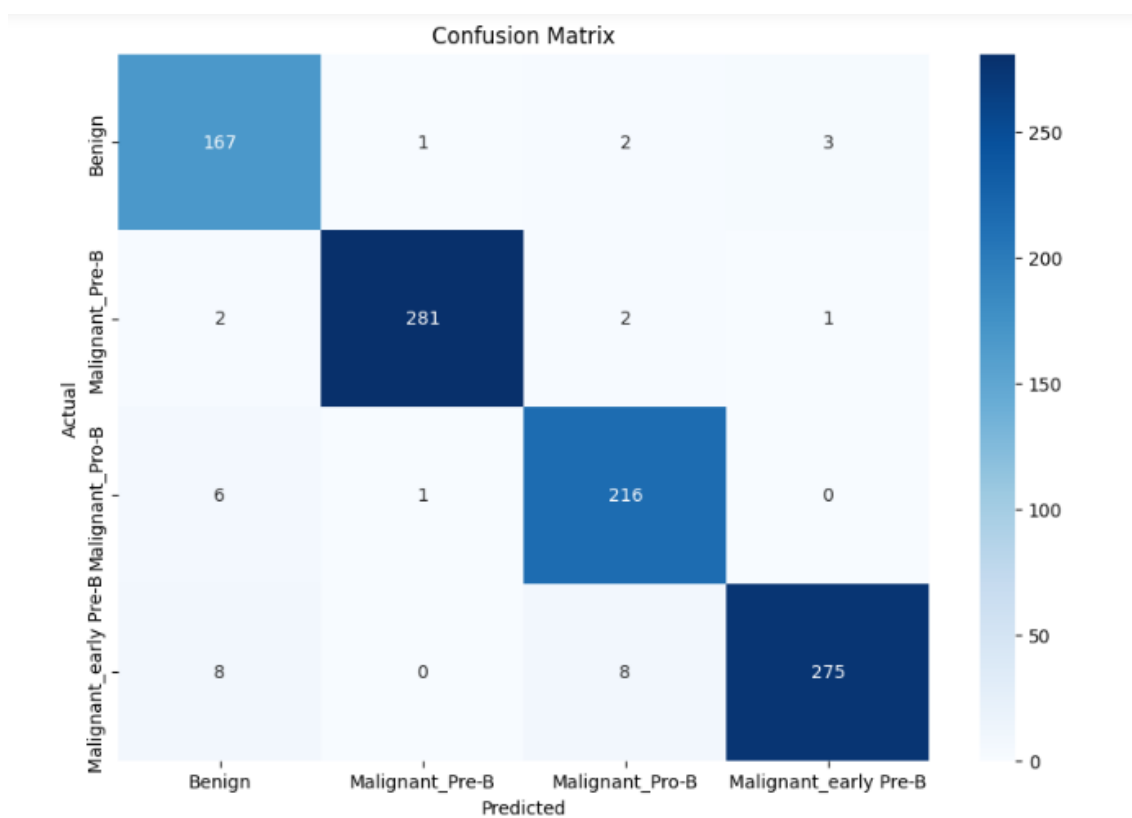


Fig 7.6. Confusion matrix

## CHAPTER 8

### CONCLUSION

This Thesis presents an improved ensemble method that combines Support Vector Machines (SVM) and Convolutional Neural Networks (CNN) for more precise blood cancer classification and detection from different cancer blood cell imaging datasets. By combining the strong classification capacity of SVMs with the deep learning capabilities of CNNs, this model significantly improves detection performance. The CNN+SVM ensemble model outperforms conventional single-model techniques, achieving an amazing 96% accuracy, according to experimental testing on a variety of datasets. This model exhibits resilience across several blood cancer cell types in addition to providing increased accuracy. Although the combination of SVM with CNN improves model performance, it also adds to the computational complexity. Nonetheless, the notable improvement in detection accuracy justifies the trade-off, rendering our ensemble model an invaluable instrument for cancer medical diagnostics.

#### **8.1 FUTURE WORK:**

We want to improve the suggested CNN+SVM ensemble model's functionality in further work to increase its usefulness in the medical domain, especially for the diagnosis of blood cancer. Developing techniques for the early diagnosis of uncommon blood cancer subtypes and integrating real-time diagnostic capabilities to dynamically track disease development will be primary areas of attention. Furthermore, our goal is to assess the model's efficacy in various scenarios by extending its use to a wider spectrum of hematological disorders. We will also investigate the integration of more sophisticated machine learning methods, such reinforcement learning for adaptive model tweaking based on fresh data. The aforementioned developments will enhance the diagnostic precision of the model and expand its application in clinical environments, hence promoting prompt and customized patient treatment.

## REFERENCES

- [1] Kanimozhi, N., Swikruti Nayak, Keshav Kumar, Vinda Manjramkar, Ranjeet Kumar, and D. Suganthi. "Blood Cancer Detection and Classification using Auto Encoder and Regularized Extreme Learning Machine." In *2023 8th International Conference on Communication and Electronics Systems (ICCES)*, pp. 1122-1127. IEEE, 2023.
- [2] Dharani, N. P., G. Sujatha, and R. Rani. "Blood Cancer Detection Using Improved Machine Learning Algorithm." In *2023 International Conference on Circuit Power and Computing Technologies (ICCPCT)*, pp. 1136-1141. IEEE, 2023.
- [3] Jayachitra, J., and N. Umakathaf. "Blood Cancer Identification using Hybrid Ensemble Deep Learning Technique." In *2023 Second International Conference on Electronics and Renewable Systems (ICEARS)*, pp. 1194-1198. IEEE, 2023.
- [4] Vomanesh, Chekuri Sai, Mediseti Venkata Sai Chaitanya, Sadala Sumanth, and Ramya Arumugam. "Detection and Classification of Leukaemia using Artificial Intelligence." In *2023 International Conference on Recent Advances in Electrical, Electronics, Ubiquitous Communication, and Computational Intelligence (RAEEUCCI)*, pp. 1-6. IEEE, 2023.
- [5] Sheet, Joydeb, Chanchal Ghosh, and Biplab Kanti Das. "Deep Learning-Based Transfer Learning for the Detection of Leukemia." In *2023 International Conference on Intelligent Systems, Advanced Computing and Communication (ISACC)*, pp. 1-6. IEEE, 2023.
- [6] Rehman, Amjad, Naveed Abbas, Tanzila Saba, Syed Ijaz ur Rahman, Zahid Mehmood, and Hoshang Kolivand. "Classification of acute lymphoblastic leukemia using deep learning." *Microscopy Research and Technique* 81, no. 11 (2018): 1310-1317.
- [7] Premalatha, S., K. Kavya, and K. Jayasudha. "An Efficient Detection and Classification of Acute Lymphoblastic Leukemia using CNN models." In *2023 2nd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, pp. 1-6. IEEE, 2023.

- [8] Aditya, Y., M. Durga, and Eswar Datta. "An Improved Leukemia Detection Model using CNN." In 2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA), pp. 261-264. IEEE, 2023.
- [9] Khairunnahar, Laila, Mohammad Abdul Hasib, Razib Hasan Bin Rezanur, Mohammad Rakibul Islam, and Md Kamal Hosain. "Classification of malignant and benign tissue with logistic regression." *Informatics in Medicine Unlocked* 16 (2019): 100189.
- [10] Seraj, Raihan, Razib Bin Hasan Rezanur, and Mohammad Abdul Hasib. "Classification of Malignant and Benign tissue with Logistic Regression." PhD diss., Department of Electrical and Electronic Engineering, Islamic University of Technology (IUT), Board Bazar, Gazipur-1704, Bangladesh, 2015.
- [11] Bhattacharjee, Subrata, Hyeon-Gyun Park, Cho-Hee Kim, Deekshitha Prakash, Nuwan Madusanka, Jae-Hong So, Nam-Hoon Cho, and Heung-Kook Choi. "Quantitative analysis of benign and malignant tumors in histopathology: Predicting prostate cancer grading using SVM." *Applied Sciences* 9, no. 15 (2019): 2969.
- [12] Ara, Sharmin, Annesha Das, and Ashim Dey. "Malignant and benign breast cancer classification using machine learning algorithms." In 2021 International Conference on Artificial Intelligence (ICAI), pp. 97-101. IEEE, 2021.
- [13] Madasamy, Kaliappan, Vimal Shanmuganathan, Nithish, Vishakan, Vijayabhaskar, Muthukumar, Balamurali Ramakrishnan, and M. Ramnath. "Benign and Malignant Cancer Prediction Using Deep Learning and Generating Pathologist Diagnostic Report." In *International Conference on IoT and Health*, pp. 73-87. Cham: Springer Nature Switzerland, 2023.
- [14] Teramoto, Atsushi, Ayumi Yamada, Yuka Kiriya, Tetsuya Tsukamoto, Ke Yan, Ling Zhang, Kazuyoshi Imaizumi, Kuniaki Saito, and Hiroshi Fujita. "Automated classification of benign and malignant cells from lung cytological images using deep convolutional neural network." *Informatics in Medicine Unlocked* 16 (2019): 100205.



**DELHI TECHNOLOGICAL UNIVERSITY**

(Formerly Delhi College of Engineering)  
Shahbad Daultpur, Main Bawana Road, Delhi-42

**PLAGIARISM VERIFICATION**

Title of the Thesis \_\_\_\_\_

Total Pages \_\_\_\_\_ Name of the  
Scholar \_\_\_\_\_ Supervisor (s)

(1) \_\_\_\_\_

(2) \_\_\_\_\_

(3) \_\_\_\_\_

Department \_\_\_\_\_

This is to report that the above thesis was scanned for similarity detection. Process and outcome is given below:

Software used: \_\_\_\_\_ Similarity Index: \_\_\_\_\_, Total Word Count: \_\_\_\_\_

Date: \_\_\_\_\_

**Candidate's Signature**

**Signature of Supervisor(s)**

PAPER NAME

**Thesis (Complete).pdf**

AUTHOR

**Shivnath**

WORD COUNT

**9144 Words**

CHARACTER COUNT

**51245 Characters**

PAGE COUNT

**44 Pages**

FILE SIZE

**1.3MB**

SUBMISSION DATE

**May 31, 2024 1:39 PM GMT+5:30**

REPORT DATE

**May 31, 2024 1:40 PM GMT+5:30**

### ● 13% Overall Similarity

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

- 8% Internet database
- 4% Publications database
- Crossref database
- Crossref Posted Content database
- 11% Submitted Works database

### ● Excluded from Similarity Report

- Bibliographic material
- Quoted material
- Cited material
- Small Matches (Less than 10 words)

## ● 13% Overall Similarity

Top sources found in the following databases:

- 8% Internet database
- 4% Publications database
- Crossref database
- Crossref Posted Content database
- 11% Submitted Works database

### TOP SOURCES

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

1	<b>dtusimilarity on 2024-05-29</b> Submitted works	2%
2	<b>ijircce.com</b> Internet	<1%
3	<b>ir.juit.ac.in:8080</b> Internet	<1%
4	<b>VNR Vignana Jyothi Institute of Engineering and Technology on 2023-...</b> Submitted works	<1%
5	<b>Southern University And A &amp; M College on 2024-04-29</b> Submitted works	<1%
6	<b>Gitam University on 2023-03-21</b> Submitted works	<1%
7	<b>University of Wolverhampton on 2020-11-28</b> Submitted works	<1%
8	<b>Joydeb Sheet, Chanchal Ghosh, Biplab Kanti Das. "Deep Learning-Base...</b> Crossref	<1%

9	<b>dspace.dtu.ac.in:8080</b> Internet	<1%
10	<b>nou.edu.ng</b> Internet	<1%
11	<b>Chekuri Sai Vomanesh, Mediseti Venkata Sai Chaitanya, Sadala Suma...</b> Crossref	<1%
12	<b>The University of the West of Scotland on 2024-04-19</b> Submitted works	<1%
13	<b>"Chapter 3 Development Processes and Frameworks", Springer Scienc...</b> Crossref	<1%
14	<b>(school name not available) on 2006-09-23</b> Submitted works	<1%
15	<b>Manipal University on 2023-10-30</b> Submitted works	<1%
16	<b>metaeye.co.uk</b> Internet	<1%
17	<b>slideshare.net</b> Internet	<1%
18	<b>East Tennessee State University on 2016-05-04</b> Submitted works	<1%
19	<b>dspace.dtu.ac.in:8080</b> Internet	<1%
20	<b>lib.buet.ac.bd:8080</b> Internet	<1%



21	<b>Arab Open University on 2012-04-10</b> Submitted works	<1%
22	<b>Gates-Chili High School on 2016-03-16</b> Submitted works	<1%
23	<b>"Table of Contents", 2023 International Conference on Circuit Power an...</b> Crossref	<1%
24	<b>American Intercontinental University Online on 2009-11-15</b> Submitted works	<1%
25	<b>Southern University And A &amp; M College on 2022-11-03</b> Submitted works	<1%
26	<b>Asia Pacific University College of Technology and Innovation (UCTI) on...</b> Submitted works	<1%
27	<b>Sikkim Manipal University on 2022-06-17</b> Submitted works	<1%
28	<b>"Proceedings of the 2nd International Conference on Big Data, IoT and ...</b> Crossref	<1%
29	<b>"Table of Contents", 2023 Second International Conference on Electron...</b> Crossref	<1%
30	<b>Bolton Institute of Higher Education on 2012-12-11</b> Submitted works	<1%
31	<b>ieeexplore.ieee.org</b> Internet	<1%
32	<b>pt.scribd.com</b> Internet	<1%

33	<b>Daffodil International University on 2023-12-18</b> Submitted works	<1%
34	<b>Institute of Technology, Nirma University on 2022-12-13</b> Submitted works	<1%
35	<b>allassignmentandessayhelp.blogspot.com</b> Internet	<1%
36	<b>fri-datascience.github.io</b> Internet	<1%
37	<b>link.springer.com</b> Internet	<1%
38	<b>researchsquare.com</b> Internet	<1%
39	<b>SASTRA University on 2022-06-20</b> Submitted works	<1%
40	<b>ethesis.nitrkl.ac.in</b> Internet	<1%
41	<b>futurecaptains.com</b> Internet	<1%
42	<b>sciencegate.app</b> Internet	<1%
43	<b>N. P. Dharani, G. Sujatha, R. Rani. "Blood Cancer Detection Using Impro...</b> Crossref	<1%
44	<b>Sri Sairam Engineering College on 2024-04-15</b> Submitted works	<1%

45	<b>University of Wales Institute, Cardiff on 2021-05-17</b> Submitted works	<1%
46	<b>d.docksci.com</b> Internet	<1%
47	<b>pdfcoffee.com</b> Internet	<1%
48	<b>ijisae.org</b> Internet	<1%