A NOVEL SOFT COMPUTING-BASED APPROACH FOR DETECTION OF LEUKEMIA DISEASE

Thesis Submitted for the Award of the Degree of

DOCTOR OF PHILOSOPHY

in

Computer Science and Engineering

By

Navpreet Kaur

Registration Number: 41800143

Supervised By

Prof. (Dr.) Amar Singh (23318)

School of Computer Applications

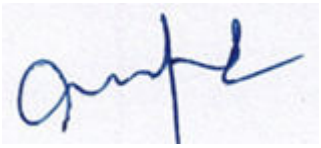
Lovely Professional University, Phagwara

LOVELY PROFESSIONAL UNIVERSITY, PUNJAB

2025

DECLARATION

I, hereby declared that the presented work in the thesis entitled **"A Novel Soft Computing-Based Approach for Detection of Leukemia Disease"** in fulfilment of degree of **Doctor of Philosophy (Ph. D.)** is outcome of research work carried out by me under the supervision of Dr. Amar Singh working as Professor, in the **School of Computer Application** of Lovely Professional University, Punjab, India. In keeping with general practice of reporting scientific observations, due acknowledgements have been made whenever work described here has been based on findings of another investigator. This work has not been submitted in part or full to any other University or Institute for the award of any degree.



Navpreet Kaur

Registration No.: 41800143

School of Computer Science and Engineering

Lovely Professional University,

Punjab, India

CERTIFICATE

This is to certify that the work reported in the Ph. D. thesis entitled **"A Novel Soft Computing-Based Approach for Detection of Leukemia Disease"** submitted in fulfillment of the requirement for the award of degree of **Doctor of Philosophy (Ph.D.)** in the **School of Computer Science and Engineering**, is a research work carried out by **Navpreet Kaur, 41800143** is a bonafide record of his/her original work carried out under my supervision and that no part of thesis has been submitted for any other degree, diploma or equivalent course.

Dr. Amar Singh

Professor

School of Computer Application

Lovely Professional University, Phagwara

ABSTRACT

Leukemia is a cancer of the blood and bone marrow that affects how blood cells are made and work in the body. Leukemia is complex, varies from person to person and it can be hard to diagnose it in the early stages. Current diagnosis methods are effective but take a lot of time and require expert knowledge, which can delay treatment. This research work focuses to solve these problems by creating new approaches that use intelligent techniques to detect leukemia more quickly and accurately.

The study started by reviewing existing leukemia diagnosis methods and identifying their main problems, showing the need for automated and reliable systems. To address this, new soft computing-based approaches namely, VGG16-PCA-PB3C, VGG19-PCA-BBBC, and HP3PGA-3PGA are proposed in this thesis. A detailed comparison of these approaches is done using four performance metrics. The proposed approaches are compared with existing approaches for leukemia detection.

The research work was focused on the following initially set objectives:

1. To study, analyze and evaluate the performance of the various existing soft computing-based approaches for detection of Leukemia disease.
2. To propose a novel soft-computing based approach for detection of Leukemia disease.
3. To compare the proposed approach with existing approaches using few performance metrics.

The research was done based on the above-mentioned objectives. This thesis covers all the work completed during the research. The thesis has six chapters, and the details of each chapter are explained below:

Chapter 1 presents the background of the research. It introduces leukemia, the causes of leukemia, its symptoms, types, and the challenges of the traditional leukemia diagnosis methods. This chapter presents the research problem and objectives. It highlights the approaches that can be used to diagnose leukemia. It also shows the main

contributions of the research and explains the structure of the thesis.

Chapter 2 provides a review of the current state of the art associated with literature. The review is mainly split into two parts. Part 1 contains traditional leukemia detection methods. The part 2 was focused on the review of artificial intelligence-based leukemia detection approaches. Further, the artificial intelligence-based approaches are divided into three categories: machine learning based approaches, deep learning-based approaches, and soft computing-based hybrid approaches. This chapter also presents the research gap found after the review of the existing approaches.

Chapter 3 proposes a new hybrid approach for leukemia detection. This approach combines Visual Geometry Group 16 (VGG16), Principal Component Analysis (PCA), and Parallel Big Bang Big Crunch (PB3C). In the proposed approach, VGG16 is used to extract features. The dimensionality of the extracted features is reduced using PCA technique. Further, the PB3C algorithm is used to select the best features. PB3C is a multipopulation based search and optimization algorithm. For the leukemia detection purpose, the deep neural network is trained on optimal features provided by PB3C algorithm. To evaluate that how well the proposed approach works, we tested it on CNMC_2019 leukemia dataset. The proposed approach is done in Python and compared with 13 other methods: Inception V3, VGG16, VGG19, SVM, Logistic Regression, Random Forest, Decision Tree, K Nearest Neighbor, and Bagging etc. It was observed that the proposed VGG16-PCA-PB3C based approach outperformed all the other 13 approaches on leukemia detection problem.

Chapter 4 proposes a new hybrid approach based on soft computing for leukemia detection. This approach combines Visual Geometry Group 19 (VGG19), Principal Component Analysis (PCA), and the Big Bang Big Crunch (BBBC) algorithm. In this approach, the deep features are extracted using VGG19 encoder. The dimensionality of the extracted features is reduced by PCA. After reducing the dimensions, a single-population-based BBBC algorithm is used to select the best principal components. The BBBC algorithm uses the best features to train a deep neural network for leukemia detection. The proposed algorithm was implemented in python and evaluated using the

CNMC_2019 leukemia dataset. The proposed algorithm is tested against 15 transfer learning and machine learning methods for leukemia detection and it is found that the proposed approach performed better than 15 other existing leukemia detection approaches.

Chapter 5 proposes a new soft computing-based algorithm namely, HP3PGA-3PGA (Hybrid Parallel Three Parent Genetic Algorithm – Three Parent Genetic Algorithm). The proposed HP3PGA-3PGA algorithm combines a multi-population-based parallel three-parent genetic algorithm (P3PGA) and a single-population-based three-parent genetic algorithm (3PGA). The proposed algorithm is tested using CEC2021 test suite consisting of 80 test functions. It was compared with other 10 recent algorithms. We also used the proposed algorithm to automatically evolve the near-optimal architecture of CNN for leukemia detection. This approach for leukemia detection is done in Python. The proposed approach is compared with 17 other methods for leukemia detection. The performance of HP3PGA-3PGA was observed to be quite good.

Chapter 6 concludes the research and explores the future possibilities of the work.

ACKNOWLEDGEMENT

I am profoundly grateful to my esteemed supervisor, Prof. Dr. Amar Singh, for his unwavering guidance, encouragement, and invaluable support throughout my journey in pursuing my Ph.D. His expertise, constructive feedback, and mentorship have been instrumental in shaping the direction of my research and academic growth. His expertise and commitment have played a crucial role in enriching the quality of this work.

I extend my sincere appreciation to the School of Computer Science and Engineering and School of Computer Application, Lovely Professional University, for providing me with the platform and resources to pursue my Ph.D. The academic environment and collaborative spirit within the department have been instrumental in fostering my research endeavours.

I would like to express my heartfelt gratitude to my husband for his unwavering support, patience, and for being a constant source of emotional strength throughout this journey. I am also deeply thankful to my parents, brother, and my entire family for their belief in my abilities, and for uplifting me with their faith and blessings during every step of this endeavour.

.

(Navpreet Kaur)

TABLE OF CONTENTS

Declaration……..……...……ii

Certificate……..……..iii

Abstract……iv-vi

Acknowledgement……...vii

List of Tables……x-xi

List of Figures……..xii

1. Introduction……...……...13-30

1.1. Research Background……....13

1.2. Blood Cancer and Leukemia……...……14-17

1.3. Diagnosis of Leukemia using Traditional method……17-18

1.4. Machine Learning……...……18-20

1.5. Convolutional Neural Networks……...……21-25

1.6. Soft Computing……...……25-28

1.7. Problem Formulation……28

1.8. Research Objectives……...……...29

1.9. Research Contributions……...……..29

1.10. Thesis Structure……29-30

2. Literature Review……...…….31-76

2.1. Methodology for Conducting of Literature Review……..31-33

2.2. Traditional Methods for Leukemia Detection……...33-38

2.3. Artificial Intelligence based Methods for Leukemia Detection……...…….38-75

2.4. Research Gap……76

2.5. Summary……...……76

3. VGG16-PCA-PB3C: A Hybrid PB3C and Deep Neural Network based approach for Leukemia Detection……...…….77-93

3.1. Introduction……...……...77-78

3.2. Visual Geometry Group 16 (VGG-16)……...……..78-83

3.3. Principal Component Analysis (PCA)……...……...83-84

3.4. Parallel Big Bang Big Crunch (PB3C)……..84-85

3.5. Proposed Soft Computing based Hybrid Leukemia Detection Approach…….85-89

3.6. Dataset……...……...89-90
3.7. Results and Discussion……...……..90-93
3.8. Summary……...……93
4. VGG19-PCA-BBBC: An Intelligent Framework for Leukemia Detection....94-111
4.1. VGG19 and its Architecture……...……95-100
4.2. Principal Component Analysis for Dimensionality Reduction……100
4.3. Big Bang Big Crunch Algorithm for Optimization……...……100-101
4.4. Proposed VGG19-PCA-BBBC Approach for Leukemia Detection…..…...101-108
4.5. Results and Discussion……...……..108-111
4.6. Summary……..……...111
5. HP3PGA-3PGA: A New Hybrid Soft Computing based algorithm for early detection of Leukemia……...112-130
5.1. Introduction……...……...113-114
5.2. Proposed Hybrid Soft Computing based Algorithm……..……...114-116
5.3. Simulation and Performance Analysis……...……...116-126
5.4. HP3PGA-3PGA based Neural Architecture Search (NAS) Approach…....126-130
5.5. Summary……...…..…130
6. Conclusion and Future Scope.....131-133
6.1. Conclusion……...…131-132
6.2. Future Scope…….132-133

References……..……....134-148
List of Publication……...……149
List of Conferences……...……..…150

LIST OF TABLES

Table 2.1 Search keywords for leukemia detection using traditional and artificial intelligence-based approaches 32
Table 2.2 Traditional leukemia detection methods 36-37
Table 2.3 Machine Learning based leukemia detection approaches 42-44
Table 2.4 Deep Learning based leukemia detection approaches 52-54
Table 2.5 Soft Computing-based hybrid leukemia detection approaches 70-75
Table 3.1 VGG16 architecture 79-82
Table 3.2 Features extracted by VGG16 encoder layers 82-83
Table 3.3 Comparison of proposed VGG16-PCA-PB3C approach with existing transfer learning- based approaches 90-91
Table 3.4 Comparison of proposed VGG16-PCA-PB3C approach with existing machine learning- based approaches 91
Table 4.1 VGG19 Convolutional layer blocks 96
Table 4.2 VGG19 layers architecture 97-98
Table 4.3 Layer wise features extracted by VGG16 and VGG19 103
Table 4.4 Comparative Analysis of VGG16 and VGG19 architecture 104
Table 4.5 Comparison of proposed VGG19-PCA-BBBC with existing transfer learning-based approaches... 109
Table 4.6 Comparison of proposed VGG19-PCA-BBBC with existing machine learning-based approaches. 109-110
Table 5.1 Various functions of CEC2021 test suite 117-118
Table 5.2 Performance results of HP3PGA-3PGA along with other algorithms on the standard test bench suite for CEC2021 119-123
Table 5.3 Comparison of HP3PGA-3PGA performance on CEC2021 benchmarks 124-125
Table 5.4 Hyper-Parameters considered for CNN optimization 127
Table 5.5 Accuracy levels for HP3PGA-3PGA based CNN on the CNMC_2019 dataset 128

Table 5.6 Comparison of proposed HP3PGA-3PGA approach with existing machine learning and transfer learning approaches 128-129

LIST OF FIGURES

Figure 1.1 Formation of blood cells from stem cell..... 15
Figure 1.2 Types of blood cancer..... 16
Figure 1.3 Leukemia blood cells..... 17
Figure 1.4 Traditional leukemia diagnosis methods 18
Figure 1.5 A CNN architecture 22
Figure 1.6 Grayscale image 23
Figure 1.7 A RGB image..... 23
Figure 1.8 Working of a convolutional block 24
Figure 1.9 Soft computing approaches 26
Figure 2.1. PRISMA approach for bibliometric analysis of leukemia detection approaches..... 33
Figure 3.1 Proposed soft computing-based approach for leukemia detection 86
Figure 3.2 Microscopic blood cell images of leukemia patient 89
Figure 3.3 Microscopic blood cell images of normal person..... 89
Figure 3.4 Comparison of VGG16-PCA-PB3C with transfer learning-based approaches 92
Figure 3.5 Comparison of VGG16-PCA-PB3C with existing machine learning-based approaches..... 92
Figure 4.1 Architecture of VGG19 model for feature extraction and classification.... 99
Figure 4.2 Customized architecture of VGG19 model for feature extraction 99
Figure 4.3 Proposed soft computing-based approach for leukemia detection 107
Figure 4.4 Comparison of VGG19-PCA-BBBC with transfer learning approaches . 110
Figure 4.5 Comparison of VGG19-PCA-BBBC with existing machine learning-based approaches..... 111
Figure 5.1 Three parent genetic algorithm concept 113

Chapter 1 Introduction

This chapter presents the foundation of our research work. It discusses the issues and challenges in the early detection of leukemia. This chapter explains the research objectives and gives a brief overview of leukemia detection methods. Section 1.1 introduces the background of the research. Section 1.2 provides a summary of blood cancer and leukemia. Section 1.3 reviews traditional leukemia diagnosis methods. Section 1.4 presents the machine learning concept. Section 1.5 explains the convolutional neural networks. Soft computing methods are explained in section 1.6. Problem formulation is presented in section 1.7. Identification of research objectives is done in section 1.8. The research contributions are described in section 1.9, and section 1.10 provides an overview of the thesis structure.

1.1. Research Background

Leukemia is a form of cancer that impacts the blood and bone marrow. It causes white blood cells (WBCs) to grow abnormally, disrupting the normal production and function of blood cells [1]. Traditional methods for diagnosing leukemia, like bone marrow biopsies, cytogenetic tests, and flow cytometry, are effective but require a lot of time and specialized knowledge. Detecting leukemia early is very important because it helps start treatment quickly, especially in fast-progressing cases like acute leukemia. Early treatment can slow the disease, improve recovery chances, and increase survival rates. Since different types of leukemia need different treatments, accurate diagnosis ensures patients get the right care. It also helps understand how the disease is progressing and responding to treatment, which is important for planning future care.

Due to the complexity and variety of leukemia, there is a need for better diagnostic methods that are faster and more accurate [2]. Modern soft computing-based techniques offer promising solutions. These technologies can help develop automated systems to identify and classify leukemia early, improving treatment and care for patients [3].

1.2. Blood Cancer and Leukemia

1.2.1. Blood Components

Blood is the basic fluid and element of life. Blood consists of nearly 55% plasma, 45% erythrocytes, and under 1% leukocytes and platelets. The bone marrow is where these blood cells are produced and then released into the bloodstream to carry out their intended functions.

Red Blood Cells (RBCs), also called erythrocytes, carry oxygen from the lungs to all parts of the body. These red cells provide the oxygen to the different cells and the tissues of the human body and takes away the carbon dioxide from the different cells of human body. The white blood cell contains a nucleus and cytoplasm, and they are involved in the protection mechanism against infections. There is 1 white blood cell to every 600-700 red cells of blood. White cells of blood produce antibodies and these cells are responsible for fighting against infections. They protect the body from the harmful bacteria, germs, and viruses. The white cells of the blood are categorized into agranulocytes (lymphocytes) and granulocytes.

The white cells of the blood which are lymphocytes produce antibodies to protect the body against disease. Lymphocytes are found in the bone marrow, lymph nodes, lymphatic tissues, spleen, and other areas. Lymphocytes are classified into three primary types i.e. B cells, T cells, and NK cells. B cells develop the antibody particles and these antibody molecules are provided to the plasma membrane. T cells help control the immune system. T cells keep the human body away from the cancer cells and various pathogens and these cells are developed in the bone marrow. Natural Killer (NK) cells protect the human body against the tumor cells.

Granulocytes are the subset of white cells of blood which are important in healing the damaged cells and these immune cells protect a human body against pathogens. Granulocytes are further classified into:

a) Neutrophils are the cells which fight against the infection and kills bacteria.
b) Eosinophils are the cells which destroy the cancer cells and these cells are responsible in wrecking the certain parasites.
c) Basophils are the cells which are involved in fighting viral infections and allergic responses.
d) Monocytes are the cells which eliminate the infected and the dead cells.

Platelets are responsible in blood clot formation and thus with the help of platelets, blood loss is prohibited. To every 20 red cells of the blood, there is a single platelet. Platelets are helpful in the process of blood coagulation and assist in controlling bleeding as they travel and accumulate to the area which bleeds. Plasma is the fluid component of blood that facilitates the transfer of nutrients, proteins, and various other molecules to ensure that various body parts can function. The whole process of the formation of plasma, platelets, and different cells of blood from the stem cell as shown in Figure 1.1 is called haematopoiesis. The different types of cells are formed and developed from the foundational microorganisms by a procedure known as separation. When blood cells are full grown and prepared to work, the matured blood cells at that point take themselves off from the bone marrow and move to the blood of an individual [4][5].

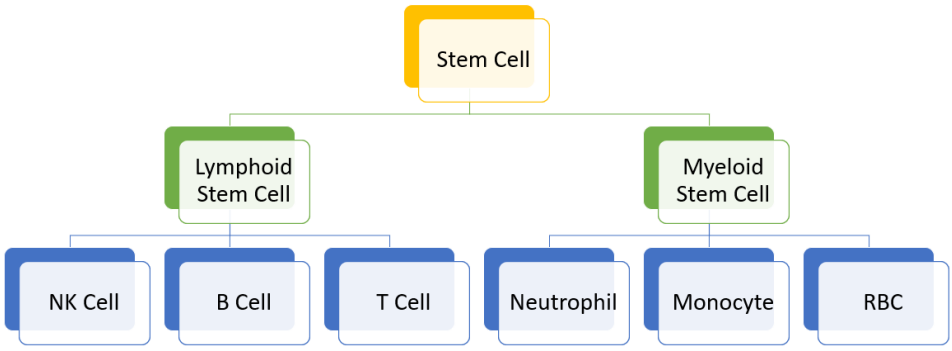


Figure 1.1 Formation of blood cells from stem cell

1.2.1.1. Blood Cancer

When abnormal cells, excessively grow, they cause blood cancer. This type of cancer

affects the development of the blood cells, and the functioning of the blood cells is also changed. Due to this, swollen lymph nodes, tiredness, nosebleeds, frequent infections, weight loss, and body pain occurs. Figure 1.2 illustrates that blood cancer can impact leukocytes, the lymphatic system, and plasma cells [6].

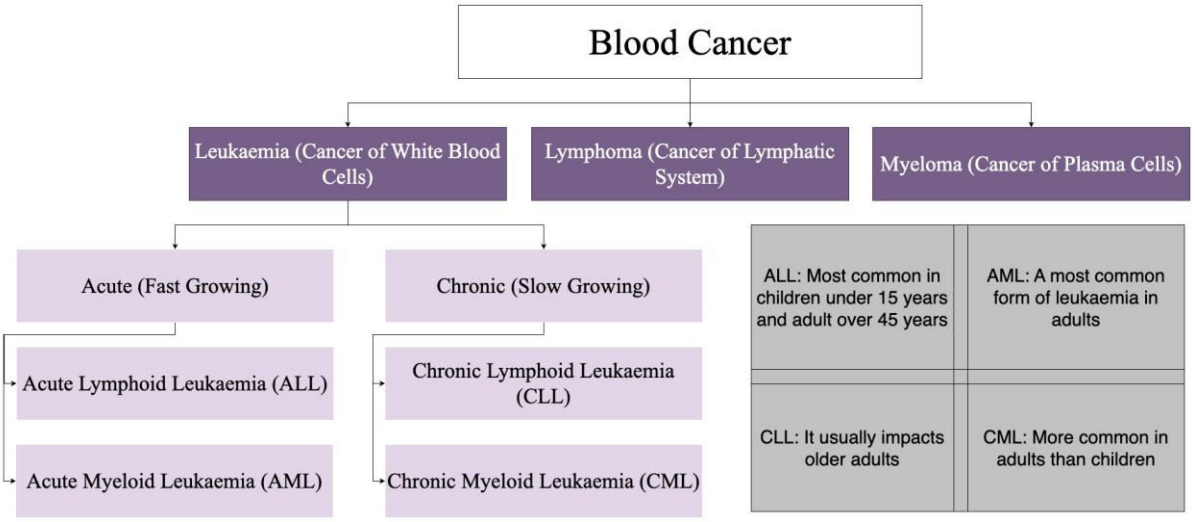


Figure 1.2 Types of blood cancer

1.2.2. Leukemia

Leukemia is a blood condition that begins in the bone marrow, a sponge-like tissue found in many bones. In individuals with Leukemia, there is an excessive generation of lymphoblasts in the bone marrow, as depicted in Figure 1.3, resulting in reduced immunity.

Leukemia, or blood cancer, is a type of cancer that affects leukocytes. It is classified by the type of cancer cells, which can change into lymphoid or myeloid. It is also divided into two types based on how fast it grows: chronic and acute. Lymphocytes, also known as lymphoid cells, are impacted in lymphocytic leukemia, while myeloid leukemia affects bone marrow's myeloid cells.

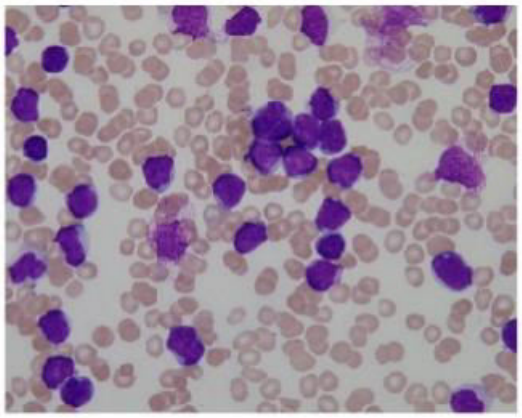


Figure 1.3 Leukemia blood cells [7]

The excess production of B-cells is found in the lymphocyte category. Myeloid cells help in the development of red blood cells, white blood cells, and blood platelets. In acute leukemia, there is a rapid increase in abnormal blood cell growth, resulting in quickly developing symptoms. Abnormal blood cells take time to form in chronic leukemia [8].

1.2.2.1. Causes of Leukemia

The exact and the certain reason of leukemia is not clearly determined. Some of the parameters which are incorporated as the causes of leukemia are ionizing radiation, smoking, earlier chemotherapy, a few synthetic concoctions, and down's ailment. The individuals with a family history of leukemia are equally at more serious hazard. After analysing the stage and type of leukemia, treatment is chosen [9]. The type of leukemia is the key factor in making the choice in the treatment, but other factors also play a very vital role.

1.3. Diagnosis of Leukemia using Traditional Method

Early detection of leukemia is a challenging issue because symptoms such as flu-like symptoms, weight loss, tiredness, and fatigue are frequently seen. Leukemia cases are typically identified by pathologists by analyzing blood smears under a microscope. The pathologists analyze a range of cells to identify leukemia. An indication of leukemia is when there is a rise in abnormal immature leukocytes and a decrease in other blood cell counts.

Many predictive models are built based on clinical knowledge and data which provide useful findings to identify the illness in its initial phases [10]. However, decision support system development for disease identification is still in its infancy stage. Figure 1.4 shows the common methods used by doctors to diagnose leukemia. As shown in Figure 1.4, the traditional leukemia detection techniques are categorized into five categories namely physical exams, blood tests, bone marrow tests, imaging tests, and other tests [11]. The detail of each category is discussed in chapter 2 of this thesis.

In the traditional method to detect leukemia, number of physical tests like bone marrow biopsy, bone marrow aspiration, cytometry etc. are conducted and the results of these tests are analysed by the haematologist oncologist whereas in convolutional neural network (CNN) and deep neural network (DNN) based approaches for the detection of leukemia, the blood cell image of the patient is provided to the model and then it is analysed whether the patient is suffering with leukemia or not.

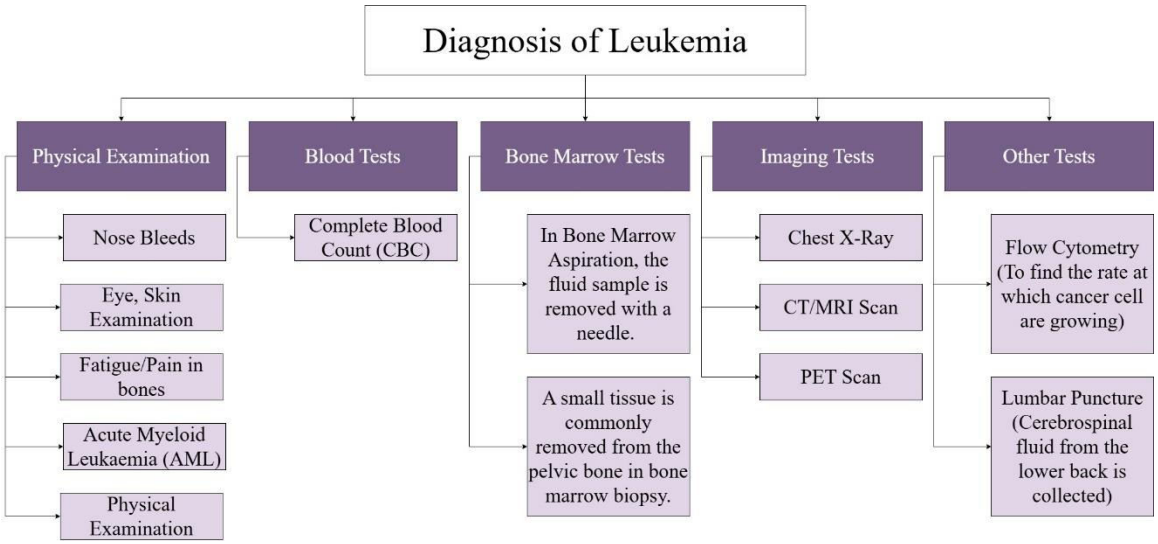


Figure 1.4 Traditional leukemia diagnosis methods

1.4. Machine Learning

According to Arthur Samuel, the definition of Machine Learning (ML) is: “the field of study that gives computers the ability to learn without being explicitly programmed.” Later, the definition was improved by Tom Mitchell as “A computer program is said to learn from experience E concerning some class of tasks T and performance measure P,

if its performance at tasks in T, as measured by P, improves with experience E." It is a rapidly evolving field that has gained significant attention and importance in various domains. The primary goal of machine learning is to assist computers in enhancing their capabilities by gaining knowledge from previous experiences or data. This is achieved through the development and utilization of various techniques and algorithms that can identify patterns, extract insights, and utilize this information to make predictions or decisions using the available data. We can categorize machine learning techniques into 3 groupings (i) Supervised Learning (SL), (ii) Unsupervised Learning (UL), and (iii) Reinforcement Learning (RL) techniques. Each category possesses distinct traits and finds its specific applications [12].

Machine learning comprises of statistical tools that learn from data. Self-learning algorithms of machine learning drive knowledge from the data. The available computational power can be used to apply advanced algorithms to uncover hidden patterns in data, infer relationships, and predict outcomes. Machine learning algorithms are used in solving the problems for which there is no mathematical solution. Machine learning is extensively utilized in email spam filters, natural language processing, voice recognition, recommendation systems, classification and clustering problems, robotics, object identification, image processing, and disease detection [13] etc.

Machine learning models learn from the previous and historical data to create a model. This model can help predict the result for a new input. A machine learning model receives various types of data such as numerical, textual, visual, or audiovisual, and produces results that can be either a decimal number (such as the cost of a product or the speed of an autonomous vehicle) or a whole number identifying a group or type [14]. The different categories of the machine learning techniques are mentioned as below:

1) **Supervised Learning:-** These algorithms find a way to produce the desired output from a given input. Labeled data is given to the machine as input for training. During training, for a given input, the intended result is predetermined. In Supervised learning, algorithm or model learns a function that connects from

x to y, based on the labeled training examples (x, y). Algorithms trained by using labeled data allow us to make predictions about the unseen and future data. The two main categories of supervised learning are classification and regression. Classification tries to analyze the category of the input data and thus predict the class label from a given list of discrete class labels. Classification can involve either binary or multiclass categories. In binary classification, the algorithm learns various set of rules to discriminate between two classes. Multiclass classification classifies the data in more than two classes (recognizing a digit from handwritten number) [15-16]. Regression on the other hand trains and predicts a continuous-valued response.

2) **Unsupervised Learning:-** Unsupervised algorithms transform data in the form that is easier for humans and other algorithms to understand. Unlike supervised algorithms, in unsupervised algorithms, data contains only inductive signals without any description attached. Unlabeled data is used in unsupervised algorithms. Unlabeled means for a given set of input attributes x_{i1}, x_{i2}, x_{i3}, ….x_{in} the output attribute y_i is not defined. Unsupervised algorithms analyze the structure of data to extract important and useful information without the supervision of a known outcome. The common application of the unsupervised algorithms is used to find the more informative representation of data by dimensionality reduction (using Principal Component Analysis) or clustering [17]. Clustering can be applied to organize unlabeled data into distinct clusters where each cluster will have some common features.
3) **Reinforcement Learning:-** In this learning, the agent or system observes the environment, performs an action, and gets a reward in return. Reinforcement learning is feedback based, based on how well an action is performed, the rewards are decided. The aim is to maximize the reward via an exploratory trial and error approach. Robots use reinforcement learning to find the path, a chess engine decides a series of moves and the outcome of the game can result in either a victory or defeat, which is the reward [18-19].

1.5. Convolutional Neural Networks

A CNN is a type of neural network which is used to find patterns in data. CNNs automatically detect patterns, like edges, shapes, and textures, in the data. This kind of network com various layers such as input, convolutional, pooling, flatten, and FC layers. A layer is a set of neurons that perform the same operation and have the same hyperparameters. A neuron is a basic function that takes several inputs and gives one output [20].

CNNs process image data more effectively and efficiently. They use convolution operations to find important features in images. CNNs used for classification tasks have two main parts: a feature extractor and a classifier. The feature extractor consists of convolutional blocks, which encompasses several convolutional layers. After these layers, a network has max pooling layers. The features in the image are found by convolutional layers and store in activation maps. The number of filters in a convolutional layer are decided during the design process, and the depth of the output depends on the number of filters. Pooling layers help reduce the size of activation maps, making the model more efficient and less likely to overfit. The classifier then uses the extracted features to calculate class probabilities through fully connected layers [21]. If there are more than two classes, a SoftMax layer is used to change the output into probabilities, showing the likelihood of each class for the image. The structure of a CNN is shown in Figure 1.5.

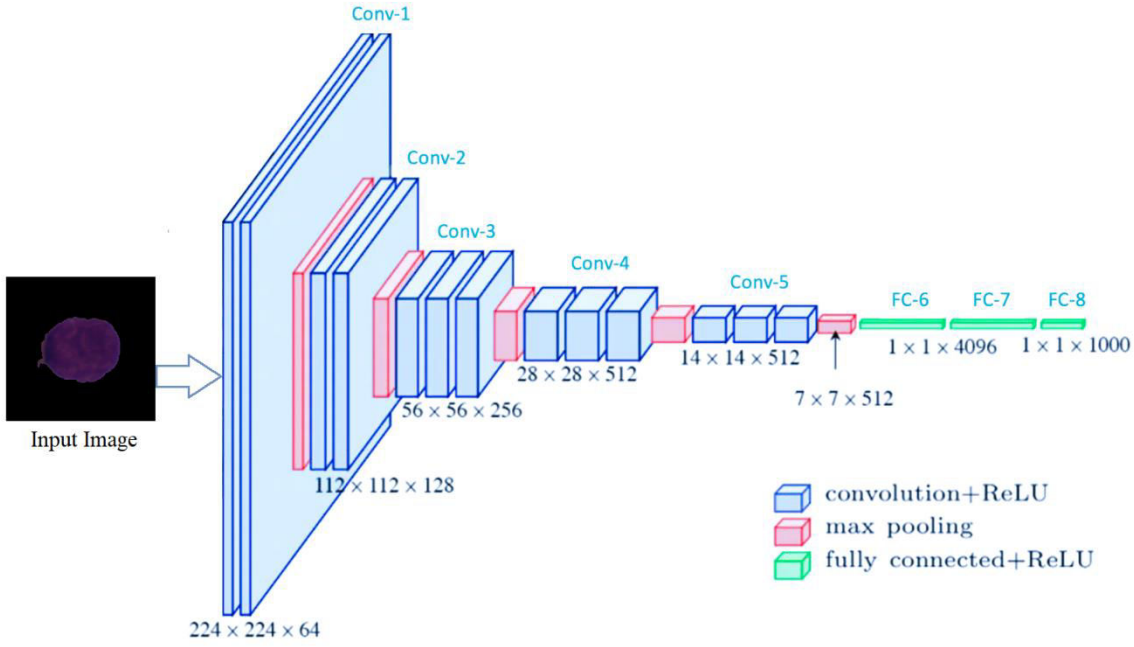


Figure 1.5. A CNN architecture

The next sections explain in detail how the layers of a convolutional neural network work.

1.5.1. Input Layer

The very first layer in this network is the input layer. The input image of the dataset is provided to this layer. The input image can either grayscale or RGB. A grayscale image has one channel that displays shades of gray from black to white. An RGB image has three channels: red, green, and blue. Each channel shows the intensity of their respective colors. Figure 1.6 and Figure 1.7 shows the grayscale and RGB image respectively.

1.5.2. Convolutional Layers

A convolutional layer works like the "eyes" of a CNN. The neurons in this layer search for specific features in the image. The first layers look for simple features like edges, textures, colors, and contrast. The middle layers detect shapes, corners, and patterns, helping the network recognize parts of objects. The deeper layers find more complex

features like parts of objects, such as eyes or wheels or a cell. In the final layers, the network can identify full objects like a person, car, or dog.

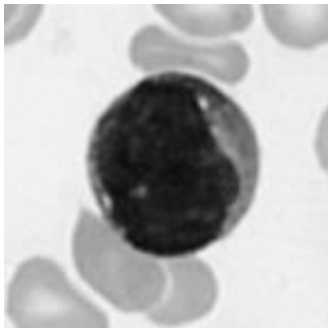


Figure 1.6 Grayscale image[22]

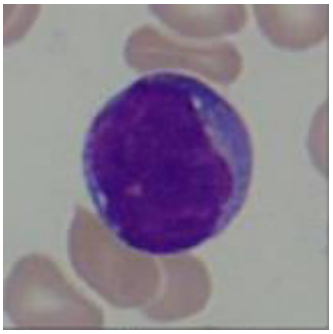


Figure 1.7 A RGB image[22]

The input to a convolutional layer is usually a 2D array, either the original image or the output from an earlier layer. The first convolutional layer receives the input image, which can be a grayscale or colour image. Convolutional layers use filters, also called kernels, to examine input images. The filter is placed over a part of the input, and its values are multiplied with the corresponding parts of the input. This gives a single number as the result. The filter then slides over the image with the stride value. Stride refers to number of pixels the kernel moves at a time. When the stride is smaller, more features are learned and when the stride is larger, fewer features are extracted. Another important parameter in CNN is padding. It adds additional pixels to the borders of the input image to keep the original size. There are various padding methods, such as zero padding, same padding, and valid padding. Zero padding is the most common because it is easy and efficient. It evenly adds zeros around the edges of the input image [23][24]. The output is processed using an activation function and stored in an activation map. The working of a convolutional block is shown in Figure 1.8. The size of the output feature in a 2D convolution is calculated using the formula:

$$O=((N-F+2P)/S)+1 \quad (1.1)$$

Where O denotes the output size and N denotes the input size (height or width), F is the size of Kernel, P indicates padding and S presents stride. The convolution operation is defined as:

$$Convolution(image, filter) = sum(image * filter) \quad (1.2)$$

The convolutional layers use the rectified linear unit (ReLU) activation function.

$$f(x) = max(0, x) \quad (1.3)$$

where x is the input value. The output of the function is the input value if it is zero or higher, and zero if it is less than zero.

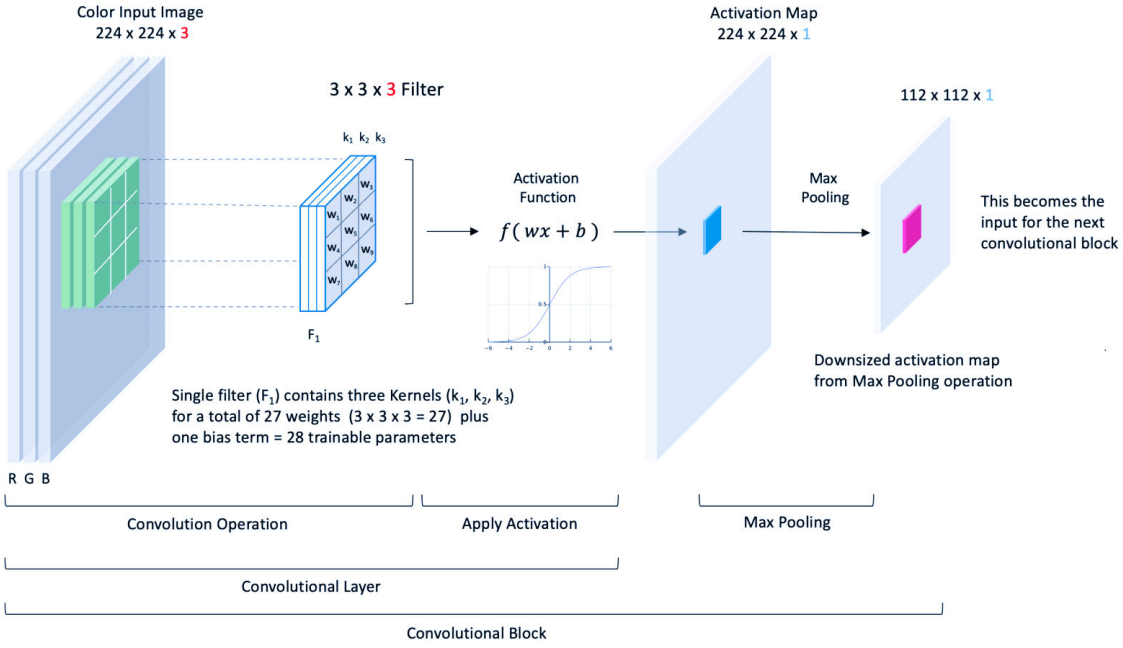


Figure 1.8 Working of a convolutional block

1.5.3. Pooling Layers

Pooling layers are included next to convolutional layers to decrease the size of the activation maps. They down-sample information from small regions in the feature map. Max pooling, which keeps the highest value from each area, and average or mean pooling, that finds the mean value are the two types of pooling. The number of parameters is decreased by the pooling layers, prevents overfitting, and focuses on the significant feature set, making the algorithm faster and more efficient [25].

1.5.4. Flatten Layer

The flatten layer in a CNN changes multi-dimensional feature maps into a single, one-dimensional line of data. This makes the data ready for the fully connected layers that

come next. After the convolutional and pooling layers find features, the flatten layer takes the output (like a 3D tensor, for example, 7*7*64) and reshapes it into a 1D vector (e.g., 3136 values). This step helps the network move from extracting features to classifying them [26]. The flatten layer makes sure all the features are kept and ready for the fully connected layers, which need 1D data for classification.

1.5.5. Fully Connected Layers

Fully connected layers (or dense layers) in a CNN are used to make the final decisions or predictions. Once the extraction of the feature set is done, the dense or FC layers combine the features to classify the data. In FC layers, every neuron is connected to the other neurons of upcoming layer. The one-dimensional input data is multiplied by different weights, and biases are added. Then, an activation function like ReLU or Softmax is used to get the output. The ReLU function is used in the first two fully connected layers. The last fully connected layer is usually followed by a softmax function in classification tasks to turn the output into probabilities, with each probability matching a class label. The fully connected layers help the network classify the image based on the features learned earlier [27].

The SoftMax function is defined as:

$$\mathbf{softmax}(\boldsymbol{x})_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \qquad (1.4)$$

1.6. Soft Computing

Soft computing is a collection of techniques designed to solve difficult problems by copying how humans think and make decisions. Unlike traditional computing, which needs exact answers, soft computing can handle uncertainty, estimation, and inaccuracy, making it useful for real-life problems. The techniques in soft computing include neural networks, fuzzy logic, genetic algorithms, and evolutionary computing [28]. These methods are used in many fields, like medical diagnosis, data analysis, and pattern recognition. Soft computing combines flexibility, adaptability, and strength to create smart systems that can solve difficult problems, like detecting diseases such as leukemia early. The various soft computing techniques are depicted in Figure 1.9.

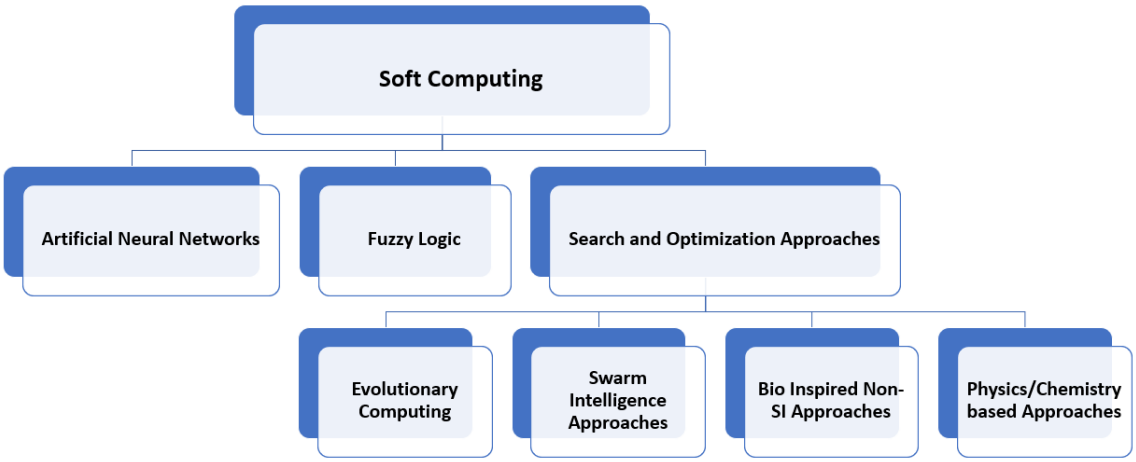


Figure 1.9. Soft computing approaches

In the past, healthcare decisions were only made by doctors or medical experts. However, a major issue in both developed and developing countries is the shortage of medical professionals in hospitals. With the development of soft computing algorithms, there has been an increased reliance on machines and intelligent systems to help with medical diagnosis. Soft computing is especially useful in medical imaging because it can handle the uncertainties in image data. Many predictive models are now being developed to diagnose diseases early based on clinic observations and data. However, disease detection systems are still in the early stages. Soft computing helps intelligent systems learn from uncertain environments and find the best solutions [29][30]. In healthcare, it can be applied in areas such as analysing clinical data, monitoring activities, diagnosing, and predicting diseases.

1.6.1. Artificial Neural Network

The structure and the functionality of the Artificial Neural Networks (ANNs) is modelled like how the human brain works. An artificial neural network is composed of a structured network with interconnected nodes, neurons, and multiple layers. Every neuron handles inputs and sends the output to the next layer. This continues until the network predicts the final output. Thus, a neuron calculates the combined weight of the input signals and measures it against a specific threshold value. If the input value is less than the threshold, the neuron will output -1, but if it is equal to or exceeds the threshold,

the output will be +1. ANN is powerful for recognizing patterns from big datasets, hence best for classification in medical diagnostics. The weights and biases of the ANN change as the training process progresses. This implies that they are highly adept at capturing the complex connections between inputs and outputs due to their nonlinear activation functions. Therefore, a well-trained neural network can make accurate predictions on new, unseen data by generalizing from previously seen data through nonlinear neuron interactions [31].

An ANN training algorithm was introduced in 1958 by Frank Rosenblatt, which provided a procedure to train a simple artificial neural network, i.e., a perceptron. To improve classification, the perceptron makes small changes to the weights to minimize the gap between the predicted result and the real output. In perceptron's training algorithm, the initial weight values are initialized, and then the perceptron is activated by applying the inputs. After that, the perceptron weights are updated, the iteration is increased, and the process keeps on repeating until convergence. Multiple layers of neurons comprise multi-layered neural networks. A multi-layered neural network has an input layer, one or more hidden layers, and an output layer [32]. The input signals are transmitted in the forward direction on a layer-by-layer approach.

1.6.2. Fuzzy Logic

In 1965, Lotfi A. Zadeh and Dieter Klaua introduced fuzzy sets. Fuzzy sets were developed by expanding the idea of traditional crisp sets. Crisp sets are based on binary logic, where an object can either belong to the set (1) or not (0). In contrast, fuzzy sets allow an object to belong to the set with a membership value that can range from 0 to 1, meaning it can be partially in the set. This makes it possible to have a more flexible and continuous representation. Operations like union, intersection, inclusion, and complement can also be used with fuzzy sets, and their properties are studied in this framework.

Fuzzy logic expands upon traditional boolean logic by introducing the theory of partial truth, allowing truth values to exist on a spectrum between completely true and false. This approach is useful in dealing with uncertainty and imprecision, which are common

in medical data [33]. Membership functions gain inputs into degrees of membership in fuzzy logic systems. Such fuzzy inputs undergo a set of rules for fuzzy output, which then can be defuzzied to get a crisp result. If-then rules in fuzzy logic help experts understand and modify complex decision-making processes [34]. Fuzzy logic uses words like high, medium, and low to describe variables instead of numbers, making it easier to understand and more like how humans think.

1.6.3. Nature Inspired Computing Algorithms

Our environment has been evolving for centuries, overcoming various challenges. These challenges inspire the creation of intelligent algorithms to solve complex problems. Algorithms inspired by natural processes are called "nature-inspired algorithms." Researchers have identified many benefits of these algorithms and divided them into categories like swarm intelligence, bio-inspired, physics/chemistry-based, and evolutionary algorithms. These nature-inspired algorithms are used to find the best solutions to problems. They are applied in fields like image recognition, pattern recognition, and finding the best routes in wireless networks [35][36].

1.7. Problem Formulation

Traditional methods for identifying leukemia, such as blood tests, bone marrow biopsies, and cytogenetic analysis, can be time-consuming and need expertise, resulting in possible treatment delays and variations in diagnostic accuracy. Furthermore, these methods require a lot of resources, making them less available in environments with limited resources. This highlights the importance of promptly and precisely identifying leukemia with the assistance of automated diagnostic tools, which can enhance precision, decrease analysis time, and improve diagnostic accessibility. Thus, there is a need for soft-computing based approaches that can select the near-optimal features from the blood smear images dataset automatically. Hence, this research work is devoted to the development of new soft-computing based image classification approaches that could apply to leukemia detection and other fields related to computer vision.

1.8. Research Objectives

The goal of this research was to focus on the following objectives:

1. To study, analyze and evaluate the performance of the various existing soft computing-based approaches for detection of Leukemia disease.
2. To propose a novel soft-computing based approach for detection of Leukemia disease.
3. To compare the proposed approach with existing approaches using few performance metrics.

1.9. Research Contributions

Major contribution of our research work is as below:

1. **VGG16-PCA-PB3C approach:** We proposed the use of the VGG16 architecture for feature extraction, PCA for feature reduction and PB3C for optimization, which enhances the accuracy in leukemia detection while reducing the computational complexity.
2. **VGG19-PCA-BBBC approach:** We proposed the VGG19-PCA-BBBC model, which integrated VGG19, PCA, and BBBC to make improvements in the precision and performance of the leukemia detection in comparison to VGG16-PCA-PB3C approach.
3. **The HP3PGA-3PGA approach:** We proposed the HP3PGA -3PGA algorithm that integrated the 3-Parent Genetic Algorithm with parallel processing. The proposed algorithm is validated on leukemia detection problem. The proposed algorithm evolves CNN architectures for leukemia detection automatically.

1.10. Thesis Structure

The following is the format of the remaining parts of this thesis.

Chapter 2 presents the state-of-the-art survey of leukemia detection approaches. This chapter focuses on the area of application of image preprocessing, machine learning, and soft-computing in leukemia detection. The review underscored the challenges associated with leukemia diagnosis problem, particularly in dealing with large datasets

and potential information loss during digitization.

Chapter 3 presents a new hybrid approach for leukemia detection which integrates Visual Geometry Group 16 (VGG16), Principal Components Analysis (PCA), and Parallel Big Bang Big Crunch (PB3C). The feature extraction process from the images using the customized VGG16 model is explained. The dimensionality reduction task of the extracted features using PCA is discussed in detail. The search and optimization based PB3C used for the optimal selection of the features is also highlighted in this chapter.

Chapter 4 presents a novel algorithm called Visual Geometry Group 19 (VGG19) - Principal Component Analysis (PCA) -Big Bang Big Crunch (BBBC). The deeper structure and the architecture of VGG19 is discussed in detail for the extraction of the deeper features from the blood smear images. The chapter also highlights the BBBC algorithm for the optimal feature selection. The results of this proposed approach are compared with the VGG16-PCA-PB3C approach and it is analysed that the VGG19-PCA-BBBC outperformed VGG16-PCA-PB3C.

Chapter 5 introduces a new algorithm called HP3PGA-3PGA (Hybrid Parallel Three Parent Genetic Algorithm – Three Parent Genetic Algorithm) for early leukemia detection. There are two stages of the algorithm in which it works in. The first stage is the multi-population stage and the second stage is the single-population. The multi-population stage explores many possible solutions to find the best ones, while the single-population stage focuses on refining the solutions when there is less need for variety. Both phases use two types of three-parent genetic algorithms: Parallel Three Parent Genetic Algorithm (P3PGA) and Three Parent Genetic Algorithm (3PGA).

Chapter 6 outlines the conclusion and potential future directions of our proposed work.

Chapter 2 Literature Review

This chapter presents a state-of-the-art survey and a detailed analysis of the traditional leukemia detection approaches and the artificial intelligence-based leukemia detection approaches. The way leukemia is diagnosed has improved a lot over time. Doctors are using both traditional methods and the new artificial intelligence techniques for the detection of leukemia. Each method has its own advantages and disadvantages, affecting how accurate, fast, and effective the diagnosis is. The traditional and the manual detection of leukemia includes examining peripheral blood smear, bone marrow tests, cytochemical staining, flow cytometry, and cytogenetic tests. The manual detection process of the leukemia is lengthy, takes more time and is expensive. The results also depend on how experienced the pathologist is in examining different cell types. On the other side, artificial intelligence-based leukemia detection techniques are based on machine learning, transfer learning, convolutional neural network, nature inspired algorithms. These methods are popular in medical diagnosis because they can work quickly with difficult and unclear information. This chapter explains various traditional and AI based methods to detect leukemia along with their strengths and weaknesses.

Section 2.1 of this chapter discusses the methodology for conducting the literature review for our research. Section 2.2 represents the literature work which is related to the traditional methods for leukemia detection. Section 2.3 presents the literature work related to artificial intelligence-based methods for leukemia detection. Section 2.4 provides the research gap and section 2.5 summarizes the overall literature review section.

2.1. Methodology for Conducting of Literature Review

We conducted a bibliometric analysis to compile and evaluate the existing research for our literature review. This review included various scientific indexing services such as Scopus, IEEE Xplore, ScienceDirect, among others. The primary focus was on leukemia

detection, with the relevant terms divided into two groups: primary and secondary keywords. Table 2.1 provides the keywords selected for current and proposed search methodology for research.

The literature review utilizes the method of Reporting Matters, such as the PRISMA method for Organized Reviews and Meta-Analyses [37][38]. This method involves inclusion criteria to choose the relevant articles to leukemia detection. The PRISMA flow diagram is shown in Figure 2.1.

Table 2.1 Search keywords for leukemia detection using traditional and artificial intelligence-based approaches

First level keyword	leukemia detection
Second level keyword using AND operator	manual detection, Artificial intelligence-based approaches, Soft Computing algorithm
Second level keyword using OR operator	bone marrow test, flow cytometry, feature extraction, dimensionality reduction, Transfer learning, machine learning, deep learning, CNN, genetic algorithm, image classification

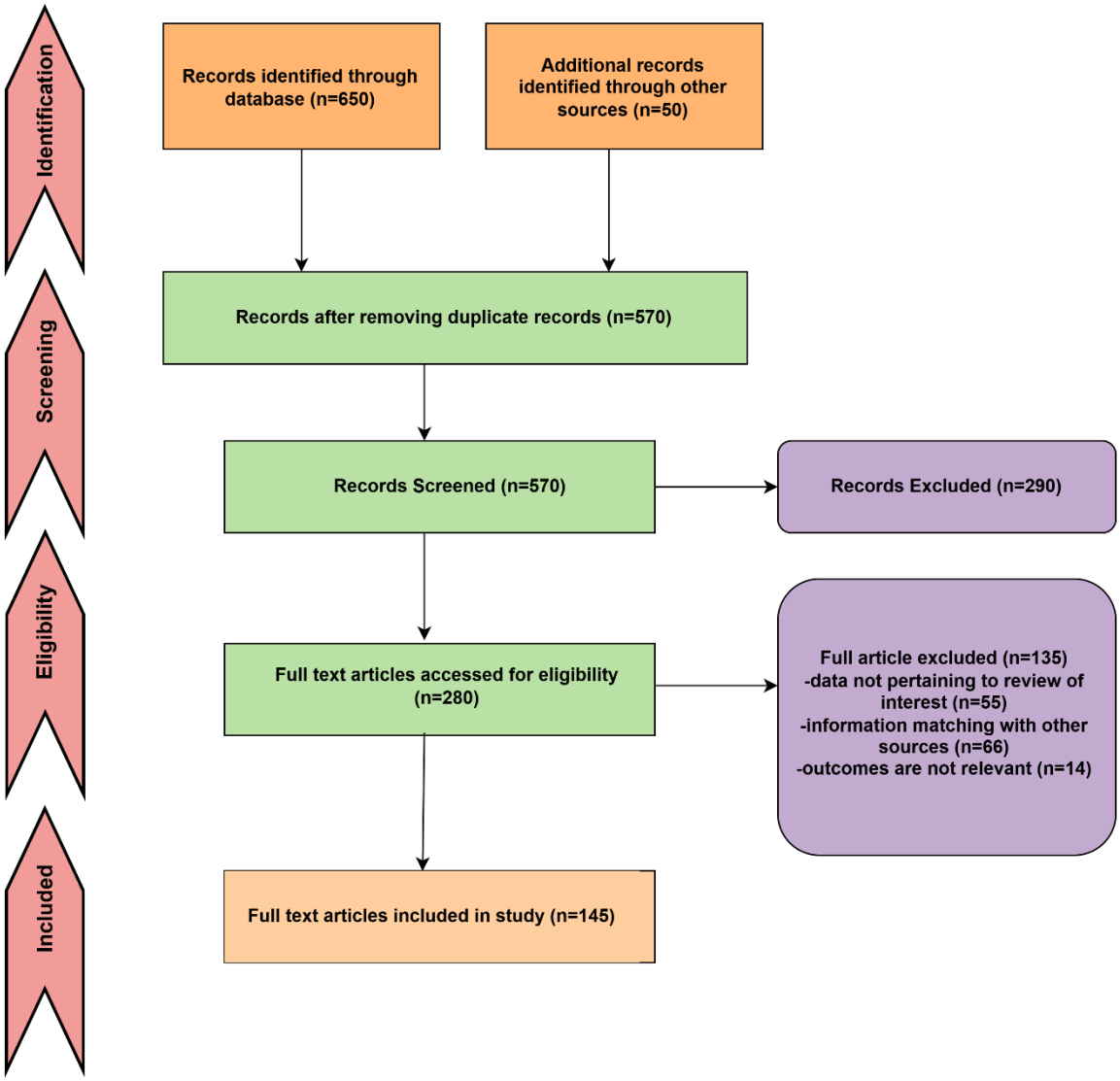


Figure 2.1 PRISMA approach for bibliometric analysis of leukemia detection approaches.

The analysis included research papers published in different journals, conferences, and book chapters. Studies that did not provide enough details about their methods were excluded. This reduced the number of documents to 145, and each of the remaining studies was then evaluated based on its methodology.

2.2. Traditional Methods for Leukemia Detection

Traditional methods have been used for many years to detect leukemia and are important

for diagnosis. These include looking at blood samples under a microscope, doing bone marrow biopsies, using special stains to find leukemia cells, and checking cell markers with flow cytometry. Doctors also study chromosome changes with cytogenetic analysis to identify leukemia types. The strengths and weaknesses of each approach are discussed in the subsections.

2.2.1. Peripheral Blood Smear

Peripheral blood smear is one of the oldest and most basic methods for detecting leukemia. In this method, a drop of blood is placed on a glass slide, stained, and examined under a microscope. Hematologists check for changes in the size, shape, and maturity of blood cells, which may indicate leukemia. This method is simple, quick, and cost-effective, making it useful for initial screening. However, it has some weaknesses. The results can vary because they depend on the examiner's experience, and it is not very effective at detecting small numbers of abnormal cells [39]. Additionally, it does not provide detailed genetic information, which is important for diagnosing and treating leukemia.

2.2.2. Bone marrow biopsy

A bone marrow biopsy usually takes a bone marrow sample from the hip bone for microscopic examination. This is an important procedure for diagnosing leukemia because it provides a detailed look at the cells in the bone marrow and can help detect leukemia cells. However, this procedure is painful and invasive, with risks of bleeding and infection. The interpretation of the results is very subjective and requires a high degree of expertise [40]. Additionally, since the sample comes from only a small part of the bone marrow, it may not fully represent the severity of the disease, which could lead to an incorrect diagnosis.

2.2.3. Cytochemical Staining

In cytochemical staining, special stains are used to highlight specific cell parts in bone marrow or blood samples. These stains help differentiate between different types of leukemia, such as periodic acid-Schiff for lymphoid cells and myeloperoxidase for myeloid cells. Cytochemical staining is useful for classifying leukemia subtypes, but it

is time-consuming and expensive because it often requires multiple steps and specific chemicals [41].

2.2.4. Flow Cytometry

This method uses special antibodies with fluorescent labels to find specific markers on cells. A laser beam is then used to examine the cells. It is very useful for identifying different types of leukemia because it provides accurate details about the cells. However, it is very expensive and needs advanced technology and trained experts. It may also miss rare cells, and understanding the results can be difficult [42].

2.2.5. Cytogenetic Evaluation

Cytogenetic analysis studies the chromosomes of leukemia cells to find genetic abnormalities. This includes techniques like fluorescence in situ hybridization (FISH) and karyotyping. These methods are especially important for detecting certain leukemia types, such as Philadelphia chromosome-positive chronic myeloid leukemia (CML). Cytogenetic analysis provides valuable genetic information, helping doctors understand the disease better. However, the process is time-consuming and requires high-quality samples. Additionally, some genetic defects may not be detected, which can lead to false-negative results [43].

2.2.6. Molecular Diagnostics

The latest methods for detecting leukemia are molecular diagnostics, such as polymerase chain reaction (PCR) and next-generation sequencing (NGS). These techniques help find the specific genetic mutations and translocations with high accuracy. Molecular diagnostics provide detailed and precise genetic information, making them very useful in diagnosis. However, they are expensive and need specialized lab equipment. Additionally, these tests produce a large amount of data, which requires advanced computer tools to analyze. This can be a challenge in some medical settings where such resources are limited [44]. Table 2.2 presents the traditional leukemia detection methods.

Table 2.2 Traditional leukemia detection methods

Method	**Diagnosis Time**	**Cost**	**Strengths**	**Weaknesses**
Peripheral Blood Smear [39]	Fast	Low	Blood smear images are simple to use, easy for diagnosis, and cost-effective, as they require no special equipment or training.	It depends on examiner's skill, may miss small changes, and does not provide genetic details which reduces accuracy.
Bone Marrow Biopsy [40]	Slow	High	A bone marrow biopsy allows doctors to closely examine the marrow's cells, giving them a clear understanding of its structure and function.	Bone marrow biopsy is painful, risky (with infection and bleeding), and may sometimes miss important details, affecting its accuracy.
Cytochemical Staining [41]	Slow	High	Cytochemical staining is useful because it helps identify different types of leukemia, allowing doctors to choose the best treatment for patients.	Cytochemical staining is time-consuming, costly, needs special chemicals and can have unclear results, making interpretation difficult.

Method	Diagnosis Time	Cost	Strengths	Weaknesses
Flow Cytometry [42]	Medium	Very High	Flow cytometry provides detailed information about cell types, helping doctors and researchers understand sample composition. It's very accurate in finding leukemia.	Flow cytometry is expensive, needs advanced machines, requires skilled staff and may miss rare cells.
Cytogenetic Evaluation [43]	Slow	High	It is helpful for diagnosing certain types of leukemia by finding the genetic problems causing the disease.	Cytogenetic analysis is slow, requires good samples, and may miss some genetic issues, leading to incomplete diagnoses.
Molecular Diagnostics [44]	Medium	Very High	Molecular diagnostics provides accurate and detailed genetic information, helping doctors understand diseases better and create targeted treatments.	It's very expensive and needs advanced equipment.

Traditional methods for detecting leukemia, such as peripheral blood smear, bone marrow biopsy, cytochemical staining, flow cytometry, cytogenetic evaluation, and molecular diagnostics, are important for diagnosis. Each method has strengths and weaknesses as presented in Table 2.2. Some methods, like blood smear, are fast and affordable, while others, like molecular diagnostics, provide very accurate genetic details but are costly and need special labs. Many of these methods take a long time, require skilled experts, and may not detect rare or small changes in cells. Because of these challenges, AI-based leukemia detection is becoming more important.

2.3. Artificial Intelligence based Methods for Leukemia Detection

Artificial Intelligence can process large amounts of medical data quickly and accurately, reducing human mistakes and making diagnosis more efficient. Various AI-based techniques, such as machine learning, deep learning, neural networks, fuzzy logic, soft computing, and optimization methods, are being used to develop hybrid models for the fast and accurate detection of leukemia. These AI techniques are useful because they function like the human brain [45-47]. They can quickly identify the patterns and connections in data. This section presents various AI-based methods that combine different techniques to improve leukemia detection, along with their strengths and weaknesses. Researchers in various studies have proposed different models for leukemia detection along with their challenges in leukemia detection. Section 2.3.1 presents a literature review of machine learning-based models for leukemia detection. Section 2.3.2 focuses on deep learning-based models for leukemia detection. Section 2.3.3 reviews various hybrid models that integrate machine learning, deep learning, and soft computing-based techniques

2.3.1 Machine Learning based model for leukemia classification

This section presents various machine learning-based studies used for leukemia detection. It also discusses different machine learning classifiers and techniques such as feature extraction, segmentation, feature selection, and data augmentation, as explored in various studies.

Mamun et al. introduced an automated system for recognizing acute lymphoblastic

leukemia using peripheral blood smears [48]. The method began by using color thresholding to separate lymphocyte blood cells from the smear. Additional post-processing methods involved using morphological operations and the watershed algorithm to segment individual lymphocyte cells. Statistical features were then extracted from these segmented cells to form a feature vector followed by a support vector machine (SVM) classifying the images based on the feature vectors. Similarly, with the use of an ALL IDB dataset, the proposed framework outperforms the baseline by reaching 99% accuracy, 98% sensitivity, 99% specificity, 99% precision, and 99% F1-score. Although the proposed model achieved outstanding accuracy, the ALL-IDB dataset contains a limited number of images.

Liu and Hu developed an approach which classifies the Acute Myeloid Leukemia automatically. This study utilized total 50 bone marrow smear images of patients suffering with AML images from the Cancer Imaging Archive database [49]. The proposed model used random forest classifier and broad learning system for the classification purpose. The results were analyzed with a random forest model, BLS approach, and subtype classification. The model can classify AML into its subtypes, AML M1 and AML M2. The model provides an accuracy of 99.8% but a very small number of bone marrow images were used for training.

Gupta et al. concentrated on the CNMC dataset, which includes more than 15,000 high-resolution microscopic images of B-Lineage Acute Lymphoblastic Leukemia (B-ALL) [50]. The study tried to fix some of the problems with current machine learning tools for diagnosing B-ALL. One of the main issues with the ML tools is that these tools rely on some hand-picking important features and using small datasets. It detailed the IEEE ISBI 2019 medical imaging challenge, where the training and test classification was conducted using the CNMC dataset. The study reviewed various machine learning methods for B-ALL diagnosis and found the best result in a recent competition achieved a score of 91%, which shows how much impact AI is making on improving B-ALL diagnosis and patient outcomes. This large, curated dataset from 118 subjects offers a solid base for building better AI tools, though its real-world impact depends on further testing with diverse cases.

Das et al. developed a new approach to detect leukemia. First, they pre-processed the input images by cleaning them up and removing extra data. Then, using an innovative blast cell segmentation technique, termed as Gini index-based Fuzzy Naïve Bayes, to identify and separate the cancer cells (blast cells) from the rest of the image [51]. This integrated the Gini Index for optimal feature selection with a Fuzzy Naïve Bayes classifier, refining the identification process. A new grading system has also been developed to accurately diagnose leukemia by counting the number of blast cells. An analysis of a large database of images revealed that this approach outperforms current methods, with an accuracy of 96%. However, this relies heavily on high quality blood smear images to sustain the preprocessing well.

Saleem et al. provided the detailed view of various phases involved in AI based leukemia detection such as pre-processing of data, segmentation, feature extraction, feature selection, and image classification [52]. It is also discussed that most of the studies are conducted using ALL-IDB1, ALL-IDB2, ASH and other local datasets which are small. The overfitting problem occurs due to the unavailability of large balanced dataset. Due to the less images and imbalanced dataset, the classification accuracy is also decreased as deep learning methods need a large dataset for training a mode. So, this study explained in detail about the steps of image classification along with the challenges associated with the small or local datasets.

Maurya et al. presented several ML and DL concepts applied for early detection and diagnosis of five types of cancers, namely: brain tumor, cervical cancer, breast cancer, skin cancer, and lung cancer [53]. This study looked at many different ML and DL methods, and compared them to each other using standard datasets and evaluation metrics. It also included a table that listed the benefits and drawbacks of each approach. Additionally, the study identified the biggest research challenges as the limited dataset availability for each type of cancer.

Bharat et al. discussed in detail about the importance of nature-inspired computing methods in improved medical diagnostics, including disease detection and management [54]. Techniques like Genetic Algorithms (GAs) and optimization methods have made

significant progress in solving complex problems. These methods create a group of possible solutions and gradually improve them through processes like selection, crossover, and mutation, like how evolution works in nature. Over several generations, these processes lead to better solutions. This approach helps in selecting important features, classifying data, and optimizing processes, improving the accuracy of leukemia diagnosis. Research of different soft computing-based search and optimization techniques for leukemia detection has been reviewed, highlighting their benefits and limitations.

Bouchet et al. explained a new approach for the segmentation of leukocytes in colour images [55]. The dataset consisted of one hundred colour images with acquisitions from the Cella vision blog. The algorithm chose a pixel identified as a white blood cell and used a similarity metric against the entire image to emphasize it over the background. The experimental results using this algorithm are of high performance, with 99.41% correctly classified leukocytes and 99.23% for the background. The accuracy of 99.32%, precision of 99.41%, and recall of 99.24%, respectively was achieved. Despite the good accuracy of the proposed model, the dataset used for training has only 100 images, which could lead to overfitting.

Acharya et al. introduced a computer-aided diagnosis model for segmenting blood smear images and identifying AML stages [56]. The process involved capturing images, dividing them, extracting/selecting features, and categorizing them. Using 800 images from Kasturba Medical College Manipal and 200 from another dataset for training, and 500 images for testing, the model utilized a new segmentation algorithm to precisely categorize AML stages, detect blast cells, extract cytoplasm, and separate overlapping cells. Feature selection used InfoGainAttributeEval and ranker search methods. The model achieved 99.81% accuracy in differentiating NRBC from WBC and an overall classification accuracy of 99.48%. The dataset used in this study is very limited and due to this there is instability in the model.

Chen et al. demonstrated cellular feature engineering, using unsupervised clustering for identifying cellular phenotypes, offers quite enhanced analytic performance to analyze

digitized pathology slides, with an accuracy rate 0.925 and an AUC was 0.978. This approach outperformed mixed and supervised features, and various methods for feature fusion and selection, and patch-based CNN feature extraction as well [57]. The effectiveness of unsupervised feature extraction was demonstrated by analyzing stability and repetition of splitting, showing it is a useful diagnostic tool to identify CLL patients who show signs of disease progression in their tissue samples. Its strength could help pathologists in tough cases, but combining it with molecular data might provide even deeper insights.

Shukla et al. suggested utilizing a Naive Bayes classifier as a fitness function to selectively identify highly effective genes to improve the accuracy of cancer classification [58]. Its effectiveness was assessed on ten biological datasets and contrasted with top computational intelligence methods for tumor prognosis. The suggested technique, which relies on experimental findings and statistical evaluations, surpasses current metaheuristic methods notably in convergence speed, classification precision, and optimal feature set identification. Among them, in six datasets, a classification accuracy of more than 98% was attained, and the maximum accuracy was achieved in the DLBCL dataset. This method's hybrid TLBOGSA design enhances global search efficiency significantly. Table 2.3 shows the machine learning based leukemia detection approaches along with the limitation of every approach.

Table 2.3 Machine Learning based leukemia detection methods

Study	Cancer Type	Dataset	Methodology	Accuracy & Findings	Limitations
Mamun et al. [48]	ALL	ALL-IDB	Color thresholding, Watershed segmentation, SVM classifier	99% accuracy, 98% sensitivity, 99% specificity, 99% precision, 99% F1-score	Small dataset size
Liu & Hu [49]	AML	Cancer Imaging Archive	Random Forest & Broad Learning System (BLS)	99.8% accuracy	Only 50 images used

Study	Cancer Type	Dataset	Methodology	Accuracy & Findings	Limitations
Gupta et al. [50]	B-ALL	CNMC dataset (15,000 images)	ML-based classification, IEEE ISBI 2019 Challenge	91% accuracy	Needs real-world validation
Das et al. [51]	Leukemia	Large database (unspecified)	Gini Index-based Fuzzy Naïve Bayes for blast cell segmentation	96% accuracy	Requires high-quality images
Saleem et al. [52]	Leukemia	ALL-IDB1, ALL-IDB2, ASH, local datasets	AI-based leukemia detection steps (pre-processing, segmentation, feature extraction, classification)	Highlights challenges of small/imbalanced datasets	Overfitting due to small datasets
Maurya et al. [53]	Brain, Cervical, Breast, Skin, Lung Cancer	Various datasets	ML & DL comparison	Discusses strengths/weaknesses of ML/DL	Limited datasets for each cancer type
Bharat et al. [54]	Leukemia	Multiple datasets	Nature-inspired computing (Genetic Algorithms, Optimization)	Enhanced leukemia diagnosis accuracy	Computational complexity
Bouchet et al. [55]	Leukemia	100 color images (Cella Vision blog)	Leukocyte segmentation using similarity metric	99.32% accuracy	Small dataset (100 images)
Acharya et al. [56]	AML	Kasturba Medical College Manipal (800 images) + external (200 images)	New segmentation algorithm, feature selection (InfoGainAttributeEval)	99.81% accuracy for NRBC vs WBC, 99.48% overall	Small dataset causes instability
Chen et al. [57]	CLL	Pathology slides	Unsupervised clustering & cellular feature engineering	92.5% accuracy, AUC 0.978	Needs molecular data integration

Study	Cancer Type	Dataset	Methodology	Accuracy & Findings	Limitations
Shukla et al. [58]	Cancer (Multiple types)	10 biological datasets	Naïve Bayes classifier + TLBOGSA hybrid for gene selection	>98% accuracy on 6 datasets	Computational complexity

Machine learning models have shown high accuracy in detecting leukemia, often over 95%. Many studies have used advanced techniques for classification, feature extraction, and augmentation to improve diagnosis. Some have also introduced new ways to segment images and select important features, making the models more effective. However, many studies still use small and unbalanced datasets. Some only used 50 or 100 images, which can make the models too focused on just that data and not work well on new data. A few did not clearly explain where their data came from. Without enough diverse data, these models may not work well in medical environments. Various research suggests that selecting better features can improve accuracy. To automatically extract the features, there was the need of deep learning-based approaches.

2.3.2 Deep Learning based approaches for Leukemia Detection

This section presents various existing deep learning-based studies. The most of these studies include CNN-based models to improve feature extraction and classification. Different deep learning architectures like ResNet, VGG, and DenseNet have been tested which automatically extract the features and no manual feature extraction is required.

Sharma et al. developed a microscopic images-based leukemia classification model using Convolutional Neural Network [59]. The designed model is trained and tested using 108 images of ALL IDB1 dataset, which is publicly available. As the ALL IDB1 dataset does not include segmented images of the blood cell, so a segmentation technique using nature inspired based approaches is also built. Using the proposed segmentation technique, the blood cells are segmented and further the training and classification is performed using CNN. The classification accuracy with the proposed technique noticed around 99%. Although, this proposed leukemia classification model provides a very

good accuracy but the training and testing is performed on a very small dataset of 108 images only. So, there is a problem of overfitting and bias in predictions.

Abhishek et al. performed a binary classification task that was able to achieve 97% accuracy based on the fine-tuned VGG16 model [60]. On the three-class classification, ResNet50 with SVM achieved 95% accuracy while DenseNet121 with SVM achieved 98.5% for AML and 97.1% for ALL. Average accuracies for three-class classification were 96.67% with ResNet50, 94% with VGG19, 93.67% with DenseNet121, and 93.67% with MobileNet. In conclusion, deep learning methods are successful at classifying acute leukemia. The purpose of this paper was to introduce this dataset for future investigation and this work introduced a novel dataset of 500 images, expanded with ALL-IDB, to support high accuracies.

Anagha et al. proposed a white blood cell detection tool using a CNN [61]. In this work, the Keras library along with TensorFlow was used as the backend. The model was subjected to training and testing on the cancer cell dataset CNMC2019, comprising images of segmented white blood cell regions from microscopic blood smears. It yielded a significant performance i.e. 91% with training set, and 87% with testing set. This solid accuracy shows CNNs can ease manual diagnosis, though larger datasets could strengthen it further.

Anil et al. have concentrated on different deep learning methods for categorizing acute lymphoblastic leukemia [62]. This study used a type of artificial intelligence called deep convolutional neural networks to divide Acute Lymphoblastic Leukemia (ALL) into different categories based on the World Health Organization's classification system. This approach was able to do this without needing to use complex and time-consuming methods like image segmentation and feature extraction. They achieved an accuracy of 94.12% in classifying B-cell and T-cell ALL images by utilizing the complete capabilities of a pre-existing CNN named AlexNet, along with the custom deep learning network, LeukNet.

Abas et al. proposed a computerized detection system for diagnosing leukemia using deep learning algorithm. They developed a CAD3 system that detects and classifies three

varieties of white blood cells [63]. The study used a modified version of the YOLO v2 algorithm, which is a type of deep learning technique. This algorithm was combined with another technique called CNNs, or convolutional neural networks. The data used to train the algorithm was specially prepared from 15 patients, and it did not use traditional image segmentation or preprocessing techniques. By dividing the problem into smaller parts, the algorithm was able to perform much better. CAD3 is capable of recognizing leukocytes with an average precision of 96% and classifying them with an accuracy of 94.3%. Moreover, CAD3 provides detailed WBC reports and its efficiency was further validated with additional datasets, including ALL-IDB1 and BCCD.

Vieira et al. addressed CNNs defined on hypercomplex algebras to lymphocyte classification in digital microscopic images of blood smears, an important task in ALL diagnosis [64]. They ran eight HvCNNs and real-valued CNNs on this classification task. In their experiments, HvCNNs have outperformed their real-valued counterparts in terms of accuracy when using significantly fewer parameters. Notably, the highest accuracies that have appeared were exactly clifford algebras and HSV-encoded images. The presented approach was able to reach a mean accuracy of 96.6% for ALL-IDB2 using a 50% train-test split, on par with the existing models but employing a much simpler network with less parameters. This simplicity could help in basic settings, but it needs testing on bigger datasets.

Rastogi et al. introduced "LeuFeatx," a fine-tuned approach for leukemia diagnosis [65]. LeuFeatx excels in extracting critical features from the single-cell leukocyte microscopic images, outperforming the original VGG16 model. Analyses on three public datasets of white blood cells showed that classifiers using LeuFeatx features had better precision and sensitivity than recent studies on AML datasets, for seven different types of white blood cells. Utilizing the ALL_IDB2 dataset for binary classification, LeuFeatx demonstrated a 96.15% accuracy, highlighting the strength of its extracted characteristics. Overall, this approach emerged as a promising tool for the automation of leukemia diagnosis, however, future work suggests the need for larger, diverse, and high-quality datasets.

Mallick et al. developed a way to identify when deep neural networks (DNNs) have converged by considering certain factors. These factors include using inputs that are not repetitive, having too many neurons in the network, and having enough hidden layers [66]. This approach employed a DNN to classify gene expression data, with a specific focus on bone marrow samples. Accordingly, a classifier with five layers is designed to differentiate ALL from AML. The network is trained with 80% of the data and provides the remaining 20% of the data for testing. The proposed DNN classifier brought an outstanding output that differentiated leukemia with an accuracy of 98.2%, sensitivity of 96.59%, and specificity of 97.9%. The proposed approach achieved good accuracy, but the dataset details are not clearly provided. Therefore, it is unclear how many images were used for training.

Ansari et al. created a new image dataset showing Acute Lymphoblastic Leukemia (ALL) and Acute Myeloid Leukemia (AML). The images in this study were obtained from Shahid Ghazi Tabatabai Oncology Center in Tabriz ,totaling 184 images of ALL and 469 images of AML . They worked closely with experts in the field to help other scientists use machine learning to study and understand these diseases better [67]. In this study a Deep Neural Network (DNN) based on combined GAN and CNN is developed to learn the optimal features. The reason for using the combination of GAN and CNN networks in this work is that the data limitation in training has been solved by using GAN and the CNN classifies acute leukemia cells by using a simple, customized, end-to-end architecture. The proposed CNN model included six layers for convolution and four layers for dense, using softmax as the activation function coupled with the Tversky loss function for classifying images of acute leukemia. The model was able to accurately distinguish between ALL and AML with a very high rate of 99%. The model was fast and precise, making it a useful tool for doctors and specialists in a clinical setting. However, its limitation is the use of a small dataset.

Ananthu et al. conducted an extensive comparative evaluation of various deep learning models used to identify ALL in blood smear cell images, including those that were trained on the ALL-IDB2 dataset [68]. It was demonstrated that the performances amongst Xception, InceptionV3, and DenseNet201 were 87.97%, 88.92%, and 88.92%,

respectively, whereas ResNet50 and MobileNet gave the best performances amongst all with an accuracy of 95.28% and 97.88%, respectively. The limitation of the study is the small dataset used for training the model.

Elhassan et al. proposed an advanced technique to deal with the imbalanced WBC distribution in blood samples by adopting an integrated approach using geometric transformation and deep convolutional autoencoder, namely the "GT-DCAE WBC augmentation model" [69]. Moreover, the perfect learning framework that incorporates WBC segmentation into deep learning for context-independent feature extraction is presented. Based on the designed architecture "two-stage DCAE-CNN a typical WBC classification model", the authors have classified a typical WBCs into eight classes. The obtained results were outstanding, classifying with 97% average accuracy, reaching a sensitivity of up to 97%, and a precision of 98%. Though the model achieved good accuracy, the dataset used to train the model is not clearly mentioned.

Sridhar et al. proposed simple enhancements to the traditional neural network architectures and produced outstanding results in classifying malignant leukocytes [70]. The developed method showed marked interest for non-malignant and malignant cell recognitions with high precision. A deep learning approach was developed for classifying leukemic B-lymphoblasts. This study had a small dataset ALL-IDB and ASH, so they used a technique called data augmentation to make the data bigger. They also used a technique called transfer learning, which helps the network learn faster. This approach worked even better than a single network, and it was able to accurately identify leukemic B-lymphoblasts 95.59% of the time.

Chand et al. presented a novel framework for the diagnosis of ALL based on convolutional neural networks [71]. It required neither the extraction of features nor pre-training, which made it particularly suitable for real-time applications. It involved only 41,626 tunable parameters, much less than those in the existing deep networks like AlexNet, VGG-Net, and ResNet 152. Despite the simplicity of this model, it showed 100% accuracy for most of the different trials with a mean of 98.17% in two experiments: 98.62% with data split 80%-20% run over 20 epochs and 97.73% for a

60-48 image split run over 30 epochs. The dataset used for training and testing the model was ALL-IDB, which included fewer than 300 images.

Muhamad et al. created a model for detecting and classifying white blood cells into five main types [72]. Manual identification of different WBC types is prone to errors, so an automatic system is preferred for accurate classification. Data were gathered from patients at the Hiwa Cancer Hospital in Sulaymaniyah, Iraq, encompassing five categories: basophil (37 images), eosinophil (440 images), lymphocyte (380 images), monocyte (421 images), and neutrophil (450 images), totaling 1,728 images. The classification utilized three distinct models, with the CNN model achieving an accuracy rate of 95.3%, MobileNetV2 reached 97.6%, and AlexNet attained 81.6%. This research created a standard for testing the accuracy of deep-learning models for detecting small blood cells in peripheral blood.

Mohsin et al. proposed a technique for blood smear classification based on several deep learning architectures: SqueezeNet, ResNet-50, and AlexNet [73]. The backbone of their work is the utilization of the ALL dataset. They presented such models as the most suitable to this domain. AlexNet with 99% accuracy outperformed all other deep learning models such as SqueezeNet and ResNet-50. Although the models presented in this study achieved very good accuracy, they were trained on a very small dataset, which may lead to the problem of overfitting.

Dibouliya et al. introduced a combined model which consisted of artificial neural networks and fuzzy logic to improve the overall performance of classification [74]. Since ResNet's backbone architecture achieved only 97% accuracy, efforts were made to create a solution with improved performance. Their new technique utilized the artificial neural network's effectiveness in task classification and integrated it with fuzzy logic to enhance feature extraction by managing the uncertainties in feature extraction. While combining these two, the hybrid model performed far better than the overall performance of these two methodologies. Simultaneously, this combinative technique covered the loophole of the traditional neural network model and achieved an accuracy rate as high as 99%. This new model shows that combining different

approaches can result in more accurate and reliable results, especially when classifying complex data. However, potential weakness in the paper includes increased computational complexity and need of validation on larger and diverse datasets.

Boreiri et al. presented a new method for ALL detection from microscope cell images using a deep convolutional neuro-fuzzy network [75]. To increase the sample size, they used a data augmentation technique before training, since overfitting was occurring. Images were pre-processed by two-stage fuzzy colour segmentation, allowing only leukocytes, reducing image sizes to 64×64 pixels containing each single nucleus. Currently, the TSK fuzzy system model has reached an impressive average accuracy level of 97.31% when detecting ALL. This research has shown that using deep network architectures on a dataset of ALL-IDB1 data with a small sample size can improve performance. In this research, two new operations have been proposed for the TSK model.

Aftab et al. proposed a way to diagnose leukemia using a special type of artificial intelligence called deep transfer learning CNN architecture [76]. They looked at small pictures of human blood cells and found that this method worked well. This model successfully recognized four kinds of leukemia. The approach utilized SBDL framework along with GoogleNet and obtained 97.33% accuracy. During the comparison, the model BigDL achieved a training accuracy of 96.42% and a validation accuracy of 92.69%. Overall, the BigDL model excels over the Keras model in performance and accuracy. The study acknowledges the potential limitations including computational complexity of the approach.

Akalin et al. addressed the diagnosis of Acute Lymphoblastic Leukemia (ALL), a major leukemia type defined by the rapid overproduction of immature and cancerous white blood cells known as lymphoblasts [77]. Due to the aggressive nature of ALL, where the disease quickly spreads through the blood and to vital organs, rapid testing for early detection is crucial. This system employed the YOLOV4 algorithm and has been trained using images from the freely available ALL-IDB dataset, which are annotated by expert oncologists. The system was tested and achieved impressive success metrics, including

a success rate of 98.87%, precision of 99.08%, recall of 99.81%, specificity of 88.88%, and an F1- score of 99.44%. The challenge in this study is also the size of the dataset. The model is trained using less than 300 images.

Chand et al. introduced a CNN for the detection of Acute Lymphoblastic Leukemia (ALL). Unlike traditional methods, the suggested approach does not require feature extraction or pre-training on external datasets, making it well-suited for real-time detection of leukemia [78]. This streamlines the process, enhancing its practicality for immediate diagnostic use. The framework is simpler compared to existing deep networks, with only 41,626 tunable parameters, significantly fewer than those in pre-trained networks like AlexNet (over 60 million), VGG-Net has a parameter size of 138 million, while ResNet 152 has a parameter size of 60.3 million. Having fewer parameters enabled the framework to function on a simple processor without requiring a GPU. Despite its lower number of parameters, the model achieved 98.17% accuracy. This method works well for smaller setups, but handling larger datasets could be a challenge.

Mondal et al. utilized deep learning methods, specifically CNNs, to detect Acute Lymphoblastic Leukemia [79]. This method highlighted creating a weighted ensemble model by combining various CNN architectures to enhance classification accuracy. Various methods for preprocessing and augmenting the data were experimented to enhance generalization abilities. Each test was conducted using the publicly available C-NMC-2019 dataset. It was surprising that the ensemble model, which assigns weights based on kappa values in decreasing order, achieved an impressive accuracy of 86.2%. This strong performance shows it could be useful in clinical settings, but testing on different patient groups could make it even more reliable.

Ghaderzadeh et al. introduced a public dataset for Acute Lymphoblastic Leukemia (ALL) detection, consisting of 3,562 peripheral blood smear (PBS) images from 89 patients, including 25 healthy people and 64 ALL patients [80]. After segmenting with colour thresholding in the HSV colour space using a two channelled network, various CNN models were utilized for extracting features. DenseNet201 demonstrated the highest performance in diagnosis and classification. The developed deep learning model,

utilizing DenseNet201 as its foundation, attained an accuracy: 99.85%, sensitivity: 99.52%, specificity: 99.89%. These near-perfect results show its diagnostic accuracy, but testing on larger, multi-center datasets could prove it works widely.

Vogado et al. introduced LeukNet, a convolutional neural network (CNN) motivated from convolutional blocks of VGG-16, however, featuring smaller dense layers [81]. The settings in LeukNet were selected after assessing different CNN models and fine-tuning techniques on 18 image datasets with varying resolution, colour, contrast, and texture features. The training dataset was expanded using data augmentation, leading to an accuracy of 98.61% in 5-fold cross-validation. To assess generalization, cross-dataset validation was used, achieving accuracies of 97.04%, 82.46%, and 70.24% on three different datasets, exceeding the current top methods. The study ended by stating that using deeper and more general CNNs might not be the best for tasks where the images used for classification are different from those used in pre-training. Also, validating across different datasets is highlighted as an excellent approach for evaluating a model's generalization capability, crucial for CAD systems. Table 2.4 presents the existing deep learning-based leukemia detection approaches along with the limitation of every approach.

Table 2.4 Deep Learning based leukemia detection methods

Study	Cancer Type	Dataset	Methodology	Accuracy & Findings	Limitations
Sharma et al. [59]	ALL	ALL-IDB1 (108 images)	Nature-inspired segmentation + CNN	99% accuracy	Very small dataset, risk of overfitting and bias
Abhishek et al. [60]	AML, ALL	Novel dataset of 500 images + ALL-IDB	VGG16, ResNet50+SVM, DenseNet121+SVM	Binary: 97%; 3-class: up to 98.5%	Small dataset; need for validation
Anagha et al. [61]	WBC Detection	CNMC2019	CNN (Keras + TensorFlow)	91% (train), 87% (test)	Limited dataset size
Anil et al. [62]	B-cell, T-cell ALL	Not specified	AlexNet + custom CNN (LeukNet)	94.12% accuracy	Dataset details missing
Abas et al. [63]	WBCs	Custom (15 patients), ALL-IDB1, BCCD	Modified YOLO v2 + CNN (CAD3 system)	Avg. precision 96%,	Not using traditional segmentation; limited samples

Study	Cancer Type	Dataset	Methodology	Accuracy & Findings	Limitations
				classification 94.3%	
Vieira et al. [64]	Lymphocyte (ALL)	ALL-IDB2	Hypercomplex CNNs (HvCNNs)	Mean accuracy: 96.6%	Needs testing on bigger datasets
Rastogi et al. [65]	ALL, AML	ALL-IDB2 + others	Fine-tuned DNN (LeuFeatx)	96.15% accuracy	Small datasets, need for diversity
Mallick et al. [66]	ALL, AML	Gene expression (unclear size)	DNN (5 layers)	98.2% accuracy, high sensitivity and specificity	Dataset size not clear
Ansari et al. [67]	ALL, AML	184 ALL, 469 AML	GAN + CNN (custom architecture)	99% accuracy	Small dataset
Ananthu et al. [68]	ALL	ALL-IDB2	CNNs (Xception, InceptionV3, etc.)	Best: MobileNet 97.88%, ResNet50 95.28%	Small dataset
Elhassan et al. [69]	WBCs	Not specified	GT-DCAE + 2-stage CNN	Avg. 97% accuracy, 98% precision	Dataset not clearly mentioned
Sridhar et al. [70]	B-lymphoblasts (ALL)	ALL-IDB, ASH	Transfer learning + Data augmentation	95.59% accuracy, improved malignant vs. non-malignant classification	Small dataset
Chand et al. [71]	ALL	ALL-IDB (<300 images)	CNN without feature extraction or pre-training	98.17% avg. accuracy, 100% in trials	Small dataset
Muhamad et al. [72]	WBC classification	Hiwa Cancer Hospital (1,728 images)	CNN, MobileNetV2, AlexNet	MobileNetV2: 97.6%, CNN: 95.3%, AlexNet: 81.6%	Small dataset for basophils (37 images)
Mohsin et al. [73]	ALL	ALL dataset	SqueezeNet, ResNet-50, AlexNet	AlexNet: 99% accuracy	Small dataset, risk of overfitting
Dibouliya et al. [74]	Leukemia	Not specified	Hybrid ANN + Fuzzy Logic	99% accuracy	High complexity, needs larger dataset validation

Study	Cancer Type	Dataset	Methodology	Accuracy & Findings	Limitations
Boreiri et al. [75]	ALL	ALL-IDB1	Neuro-fuzzy CNN + TSK fuzzy segmentation	97.31% accuracy	Small dataset
Aftab et al. [76]	Leukemia (4 types)	Not specified	SBDL + GoogleNet, BigDL	GoogleNet: 97.33%, BigDL: 96.42%	Computationally complex
Akalin et al. [77]	ALL	ALL-IDB (<300 images)	YOLOV4	98.87% accuracy, F1: 99.44%	Small dataset
Chand et al. [78]	ALL	ALL-IDB	Lightweight CNN (41,626 params)	98.17% accuracy	Struggles with large datasets
Mondal et al. [79]	ALL	C-NMC-2019	Weighted CNN ensemble	86.2% accuracy	Needs testing on different patient groups
Ghaderzadeh et al. [80]	ALL	3,562 PBS images (89 patients)	DenseNet201 + HSV thresholding	99.85% accuracy, high sensitivity & specificity	Needs multi-center validation
Vogado et al. [81]	ALL	18 datasets incl. ALL-IDB	LeukNet (light VGG-style CNN)	Cross-validation: 98.61%, Cross-dataset: up to 97.04%	Lower accuracy on diverse datasets

Deep learning models, especially CNNs, are highly accurate in detecting leukemia as they perform automatic feature extraction. However, some models are complex and require a lot of computing power. Further, the performance and the stability of these models can be enhanced by hyperparameter tuning and integrating optimization algorithms.

2.3.3 Soft Computing based hybrid approaches

This section reviews several studies that use techniques like CNN models, transfer learning, genetic algorithms, and hybrid machine learning methods. These studies focus on improving feature extraction, classification, and model performance. They use advanced methods such as nature-inspired algorithms, ensemble learning, and boosting techniques to achieve better results.

Chadaga et al. utilized three nature inspired-based feature selection techniques i.e. Harris

Hawks optimization, Salp Swarm optimization and mutual information for the selection of the features. Based on the selected features, different machine learning and explainable artificial intelligence (XAI) methods such as SHAP, LIME, EL15 and QLattice are used for assessing the survival rates among patients receiving hematopoietic stem cell transplantation [82]. A blood cell images dataset of 186 cancerous and non-cancerous patients is considered in this study. Harris Hawks Optimization algorithm has selected 10 features, salp swarm optimization has selected 13 features and Mutual information has selected total 10 features. Random forest, logistic regression, decision tree, K-nearest neighbors classifiers were stacked together to design the model STACKA. Adaboost, catboost and xgboost boosting algorithms were stacked together to design the model STACKB. STACKA and STACKB were combined to designed another model STACKC which provided the accuracy of 89% to predict the effectiveness of bone marrow transplant. This study provided a best view of various feature selection techniques, machine learning classifiers and the boosting algorithms but the size of the dataset used is small.

Vishwaraj et al. discussed machine learning platform for cancer cell detection based on Waveguide Bragg Gratings [83]. Feature reduction is done with PCA and the neural network with multi-layer perceptron is utilized and are further connected to fully connected layers. The model used light reflection to find cancer and achieved an accuracy around 95%. It is mentioned that 666 images dataset was generated with the variations in the RI of the sample solution. Since, this study only used the generated data, it has not been checked with real samples.

Wu et al. introduced a soft-computing approach for analyzing histopathology images with overlapped cells, specifically targeting lymphoma [84]. Their technique combines a deep learning network with a genetic algorithm to detect and separate overlapping cells and enhance the elliptical pattern of each cell. The uniqueness of this method is found in its implementation to medical image segmentation and analysis. Tested on a dataset of 50 lymphoma images, the method attained an average F1-score of 0.87 for cell segmentation and a Jaccard index of 0.77 for cell splitting. This approach could also be

further extended to other histopathology images for tasks like cell classification and feature extraction. A major limitation of this study is that it needs someone to manually go through the training data and label it, which can be a time-consuming and tedious task. Additionally, the study only used a small dataset to evaluate the model, which may not be representative of a larger population. Finally, the study did not compare its results to other advanced methods, which makes it difficult to know how it compares to other approaches.

Priyadarshini et al., in their critical review related to research in skin cancer detection and classification, had focused on enhancing computer vision by using a machine learning approach [85]. They proposed a hybrid algorithm, named ELM-TLBO, by combining Extreme Learning Machine (ELM) with Teaching-Learning-Based Optimization (TLBO). This approach includes median filtering and fuzzy C-Means for image segmentation. The paper looked at what has already been done in this area and suggests some ideas for future research. It is analyzed that future research should focus on making the detection of leukemia more accurate and faster. The paper also mentioned that deep learning is being used more and more to help machines recognize and classify medical images. Experimental results showed high accuracy in melanoma detection. 300 skin images were used for the testing and the proposed approach achieved 93.18% accuracy, outperforming several existing methods. The method's strength lies in its quick training and effective parameter tuning, though it may stumble with complex image patterns.

Nematzadeh et al. compared the GA with GWO and hence with EGS to make a point that metaheuristic methods were much more efficient and much faster to reach the best hyper-parameters [86]. This study has shown that GWO outperforms GA by obtaining a better performance with faster convergence. It improved performance in different biological and biomedical datasets, and it is directly recommended for non-machine learning expert users. A practical and effective way has been proposed for performing the tuning of hyper-parameters with better applications towards reaching accuracy in machine learning models. The testing was performed using 11 diverse datasets with 11

algorithms. It proves versatile, though its effectiveness may vary with highly complex or imbalanced data.

Baby et al. concentrated on refining an ALL-detection system based on transfer learning that is efficient and simple [87]. One of the most advanced deep learning models, EfficientNet, was used to tackle the problems associated with feature extraction. This study made use of eight different variations of EfficientNet for feature extraction and presented a comparison of their results in terms of classification accuracy. An ensemble approach was executed, which combined three advanced classifiers: SVM, random forest, and logistic regression. Therefore, the suggested system reached a classification accuracy of 98.5%. The potential for reliable diagnosis of ALL at high accuracy levels in clinical settings allowed the proposed system to operate quickly. The challenge associated with this study is the missing details of the dataset.

Devi et al. presented a Convolutional Leaky RELU design integrated with CatBoost and XGBoost for important feature extraction in image segmentation followed by classification [88]. The recommended approach involves binary classification and gradient boosting methods using CatBoost and XGBoost. The goal of developing convolutional leaky RELU with CatBoost is to reduce bias and achieve high accuracy. Conversely, combining Convolutional Leaky RELU with XGBoost is effective for addressing classification and regression issues, leading to improved algorithm performance and speed. In this work, test images are classified as ALL by using the CLR-CXG method. The proposed model achieves an accuracy of 97.12%; however, a local dataset was used for training, and its details are not provided.

Rejula et al. introduced an enhanced ANFIS (Improved ANFIS or I-ANFIS) model for predicting leukemia data, utilizing Euclidean distance to compare trained and test feature data [89]. A novel ANFNN has been introduced, aiming to reduce computational complexity by partitioning the input space into localized regions using fuzzy clustering. The total count of fuzzy rules is determined by cluster separation and compactness using a validity function. A hybrid learning algorithm was applied to fine-tune the parameters of the model, integrating forward and backward passes. In the forward pass, node outputs

advance up to layer 4, and the remaining parameters are determined using the Least Square Estimate (LSE) technique. Error metrics are calculated for each individual node. During the backward pass, gradient descent updated the principal parameters. The Improved ANFIS model achieved high accuracy (97.14%), sensitivity (96%), and specificity (90%), and accurately classified all cell types, particularly in the microscopic blood cell ALL IDB dataset. There might be potential risks of overfitting due to reliance on a single dataset.

Fauzi et al. compared how effective combining Principal Component Analysis (PCA) with Fuzzy Support Vector Machines (FSVM) is for cancer classification, in comparison to using FSVM by itself [90]. Experimental findings showed that the FSVM technique achieved an 87.69% accuracy without using PCA. However, when PCA is applied to the cancer data, reducing the feature set to 60, the classification accuracy significantly improved to 96.92%. Here, the study's strengths lie in its innovative combination of FSVM with PCA, boosting accuracy by reducing noise and redundancy. The study acknowledges need for further validation on large datasets and insufficient discussion on how varying the PCA parameters might affect the overall performance.

Ramya et al. introduced aa FBW based approach i.e. FBW-NN to identify AML [91]. The AML region is divided using a model called Adaptive Fuzzy Entropy (AFE), combining an ACM with FCM clustering. Following segmentation, feature sets are extracted at both the statistical and image level. The performance of the Artificial Neural Network (ANN) was enhanced using the Fractional Black Widow Optimization technique. By incorporating pre-trained models, the proposed method surpassed other advanced techniques when using the Munich AML Morphology dataset, achieving 0.96 accuracy, 0.97 precision, and 0.97 recall. The results demonstrated that the FBW-NN outperforms other state-of-the-art methods, including pre-trained Deep Convolutional Neural networks (DCNN), Naïve Bayes, Convolutional Neural Networks (CNN), and Chronological Sine Cosine-based Actor-Critic Neural Networks (CSC-ACNN). However, the study did not extensively discuss the computational complexity or potential limitations of the FBWO algorithm.

Bai et al. introduced the method of deep neural network and the process starts with the preprocessing and segmentation of the input image [92]. In the mapper phase, the most important features were extracted to enhance the performance of the classification, based on the HHOA and WOA, which are combined into the algorithm named HHWO. The method is concluded by the severity analysis based on classification of levels of leukemia for improved treatment. The proposed technique of HHWO-enabled DNFN performed better 95.9% Accuracy, 96.5% sensitivity, and the specificity of 96.6%. The study's strengths lie in the use of HHWO algorithm for feature extraction and integration of DNFN with Map Reduce framework, which enhanced the computational efficiency.

Narayanan et al. introduced a technique for evaluating the accuracy of Acute Leukemia classification using two distinct ML classifiers: a combination of HFCM with a RF classifier, and a SVM classifier [93]. In this paper, the algorithms proposed for detecting and classifying Acute Leukemia include HyFMr and SVM. The Hybrid FCM and RF algorithm demonstrated 99.06% accuracy, 99.4% sensitivity, and 97.8% specificity in the experimental results. ROC curve analysis demonstrated the high effectiveness of the Hybrid FCM and RF classifier in diagnosing and classifying Acute Leukemia. The study used MATLAB for its development. The performance metrics for the SVM classifier have not been discussed.

Dhal et al. presented the Enhanced Slime Mould Algorithm (ESMA), which integrated opposition-based learning and the mutation strategy of differential evolution, to attain segmentation of white blood cells (WBCs) without the need for illumination [94]. ISMA addressed the issue of local optima trapping in partitional clustering techniques. This research also looked at how lighting affects the way we group images in pathology, which is important for diagnosing diseases. They did this by studying the different colours in different colour spaces. Results indicated that clustering methods based on illumination or colour components could be successful for segmenting images. In particular, the combination of ISMA-KM with the "ab" colour channels in the CIELab colour space achieved more than 99% accuracy for segmenting nuclei. Additionally,

ISMA-KM showed the shortest execution time, whereas ISMA-RKM had the longest. ISMA demonstrated competitive performance on CEC2019 benchmark test functions compared to other recent nature-inspired optimization algorithms (NIOAs). This study lacks real-world noise handling, has execution time trade-offs (ISMA-RKM is slow), and does not compare with deep learning models, raising concerns about generalizability and parameter sensitivity.

Jha et al. developed a leukemia detection technique which is based on sine/cosine method [95]. The blood smear images were segmented using a hybrid model based on entropy, resulting in the extraction of image-level and statistical features. These characteristics were fed into the classifier to identify leukemia. The Chrono-SCA-ACNN enhanced weight optimization by incorporating the concept of time order into the Sine Cosine Algorithm (SCA). Tests carried out with the ALL-IDB2 dataset prove the efficiency of the approach, reaching an accuracy of 0.99, surpassing current classification methods. However, potential limitations include overfitting due to complex weight optimization and reliance on a single dataset (ALL-IDB2). Additionally, the computational cost of the entropy-based segmentation and actor-critic neural network may be high.

Zakir et al. presented a non-invasive diagnostic technique that utilized medical images, employing a convolutional neural network (CNN) [96]. The suggested solution combined a CNN model with an ECA module and the VGG16 structure to improve feature extraction, increasing both feature representation and classification accuracy. The ECA module effectively distinguished between ALL cancer and healthy cell images despite their morphological similarities. Furthermore, different methods for enhancement are utilized to improve both the quality and quantity of the training data. With the CNMC dataset categorized into seven folds to consider subject-level differences, the model reached an accuracy of 91.1% in distinguishing between normal and cancerous cells. These results show that the new method can assist pathologists in finding key features in blood cell images to accurately diagnose ALL. The dataset used for training the model is not clearly mentioned.

Mounika et al. aimed to reduce identification time, improve diagnosis accuracy, and reduce costs related to specialized care in Acute Lymphoblastic Leukemia [97]. In this study, researchers used a dataset of 108 images of lymphocyte cells. To make it even bigger, they used a technique called image augmentation, which added more images to the dataset. This increased the total number of images to 3240. These augmented images were used for feature extraction with the MobileNetV2 model, while the XGBoost classifier was trained on the prediction of the respective labels. Advanced Models: GoogLeNet, ResNet50, and MobileNetV2 + SVM have shown that the proposed approach of MobileNetV2 + XGBoost achieved an accuracy rate of 99.07%, a precision rate of 99.35%, and a recall rate of 98.72%. The limitation of the study is the smaller number of images used for the training purpose.

Khatter et al. focused on the development of an automated system for detecting and identifying blood diseases using patient data [98]. It emphasized the necessity of having an intelligent system that can analyze medical reports and assist doctors globally. The system, known as S-ANFIS, combined the Adaptive Neuro-Fuzzy Inference System (ANFIS) with content curation and intelligence analysis. By analyzing different models and case studies, especially relating to diabetes, researchers customized S-ANFIS to outperform current techniques in forecasting chronic illnesses, achieving an accuracy level of 88.6%. The proposed system was tested on 1000 real blood samples.

Rodrigues et al. proposed the joint approach based on a genetic algorithm combined with the ResNet-50V2 residual convolution neural network for predicting Acute Lymphoblastic Leukemia from microscopy images in ALL-IDB dataset [99]. Optimization of hyperparameters using GA is an important issue for this study since the manual tuning is challenging. The results were compared by using GA optimization against many other methods such as random search and Bayesian optimization. The findings showed that with GA, the model significantly improves its accuracy to 98.46%. The current findings pointed out the feasibility of computer vision strategies for practical applications in leukemia identification.

Mahesh et al. proposed a new approach for the prediction of leukemia using microarray

gene data with Hybrid Ant Lion Mutated Ant Colony Optimization combined with Particle Swarm Optimization [100]. The main purpose of the hybrid model was selecting the best features for the purpose of classification. The key effectiveness of this novel approach was supported through an excellent prediction accuracy of 87.88%. Hence, this study underlines the potential of combining evolutionary algorithms in enhancing computational efficiency and accuracy about medical predictions.

Sallam et al. described in their work how machine-learning classifiers could be useful in the diagnosis of ALL whether malignant or benign [101]. First, this method improved images through an adaptive threshold method, reducing errors and enhancing contrast. The grey wolf optimization approach was utilized to pick the key important feature sets from the images. Later, the ALL (acute lymphoblastic leukemia) data was classified using different classification techniques. The model turned in very impressive performance metrics, with accuracy rated 99.69%, sensitivity rated 99.5%, and specificity rated 99%. The limitation of this study is the very small size of the dataset used for training the model.

Hosseinzadeh et al. presented a technique for identifying Acute Lymphoblastic Leukemia (ALL) [102]. The research examined various transfer learning models and feature selection was then carried out using various methods such as Genetic Algorithm, PCA, and ANOVA etc. Various classifiers have been tried out of which Multilayer Perceptron gave the best results. Further, the presented method has been used on classification of ALL and HEM using CNMC 2019 dataset. It scored a reasonable accuracy rated 90.71% and a sensitivity rated 95.76%, the best so far from many methods using the same dataset. It is discussed in the future work that hybrid models could be explored further to improve accuracy.

Agustin et al. proposed a two-stage ANN integrated with PSO to classify immature white blood cells of the patients with ALL [103]. The first step deal with binary classification of lymphoid cell type, while the second step considers a binary classification of lymphoblast cell type. Five peripheral blood samples of ALL from the Sardjito Hospital were used to identify the model. Data preparation, selection of

relevant features, important characteristic extraction, and an application for the two-step classification were the basic steps involved in this series since each is very important to assure the accuracy and efficiency of the obtained results. Its precision was later validated with the common methods of multiclass NN-BP and multiclass NN-PSO, which utilized a particle swarm optimization algorithm. The proposed method gave good accuracy at 86.92%, proving quite promising for development in further scenarios. This method makes it easier to find all the cells, but it hasn't been tested on many samples yet.

Sallam et al. introduced an improved technique for categorizing all subtypes of Acute Lymphoblastic Leukemia (ALL) by employing the k-means clustering algorithm [104]. The process begins with preprocessing of the images, followed by extracting key features that describe the images in detail. Next, the Enhanced Grey Wolf Optimization (EGWO) algorithm is applied to select the most significant features related to blood cell morphology. This is an approach where k-means clustering has identified optimal cluster centers based on certain parameters. In the presented approach here, several leading supervised classifiers have been compared, namely RF, KNN, SVM, and NB. It showed remarkable results with an accuracy of 99.22%, the precision rate of 99%, and the sensitivity of 99%. This method is very accurate compared to others, but it's based on a dataset of 3,189 images.

Veeraiah et al. suggested Mayfly optimization combined with a Generative Adversarial Network to improve the process of feature extraction and classification of various types of blood cancers [105]. A Hybrid Generative Adversarial System combined with Principal Component Analysis was also used within the feature extraction model. The model leverages semantic techniques and morphological procedures based on geometric features to carry out leukemia cell segmentation. The various leukemia types along with abnormal WBCs, were categorized by the MayGAN model. This model has proved to be highly efficient in finding out these different types of leukemia with high accuracy. MayGAN has been able to show improved performance as compared to existing and prevailing techniques in detecting white blood cell abnormalities through

geometric feature analysis. It exhibited excellent performance metrics: 99.8% as accuracy, 98.5% for precision, 99.7% for recall, and 97.4% for the F1-score. These good results are based on a dataset of 1,200 images, but more testing is needed on unstained samples.

Balasubramanian et al. used deep learning method with an evolutionary algorithm to enhance white blood cell classification [106]. They adapted the PSO algorithm for hyperparameter optimization of considered CNN model. After the simplification process, the dataset included pictures from both LISC and BCCD datasets. This resulted in a model with an accuracy of 99.2%, sensitivity at 94.56%, specificity at 98.78%, and an AUC of 0.982. The findings were contrasted with alternative optimization methods like GAs, Differential Evolution, and GWO. In general, the findings suggested that PSO is highly efficient in optimizing CNN parameters to enhance sensitivity, making it a dependable tool for assessing blood cell count. This combination of PSO and CNN worked well for detecting the five WBC types, but larger datasets could help test its limits.

Ahmad et al. have proposed a deeply advanced hybrid method for the efficient WBC subtypes classification [107]. Initially, the method acquired the optimal characteristics from the images of the leukocytes through transfer learning with pre-existing DenseNet201 and Darknet53 models. The extracted feature vector was optimized by applying an entropy-controlled marine predator algorithm meta-heuristic technique that could pick out only the most important features and reject the less informative ones. The refined feature set after this reduction was classified using various baseline classifiers with different kernel configurations. On testing upon public dataset with 5000 images from five WBC subtypes, achieved an overall accuracy rated 99.9% along with minimal size of the feature vector by more than 95%. The superior convergence rates for the ECMPA compared with traditional meta-heuristic methods were also demonstrated. The sources of the dataset used are not clearly mentioned.

Kumar et al. proposed a classification approach using CNN that automatically classified five types of peripheral blood cells: eosinophils, basophils, lymphocytes, monocytes,

and neutrophils, without any human interference [108]. They proposed the use of an improved SSO along with PSO for optimization on a large database of blood cell images that contributed to a significantly improved performance in the classification. They employed the CNN based on the architecture of VGG19, then trained by a self-developed method to classify data with an accuracy rate of 98%. After optimizing this model by using the SSPSO technique, they have proposed a very accurate automated system for peripheral blood cell classification. The authors refined the model using 10,674 images acquired from a clinical environment, which further improved the classification accuracy to 99%, along with significant enhancements in precision and the F1-score.

Alrefai et al. proposed a method of cancer classification based on the ensemble learning approach, combining particle swarm optimization for feature selection [109]. The experiments showed high accuracy of this method on microarray datasets: 100% on leukemia, 92.86% on colon cancer, 86.36% on breast cancer, 100% on ovarian cancer, and 85.71% on central nervous system cancer. These results outperformed many of the prevailing methods and gives an indication that the model works well when compared with other existing methods. This approach improved performance by 12% over basic ensemble methods, but it depends on microarray data, which limits its use in other imaging areas.

Alabdulqader et al. created a blood cancer model through supervised machine learning techniques [110]. The leukemia microarray gene dataset consisted of 22,283 genes. The promising challenges were imbalanced and high-dimensional data in the dataset were handled by applying the Chi-squared feature selection technique. The SMOTE-Tomek is a resampling technique that balances the dataset by generating synthetic samples, while Chi2 selects the most relevant genes for the model's training. This contribution presented the novelty of the weighted CNN model, which integrated support for three different CNN architectures for classification. Through comprehensive testing and comparisons with other advanced methods, the weighted CNN, when combined with SMOTE-Tomek and Chi2, achieved a remarkable accuracy of 99.9%, showcasing its

superior effectiveness. This method works well with gene data, but testing on images could make it more useful.

Amin et al. presented a method for early-stage segmentation and classification of white blood cells (WBC) by converting RGB images to HSV and applying dual thresholding to the saturation component for WBC segmentation [111]. Deep learning models extracted features from AlexNet (FC8 layer), MobileNetV2 (Logits layer), ShuffleNet (node_202 layer), and ResNet-18 (FC1000 layer). A non-dominated sorting genetic algorithm (NSGA) was utilized to combine and optimize these feature vectors. The method was tested on the LISC, ALL_IDB1, and ALL_IDB2 datasets, achieving perfect accuracy (1.00) for classifying blast/non-blast cells, lymphocyte, neutrophil, monocyte, and eosinophil cells, and near-perfect accuracy (0.9992) for basophil cells, showing its capability to precisely recognize different types of WBC.

Ay et al. presented a model for predicting heart disease and heart failure survival by integrating meta-heuristic feature selection algorithms: cuckoo search (CS), flower pollination (FPA), whale optimization (WOA), and harris hawk's optimization (HHO) [112]. The model was tested on the cleveland heart disease and Faisalabad Institute of Cardiology datasets to determine the best population size for feature selection. For heart disease, the original dataset's highest F1-score is 88% using K-nearest neighbour (KNN), but the proposed method achieves 99.72% with KNN, FPA, and eight features. For heart failure, the original dataset's top F1-score is 70% with logistic regression (LR) and random forest (RF), while the new approach reaches 97.45% with KNN, HHO, and five features. The study highlights the value of meta-heuristic optimization, but applying it to other heart-related datasets, like ECG signals, could make it more useful.

Shahzad et al. proposed an enhanced WBC classification technique including several preprocessing steps, CNN architectures, feature selection methodologies, and classifiers [113]. In the preprocessing step, the authors applied CLAHE to enhance the input images. Internally developed CNN architecture, ResNet50, and EfficientNetB0 are used for feature extraction. Here, it used the selection of the most important features through ant colony optimization, combining them and classifying them with models

like SVM and QDA. Then it yields an accuracy of 98.44% on the blood cell images dataset. This approach shows strong feature fusion, but adding more WBC types, like basophils, could improve its diagnostic ability.

Haznedar et al. compared the performance against some algorithms, such as backpropagation, some hybrid algorithms, genetic algorithms, Bayesian networks, support vector machines, and J48 decision trees [114]. The average accuracy of the obtained cancer types was 96.28% with FCM-based ANFIS optimized by the SA algorithm. This method showed significant improvement in classifying cancer gene expression data from DNA microarrays compared to the other algorithms. It worked well with five cancer datasets, which suggests it could be used for other gene expression problems, but it may need to be faster for larger datasets.

Sampath et al. introduced a method to select key genes from cancer gene expression data for treatment by utilizing a tailored bio-inspired algorithm called cuckoo search with crossover [115]. This approach aimed to accurately classify various cancer sub-types from microarray technology. Studies were carried out on five standard cancer gene expression datasets. The findings demonstrated that CSC is more effective than conventional cuckoo search (CS) and other popular methods, with a 99% accuracy rate in classifying prostate, lung, and lymphoma data sets by utilizing the top 200 genes. CSC obtained accuracies of 96.98% and 98.54% for the leukemia. This high accuracy shows it could be useful for precision medicine, but testing the selected genes biologically could make it more effective for treatment.

Li et al. introduced a new non-invasive testing technique that integrated AlexNet and Extreme Learning Machine networks to enhance diagnostic outcomes [116]. The technique was further improved by utilizing an upgraded edition of the Grasshopper Optimization Algorithm (GOA). Comparing to other advanced methods, simulations indicated that the suggested technique improves efficiency, reaching a 98% accuracy and a 93% sensitivity.

Meenakshi et al. presented a Deep Features based CNN (DFCNN) for counting (WBC) from an image database, comprising three phases: feature extraction, selection, and

classification [117]. During feature extraction (AlexNet, GoogLeNet, and ResNet-50) a combined CNN structure, extracted almost 3000 features from the input data of images. To select features, a combination of Mayfly Algorithm and particle swarm optimization (HMA-PSO) was used, where PSO aids in updating the velocity of mayflies to select essential features. Different characteristics were categorized using a Recurrent Neural Network with Long Short-Term Memory (RNN-LSTM) into different types of WBCs. Utilizing MATLAB, the suggested approach has been evaluated using different metrics such as accuracy, recall, precision, specificity, and F1-score, surpassing the performance of current methods (MA-RNN and PSO-RNN) with a recall rate of 0.98, precision rate of 0.9, and accuracy rate of 0.97.

Chen et al. introduced a model called Resnet101-9 ensemble, which utilized nine Resnet-101 models trained and combined using majority voting to classify Acute Lymphoblastic Leukemia (ALL) in microscopic images [118]. Each Resnet-101 model was fine-tuned using transfer learning, integrating pre-trained Resnet-101 models, and optimizing algorithm hyperparameters via the Taguchi experimental method. These models have been trained and performance was evaluated based on microscopic images from the C-NMC dataset. Accordingly, the Resnet101-9 ensemble had superior performances at reaching an accuracy of 85.11% and F1-score of 88.94%, experimental tests showed.

Salama et al. created deep neural networks (DNNs) that were trained using the 10-colour CLL MRD panel for CLL patients who received treatment [119]. The DNNs were used sequentially in a hybrid approach to classify CLL MRD, with expert analysis serving as the reference point. In another separate group of 34 samples, this combination method yielded an accuracy rate of 97.1%. Furthermore, there was a strong correlation between DNN and expert examination in detecting CLL cells as a percentage of total WBCs. The DNN decreased the time needed for gating to 12 seconds per case, in contrast to the 15 minutes per case required manually.

Taino et al. suggested a method utilizing genetic algorithms [120]. The genetic algorithm's chromosome structure included four genes, with evaluation, selection, crossover, and mutation processes defined to optimize group separation using the highest

AUC scores and the fewest features. From histological images, 1512 features were extracted; different population sizes and iteration numbers were tested. The highest AUC with a value of 0.984 was derived when the initial population was set to 50 and iterations set to 50. On the other hand, the highest level of success i.e. AUC score of 0.947 was attained. This method, by explaining methods, features, and their optimal configuration, is useful for recommendations for researchers studying pattern recognition in colorectal cancer and lymphoma studies.

Alsuliman et al. improved the classifying autism spectrum disorder through the development of 16 optimized machine learning models including GWO-NB, GWO-SVM and GWO-KNN [121]. Four optimization algorithms were used for feature selection: GWO, flower pollination algorithm, Bat Algorithm, and Artificial Bee Colony. These algorithms have been applied to optimize the wrapper feature selection method by identifying the most important features and enhancing model performances. The results demonstrated excellent performance, with the GWO-SVM model achieving the highest accuracies, reaching 99.66% on the PBC dataset and 99.34% on the GE dataset.

Oyelade et al. introduced an innovative hybrid binary optimization technique for choosing important features in large datasets, utilizing a two-tier optimization framework with a sub-population selective mechanism [122]. In this approach, the optimizer with level-1 employs the binary Ebola optimization search algorithm (BEOSA) to mutate selected items, which were subsequently passed to a level-2 optimizer that utilized either simulated annealing (SA) or firefly (FFA) algorithms. Additionally, nested transfer (NT) functions were incorporated to guide the behaviour of the level-1 optimizer. The proposed hybrid models, namely HBEOSA-SA, HBEOSA-FFA, and their NT-enhanced variants, have been tested on high-dimensional datasets. Experimental results on large, medium, and small datasets furnished classification accuracies: 0.995 by HBEOSA-FFA, 0.967 by HBEOSA-FFA-NT, and 0.953 by HBEOSA-FFA.

Aher et al. proposed a new approach, including a Recurrent Neural Network classifier

optimized by the Rider Chicken Optimization algorithm, to carry out efficient classification and detection of cancer [123]. Preprocessing of gene expression data was performed to make it ready for the classification stage, while dimensionality reduction was achieved by entropy-based gene selection. Then, by using the RNN that was trained with the RCO algorithm-a hybrid of Chicken Swarm Optimization (CSO) and Rider Optimization Algorithm (ROA)-the selected genes were classified. The performance of discussed method was evaluated with three datasets: Leukemia database, Small Blue Round Cell Tumor dataset, and Lung Cancer dataset. Its performance was measured against existing key performance metrics: sensitivity, specificity, and accuracy. RCO-RNN obtained up to 95% in all these metrics, hence proving to be very effective for the classification task at hand.

Table 2.5. Soft Computing-based hybrid leukemia detection approaches

Study	Cancer Type	Dataset	Methodology	Accuracy & Findings	Limitations
Chadaga et al. [82]	Post-transplant survival (Blood cancers)	Blood cell images of 186 cancerous and non-cancerous patients	Feature selection (HHO, Salp Swarm, Mutual Info) + XAI (SHAP, LIME) + STACKA, STACKB, STACKC	STACKC achieved 89% accuracy in predicting bone marrow transplant success	Small dataset
Vishwaraj et al. [83]	General Cancer Detection	666 synthetic images	ML on Waveguide Bragg Gratings + PCA + MLP	~95% accuracy using light reflection	No real sample testing
Wu et al. [84]	Lymphoma	50 histopathology images	Deep learning + Genetic algorithm for overlapping cell segmentation	F1-score: 0.87, Jaccard index: 0.77	Manual annotation, small dataset, no method comparison
Priyadarshini et al. [85]	Skin Cancer (Melanoma)	300 skin images	ELM-TLBO hybrid + fuzzy C-means + median filtering	93.18% accuracy, fast training, good parameter tuning	Struggles with complex patterns

Study	Cancer Type	Dataset	Methodology	Accuracy & Findings	Limitations
Nematzadeh et al. [86]	Multiple cancers (general ML tuning)	11 biological & biomedical datasets	GA, GWO, EGS for hyperparameter tuning	GWO outperformed GA with faster convergence	May vary with complex/imbalanced data
Baby et al. [87]	ALL	Dataset details not provided	Transfer learning using 8 EfficientNet variants + SVM, RF, LR ensemble	98.5% classification accuracy	Dataset details missing
Devi et al. [88]	ALL	Local dataset (unspecified)	Convolutional Leaky ReLU + CatBoost + XGBoost	97.12% accuracy	Dataset details not provided
Rejula et al. [89]	Leukemia (ALL)	ALL-IDB dataset	Improved ANFIS (I-ANFIS) using fuzzy clustering + LSE + gradient descent	Accuracy: 97.14%, Sensitivity: 96%, Specificity: 90%	Risk of overfitting, single dataset
Fauzi et al. [90]	Cancer (general)	Not specified	FSVM alone vs. FSVM + PCA	Accuracy improved from 87.69% to 96.92% using PCA	Needs larger validation, PCA parameter tuning not explored
Ramya et al. [91]	AML	Munich AML Morphology dataset	FBW-NN using Adaptive Fuzzy Entropy + FCM + ANN optimized with Fractional Black Widow	Accuracy: 0.96, Precision: 0.97, Recall: 0.97	Computational complexity and FBWO limitations not discussed
Bai et al. [92]	Leukemia	Not specified	HHWO algorithm (HHO + WOA) + DNFN + MapReduce	Accuracy: 95.9%, Sensitivity: 96.5%, Specificity: 96.6%	Dataset not specified
Narayanan et al. [93]	Acute Leukemia	Not specified	HFCM + RF and SVM classifiers	HFCM+RF: Accuracy: 99.06%, Sensitivity: 99.4%, Specificity: 97.8%	SVM performance not discussed

Study	Cancer Type	Dataset	Methodology	Accuracy & Findings	Limitations
Dhal et al. [94]	Leukemia (WBC Segmentation)	Not specified	Enhanced Slime Mould Algorithm + clustering in CIELab color space	ISMA-KM achieved >99% segmentation accuracy	No real-world noise handling, slow ISMA-RKM, no DL comparison
Jha et al. [95]	Leukemia	ALL-IDB2	Hybrid entropy segmentation + Chrono-SCA-ACNN	Accuracy: 0.99	Overfitting risk, high computational cost, single dataset
Zakir et al. [96]	ALL	CNMC (7-fold categorized)	CNN + ECA module + VGG16 + enhancement techniques	Accuracy: 91.1%	Dataset not clearly mentioned
Mounika et al. [97]	ALL	108 lymphocyte cell images augmented to 3240	MobileNetV2 + XGBoost	Accuracy: 99.07%, Precision: 99.35%, Recall: 98.72%	Small original dataset
Khatter et al. [98]	Blood diseases (general)	1000 real samples	S-ANFIS combining ANFIS with content curation	Accuracy: 88.6%	Limited to forecasting, specific case studies
Rodrigues et al. [99]	ALL	ALL-IDB	GA + ResNet-50V2	Accuracy: 98.46%	Focus on hyperparameter tuning, not broader model comparison
Mahesh et al. [100]	Leukemia	Microarray gene data	Hybrid ALO + ACO + PSO for feature selection	Accuracy: 87.88%	No detailed dataset or comparison with DL methods
Sallam et al. [101]	ALL	Not specified	Adaptive thresholding + GWO + multiple classifiers	Accuracy: 99.69%, Sensitivity: 99.5%, Specificity: 99%	Very small dataset
Hosseinzadeh et al. [102]	ALL and HEM	CNMC 2019	Transfer learning + feature selection (GA, PCA,	Accuracy: 90.71%, Sensitivity: 95.76%	Further hybrid models needed for better results

Study	Cancer Type	Dataset	Methodology	Accuracy & Findings	Limitations
			ANOVA) + MLP		
Agustin et al. [103]	ALL	5 samples from Sardjito Hospital	Two-stage ANN with PSO for binary classification	Accuracy: 86.92%; better than NN-BP and NN-PSO	Limited number of samples used
Sallam et al. [104]	ALL (All Subtypes)	3,189 images	K-means clustering + EGWO + RF, KNN, SVM, NB	Accuracy: 99.22%; Precision: 99%; Sensitivity: 99%	Based on a single dataset
Veeraiah et al. [105]	Various Leukemia Types	1,200 images	Mayfly Optimization + GAN + PCA (MayGAN)	Accuracy: 99.8%; Precision: 98.5%; Recall: 99.7%; F1: 97.4%	Needs testing on unstained samples
Balasubramanian et al. [106]	WBC classification	LISC and BCCD	CNN with PSO for hyperparameter tuning	Accuracy: 99.2%; Sensitivity: 94.56%; Specificity: 98.78%; AUC: 0.982	Larger datasets needed for testing
Ahmad et al. [107]	WBC Subtypes	5,000 images (public)	DenseNet201 & Darknet53 + ECMPA + various classifiers	Accuracy: 99.9%; 95%+ feature vector reduction	Dataset source not clearly mentioned
Kumar et al. [108]	Blood Cell Types (5 classes)	10,674 clinical images	VGG19-based CNN + SSO-PSO optimization (SSPSO)	Accuracy: 99%; high F1-score and precision	None major, very large dataset used
Alrefai et al. [109]	Leukemia & Other Cancers	Microarray datasets	Ensemble learning + PSO for feature selection	Accuracy: 100% (Leukemia); 92.86% (Colon); others also high	Limited to microarray data, not imaging

Study	Cancer Type	Dataset	Methodology	Accuracy & Findings	Limitations
Alabdulqader et al. [110]	Leukemia	Microarray dataset (22,283 genes)	Chi2 + SMOTE-Tomek + weighted CNN (3 CNNs)	Accuracy: 99.9%	Needs validation on imaging datasets
Amin et al. [111]	WBC (Various Types)	LISC, ALL_IDB1, ALL_IDB2	HSV conversion + thresholding + AlexNet, MobileNetV2, ShuffleNet, ResNet18 + NSGA	Accuracy: 1.00 for most cells, 0.9992 for basophils	None mentioned, very high performance
Ay et al. [112]	Heart Disease & Heart Failure	Cleveland & Faisalabad datasets	KNN + meta-heuristics (CS, FPA, WOA, HHO)	F1-score: 99.72% (Heart Disease), 97.45% (Heart Failure)	Needs testing on ECG and other heart datasets
Shahzad et al. [113]	WBC	Blood cell images	CLAHE preprocessing, ResNet50 & EfficientNetB0, ACO for feature selection, SVM/QDA classifiers	98.44% accuracy	Limited WBC types, e.g., basophils missing
Haznedar et al. [114]	Multiple cancers	Cancer gene expression (microarray)	FCM-based ANFIS optimized with SA	96.28% accuracy	May be slower with large datasets
Sampath et al. [115]	Prostate, Lung, Lymphoma, Leukemia	Microarray gene expression	Cuckoo Search with Crossover (CSC) for key gene selection	Up to 99% accuracy	Biological validation needed for gene selection
Li et al. [116]	General cancer	Microscopy images	AlexNet + ELM + improved GOA	98% accuracy, 93% sensitivity	Limited testing on varied datasets
Meenakshi et al. [117]	WBC	WBC image dataset	AlexNet, GoogLeNet, ResNet-50 + HMA-PSO + RNN-LSTM	97% accuracy, 98% recall	Tested only on MATLAB environment

Study	Cancer Type	Dataset	Methodology	Accuracy & Findings	Limitations
Chen et al. [118]	ALL	C-NMC	Nine ResNet101 models ensemble with Taguchi method	85.11% accuracy, 88.94% F1-score	Moderate accuracy, specific to ALL
Salama et al. [119]	CLL	CLL MRD panel	DNNs hybrid model, compared with expert analysis	97.1% accuracy	Limited dataset (34 samples)
Taino et al. [120]	Colorectal & Lymphoma	Histological images	Genetic Algorithm, AUC-based optimization	0.984 AUC, 0.947 success	Focus on configuration, may lack biological insight
Alsuliman et al. [121]	Autism	PBC & GE datasets	16 optimized ML models, GWO-based feature selection	Up to 99.66% accuracy	Limited to autism datasets
Oyelade et al. [122]	Various	High-dimensional datasets	HBEOSA-SA, HBEOSA-FFA hybrid models	Up to 99.5% accuracy	Complex setup, may be hard to scale
Aher et al. [123]	Leukemia, Lung Cancer, SBRCT	Gene expression data	RCO-RNN (Rider Chicken Optimization with RNN)	95% accuracy, sensitivity, specificity	Performance based on 3 datasets only

Table 2.5 present soft computing-based hybrid leukemia detection approaches along with the limitation of every approach. From the existing studies shown in Table 2.5, we observed that many of the existing studies used very small or unclear datasets which may lead to biased results and overfitting. Many approaches also rely on manual data labeling, which is time-consuming. Some models might be overfitting because they were tested on just one small dataset. In some cases, important details like dataset information or method comparisons were missing. Computational complexity is another issue, as advanced models require high processing power and time. So, it may be hard to use in real hospitals or labs. Finally, even if some models showed high accuracy, they did not deal with problems like large data handling.

2.4. Research Gap

This literature survey highlights key problems in the existing leukemia detection methods. The traditional leukemia detection methods are invasive, painful, and expensive. They are also time consuming and results vary as per the expertise of the pathologist or doctor. The artificial intelligence-based approaches are highly applicable but the major issue with these approaches is the lack of enough diverse dataset, which makes it hard for models to work well. Further, we observed that deep neural network-based approaches are providing promising results but to achieve the good results, the sizes of the models are not considered. We analyzed that the focus of most existing research works is on accuracy only. From this survey, we observed that there are very few approaches available that are automatically evolving the architecture of the specific leukemia detection models. Thus, there is a need of new neural architecture search-based approaches for leukemia detection.

2.5. Summary

This chapter classifies the leukemia detection approaches into two broader groups, traditional methods, and artificial intelligence-based leukemia detection methods. Further, artificial intelligence-based techniques are divided into three groups, machine learning based, deep learning based, and hybrid approaches. An extensive survey has been carried out for the available leukemia detection approaches. In this survey, we observed that the traditional leukemia detection methods are painful and time consuming. The computer based intelligent approaches can be implemented for leukemia detection but they require large and diverse datasets, focus on both accuracy and sizes of the model, and requires the expertise to decide the architecture of an intelligent leukemia detection model.

Chapter 3 VGG16-PCA-PB3C: A Hybrid PB3C and Deep Neural Network based approach for Leukemia Detection

This chapter presents a new hybrid soft computing-based approach for leukemia detection. The proposed approach integrates Visual Geometry Group 16 (VGG16), Principal Components Analysis (PCA), and Parallel Big Bang Big Crunch (PB3C) for leukemia detection. In the proposed approach, the VGG16 is used to analyse and extract the important features from the blood smear images. PCA is used with VGG16 to decrease the dimensionality of the extracted feature sets and preserves the important features required for leukemia detection. Further, all important principal components are picked by the PB3C optimization approach. These are near optimal features to be used by a deep neural network-based classifier for the detection of leukemia. The VGG16-PCA-PB3C hybrid model is designed to increase accuracy and make the model training easier.

3.1. Introduction

Classification is the process of categorizing an object or instance based on its features or characteristics. There are many features describing each instance, so the classification problems deal with high-dimensional data. High-dimensional data often includes features that are unnecessary or irrelevant. These features may give repeated or even misleading information about the class of an instance. This can make classification models less accurate and slower. As the number of features increases, the "curse of dimensionality" becomes a problem. This means that training a classifier effectively requires many more examples, which can be very challenging. Feature selection (FS) helps to solve this problem. It involves picking a smaller set of important and useful features by removing the ones that are irrelevant or redundant [124]. This makes classification models more efficient and accurate. This chapter proposed the VGG16, PCA and PB3C integrated approach for the leukemia detection.

The major contribution of this chapter is as below:

1. Proposed a hybrid soft computing-based VGG16-PCA-PB3C approach to detect leukemia from blood cell images.
2. The proposed leukemia detection approach is tested on the CNMC_2019 dataset.
3. The proposed algorithm is compared with the existing 13 leukemia detection approaches.

Section 3.1 presents the introduction. Section 3.2 discusses the Visual Geometry Group 16 (VGG16) technique. Section 3.3 presents the Principal Component Analysis approach for dimensionality reduction. The PB3C optimization algorithm is presented in Section 3.4. Section 3.5 presents the proposed soft computing-based hybrid approach for leukemia detection. Section 3.6 discusses the CNMC_2019 dataset. Section 3.7 presents the results and the discussion. Section 3.8 summarizes the chapter.

3.2.Visual Geometry Group 16 (VGG16)

VGG16 is a Convolutional Neural Network which is pre-trained and it uses transfer learning. In transfer learning, the feature extractor of the pre-trained model also called encoder is used to extract the features and a classifier is trained to learn the weights for the new dataset. VGG16 was designed by the Visual Geometry Group (VGG) at Oxford University in 2014. The name "VGG16" comes from the architecture's 16 layers, which comprises of 13 convolutional, 5 max pooling, and 3 FC layers [125][126]. In VGG16, the features are obtained by the convolutional and pooling layers from the images and the FC layers are utilized for classification. VGG16 was trained on a large dataset called ImageNet, which contains over 14 million images labelled with 1000 various classes, such as objects, and scenes. Since VGG16 was trained on a large and diverse dataset, it learned to identify various features like edges of the object, the object’s texture, and different shapes or patterns of the object, which can be useful for recognizing various objects in new images. The architecture of VGG-16 is presented in Table 3.1.

Table 3.1 VGG16 architecture

VGG-16 Layers	No. of Filters	Filter Size	No. of Stride	No. of Padding	Output Dimension	Equation
Input Layer	It takes the image of size 224*224*3, uses no function and having same output dimension i.e. 224*224*3					
Convolutional Layer 1	64	3*3*3	1	1	224*224*64	$Z_1 = f(X * W_1 + b_1)$
Convolutional Layer 2	64	3*3*64	1	1	224*224*64	$Z_2 = f(A_1 * W_2 + b_2)$
Max-Pooling Layer 1	Applies max operation over a 2*2 window		2	-	112*112*64	$A_1 = P_1(Z_1)$
Convolutional Layer 3	128	3*3*64	1	1	112*112*128	$Z_3 = f(A_2 * W_3 + b_3)$
Convolutional Layer 4	128	3*3*128	1	1	112*112*128	$Z_4 = f(Z_3 * W_4 + b_4)$
Max-Pooling Layer 2	Applies max operation over a 2*2 window		2	-	56*56*128	$A_2 = P_2(Z_2)$

VGG-16 Layers	No. of Filters	Filter Size	No. of Stride	No. of Padding	Output Dimension	Equation
Convolutional Layer 5	256	3*3*128	1	1	56*56*256	$Z_5 = f(A_3 * W_5 + b_5$
Convolutional Layer 6	256	3*3*256	1	1	56*56*256	$Z_6 = f(Z_5 * W_6 + b_6)$
Convolutional Layer 7	256	3*3*256	1	1	56*56*256	$Z_7 = f(Z_6 * W_7 + b_7)$
Max-Pooling Layer 3	Applies max operation over a 2*2 window		2	-	28*28*256	$A_3 = P_3(Z_4)$
Convolutional Layer 8	512	3*3*256	1	1	28*28*512	$Z_8 = f(A_4 * W_8 + b_8)$
Convolutional Layer 9	512	3*3*512	1	1	28*28*512	$Z_9 = f(Z_8 * W_9 + b_9)$
Convolutional Layer 10	512	3*3*512	1	1	28*28*512	$Z_{10} = f(Z_9 * W_{10} + b_{10})$

VGG-16 Layers	No. of Filters	Filter Size	No. of Stride	No. of Padding	Output Dimension	Equation
Max-Pooling Layer 4	Applies max operation over a 2*2 window		2	-	14*14*512	$A_4 = P_4(Z_7)$
Convolutional Layer 11	512	3*3*512	1	1	14*14*512	$Z_{11} = f(A_5 * W_{11} + b_{11})$
Convolutional Layer 12	512	3*3*512	1	1	14*14*512	$Z_{12} = f(Z_{11} * W_{12} + b_{12})$
Convolutional Layer 13	512	3*3*512	1	1	14*14*512	$Z_{12} = f(Z_{12} * W_{13} + b_{13})$
Max-Pooling Layer 5	Applies max operation over a 2*2 window		2	-	7*7*512	$A_5 = P_5(Z_{10})$
Fully Connected Layer 1	It flattens the input into a vector of size 25088 and performs a linear transformation to the input using a bias vector of size 4096 and a weight matrix of size 25088*4096. The output dimension is 4096.					$Y_1 = f(Z_{13} \cdot V_1 + c_1)$
Fully Connected Layer 2	It transforms the input linearly using a bias vector of size 4096 and a weight matrix of size 4096*4096. The output dimension is 4096.					$Y_2 = f(Y_1 \cdot V_2 + c_2)$

VGG-16 Layers	No. of Filters	Filter Size	No. of Stride	No. of Padding	Output Dimension	Equation
Fully Connected Layer 3	It transforms the input linearly using a bias vector of size 1000 and a weight matrix of size 4096*1000. The output dimension is 1000.					$Y_3 = f(Y_2 \cdot V_3 + c_3)$
SoftMax Layer	The SoftMax function is used on the input to turn it into probabilities for 1000 different classes. The result has 1000 values.					$S = \sigma(Y_3)$

The layer wise features extracted by the VGG16 encoder are shown in Table 3.2.

Table 3.2 Features extracted by VGG16 encoder layers

VGG 16 Encoder Layers	Convolution Layers Included	Features Extracted
Input Image	-	224*224*3 = 150528
First Convolution Block	Conv 1_1, Conv 1_2	222*222*64 = 3154176
Second Convolution Block	Conv 2_1, Conv 2_2	220*220*64 = 3097600
Max Pooling	-	110*110*64 = 774400
Third Convolution Block	ConV 3_1, ConV 3_2, ConV 3_3	108*108*128 = 1492992
Max Pooling	-	54*54*128 = 373248
Fourth Convolution Block	ConV 4_1, ConV 4_2, ConV 4_3	50*50*256 = 640000
Max Pooling	-	25*25*256 = 160000

Fifth Convolution Block	ConV 5_1, ConV 5_2, ConV 5_3	23*23*512 = 270848
Max Pooling	-	7*7*512 = 25088

3.3.Principal Component Analysis (PCA)

PCA (Principal Component Analysis) is a common method used to reduce the number of features in a dataset. It helps simplify the data by focusing on the most important information. It works by transforming the original dataset, which may have many variables, into a new dataset with fewer variables, called principal components (PCs). These PCs capture the most important variations in the data. The PCs are created by combining the original features in such a way that they are uncorrelated [127][128]. The Principal Component Analysis (PCA) algorithm works by reducing the dimensionality of a dataset while retaining the most important information. The working of the PCA algorithm is shown in Algorithm 3.1.

Algorithm 3.1: Principal Components Analysis Algorithm

This database is shown as a matrix $D = N \times M$ matrix:
Let i^{th} row of this matrix D be represented by a vector $D_i = x_1, x_2, \ldots, x_M$.

Step 1: Find the Mean of D_i as given below:

$$\mu_i = \frac{1}{M} \sum_{j=1}^{M} D_{ij} \quad (3.1)$$

Step 2: Calculate the $N \times M$ matrix Φ as follows:

$$\phi_{ij} = D_{ij} - \mu_i \quad (3.2)$$

Step 3: Find the covariance matrix:

$$C = \Phi \times \phi^T \quad (3.3)$$

Step 4: Determine the eigenvalues of C : $\lambda_1 > \lambda_2, >, \ldots, > \lambda_M$

Step 5: Find the eigenvectors of C : $u_1, u_2, \ldots, u_M$

Step 6: Sort the eigenvectors from highest to lowest and choose the top "k" eigenvalues.

The first step in the PCA algorithm is to find the average of all the features in the

dataset. These mean features are subtracted from each feature of the dataset. This step will provide us zero mean features. These zero mean data is used to calculate the covariance matrix, which measures the relationships between the different features (or pixels) in the dataset. Next, the algorithm computes the eigenvectors and eigenvalues of the covariance matrix. The eigenvectors show the directions where the data varies the most (these are called the principal components), and the eigenvalues tell us how much variation each of these directions explains. The eigenvectors are arranged from highest to lowest according to their eigenvalues, and the top k eigenvectors (with the largest eigenvalues) are chosen for making decisions. This process reduces set of features and retains the essential feature set which makes the data easier to work with and improving model performance. The output of this step is a matrix with reduced dimensions, where each column represents a principal component capturing a specific aspect of the original data's variation [129].

3.4.Parallel Big Bang Big Crunch (PB3C)

The Parallel Big Bang Big Crunch Algorithm (PB3C) is a multi-population algorithm used for feature selection in machine learning. Feature selection is the process of picking the most useful features from a large set of features to improve the performance of a model. The algorithm is inspired by the big bang and big crunch theories from cosmology. PB3C is an improved version of single population based BBBC algorithm which is discussed in detail in chapter 4 of this thesis.

The PB3C algorithm works by iterating between two main phases: the big bang and the big crunch. In the big bang phase, multiple populations of random candidate solutions (set of features) are created. In the big crunch phase, every candidate solution’s fitness is evaluated in all the populations, and the solutions are ranked based on their performance. The best solutions are selected, and the search focuses on these solutions to refine them further. The best optimal solution from each population, called the "local best" i.e. l_{best} is selected. Then, the g_{best} solution is chosen from the l_{best} candidate solutions [130][131]. The algorithm then updates the local best solutions by exchanging their genes with the global best solution with a certain probability. Afterward, new set

of features are created by BB step. These new feature sets are created by making small random adjustments to the global best solution. This process continues until a stopping criterion is met, indicating that a near-optimal solution has been found. The working of parallel big bang big crunch algorithm is shown in Algorithm 3.2.

Algorithm 3.2: Parallel Big Bang Big Crunch Algorithm

Begin

/* Big Bang Phase Starts */

Create 'N' random populations, with each population containing 'M' candidate solutions

/* Big Bang Phase Ends */

While TC is not reached // TC is termination condition

/* Big Crunch Phase Starts */

For i=1 to N

a) Calculate the fitness of all the candidate solutions of i^{th} population.
b) Sort the population from highest to lowest based on their fitness value.
c) Select the local best candidate solution l_{best} (i) from the i^{th} population.

End

From amongst "N" l_{best} candidate solutions, select the globally best g_{best} candidate Solution

With a certain probability, replace the gene of l_{best} (i) candidate solution with the gene of g_{best} candidate solution

/* Big Crunch Phase Ends */

/* Big Bang Phase Starts */

Calculate new candidate solutions around g_{best} by adding/subtracting a small random number.

/* Big Bang Phase Ends */

End while

End

3.5. Proposed Soft computing Based Hybrid Leukemia Detection Approach

This section proposes a new hybrid soft computing-based leukemia detection approach. The proposed approach combines VGGG16 neural network architecture with PCA and the PB3C algorithm to select the near-optimal features for the detection of leukemia.

The working of the proposed approach is shown in Figure 3.1 and Algorithm 3.3. As shown in Figure 3.1, the proposed approach works in four phases, i) Feature extraction using VGG16, ii) Dimensionality reduction using PCA, iii) Optimal feature selection using PB3C, and iv) Classification of leukemia using fully connected layers.

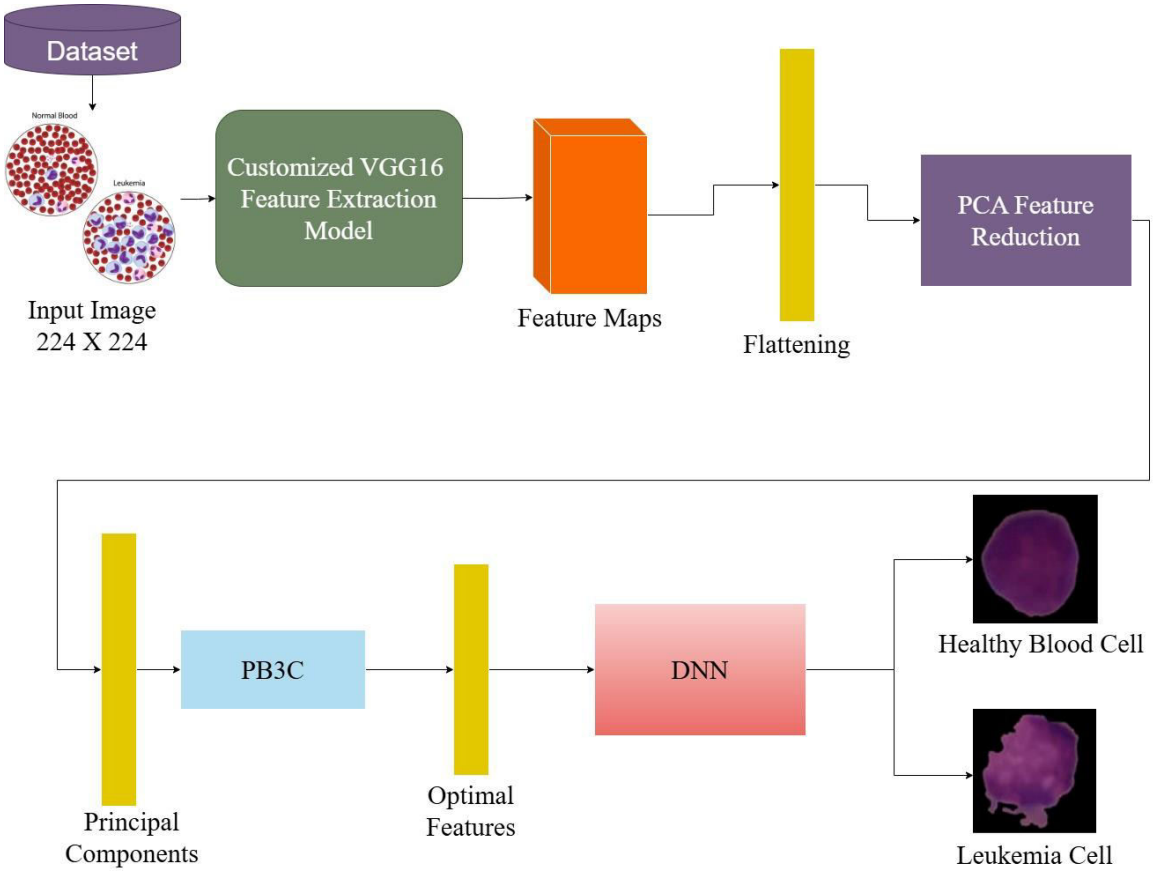


Figure 3.1 Proposed soft computing-based approach for leukemia detection

As shown in Figure 3.1, the proposed approach starts with feature extraction phase. The features from the dataset are extracted using the VGG16 convolutional neural network. For the feature extraction purpose, the blood cell images from the pre-processed CNMC_2019 dataset are fed into the convolutional and pooling layers of the VGG16 model. The details of the CNMC_2019 dataset is mentioned in section 3.6 of this thesis. The flattening layers of the VGG16 model extracts 25088 features per image from the leukemia dataset. The VGG16 model's fully connected layers are not used during this

feature extraction process. Instead, these layers are saved for the final step of the proposed approach, where they help in detecting leukemia. The number of features extracted by the VGG16 model are very large. After feature extraction, some features may be redundant, irrelevant, or contain overlapping information that does not effectively support the classification task. To overcome this, Principal Component Analysis (PCA) is used to reduce dimensionality by retaining the most significant features and eliminating those that are less useful.

The second phase of the proposed approach is the dimensionality reduction using PCA algorithm. In PCA, the relationships between features are measured using a covariance matrix. Then, principal components are found using eigenvalues and eigenvectors. The eigenvectors show the main directions of variation, and eigenvalues indicate how much variation they explain. The best principal components with the most variations are kept, making the feature maps smaller and simpler. For the dimensionality reduction purpose, the features extracted by VGG16 are provided to PCA. The PCA considered 1145 principal components out of 25088 features extracted by VGG16. All 1145 principal components are representing 99% of the variance. To reduce dimensionality while preserving most of the important information, PCA was applied with a variance retention threshold of 99%, resulting in the selection of 1145 principal components. It captures nearly all the essential patterns in the data while eliminating less informative components. After feature reduction using PCA, we regenerated the data from principal components. This reconstruction step helps evaluate how much of the original information is preserved after dimensionality reduction and allows us to estimate the extent of data loss introduced by PCA. The regenerated data was compared with actual data (features selected by VGG16) using Mean Squared Error (MSE) metric. MSE measures the average squared difference between the original and reconstructed values. The computed MSE was 0.0450%.

Since, PCA is an unsupervised method, it does not know which features help with classification. It only keeps the ones that show the most variation in the data. So, the features reduced by PCA are given to PB3C for further selection.

The third phase of the proposed method is selection of important features using the PB3C algorithm. For the feature selection purpose, we created four populations. Each population consists of four feature subsets i.e. the candidate solutions. Each candidate solution contains 200 features. The size of the feature subset remains same for all the populations. PB3C algorithm provides the 200 near optimal features. We selected the feature set size as 200 because increasing it beyond 200 did not improve the classification accuracy, and decreasing it below 200 reduced the accuracy.

In the final phase of the proposed approach, the DNN trains the model and detects leukemia. The optimized feature set from the PB3C algorithm is sent to the dense layers for classification. The first two fully connected layers use the ReLU activation function, and the last layer uses the softmax function to classify the data.

Algorithm 3.3: VGG16-PCA-PB3C algorithm

Step 1: Let D be a dataset comprising N-sized images, D = {I_1, I_2, I_3, …., I_N} where every image I_i is labeled with $O_i \in$ {True, False}, shows whether leukemia is present or not. The objective is to train an intelligent model to diagnose leukemia in an unseen image.

Step 2: Preprocess the images dataset D. In preprocessing resize each image in dataset D to 224 × 224 pixel and normalized it. The preprocessing function $\mathcal{P}$ can be defined as mentioned in equation 3.1:

$$I_i' = \mathcal{P}(I_i) = \frac{resize(I_i, 224, 224)}{255} \quad (3.1)$$

Extract features from dataset D using a pretrained VGG model denoted as ϑ. The VGG16 model (ϑ) is employed without final classification layer and is defined in equation 3.2.

$$Fi = \vartheta(I_i') \quad (3.2)$$

Step 3: Reduce dimensionality of extracted features using PCA algorithm. The PCA transformation τ reduces the dimensionality of features to κ components and it is described in equation 3.3:

$$\psi_i = \tau(F_i) = F_i W \quad (3.3)$$

Where W is a weight matrix consists of top eigenvectors of the covariance matrix computed from feature vector.

Step 4: Select optimal set of features (S) using bio inspired Parallel Big Bang Big Crunch (PB3C) Algorithm using equation 3.4:

$$Z_i = S(\psi_i) \quad (3.4)$$

Train a Deep Neural Network 'N' of selected optimal Z_i features

Step 5: Evaluate the performance of developed model using different measures like accuracy, precision and recall.

End

The proposed algorithm begins with feature extraction using the VGG16 model, where preprocessed blood cell images from the CNMC_2019 dataset are resized to 224x224 pixels and passed through the VGG16 model to extract feature maps. Next, dimensionality reduction is performed using PCA by calculating the covariance matrix, computing eigenvalues and eigenvectors, ranking them, and selecting the top k eigenvectors to form a reduced feature set. The PB3C algorithm is then applied for feature selection, alternating between the BB and the BC phase. This process keeps going until a stopping rule is reached and the optimal feature set is returned. Finally, the optimized features are used to train a deep neural network (DNN) for leukemia detection.

3.6. Dataset

For the validation purpose, we used CNMC_2019 (Classification of Normal versus Malignant Cells) leukemia patients' dataset. The dataset is prepared by Laboratory Oncology, AIIMS, New Delhi. The CNMC_2019 dataset consists of 15384 pre-segmented images. The dataset consists of two classes i.e. ALL and HEM. Here ALL represents the blood smear images of acute lymphoblastic leukemia patients and HEM represents the blood smear images of normal people [132]. Each image in the dataset has a size of 450×450 pixels.

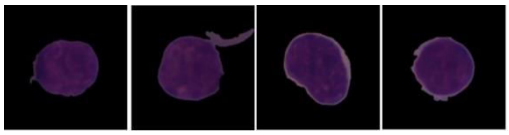


Figure 3.2 Microscopic blood cell images of leukemia patient

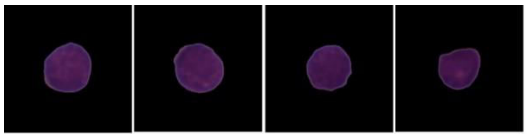


Figure 3.3 Microscopic blood cell images of normal person

The CNMC_2019 dataset deals only with one type of leukemia i.e. ALL. The sample

blood cell images from the CNMC_2019 dataset is shown in Figure 3.2 and 3.3. Figure 3.2 displays the microscopic blood cell images of leukemia patients and Figure 3.3 shows the blood smear images of normal person.

3.7. Results and Discussion

The proposed algorithm is implemented using python for the testing purpose. We compared how well the proposed approach performs with 13 existing leukemia detection methods for the performance analysis purpose. We compared the proposed approach with 6 transfer learning-based approaches and 7 machine learning approaches. Table 3.3 and Figure 3.4 show the comparison of proposed approach with 6 existing transfer learning-based approaches. Table 3.4 and Figure 3.5 show the comparison of proposed approach with other 7 machine learning-based approaches. The proposed approach is compared on different performance metrics namely accuracy, precision, recall, and F1-score. From Table 3.3, and Table 3.4, Figure 3.4, and Figure 3.5, it is noticed that the proposed approach VGG16-PCA-PB3C achieved 95.38% accuracy, 97.75% precision, 95.60% recall and 96.67% F1-score whereas without PB3C i.e. VGG16-PCA achieved 91.04% accuracy, 96.77% precision, 90.91% recall and 93.75% F1-score. VGG16 achieved 89.05% accuracy, 96.77% precision, 88.24% recall and 92.31% F1-score. These results clearly show that the proposed method is better than the existing methods.

Table 3.3 Comparison of proposed VGG16-PCA-PB3C approach with existing transfer learning-based approaches

CNN Architecture	Accuracy	Precision	Recall	F1 Score
Inception V3	86.01%	94.68%	85.58%	89.90%
CNN+ECA+VGG16 [96]	91.1%	-	-	-
TL+(GA,PCA)+MLP [102]	90.71%	-	95.26%	-

CNN Architecture	Accuracy	Precision	Recall	F1 Score
ResNet101 ensembled models [118]	85.11%	-	-	88.94%
VGG16	89.05%	96.77%	88.24%	92.31%
VGG19	90.65%	96.63%	89.58%	92.97%
VGG16-PCA	91.04%	96.77%	90.91%	93.75%
VGG16-PCA-PB3C	95.38%	97.75%	95.60%	96.67%

Table 3.4 Comparison of proposed VGG16-PCA-PB3C approach with existing machine learning based approaches

Classifier	Accuracy	Precision	Recall	F1 Score
SVM	74.8%	69.91%	76.78%	73.19%
Random Forest	81.20%	79.67%	81.66%	80.65%
Logistic Regression	83.20%	82.11%	83.47%	82.78%
K Nearest Neighbour	62%	71.54%	59.45%	64.94%
Decision Tree	70%	69.10%	69.67%	69.38%
Bagging	67.60%	69.91%	66.15%	67.98%
ML based Classification, IEEE ISBI 2019 Challenge [50]	91%	-	-	-
VGG16-PCA-PB3C	95.38%	97.75%	95.60%	96.67%

As shown, the proposed DNN gives much better accuracy, sensitivity, specificity, and F1-score compared to the existing methods. The performance of the proposed model is represented in Figure 3.4 and Figure 3.5 respectively.

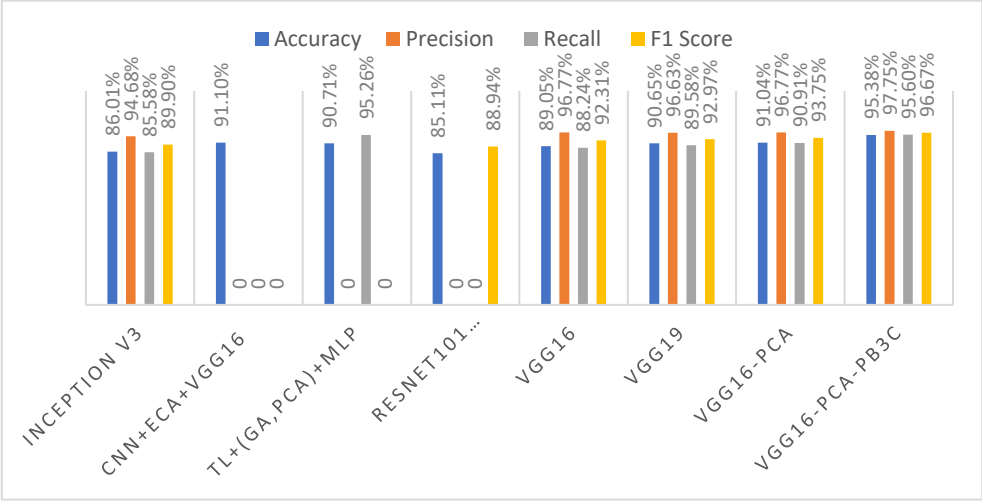


Figure 3.4 Comparison of VGG16-PCA-PB3C with transfer learning-based approaches

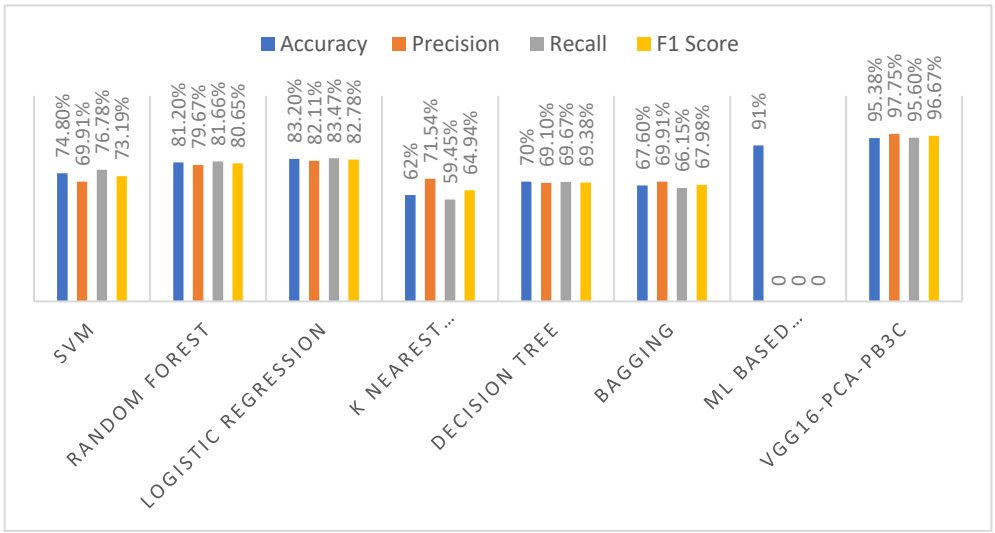


Figure 3.5 Comparison of VGG16-PCA-PB3C with existing machine learning-based approaches

The combination of the pre-trained VGG-16 model, PCA layer, PB3C algorithm, and fully connected classification layer leads to great performance.

3.8. Summary

This chapter proposed a new soft computing-based approach called VGG16-PCA-PB3C for leukemia detection from blood cell images. The proposed approach uses the VGG16 model to extract features and applies PCA to reduce the size of the leukemia dataset's feature map. The best set of features are then picked by PB3C approach. The proposed approach was trained and tested using the CNMC dataset, which contains labelled blood cell images. The VGG16-PCA-PB3C approach is implemented in python and compared with existing 13 leukemia detection approaches. The performance results showed that the proposed approach outperformed the existing approaches on leukemia detection problem.

Chapter 4 VGG19-PCA-BBBC: An Intelligent Framework for Leukemia Detection

Feature selection (FS) is an important step in data preparation that picks out a smaller set of useful features to improve how well machine learning models perform. However, the process is quite challenging due to the vast number of feature combinations to explore. This chapter focuses on the extraction and the selection of optimal feature set from the images of the blood cell for leukemia classification. This chapter proposes a new VGG19-PCA-BBBC based approach for leukemia detection. The proposed approach integrates Visual Geometry Group 19 (VGG19), Principal Component Analysis (PCA) and Big Bang Big Crunch (BBBC) algorithm for leukemia detection. The VGG19, with its deeper structure and additional convolutional layers, can extract deep features effectively. PCA minimizes the dimensions of the extracted features and find the principal components. The BBBC method is used to select the near optimal features and make the process faster. Finally, the FC layers are utilized for the classification purpose of leukemia.

The proposed approach is implemented in Python and compared with 15 other leukemia detection methods. From performance results, we observed that the VGG19-PCA-BBBC approach outperformed all other existing approaches of leukemia detection.

The major contribution of the chapter is given below.

1. Proposed a VGG19-PCA-BBBC architecture for leukemia detection.
2. The proposed leukemia detection approach is validated on the CNMC_2019 dataset.
3. The proposed approach is compared with 15 other leukemia detection approaches.

Section 4.1 introduces the VGG19 and its architecture, section 4.2 presents dimensionality reduction with principal component analysis, and section 4.3 explains optimization with the BBBC algorithm. Section 4.4 presents the proposed method for

leukemia detection. Section 4.5 explains the results and section 4.6 provides a summary of the chapter.

4.1. VGG19 and its Architecture

VGG19 is a pre-trained convolutional neural network which is used for the image recognition tasks. It was created by researchers in 2014 at Oxford University for the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). This DNN was trained on the ImageNet dataset and has 19 layers in total. There are 16 convolutional layers and 3 FC layers [133]. The conv layers use small 3x3 filters to obtain the features from the images. These small filters help to get detailed features with less computational effort. The ReLU (Rectified Linear Unit) activation function is used after every convolutional layer. VGG19 model also includes the max pooling layers. The obtained feature map's spatial dimensions are then decreased by the max pooling layers. Based on the features obtained, the final three FC layers of the VGG19 model are used for the classification purpose. The architecture of VGG19 includes:

i) Input Layer
ii) Convolutional Layers
iii) Filters
iv) Padding
v) Activation Function
vi) Max-pooling layers
vii) Fully Connected Layers
viii) Softmax Activation

The VGG19 takes as input an image with dimensions 224x224x3, where 3 represents the RGB colour channels. VGG19 consists of five convolutional blocks as described in Table 4.1.

Table 4.1 VGG19 Convolutional layer blocks

Block 1	Conv layers: 2 with each layer having 64 filters.	Max pooling layer follows filters.
Block 2	Conv layers: 2 with each layer having 128.	
Block 3	Conv layers: 4 with each layer having 256 filters.	
Block 4	Conv layers: 4 with each layer having 512 filters.	
Block 5	Conv layers: 4 with each layer having 512 filters followed by layer: max pooling.	

All the convolutional layers of block 1, 2, 3, 4, and 5 use 3x3 filters. As the network deepens, the number of the filters in every layer increases. Stacking multiple convolutional layers helps the model to understand complex and detailed patterns within the images. VGG19 uses the same padding to ensure the image size stays the same after each convolution. This means adding zeros around the edges of the input. This way, the output size remains the same as the input size. After each convolutional block, max-pooling is applied with filter size 2x2 and stride 2. This layer makes the image smaller by reducing its size (width and height) by half, while retaining the most important features found by the convolutional layers. Max pooling selects the largest value from a 2x2 area in the input. This reduces the size of the feature map. The problem of overfitting can be prevented by reducing the number of parameters. After going through both convolutional and max-pooling layers, the final feature map from the max-pooling layer is then flattened to a single dimension. Further, the flattened vector acts as an input for a fully connected layer. The last fully connected layer uses the softmax function to classify and generate the result [134-135].

The layered architecture of VGG19 is represented in Table 4.2.

Table 4.2 VGG19 layers architecture

Layer Type	No. of Filters	Filter Size	Stride	Padding	Output Dimensions	Equation
Input	Input size 224 * 224 * 3, uses no function, and same output dimension as that input: 224 * 224 * 3.					
Conv Layer 1	64	3 x 3	1	1	224 x 224 x 64	$X_1=f(W_1*X+b_1)$
Conv Layer 2	64	3 x 3	1	1	224 x 224 x 64	$X_2=f(W_2*X_1+b_2)$
Max Pooling 1	-	2 x 2	2	0	112 x 112 x 64	$X_3=MaxPool(X_2)$
Conv Layer 3	128	3 x 3	1	1	112 x 112 x 128	$X_4=f(W_3*X_3+b_3)$
Conv Layer 4	128	3 x 3	1	1	112 x 112 x 128	$X_5=f(W_4*X_4+b_4)$
Max Pooling 2	-	2 x 2	2	0	56 x 56 x 128	$X_6=MaxPool(X_5)$
Conv Layer 5	256	3 x 3	1	1	56 x 56 x 256	$X_7=f(W_5*X_6+b_5)$
Conv Layer 6	256	3 x 3	1	1	56 x 56 x 256	$X_8=f(W_6*X_7+b_6)$
Conv Layer 7	256	3 x 3	1	1	56 x 56 x 256	$X_9=f(W_7*X_8+b_7)$
Conv Layer 8	256	3 x 3	1	1	56 x 56 x 256	$X_{10}=f(W_8*X_9+b_8)$
Max Pooling 3	-	2 x 2	2	0	28 x 28 x 256	$X_{11}=MaxPool(X_{10})$
Conv Layer 9	512	3 x 3	1	1	28 x 28 x 512	$X_{12}=f(W_9*X_{11}+b_9)$
Conv Layer 10	512	3 x 3	1	1	28 x 28 x 512	$X_{13}=f(W_{10}*X_{12}+b_{10})$
Conv Layer 11	512	3 x 3	1	1	28 x 28 x 512	$X_{14}=f(W_{11}*X_{13}+b_{11})$

Layer Type	No. of Filters	Filter Size	Stride	Padding	Output Dimensions	Equation
Conv Layer 12	512	3 x 3	1	1	28 x 28 x 512	$X_{15}=f(W_{12}*X_{14}+b_{12})$
Max Pooling 4	-	2 x 2	2	0	14 x 14 x 512	$X_{16}=MaxPool(X_{15})$
Conv Layer 13	512	3 x 3	1	1	14 x 14 x 512	$X_{17}=f(W_{13}*X_{16}+b_{13})$
Conv Layer 14	512	3 x 3	1	1	14 x 14 x 512	$X_{18}=f(W_{14}*X_{17}+b_{14})$
Conv Layer 15	512	3 x 3	1	1	14 x 14 x 512	$X_{19}=f(W_{15}*X_{18}+b_{15})$
Conv Layer 16	512	3 x 3	1	1	14 x 14 x 512	$X_{20}=f(W_{16}*X_{19}+b_{16})$
Max Pooling 5	-	2 x 2	2	0	7 x 7 x 512	$X_{21}=MaxPool(X_{20})$
Fully Connected 1	This flattens the input into a vector of size 25088, and linear transformation to input with bias vector of size 4096, and weight matrix of size 25088 * 4096 results in 4096-dimensional output.				1 x 4096	$X_{22}=f(W_{17}\cdot X_{21}+b_{17})$
Fully Connected 2	It executes linear transformation of the input; the bias vector is of size 4096 and the weight matrix of the size 4096 * 4096. Output dimension: 4096.				1 x 4096	$X_{23}=f(W_{18}\cdot X_{22}+b_{18})$
Fully Connected 3	It performs a linear transformation of the input; the bias vector is of size 4096 and the weight matrix is of size 4096 * 4096. Output dimension: 2.				1 x 2 (For leukemia detection)	$Y=f(W_{19}\cdot X_{23}+b_{19})$
Softmax	Apply the SoftMax function over the input; the outcome is to normalize the vector into a probability distribution over 2 classes. Output dimension: 2.				1 x 2	$S=\sigma(Y)$

The layered architecture of VGG19 model in the graphical form is presented in Figure 4.1 whereas the custom VGG19 architecture used in the proposed method for feature extraction is shown in Figure 4.2.

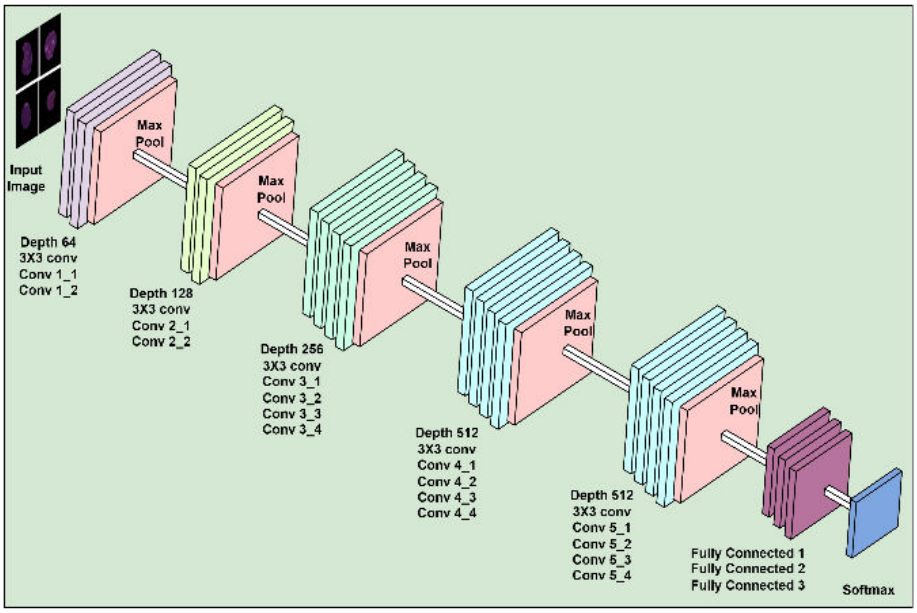


Figure 4.1 Architecture of VGG19 model for feature extraction and classification

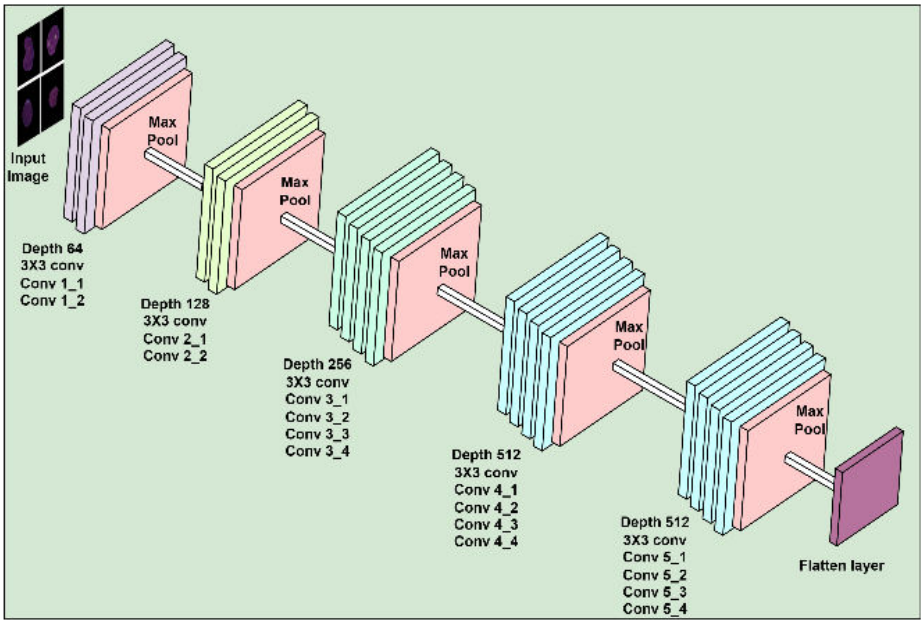


Figure 4.2 Customized architecture of VGG19 model for feature extraction

In the proposed approach, convolutional layers, max pooling layers and a flattening layer are used to extract features. The pretrained CNN's fully connected layers are not used during feature extraction.

4.2. Principal Component Analysis for Dimensionality Reduction

The PCA method is used to make the dataset smaller by reducing the number of features. The working of the PCA technique is already discussed in the chapter 3 of this thesis.

4.3. Big Bang Big Crunch Algorithm for Optimization

The big bang big crunch algorithm is an optimization method inspired by the universe's evolutionary processes, specifically the "Big Bang" and "Big Crunch" theories. This algorithm mimics two major phases in the creation and contraction of the universe: the "Big Bang" and the "Big Crunch." The big bang theory was first proposed by Georges Lemaitre in 1921. The big bang theory says that the Universe began as a very hot and dense point, called a singularity, around 13.8 billion years ago. This point suddenly expanded in a huge explosion, causing the universe to grow and cool down. As the universe expanded, matter and atoms began to form, and over time, gravity pulled them together to create galaxies, stars, and planets. The big crunch phase suggests that the particles converge towards a single point, representing the collapse of the universe to a singularity. It suggests that if the universe's expansion slows down, gravity might eventually pull everything back together, causing the universe to collapse into a single point again, like its original state. In this scenario, galaxies would move closer together and eventually merge [136]. This process of exploration and exploitation makes the BBBC algorithm an effective tool for optimization tasks in various fields, such as feature selection and optimizing machine learning models [137-139]. The working of BBBC is presented in algorithm 4.1.

Algorithm 4.1: Big Bang Big Crunch

Begin

Step 1: Initialize population size 'N', maximum iterations 'T', and bounds of search space. Generate initial population of candidate solutions randomly. Each candidate solution consists of 'G' genes.

Step 2: while termination criteria not met

* The termination criteria is maximum number of iterations*\

Step 3: for t = 1 to T:

Evaluate fitness of random candidate solutions in the population.

Step 4: Select the best candidate solution from the population. The best candidate solution is called as elite.

Step 5: Create a new population around the elite by adding or subtraction a small random number in it.

Step 6: Check the limits and violations and correct it if needed.

Step 7: End While

Step 8: Return elite.

End

4.4. Proposed VGG19-PCA-BBBC Approach for Leukemia Detection

This section proposes a new VGG19-PCA-BBBC based approach for leukemia detection. The proposed approach is the combination of VGG19-PCA and BBBC algorithm where VGG19 is used for feature extraction, PCA for dimensionality reduction and BBBC algorithm for near optimal feature selection. This approach builds upon the VGG16-PCA-PB3C method discussed in Chapter 3, where VGG16, with 13 convolutional and 3 fully connected layers was used to extract relevant image features. While VGG16 already provides a substantial number of features, the proposed approach uses VGG19, which includes three additional convolutional layers (16 in total). These added layers enable VGG19 to extract finer and more expressive features.

Since VGG19 already captures high-quality features, it requires a less complex optimization strategy. PB3C works well and gives good results. Since it uses the Big Bang Big Crunch process in parallel, it becomes more complex and requires more computational resources and time. BBBC is a single population-based search and optimization algorithm. Thus, BBBC algorithm is used to choose the most useful features from the features extracted by VGG19. BBBC is an ideal fit, offering efficient and effective feature selection without the added computational complexity of multi-population methods.

The working of the proposed approach is shown in Figure 4.3 and Algorithm 4.2. As shown in Algorithm 4.2 and Figure 4.3, the proposed approach works in four steps. The first step is extracting features. The second step is reducing the dimensions. The third step is selecting the important features, and the fourth step is detecting leukemia.

The first step in the proposed approach is to extract features which is accomplished by VGG19 convolutional neural network. The pre-processed blood smear images of CNMC_2019 dataset is passed through the convolutional and pooling layers of VGG19. The output received from final convolutional layer is a high-dimensional feature map, representing the most informative aspects of the image. Max pooling layers are used to reduce the size of the feature maps while keeping the important information. During the feature extraction step, the fully connected layers of VGG19 are not used. These layers are used later in the final step to detect leukemia.

For the leukemia detection purpose, we provided the images dataset to convolutional layers of VGG19. VGG19 extracts features from the dataset. The extracted features are useful for further classification task. The details of the features extracted by the convolutional and pooling layers of VGG16 and VGG19 are mentioned in Table 4.3.

Table 4.3 Layer wise features extracted by VGG16 and VGG19

Layer Name	Output Shape (VGG16)	Number of Extracted Features (VGG16)	Output Shape (VGG19)	Number of Extracted Features (VGG19)
block1_conv1	224×224×64	3,211,264	224×224×64	3,211,264
block1_conv2	224×224×64	3,211,264	224×224×64	3,211,264
block1_pool	112×112×64	802,816	112×112×64	802,816
block2_conv1	112×112×128	1,605,632	112×112×128	1,605,632
block2_conv2	112×112×128	1,605,632	112×112×128	1,605,632
block2_pool	56×56×128	401,408	56×56×128	401,408
block3_conv1	56×56×256	802,816	56×56×256	802,816
block3_conv2	56×56×256	802,816	56×56×256	802,816
block3_conv3	56×56×256	802,816	56×56×256	802,816
block3_conv4	—	—	56×56×256	802,816
block3_pool	28×28×256	200,704	28×28×256	200,704
block4_conv1	28×28×512	401,408	28×28×512	401,408
block4_conv2	28×28×512	401,408	28×28×512	401,408
block4_conv3	28×28×512	401,408	28×28×512	401,408
block4_conv4	—	—	28×28×512	401,408
block4_pool	14×14×512	100,352	14×14×512	100,352
block5_conv1	14×14×512	100,352	14×14×512	100,352
block5_conv2	14×14×512	100,352	14×14×512	100,352
block5_conv3	14×14×512	100,352	14×14×512	100,352
block5_conv4	—	—	14×14×512	100,352
block5_pool	7×7×512	25,088	7×7×512	25,088

Table 4.3 presents the various layers, output shape, and number of extracted features of VGG16 and VGG19. There are five different blocks each consist of convolutional and pooling layers. Both block 1 and block 2 contains two convolutional layers and one

pooling layer in VGG16 and VGG19. Block 3, block 4 and block 5 contains three convolutional and one pooling layer in case of VGG16 and four convolutional layers and one pooling layer in case of VGG19.The total number of features extracted by VGG16 and VGG19 remain same i.e. 25088. The difference between VGG16 and VGG19 is in the extraction of the deeper feature by the extra layers of VGG19 which were not present in VGG16. VGG19 extracts more refined features through extra convolutional layers, even though the final feature vector is the same shape. The details of the VGG16 and VGG19 architectures is presented in Table 4.4.

Table 4.4 Comparative Analysis of VGG16 and VGG19 architecture

Aspect	**VGG16**	**VGG19**
Conv Layers	13	16
Final Feature Map Shape	7×7×512	7×7×512
Total Features Extracted	25,088 per image	25,088 per image
Depth of Representation	Shallower	Deeper
Feature Richness	High	Slightly Higher
Trainable Parameters	~138M	~144M
Training Time	Faster	Slower

VGG16 has 13 convolutional layers, and VGG19 has 16. The extra 3 layers in VGG19 are added in blocks 3, 4, and 5. Both models give the same final output size (7×7×512), but VGG19 applies more convolution operations before each pooling step. This helps it learn better and more detailed patterns. More layers mean the model passes through more ReLU activations and can understand more complex shapes and textures. Each extra layer enables deeper neurons in VGG19 to see a slightly larger portion of the image, which helps capture finer details. But VGG19 is bigger and slower, with about 144 million parameters to learn, while VGG16 has 138 million. So, VGG19 may need more time to train and can overfit if the dataset is small. Due to the larger size and slower speed of VGG19, we are using the simple Big Bang–Big Crunch (BB-BC) optimization algorithm in the VGG19-PCA-BBBC approach instead of the parallel version of BB-BC to reduce computational load and manage processing time effectively.

The second phase in the proposed approach is dimensionality reduction. After feature extraction, the obtained feature maps are very large and may contain redundant information. To reduce the feature map's complexity and prevent overfitting, PCA is applied. First, the covariance matrix of the feature set is computed to measure the relationships between different features. Eigenvalues and eigenvectors are determined by performing calculations using the covariance matrix. The principal components are represented by the eigenvectors and they are perpendicular to each other. The principal components are ranked by their corresponding eigenvalues, which reflect how much variance each component explains. Only the best principal components that capture the most variance are retained, reducing the feature set's dimensionality.

After reducing the dimensionality, PCA selected 1222 principal components from the 25088 features extracted by VGG19. To select the near optimal features for classification, the principal components selected by PCA are provided to the optimization algorithm BBBC. The MSE is 0.0358%, which means the data loss was very small.

The third phase of the proposed approach is the selection of the near-optimal features. BBBC algorithm is used for the feature optimization and selection. As shown in algorithm 4.2, for the feature selection purpose, the BBBC algorithm starts with the Big Bang phase. In Big Bang phase, a population of feature sets is created by randomly generating the candidate solutions. These candidate solutions represent various subsets of the features. The fitness of each feature subset is calculated. After the exploration in the Big Bang phase, the algorithm enters a Big Crunch phase where it selects the feature set with the best fitness. The best candidate solution is called the elite. Then, a new population is created by adding or subtracting a small random number to the elite. Now, the fitness of each feature set is calculated and the feature set with best fitness is considered for the further exploration.

The algorithm keeps repeating the big bang and big crunch phases until a stopping condition is met. Common stopping conditions are reaching the maximum number of iterations, when the population becomes stable, or when a solution is found that meets

the desired classification accuracy for leukemia detection. The switching between exploration and exploitation allows the BBBC algorithm to find the near-optimal solutions quickly.

For the feature selection purpose, we considered ten candidate solutions in the population of feature subsets. Each feature subset contains 200 features. After optimization, BBBC selected 200 optimized features out of the original 1222 principal components. These selected features are then passed to the final classifier for leukemia detection.

If BBBC is not used after PCA, all 1222 features are used, which increases training time and may include less useful data. This can reduce accuracy and make the model slower. BBBC selects only the most helpful 200 features, removing noise and saving time. It helps the model run faster, use less memory, and give better results. The results of the proposed approach with and without BBBC are shown in Table 4.5.

In the last step of the proposed approach, the model is trained based on the best selected features. The detection of leukemia is also done in this step by the fully connected layers. Once the feature set has been optimized by the BBBC algorithm, it is directed to fully connected layers for classification purposes. The input which is transferred to dense layers is an optimized feature set. The activation function ReLU (Rectified Linear Unit) is used in the first two fully connected layers, and a softmax function is applied in the final fully connected layer.

The methodology presented in Figure 4.3 utilized the population-based BBBC algorithm to find the best solution by optimizing the features extracted with the customized VGG19 model and then reduced by PCA.

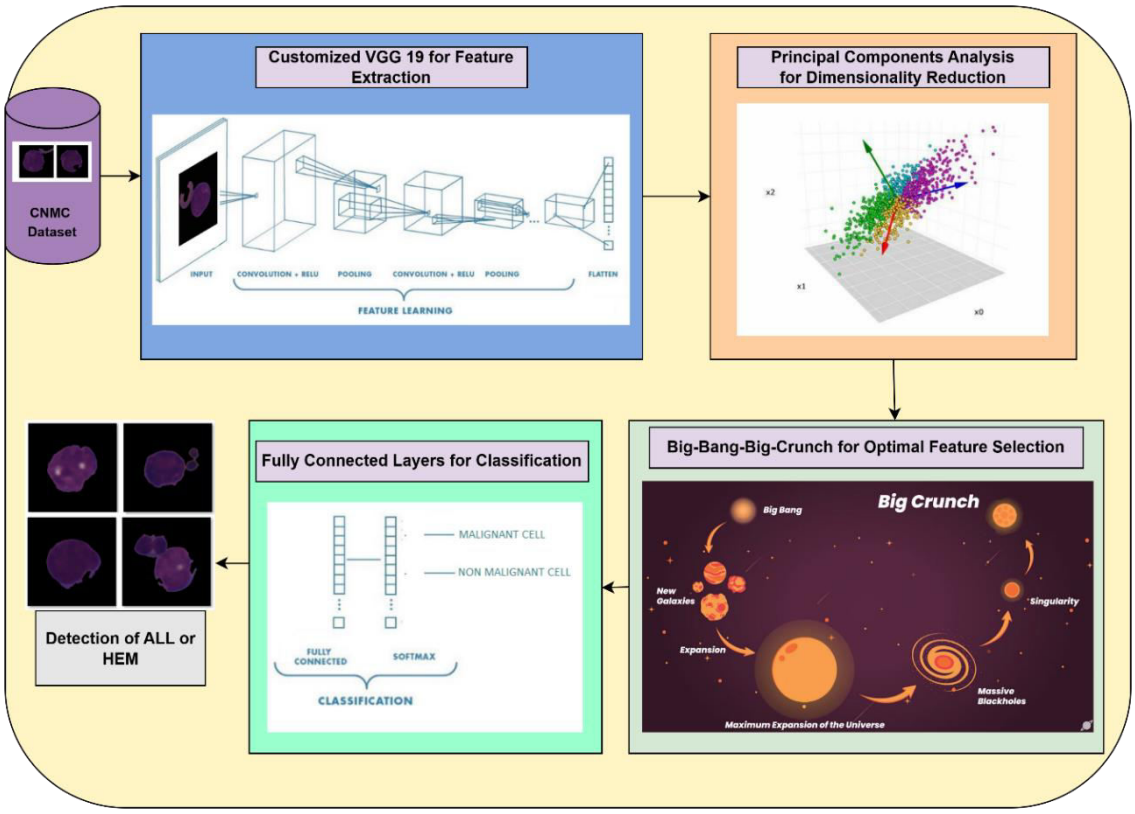


Figure 4.3 Proposed soft computing-based approach for leukemia detection

Algorithm 4.2 represents the working of proposed soft computing-based approach.

Algorithm 4.2: VGG19-PCA-BBBC algorithm

BEGIN

Step 1: Feature Extraction using VGG19

a) Preprocess the CNMC_2019 images dataset named as I to resize each image of I to 224x224 pixels. This preprocessed dataset is called as $I_{preprocessed}$.
b) Pass $I_{preprocessed}$ through the customized VGG19 architecture to extract feature map F_{pooled}

Step 2: Dimensionality Reduction using PCA

a) Compute covariance matrix
C=Cov(F_{pooled})
b) Compute eigenvalues λ_i and eigenvectors v_i of the covariance matrix
$Cv_i=\lambda_i v_i$ for all i
c) Rank the eigenvectors based on eigenvalues λ_i:
Ranked_eigenvectors=Sort(v_i,λ_i)
d) Select top k eigenvectors to form $F_{reduced}$
$F_{reduced}$=SelectTop(v_i,k)

Step 3: Feature Selection using BBBC Algorithm

1. **BIG BANG PHASE**

a) Generate initial population of candidate solutions $S_1, S_2, \ldots, S_n$ by randomly selecting feature subsets from $F_{reduced}$.
b) Evaluate fitness $f(S_i)$ of each candidate solution S_i
$f(S_i)$=Fitness(S_i)

2. **BIG CRUNCH PHASE**

a) Sort all the candidate solutions of the population as per their fitness $f(S_i)$.
b) Select the candidate solution with best fitness value (elite).
S'=elite

3. **BIG BANG PHASE**

a) Create a new population around S' by adding or subtracting a small number to/from S'.

4. **REPEAT STEP 2 AND STEP 3 UNTIL STOPPING CONDITION IS NOT MET**

5. **RETURN ELITE.**

Step 4: Model Training using dense layer for Leukemia Detection

END

4.5. Results and Discussion

For the testing purpose, we implemented the proposed algorithm in python. The proposed approach is tested on CNMC_2019 dataset. To analyse its performance, we compared the results of the proposed approach with 15 other leukemia detection approaches. For the performance analysis purpose, we compared the proposed approach with 8 transfer learning-based approaches and 7 machine learning approaches. Table 4.5 and Figure 4.4 displays a comparison between the proposed approach and 8 existing transfer learning-based methods. Table 4.6 and Figure 4.5 compare the proposed approach with 7 other machine learning-based models. The proposed approach is evaluated using performance metrics like accuracy, recall, precision, and F1-score. From Table 4.5, Table 4.6, Figure 4.4, and Figure 4.5, we can see that the proposed approach VGG19-PCA-BBBC achieved 96.24% accuracy, 97.83% precision, 96.77% recall, and 97.30% F1-score. In comparison, VGG16-PCA-PB3C achieved 95.38% accuracy, 97.75% precision, 95.60% recall, and 96.67% F1-score. VGG19-PCA achieved 93.02% accuracy, 95.51% precision, 94.44% recall and 94.97% F1-score.

These results clearly indicate the supremacy of the proposed approach in comparison with existing approaches.

Table 4.5 Comparison of proposed VGG19-PCA-BBBC with existing transfer learning-based approaches

CNN Architecture	Accuracy	Precision	Recall	F1 Score
Inception V3	86.01%	94.68%	85.58%	89.90%
CNN+ECA+VGG16 [96]	91.1%	-	-	-
TL+(GA,PCA)+MLP [102]	90.71%	-	95.26%	-
ResNet101 ensembled models [118]	85.11%	-	-	88.94%
VGG16	89.05%	96.77%	88.24%	92.31%
VGG19	90.65%	96.63%	89.58%	92.97%
VGG16-PCA	91.04%	96.77%	90.91%	93.75%
VGG19-PCA	93.02%	95.51%	94.44%	94.97%
VGG16-PCA-PB3C	95.38%	97.75%	95.60%	96.67%
VGG19-PCA-BBBC	96.24%	97.83%	96.77%	97.30%

Table 4.6 Comparison of proposed VGG19-PCA-BBBC with existing machine learning-based approaches

Classifier	Accuracy	Precision	Recall	F1 Score
SVM	74.8%	69.91%	76.78%	73.19%

Classifier	Accuracy	Precision	Recall	F1 Score
Random Forest	81.20%	79.67%	81.66%	80.65%
Logistic Regression	83.20%	82.11%	83.47%	82.78%
K Nearest Neighbour	62%	71.54%	59.45%	64.94%
Decision Tree	70%	69.10%	69.67%	69.38%
Bagging	67.60%	69.91%	66.15%	67.98%
ML based Classification, IEEE ISBI 2019 Challenge [50]	91%	-	-	-
VGG16-PCA-PB3C	95.38%	97.75%	95.60%	96.67%
VGG19-PCA-BBBC (Proposed Approach)	96.24%	97.83%	96.77%	97.30%

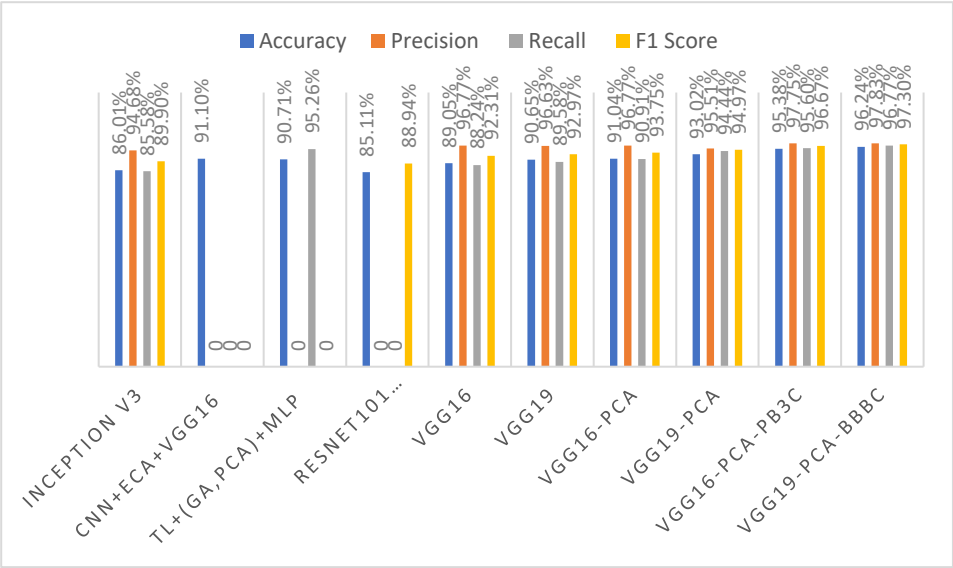


Figure 4.4 Comparison of VGG19-PCA-BBBC with transfer learning approaches

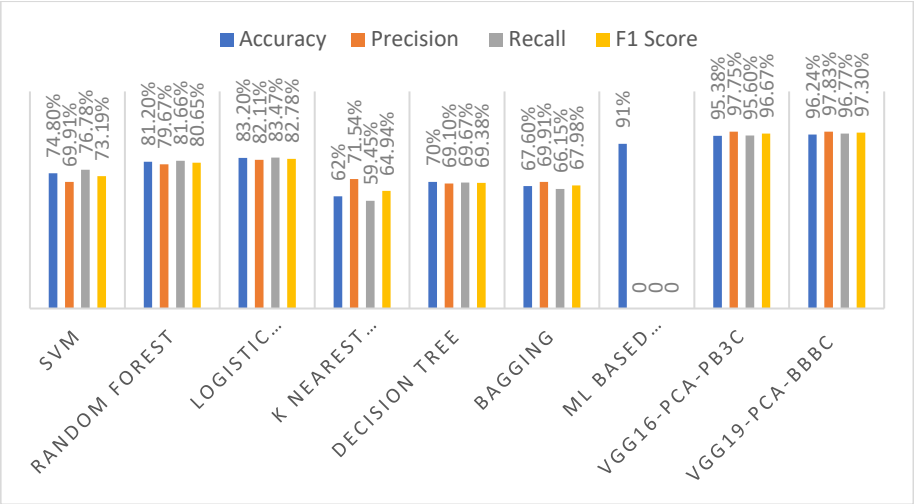


Figure 4.5 Comparison of VGG19-PCA-BBBC with existing machine learning-based approaches

This high performance is likely due to the combination of the VGG19 architecture, PCA, BBBC algorithm, and fully connected classification layer. This indicates that the deeper VGG19 architecture combined with the BBBC optimization algorithm can capture more discriminative features for leukemia detection, resulting in more accurate classifications.

4.6. Summary

This chapter proposed a new soft computing-based approach, named as VGG19-PCA-BBBC for detecting leukemia detection from peripheral blood cell images. The proposed approach utilizes VGG19 architecture to extract deep features, applies PCA to reduce the size of the leukemia dataset. Further, the near-optimal features are selected using BBBC algorithm. The CNMC dataset, which contains labelled peripheral blood cell images, was used to train and test this proposed soft computing-based approach. The results showed that the proposed soft computing-based approach performed better than other existing leukemia detection approaches.

Chapter 5 HP3PGA-3PGA: A New Hybrid Soft Computing based Algorithm for Early Detection of Leukemia

This chapter proposes a new hybrid soft computing-based algorithm for early detection of leukemia. The proposed algorithm works in two different phases: a multi-population phase and a single population phase. The multi-population phase focuses on exploring a wide range of near-optimal solutions, while the single-population phase is suitable for refining a solution when diversity is less of a concern. Both phases are implemented using two variations on three parent genetic algorithms namely Parallel Three Parent Genetic algorithm (P3PGA) and Three Parent Genetic Algorithm (3PGA). The proposed algorithm is implemented in MATLAB software and tested on 80 standard benchmark functions of CEC 2021 test suite. The performance of the proposed algorithm is compared with the 10 recent algorithms. The performance results clearly show that the proposed algorithm is better than the other 10 algorithms.

Further, we validated the proposed algorithm to evolve the near optimal architecture of CNN. The HP3PGA-3PGA based neural architecture search method is implemented in python and tested against 17 other methods for diagnosing leukemia.

The major contributions of this chapter are as below:

1. Proposed a new hybrid bio inspired HP3PGA-3PGA search & optimization algorithm.
2. The proposed algorithm is tested on 80 functions from the CEC-2021 Benchmark suite and compared with 10 recent algorithms.
3. Proposed a HP3PGA-3PGA based Neural Architecture Search (NAS) approach to evolve near optimal CNN architecture for leukemia detection in blood smear images.
4. The proposed NAS approach is compared with existing 17 leukemia detection approaches.

Section 5.1 introduces genetic algorithm and the 3-parent genetic algorithm. In Section

5.2, the HP3PGA-3PGA algorithm is proposed. Section 5.3 includes the simulation and performance analysis. Section 5.4 presents the HP3PGA-3PGA based neural architecture search approach for leukemia detection. Section 5.5 concludes the chapter.

5.1. Introduction

A genetic algorithm is a simple way to solve a problem by copying how nature works. It starts with variety of possible solutions, called chromosomes. The best solutions are picked and slightly changed to make new ones by using the methods inspired by natural processes. These strategies consist of selection (choosing the optimal solutions from the existing population), crossover (merging two or more solutions to create a new one), and mutation (introducing minor, random alterations to a solution for diversity) [140].

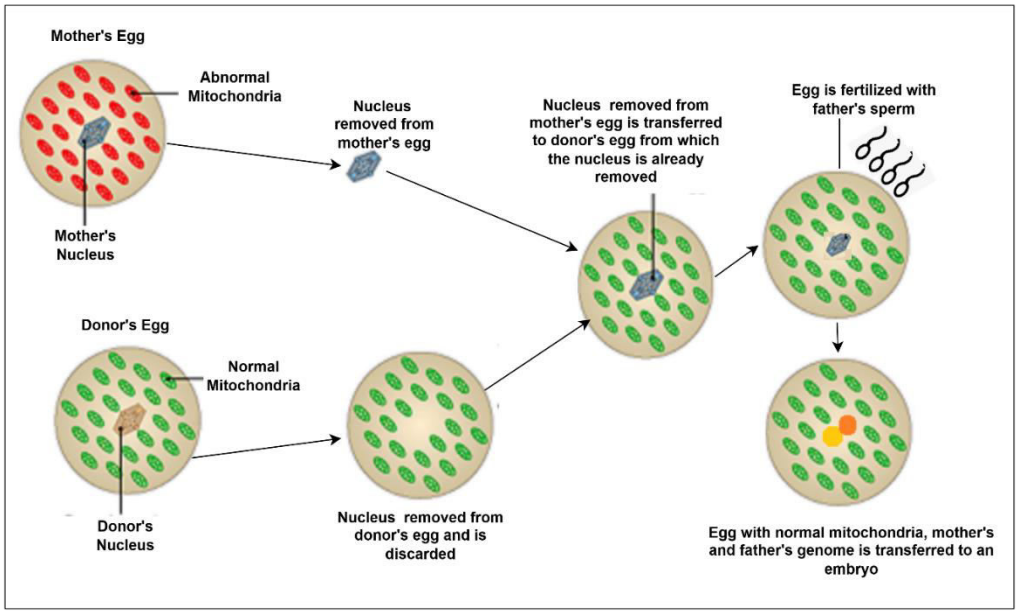


Figure 5.1 Three parent genetic algorithm concept

A new set of solutions is made to replace some or all the old ones. This cycle of evaluation, selection, crossover, mutation, and replacement keeps repeating. It continues until a stopping condition is met, such as reaching a set number of generations or achieving an optimal fitness level. The algorithm keeps improving the solutions step by step until an optimal solution is found.

The concept of 3PGA as shown in Figure 5.1 is an enhancement of the traditional GA by introducing a third parent into the recombination process [141-144]. The extension is to provide increased genetic diversity and convergence rates by the recombination of genes from three rather than two parents, as generally the case in traditional GAs.

5.2. Proposed Hybrid Bio Inspired Algorithm

This section introduces the new Hybrid P3PGA-3PG Algorithm, which combines the strengths of both parallel population and single population-based search & optimization algorithms. It brings together the operations of three-parent and parallel three-parent algorithms. The proposed hybrid algorithm has proven valuable in optimization, as it attempts to reduce the possibility of being trapped by a local optimum and to increase the possibilities of finding the global optimum. P3PGA's multi population based global search capability allows for a thorough investigation of the solution space, while the hybridization techniques refine local searches by integrating it with 3PGA. The working of proposed P3PGA-3PGA is shown in algorithm 5.1.

Algorithm 5.1: HP3PGA-3PGA Algorithm

Begin

Step1: Let **NP** be the number of populations, **N** be the number of individuals (candidate) in each population, i ∈ {1,2,…..,**NP}**
Initialize each population randomly by respecting all bounds and violations
Set gen = 1

WHILE gen <= TC:

for i = 1 to NP:
Apply mitochondrial modification on the i_{th} 2-parent (2P) population to produce i_{th} 3-parent (3P) population. This is done by adding and/or subtracting a small random number in it.
Combine the 2P population and 3P population using equation 5.1

$$P_{combined}[i]=P_2[i] \cup P_3[i] \quad (5.1)$$

Evaluate and record fitness of each individual of $P_{combined}[i]$.
Apply Genetic process to generate new 2P.

a) Select top N best individuals from the combined population $P_{combined}$ that maximize fitness using equation 5.2.

$$2P[i] = top_N(P_{combined}[i]) \quad (5.2)$$

where top_N is the N top features of $P_{combined}[i]$

b) With high probability P_c, perform crossover in 2P[i].
c) With a low probability, mutate offsprings.
d) Evaluate fitness of 2P[i]. Idenfity and update the local best individual l_{best} in $P_{combined}$[i]
e) Replace the weak candidate solutions with stronger offspring by keeping the size fixed of 2P[i].
f) Check and correct the bounds of 2P[i] (if needed).

end for (i)

For i = 1 to NP:
Replace the gene of l_{best}[i] with corresponding gene of g_{best}[i], with a certain probability.

End for (i)

Step 2: Evaluate fitness of each population and update the globally best individual g_{best} amongst all populations. Set counter1 = 0 and counter2 = 0

Step 3: IF 3PGA_switch_flag is true then go to step 4; otherwise, go to step 5

Step 4: IF counter1 = 0 THEN
Select g_{best} candidate solution and generate a new population around it. (The new population is two parents "2P") for the 3PGA algorithm.
Set counter1 = 1
END_IF
Perform mitochondrial changes in "2P" population and create a new "3P" population.
Evaluate the fitness of all candidate solutions in the population of "3*P*".
Select "N" best candidate solution from "3P" population and again create "2P" population using the general genetic process.
Compute g_{best} from "2*P*" population and set counter2 = 1
Go to step 3.
END_IF

Step 5: IF counter2 = 1 THEN
Create population "NP" population around 3PGA g_{best}

ENDIF

Step 6: Go to step 2.

End_while

End

The proposed hybrid P3PGA-3PGA algorithm can be considered one step forward in genetic algorithms, which iteratively evolves populations of candidate solutions by switching between multi population and single population behaviour of P3PGA and 3 PGA algorithms. It first initializes several random populations. Each population is a set

of individuals or candidate solutions within prescribed bounds. In each generation, a 3P population is made by adding or subtracting a small random number to the current 2P populations. This process is performed to correct the mitochondrial defects in each population. Further for every population, the general genetic operations like selection, crossover, and mutation are performed. After these genetic operations, the local best and global best are found. Now, the gene of the local best from each population is replaced with the gene of the global best solution based on a decided probability. This is known as combination operation between the lbest and gbest candidate solutions. After the specific number of iterations, switch to 3PG algorithm. For, 3PG algorithm, form the new population by slightly adjusting the global best through the addition or subtraction of a small random number. Now the single population has been created. This single population is called as 2P of 3PGA. Generate a new 3P population around the 2P population. After this, evaluate the performance of new 3P population and perform all genetic operations on it. The alternation between 3PGA and P3PGA continues for the maximum number of iterations.

5.3. Simulation and Performance Analysis

5.3.1. Simulation Results

For the testing purpose, we implemented HP3PGA-3PGA algorithm in Matlab. The proposed algorithm is tested on different benchmark functions of the CEC-2021 test bench suite. CEC 2021 test suite consists of eight category and each category contains 10 different types of benchmark functions. Thus, CEC 2021 test suite consists of 80 benchmark functions. We conducted 25 trials on each function of CEC 2021 test suite. Thus, a total number of 2000 trials (25*80) were conducted to evaluate the performance of the proposed algorithm. To evaluate the performance of the proposed algorithm, we calculated the average error across 25 trials for each function. Simulations were carried out on desktops with a 3.8GHz Core i7 processor, and 16GB RAM. Table 5.1 lists all the functions from the test bench suite CEC 2021 [145]. The performance results of the proposed algorithm are compared with 10 recent advanced algorithms, namely, APGSK_IMODE (Adaptive Parameters Gaining Sharing Knowledge and Improved Multi operator Differential Evolution algorithm), DEDMNA (Differential Evolution

with Distance-based Mutation-selection), MadDE (Improved DE through Bayesian Hyperparameter Optimization), RB_IPOP_CMAES_PPMF, J21 (Self-adaptive Differential Evolution Algorithm with Population Size Reduction for Single Objective Bound-Constrained Optimization), NL-SHADE-RSP (LSHADE algorithm with Adaptive Archive and Selective Pressure), SOMA-CLP (Self-organizing Migrating Algorithm with Clustering-aided migration and adaptive Perturbation), MLS-LSHADE (Multi-start Local Search Algorithm with L-SHADE), L-SHADE-OrdRW (LSHADE based on ordered and roulette-wheel-based mutation), and PBO (Pollination Based Optimization).

Table 5.1 Various functions of CEC2021 test suite.

Name of Function	**Number**	**Origin of Function**	$\boldsymbol{Fi*}$
Unimodal Function, also known as Sphere Function	F1	CEC2017	100 score
Basic Functions, also known as Rastrigin Function	F2	CEC2014	1100 score
Basic Functions, also known as Ackley Function	F3	CEC2017	700 score
Basic Functions, also known as Griewank Function	F4	CEC2017	1900 score
Hybrid Functions, also known as Rosenbrock Function	F5	CEC2014	1700 score
Hybrid Functions, also known as Weierstrass Function	F6	CEC2017	1600 score
Hybrid Functions, also known as Schwefel Function	F7	CEC2014	2100 score
Composition Functions Set – 1	F8	CEC2017	2200 score
Composition Functions Set – 2	F9	CEC2017	2400 score
Composition Functions Set – 3	F10	CEC2017	2500 score

Search range: $[-100,100]^D$

As shown in Table 5.1, there are total number of 10 functions of CEC2021 test suite. These functions are used to evaluate single-objective optimization algorithms and compare the performance of new algorithms with the existing ones. These functions are built upon CEC2017 and CEC2014 test suites. Each function is represented by an identifier from F1 to F10. CEC2021 test suite functions are categorized into four types:

1. Unimodal function (also known as sphere function and is represented by F1)
2. Basic functions (Rastrigin function - F2, Ackley function – F3 and Griewank function -F4)
3. Hybrid functions (Rosenbrock function – F5, Weierstrass function – F6 and Schwefel function – F7)
4. Composition function (Set 1 - F8, Set 2 - F9 and Set 3 - F10).

The unimodal function is used to test the convergence ability of optimization algorithms. Basic and hybrid functions test the exploration ability, while composition functions evaluate the optimization algorithm's adaptability to complex landscapes. Fi* represents the best-known function value after transformations like shifting and rotation. A higher Fi* value indicates that the function is more complex. The search range defines the domain of input variables and specifies that all input variables for these functions must lie within the specified range.

The performance of the HP3PGA-3PGA along with other algorithms of the standard test bench suite for CEC2021 is shown in Table 5.2. The evaluation is conducted on 10 benchmark functions (F1 to F10) (as shown in Table 5.1). These functions undergo eight different transformations:

1. Basic
2. Bias
3. Shift
4. Rotation
5. Bias and Shift
6. Bias and Rotation

7. Shift and Rotation
8. Bias, Shift, and Rotation

Each transformation makes the function more complex. The numerical values represent the error or deviation from the optimal solution. The smaller numbers mean the result is better.

Table 5.2 Performance results of HP3PGA-3PGA along with other algorithms on the standard test bench suite for CEC2021

Function	**Fn 1**	**Fn 2**	**Fn 3**	**Fn 4**	**Fn 5**	**Fn 6**	**Fn 7**	**Fn 8**	**Fn 9**	**Fn 10**
	Basic									
APGSK_ IMODE	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DEDMNA	0.000000	0.000000	2.18E+00	1.28E-01	0.000000	3.56E-03	5.68E-04	0.000000	0.000000	4.80E+01
MadDE	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
RB_IPOP _CMAES _PPMF	0.000000	2.67E-01	9.23E+00	1.02E+00	3.78E+01	1.43E+00	7.50E+00	0.000000	1.94E-07	4.80E+01
J21	0.000000	0.000000	5.63E+00	2.43E-01	0.000000	3.40E-02	7.97E-03	0.000000	0.000000	4.64E+01
NL-SHADE-RSP	0.000000	0.000000	0.000000	1.43E-02	0.000000	6.87E-03	1.38E-03	0.000000	0.000000	1.93E-03
SOMA-CLP	0.000000	1.04E-01	3.61E+08	3.60E-01	0.000000	3.11E-02	2.15E-03	0.000000	0.000000	7.71E+02
MLS-LSHADE	0.000000	0.000000	2.25E+00	6.58E-03	0.000000	1.83E-03	0.000000	0.000000	0.000000	6.92E-03
L-SHADE-OrdRW	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	4.09E-06
PBO	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
HP3PGA-3PGA	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
Function	**Fn 1**	**Fn 2**	**Fn 3**	**Fn 4**	**Fn 5**	**Fn 6**	**Fn 7**	**Fn 8**	**Fn 9**	**Fn 10**
	Bias									
APGSK_ IMODE	0.00 E + 00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.99E-04
DEDMNA	0.000000	1.50E+01	9.82E+00	4.42E-01	3.58E + 00	3.74E – 01	1.57E - 01	3.62E + 00	0.000000	5.14E+01
MadDE	0.000000	0.000000	0.000000	0.000000	0.000000	8.73E-06	0.000000	0.000000	0.000000	0.000000
RB_IPOP _CMAES _PPMF	0.000000	3.06E-01	8.87E+00	1.01E+00	3.49E+01	1.20E+00	1.45E+01	0.000000	1.98E-07	6.69E+01
J21	0.000000	1.62E+01	1.16E+01	8.44E-01	4.35E+00	1.32E+00	3.53E-01	1.59E+01	0.000000	5.16E+01

Function	Fn 1	Fn 2	Fn 3	Fn 4	Fn 5	Fn 6	Fn 7	Fn 8	Fn 9	Fn 10
NL-SHADE-RSP	0.000000	5.46E+00	5.05E+00	3.83E-01	3.31E+00	4.19E-01	1.93E-01	4.24E+01	0.000000	4.81E+01
SOMA-CLP	5.07E-08	1.61E+06	1.60E+03	9.94E-01	7.16E+07	4.52E-01	6.99E-01	1.83E+01	0.000000	5.27E+01
MLS-LSHADE	0.000000	0.000000	0.000000	0.000000	0.000000	7.81E-07	0.000000	0.000000	0.000000	4.48E-03
L-SHADE-OrdRW	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	2.62E-08	0.000000	0.000000	1.94E-06
PBO	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	5.18E+01
HP3PGA-3PGA	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
Function	**Fn 1**	**Fn 2**	**Fn 3**	**Fn 4**	**Fn 5**	**Fn 6**	**Fn 7**	**Fn 8**	**Fn 9**	**Fn 10**
	Shift									
APGSK_IMODE	0.000000	0.000000	1.05E+01	2.77E-01	0.000000	1.91E – 02	1.27 E-03	4.06 E+01	9.33E+01	3.90E+02
DEDMNA	0.000000	0.000000	4.81E+00	1.56E-01	0.000000	5.31E-03	5.10 E-04	1.40E+01	8.67E+01	3.73 E+02
MadDE	0.000000	0.000000	1.09E+01	1.88E-01	0.000000	1.62E-02	1.42E-03	8.79E+01	9.33E+01	4.00E+02
RB_IPOP_CMAES_PPMF	0.000000	2.75E+02	1.12E+01	1.06E+00	1.25E+02	5.06E+01	3.04E+01	9.72E+01	2.56E+02	4.00E+02
J21	0.000000	2.08E-03	1.02 E+01	2.54E - 01	0.000000	2.54E-02	5.62E-03	0.000000	1.10E+02	3.63E+02
NL-SHADE-RSP	0.000000	0.000000	1.02E+01	9.42E-02	0.000000	7.57E-03	2.01E-03	5.45E-01	8.01E+01	3.90E+02
SOMA-CLP	0.000000	9.23E-02	2.93E+06	3.58E-01	7.92E-06	2.42E-02	2.65E-03	7.19E-01	1.52E+02	3.94E+02
MLS-LSHADE	0.000000	2.08E-02	1.01E+01	1.21E-01	0.000000	4.45E-02	7.03E-03	6.23E+01	2.14E+02	3.87E+02
L-SHADE-OrdRW	0.000000	4.37E-02	1.09E+01	1.89E-01	6.17E+00	2.71E-01	2.98E-01	1.00E+02	3.19E+02	4.00E+02
PBO	0.000000	6.95E+00	1.15 E+01	4.74 E-01	1.18E+01	6.73 E-01	4.47 E-01	0.000000	1.00E+02	4.00E+02
HP3PGA-3PGA	0.000000	6.25E-02	0.000000	5.39E-01	0.000000	7.92E-02	1.11E-02	0.000000	3.27E+02	4.00E+02
Function	**Fn 1**	**Fn 2**	**Fn 3**	**Fn 4**	**Fn 5**	**Fn 6**	**Fn 7**	**Fn 8**	**Fn 9**	**Fn 10**
	Rotation									
APGSK_IMODE	0.000000	1.84 E+01	1.34 E+01	2.14E-01	4.10E+00	2.74 E-01	1.12E-01	6.25E+01	8.84E+01	2.79E+02
DEDMNA	0.000000	1.63E+01	1.16 E+01	4.37E-01	1.20 E+01	1.76E-01	1.88E-01	5.48 E+01	8.33E+01	3.68E+02
MadDE	0.000000	1.23E+01	1.36E+01	3.51E-01	1.10E+00	3.26E-01	2.33E-01	9.60E+01	9.00E+01	3.98E+02
RB_IPOP_CMAES_PPMF	0.000000	4.98E+02	1.14E+01	7.86E-01	1.14E+02	4.68E+01	7.50E+01	9.61E+01	2.05E+02	4.18E+02

Function	Fn 1	Fn 2	Fn 3	Fn 4	Fn 5	Fn 6	Fn 7	Fn 8	Fn 9	Fn 10
J21	0.000000	2.16 E+01	1.17E+01	7.96E-01	1.73E+00	5.17E-01	1.37 E+00	9.31 E+00	9.00 E+01	3.18E+02
NL-SHADE-RSP	0.000000	1.21E+01	1.33E+01	1.30E-01	6.20E+00	2.97E-01	7.83E-02	2.39E+01	7.52E+01	3.98E+02
SOMA-CLP	2.10E-07	4.34E+06	1.38E+01	2.15E-01	1.77E+06	3.33E-01	1.96E-01	3.35E+01	1.91E+02	3.99E+02
MLS-LSHADE	0.000000	1.87E+01	1.31E+01	4.01E-01	6.33E+00	6.39E-01	4.38E-01	6.55E+01	1.00E+02	3.88E+02
L-SHADE-OrdRW	0.000000	5.20E+00	1.18E+01	3.36E-01	2.90E+01	5.81E-01	2.29E+00	1.00E+02	2.90E+02	4.28E+02
PBO	1.56E+01	1.51E+01	1.07E+01	5.39E-01	7.79E+01	1.32E+00	1.36E+00	1.14E+01	3.79E-04	3.98E+02
HP3PGA-3PGA	0.000000	3.19E+02	2.84E+01	1.37E+00	3.00E+02	1.21E+02	4.20E-01	1.02E+02	3.78E+02	3.99E+02
Function	**Fn 1**	**Fn 2**	**Fn 3**	**Fn 4**	**Fn 5**	**Fn 6**	**Fn 7**	**Fn 8**	**Fn 9**	**Fn 10**
	Bias and Shift									
APGSK_IMODE	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
DEDMNA	0.000000	0.000000	2.14E+00	1.28E-01	0.000000	3.56E-03	4.75E-04	0.000000	0.000000	4.80E+01
MadDE	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
RB_IPOP_CMAES_PPMF	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J21	0.000000	0.000000	4.71E+00	2.39E-01	0.000000	3.27E-02	1.98E-02	0.000000	0.000000	4.64E+01
NL-SHADE-RSP	0.000000	0.000000	0.000000	1.34E-02	0.000000	7.48E-03	1.12E-03	0.000000	0.000000	2.07E-03
SOMA-CLP	0.000000	1.19E-01	3.46E+07	3.08E-01	0.000000	2.84E-02	2.95E-03	0.000000	0.000000	4.98E+07
MLS-LSHADE	0.000000	0.000000	2.32E+00	3.29E-03	0.000000	1.53E-04	0.000000	0.000000	0.000000	7.21E-03
L-SHADE-OrdRW	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
PBO	0.000000	3.12E-01	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	4.80E+01
HP3PGA-3PGA	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	3.01E-03
Function	**Fn 1**	**Fn 2**	**Fn 3**	**Fn 4**	**Fn 5**	**Fn 6**	**Fn 7**	**Fn 8**	**Fn 9**	**Fn 10**
	Bias and Rotation									
APGSK_IMODE	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	3.88E-04
DEDMNA	0.000000	1.50E+01	9.82E+00	4.42E-01	3.58E+00	3.74E-01	1.57E-01	3.62E+00	0.000000	5.14E+01
MadDE	0.000000	0.000000	0.000000	0.000000	0.000000	3.15E-05	0.000000	0.000000	0.000000	0.000000

Function	Fn 1	Fn 2	Fn 3	Fn 4	Fn 5	Fn 6	Fn 7	Fn 8	Fn 9	Fn 10
RB_IPOP_CMAES_PPMF	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J21	0.000000	1.62E+01	1.13E+01	8.39E-01	4.34E+00	1.33E+00	3.41E-01	1.24E+01	0.000000	5.16E+01
NL-SHADE-RSP	0.000000	6.46E+00	5.27E+00	4.30E-01	2.10E+00	4.23E-01	2.13E-01	4.41E+01	0.000000	5.17E+01
SOMA-CLP	6.97E-08	2.41E+07	1.56E+07	1.08E+00	7.19E+07	4.62E-01	3.07E-01	3.77E+01	0.000000	5.21E+01
MLS-LSHADE	0.000000	0.000000	0.000000	0.000000	0.000000	8.75E-07	0.000000	0.000000	0.000000	4.29E-03
L-SHADE-OrdRW	0.000000	0.000000	0.000000	0.000000	0.000000	6.29E-08	0.000000	0.000000	0.000000	1.37E-06
PBO	0.000000	3.12E-01	0.000000	3.25E-01	0.000000	4.49E-01	0.000000	0.000000	0.000000	5.16E+01
HP3PGA-3PGA	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
Function	**Fn 1**	**Fn 2**	**Fn 3**	**Fn 4**	**Fn 5**	**Fn 6**	**Fn 7**	**Fn 8**	**Fn 9**	**Fn 10**
	Shift and Rotation									
APGSK_IMODE	0.000000	0.000000	1.09E+01	2.61E-01	0.000000	1.83E-02	1.09E-03	5.00E+01	9.67E+01	3.70E+02
DEDMNA	0.000000	0.000000	4.81E+00	1.56E-01	0.000000	5.31E-03	5.08E-04	1.40E+01	8.67E+01	3.73E+02
MadDE	0.000000	8.33E-03	1.09E+01	1.93E-01	0.000000	1.45E-02	1.55E-03	9.40E+01	9.00E+01	4.00E+02
RB_IPOP_CMAES_PPMF	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
J21	0.000000	2.08E-03	9.85E+00	2.49E-01	0.000000	3.38E-02	7.72E-03	0.000000	1.04E+02	3.89E+02
NL-SHADE-RSP	0.000000	0.000000	1.09E+01	1.01E-01	0.000000	5.71E-03	4.90E-03	0.000000	7.60E+01	4.00E+02
SOMA-CLP	0.000000	1.06E-01	9.79E+02	3.48E-01	0.000000	2.94E-02	2.91E-03	3.42E-03	1.27E+02	3.87E+02
MLS-LSHADE	0.000000	3.12E-02	9.82E+00	1.58E-01	0.000000	3.24E-02	5.70E-03	5.20E+01	1.95E+02	3.90E+02
L-SHADE-OrdRW	0.000000	3.33E-02	1.09E+01	1.94E-01	6.16E+00	3.98E-01	2.47E-01	1.00E+02	3.19E+02	4.00E+02
PBO	0.000000	3.54E+00	2.44E+00	3.22E-01	4.38E-01	6.49E-01	7.27E-01	0.000000	1.00E+02	4.00E+02
HP3PGA-3PGA	0.000000	2.50E-02	1.12E+01	5.08E-01	0.000000	3.51E-01	3.48E-01	0.000000	1.04E+02	4.00E+02
Function	**Fn 1**	**Fn 2**	**Fn 3**	**Fn 4**	**Fn 5**	**Fn 6**	**Fn 7**	**Fn 8**	**Fn 9**	**Fn 10**
	Bias, Shift and Rotation									
APGSK_IMODE	0.000000	9.67E+00	1.38E+01	1.78E-01	3.93E+00	3.57E-01	5.43E-02	6.62E+01	9.41E+01	2.79E+02
DEDMNA	0.000000	1.63E+01	1.16E+01	4.37E-01	1.20E+01	1.76E-01	1.88E-01	5.48E+01	8.33E+01	3.68E+02

Function	Ƒn 1	Ƒn 2	Ƒn 3	Ƒn 4	Ƒn 5	Ƒn 6	Ƒn 7	Ƒn 8	Ƒn 9	Ƒn 10
MadDE	0.000000	2.16E+01	1.40E+01	3.72E-01	9.63E-01	3.02E-01	1.72E-01	9.13E+01	9.00E+01	3.98E+02
RB_IPOP_CMAES_PPMF	0.000000	5.53E+02	1.05E+01	8.66E-01	9.35E+01	4.97E+01	1.21E+02	9.71E+01	1.72E+02	4.15E+02
J21	0.000000	2.16E+01	1.15E+01	7.95E-01	1.70E+00	5.17E-01	1.91E+00	1.14E+01	9.00E+01	3.38E+02
NL-SHADE-RSP	0.000000	1.45E+01	1.31E+01	1.38E-01	5.03E+00	3.01E-01	2.78E-02	3.24E+01	7.97E+01	3.88E+02
SOMA-CLP	7.61E-08	6.91E+06	1.35E+01	2.00E-01	3.15E+08	2.73E-01	1.84E-01	2.81E+01	2.63E+02	3.89E+02
MLS-LSHADE	0.000000	1.82E+01	1.26E+01	3.80E-01	5.15E+00	7.30E-01	3.02E-01	7.06E+01	9.12E+01	3.84E+02
L-SHADE-OrdRW	0.000000	4.06E+00	1.15E+01	3.47E-01	2.89E+01	5.75E-01	2.80E+00	1.00E+02	2.92E+02	4.28E+02
PBO	4.49E+00	3.95E+00	1.45E+01	2.29E-01	4.41E+01	6.77E-01	2.40E+00	1.29E+01	1.00E+02	3.98E+02
HP3PGA-3PGA	0.000000	3.19E+00	2.84E+01	1.73E-01	3.00E+01	1.79E-01	1.89E+00	1.21E+01	9.90E+01	3.99E+02

As shown in table 5.2, the performance of different algorithms was evaluated under different transformations. In the basic transformation, HP3PGA-3PGA gave perfect results with zero error for all functions. Some other algorithms, like MadDE and PBO, also did well with mostly zero errors. But DEDMNA and RB_IPOP_CMAES_PPMF had problems with some functions. In the bias transformation, where a constant value is added to the function, HP3PGA-3PGA was not affected and still gave zero error. But algorithms like DEDMNA and J21 had higher error values, which means they found it harder to handle biased functions.

In the shift transformation, where the function is moved to another position, HP3PGA-3PGA still worked well overall but showed some errors in functions like F10. RB_IPOP_CMAES_PPMF had large errors in several functions. In the rotation transformation, which makes the function more complicated, HP3PGA-3PGA had some trouble with F2 and F9. RB_IPOP_CMAES_PPMF did even worse, especially on F2. When bias and shift transformations were used together, HP3PGA-3PGA still gave very good results with almost zero error. But SOMA-CLP showed very high error values and struggled more. In the bias and rotation transformation, HP3PGA-3PGA

again gave zero error, showing that it is strong in handling such changes. However, DEDMNA and NL-SHADE-RSP were affected badly and gave high errors.

HP3PGA-3PGA is one of the best algorithms overall. It gives excellent results in many transformations and often gets zero error, especially in bias transformations. It struggles slightly in specific cases, such as function F10 in the shift transformation, and functions F2 and F9 in the rotation transformation. But it still outperforms most other algorithms.

5.3.2. Performance Analysis

The HP3PGA-3PGA algorithm consistently performed well, especially in situations where other algorithms faced difficulties, such as when bias, shift, and rotation were combined. Table 5.3 summarizes how the HP3PGA-3PGA performed on the CEC2021 benchmarks. As shown in Table 5.3, the algorithm HP3PGA-3PGA has higher performance on 49 benchmark functions of CEC2021 test bench suite. Thus, it is ranked on number 1. APGSK_IMODE on the other hand performed best on 48 benchmark functions of CEC2021 and ranked on number 2. Similarly, MadDE performed well on 47 benchmark functions which made it ranked on number 3 and L-SHADE-OrdRW and PBO performed best on total 41 benchmark functions and ranked on number 4. The overall results of the simulations demonstrate the outstanding performance of HP3PGA-3PGA.

Table 5.3 Comparison of HP3PGA-3PGA performance on CEC 2021 benchmarks

Benchmark Algorithm	Best (Unmatched)	Best (Matched)	Overall Best	Ranking in Total
HP3PGA-3PGA	1	48	49	1
APGSK_IMODE	2	46	48	2
MadDE	2	45	47	3
L-SHADE-OrdRW	2	39	41	4
PBO	3	38	41	4
RB_IPOP_CMAES_PPMF	8	23	31	5

Benchmark Algorithm	Best (Unmatched)	Best (Matched)	Overall Best	Ranking in Total
NL-SHADE-RSP	4	17	21	6
MLS-LSHADE	0	20	20	7
DEDMNA	3	15	18	8
J21	1	14	15	9
SOMA-CLP	0	7	7	10

The ranking is based on four key performance indicators:

1. Best (Unmatched) – Number of times an algorithm was the best but not matched by others.
2. Best (Matched) – Number of times an algorithm was the best but shared the spot with others.
3. Overall Best – Total number of times the algorithm performed at the highest level (sum of unmatched and matched).
4. Ranking in Total – Final ranking based on overall performance.

The HP3PGA-3PGA algorithm is the best performer in the CEC 2021 benchmark tests. It ranks first and outperformed all other optimization algorithms. It has 49 best scores, with 48 of them being matched (where other algorithms also performed well) and 1 unmatched (where it was the only one to perform best). This shows that HP3PGA-3PGA is highly reliable. Among other algorithms, APGSK_IMODE and MadDE performed well. APGSK_IMODE ranked second and MadDE ranked third. APGSK_IMODE has 48 best scores, with 46 of them being matched (where other algorithms also performed well) and 2 unmatched (where it was the only one to perform best). MadDE has 47 best scores, with 45 of them being matched (where other algorithms also performed well) and 2 unmatched (where it was the only one to perform best). Both algorithms performed well, but they are not as consistent and reliable as HP3PGA-3PGA.

L-SHADE-OrdRW, PBO, and RB_IPOP_CMAES_PPMF are average-performing

algorithms. L-SHADE-OrdRW and PBO ranked fourth with 41 top score whereas RB_IPOP_CMAES_PPMF ranked fifth with 31 top score. These algorithms perform well but are not as strong as the top three. They use traditional optimization methods, which may limit their flexibility in solving complex problems. Algorithms like NL-SHADE-RSP and MLS-LSHADE learn slowly and did not perform well. SOMA-CLP got very less score, showing it has trouble with complex problems.

The results show that HP3PGA-3PGA is the best algorithm for solving complex optimization problems. It performs well in finding new solutions and improving old ones. APGSK_IMODE and MadDE are strong competitors, but they are less stable and reliable. Traditional optimization algorithms like L-SHADE and RB_IPOP_CMAES_PPMF are still competitive but not as adaptable as newer methods. The lower-ranked algorithms struggle with difficult problems. Overall, HP3PGA-3PGA is the most powerful and reliable algorithm in this comparison.

5.4. HP3PGA-3PGA based Neural Architecture Search (NAS) Approach

This section proposed a HP3PGA-3PGA based NAS approach for leukemia detection from blood smear images. We implemented the proposed HP3PGA-3PGA approach to create a near-optimal CNN architecture for leukemia detection. For the implementation purpose, the proposed approach optimized the different hyperparameters of CNN. For the testing purpose, we used CNMC_2019 blood smear images dataset. The proposed approach begins with the initialization of different populations of CNN architectures. Each population contains different candidate solutions. Every candidate solution is a CNN architecture. In one candidate solution (CNN architecture), there could be different genes. In our proposed neural architecture search approach, the genes are referred as hyperparameters. The hyperparameters [146] and their boundary values are shown in Table 5.4. For the optimization purpose, we used same steps as mentioned in Algorithm 5.1. The proposed approach optimizes the hyperparameters of CNN on different numbers of convolutional layers. First, it evolves the architecture of CNN on 1 convolutional layer. If it achieves the acceptable accuracy, then provide the optimal architecture otherwise increment in the convolutional layer by 1 and evolve the neural

architecture again. This process continues until the acceptable results are not achieved. In our proposed approach, the best candidate solution means the best accuracy of the CNN architecture. For the leukemia detection purpose, the termination criteria to stop the training process of proposed approach is as below:

The model training process would be stopped if any of the following condition is satisfied:

- Adding a new hidden (Con2d) layer did not improve the test accuracy.

or

- If greater than 98% test accuracy is achieved.

Table 5.4 Hyper-Parameters considered for CNN optimization

Hyper-Parameter	**Value Range**
Number of Convolutional Layers	Between 1 and 10
Number of Filters	Between 1 and 64
Size of Filters	Between 1 and 10
Neurons in Fully Connected Layers	From 32 to 1024
Batch Size	From 8 to 512 (in powers of 2)
Number of Epochs	Between 1 and 20
Optimizers Available	SGD, Adadelta, Adam, Adagrad, RMSprop, Ftrl, Nadam, Adamax

5.4.1. Implementation and Performance of the Proposed Leukemia Detection Approach

To check how well the proposed leukemia detection method works, we implemented it in Python, along with other existing methods. The proposed hybrid bio-inspired leukemia detection algorithm is evaluated on CNMC_2019 blood smear images dataset. The model performance was tested with 100 iterations. The proposed method obtained an accuracy of 98.99% in generation 82. Table 5.5 gives a view of best accuracies

obtained by CNN on CNMC_2019 dataset.

Table 5.5 Accuracy levels for HP3PGA-3PGA based CNN on the CNMC_2019 dataset

Number of Convolutional 2D Layers	Generation with Best Accuracy	Best Accuracy (%)
1	65	81.11%
2	75	89.21%
3	82	98.99%

From Table 5.6, we observed that the proposed HP3PGA approach achieved 98.99% accuracy whereas VGG19-PCA-BBBC and VGG16-PCA-PB3C achieved 96.24% and 95.38% accuracy respectively. VGG19-PCA and VGG16-PCA models achieved 93.02% and 91.04% accuracy. Further, VGG19 and VGG16 models achieved 90.65% and 89.05% accuracy. From these results, we can see that the proposed approach is ranked number 1 compared to the other 17 methods. VGG19-PCA-BBBC and VGG16-PCA-PB3C scored the second and third positions. VGG19-PCA and VGG16-PCA ranked on fourth and fifth positions respectively.

Table 5.6 Comparison of proposed HP3PGA-3PGA approach with existing machine learning and transfer learning approaches

Model	Accuracy	Precision	Recall	F1 Score
SVM	74.80%	69.91%	76.78%	73.19%
Random Forest	81.20%	79.67%	81.66%	80.65%
Logistic Regression	83.20%	82.11%	83.47%	82.78%
K Nearest Neighbour	62%	71.54%	59.45%	64.94%
Decision Tree	70%	69.10%	69.67%	69.38%
Bagging	67.60%	69.91%	66.15%	67.98%
ML based Classification, IEEE	91%	-	-	-

Model	Accuracy	Precision	Recall	F1 Score
ISBI 2019 Challenge [50]				
Inception V3	86.01%	94.68%	85.58%	89.90%
CNN+ECA+VGG16 [96]	91.10%	-	-	-
TL+(GA,PCA)+MLP [102]	90.71%	-	95.26%	-
ResNet101 ensembled models [118]	85.11%	-	-	88.94%
VGG16	89.05%	96.77%	88.24%	92.31%
VGG19	90.65%	96.63%	89.58%	92.97%
VGG16-PCA	91.04%	96.77%	90.91%	93.75%
VGG19-PCA	93.02%	95.51%	94.44%	94.97%
VGG16-PCA-PB3C	95.38%	97.75%	95.60%	96.67%
VGG19-PCA-BBBC	96.24%	97.83%	96.77%	97.30%
Proposed Approach	98.99%	99.20%	98.40%	98%

From Table 5.6, we observed that the proposed HP3PGA-3PGA approach outperforms all other approaches in the different performance evaluation metrics namely, accuracy, precision, recall, and F1 score. With an accuracy of 98.99%, precision of 99.20%, recall of 98.40%, and an F1 score of 98%, the proposed approach clearly overcomes the performance of traditional machine learning models like Logistic Regression (83.20% accuracy), Random Forest (81.20% accuracy) and SVM (74.80% accuracy), which show much lower performance. Even compared to transfer learning models, the proposed approach was giving better performance. We considered different transfer learning-based hybrid models like VGG19-PCA-BBBC (96.24% accuracy), VGG16-PCA-PB3C (95.38% accuracy), VGG19-PCA (93.02 % accuracy), CNN-ECA-VGG16 (91.1% accuracy), VGG16-PCA (91.04% accuracy) , TL-PCA-MLP (90.71%

accuracy) and VGG19 (90.65% accuracy). While these models perform well, they are not as accurate or reliable as compared to the proposed approach. Overall, the proposed approach excels in feature extraction, accuracy, and handling complex problems, making it the most powerful and reliable solution among all the models tested.

5.5. Summary

This chapter proposed a new hybrid P3PGA and 3PGA bio-inspired search & optimization algorithm. The performance for proposed algorithm was evaluated against 80 standard benchmark functions of CEC2021 and the results were compared to 10 existing benchmarking algorithms. In addition, the proposed hybrid bio-inspired algorithm has been used to develop the near-optimal architecture for CNN. The results of NAS approach revealed that proposed algorithm is significantly outperforming other 17 leukemia detection algorithms. The proposed approach improves various parameters such as Accuracy, Recall, Precision and F1-score for the task of leukemia detection. The highest accuracy in this research work is observed as 98.99%, demonstrating our model's strong performance and outperforming existing artificial intelligence models in the field. In summary, the hybrid P3PGA and 3PGA method provides robust and promising outcome for various prevailing optimization issues, with the characteristics and capabilities to expand the field of evolutionary computation.

Chapter 6 Conclusion and Future Scope

6.1. Conclusion

In this thesis, one hybrid soft computing-based search and optimization algorithm and three soft computing-based leukemia detection approaches are proposed. The proposed soft computing-based hybrid algorithm is named as HP3PGA-3PGA. HP3PGA-3PGA is the hybrid version of 3PGA and P3PGA algorithm. The proposed algorithm was tested on 80 benchmark functions of CEC2021 test suite. The HP3PGA-3PGA was contrasted with 10 existing benchmarking algorithms.

This thesis also proposed three soft computing-based leukemia detection approaches. The proposed three approaches are named as VGG16-PCA-PB3C, VGG19-PCA-BBBC, HP3PGA-3PGA neural architecture search approach. All the three approaches are tested on CNMC_2019 dataset.

Chapter 1 presents the motivation behind our research work. It introduces leukemia, traditional leukemia detection methods and challenges in leukemia detection. This chapter also discusses machine learning and its different techniques. The chapter presents problem formulation and objectives of the thesis.

Chapter 2 presents an overview of the related work in the field. The review of the literature is divided into two main parts. Part 1 contains various traditional methods for leukemia detection. This study was carried out to recognize the constraints of the conventional methods for detecting leukemia. Traditional methods are painful, expensive, slow, and rely heavily on a doctor's skill. Part 2 of the survey contains various AI-based methods for leukemia detection. Computer vision approaches are useful but struggle due to small and non-diverse datasets. This shows the need for new approaches that can automatically create better models for leukemia detection.

Chapter 3 presents a new soft computing-based hybrid approach VGG16-PCA-PB3C for leukemia detection. The proposed approach integrates VGG16, PCA, PB3C. The features from the blood cell images are extracted by VGG16. PCA is used to make the

extracted features simpler by reducing their number. From the extracted features, the near optimal features are selected by the multi population based PB3C algorithm. The deep neural network is trained for leukemia detection using the selected features. Python is used to implement the proposed approach. The proposed approach is compared with 13 other leukemia detection algorithms, and it is found that the proposed approach with 95.38% accuracy performed better than all of them.

Chapter 4 presents a novel soft computing-based approach VGG19-PCA-BBBC for leukemia detection. In this proposed approach, VGG19, PCA, and the Big Bang Big Crunch (BBBC) algorithms are integrated. VGG19 was used to extract deep features, PCA was used to reduce the size of feature maps, and BBBC, a single-population optimization algorithm, was used to select the optimal features. The chosen features were used to train a deep neural network for detecting leukemia. The method was implemented in Python and tested on the CNMC_2019 leukemia dataset of blood cell images. The proposed approach was compared with 15 other leukemia detection methods, and it was found that the proposed approach with 96.24% accuracy performed better than other existing approaches.

Chapter 5 proposes a new soft computing-based hybrid algorithm called Hybrid Parallel Three Parent Genetic Algorithm – Three Parent Genetic Algorithm (HP3PGA-3PGA). This algorithm combines a multi-population parallel algorithm P3PGA with a single-population algorithm 3PGA. The HP3PGA-3PGA algorithm was tested on the CEC2021 test suite, which includes 80 test functions, and compared with 10 other benchmarking algorithms. The algorithm is also used to automatically design a near-optimal architecture of a CNN for leukemia detection. The results showed that HP3PGA-3PGA (98.99% accuracy) outperformed all the other 17 leukemia detection approaches.

Chapter 6 concludes the study and outlines the future possibilities for further research.

6.2. Future Scope

This thesis has presented one bio inspired computing algorithm and three soft

computing-based leukemia detection approaches. The proposed leukemia detection approaches are validated on CNMC_2019 dataset of blood cell images. The flexibility and adaptability of the models developed herein can be easily applied toward other forms of cancer, or even to other types of diseases where early detection is so important. It is expected that all three proposed leukemia detection approaches could be used to solve different kinds of machine learning problems. In future, the proposed approaches could be applied to deal with highly complex and non-linear problems such as plant disease detection, plant classification, landmark recognition, and natural language processing. The proposed HP3PGA-3PGA algorithm can also be used for routing in wireless mess networks, routing in integrated circuits, and power efficiency in wireless sensor networks. In future, the proposed soft computing-based search and optimization algorithm can be converted into multiobjective based algorithm. The multi objective algorithm can evolve the near optimal architecture with less complexity and high accuracy. These multi objective algorithm-based models would be light weighted and suitable for low configuration mobile devices.

References

1. Dong, Ying, et al. "Leukemia incidence trends at the global, regional, and national level between 1990 and 2017." *Experimental hematology & oncology* 9 (2020): 1-11.
2. Das, Pradeep Kumar, et al. "A systematic review on recent advancements in deep and machine learning based detection and classification of acute lymphoblastic leukemia." *IEEE access* 10 (2022): 81741-81763.
3. Kaur, Navpreet, and Amar Singh. "VGG16-PCA-PB3C: A hybrid PB3C and deep neural network-based approach for leukemia detection." *International Journal of Information Technology* (2024): 1-11.
4. Cruz, Dela, et al. "Determination of blood components (WBCs, RBCs, and Platelets) count in microscopic images using image processing and analysis." *2017IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*. IEEE, 2017.
5. Koury, Mark J., and Lionel Blanc. "Red blood cell production and kinetics." *Rossi's Principles of Transfusion Medicine* (2022): 131-142.
6. Bain, Barbara J. *Blood cells: a practical guide*. John Wiley & Sons, 2021.
7. https://www.kaggle.com/datasets/sizlingdhairya1/all-idb-images
8. M. Iswarya, S. A, A. M. R and K. A, "Detection of Leukemia using Machine Learning," *2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, 2022, pp. 466-470, doi: 10.1109/ICAAIC53929.2022.9792725.
9. M. A. Hossain, A. K. M. M. Islam, S. Islam, S. Shatabda and A. Ahmed, "Symptom Based Explainable Artificial Intelligence Model for Leukemia Detection," in *IEEE Access*, vol. 10, pp. 57283-57298, 2022, Doi: 10.1109/ACCESS.2022.3176274.
10. Shah, Afshan, et al. "Automated diagnosis of leukemia: a comprehensive review." *IEEE Access* 9 (2021): 132097-132124.

11. Zhong, Pengqiang, et al. "Diagnosis of acute leukemia by multiparameter flow cytometry with the assistance of artificial intelligence." *Diagnostics* 12.4 (2022): 827.
12. Ananth, Christo, et al. "Blood Cancer Detection with Microscopic Images Using Machine Learning." *Machine Learning in Information and Communication Technology: Proceedings of ICICT 2021, SMIT*. Singapore: Springer Nature Singapore, 2022. 45-54.
13. Sarker, Iqbal H. "Machine learning: Algorithms, real-world applications and research directions." *SN computer science* 2.3 (2021): 160.
14. Mahesh, Batta. "Machine learning algorithms-a review." *International Journal of Science and Research (IJSR).[Internet]* 9.1 (2020): 381-386.
15. Jiang, Tammy, Jaimie L. Gradus, and Anthony J. Rosellini. "Supervised machine learning: a brief primer." *Behavior therapy* 51.5 (2020): 675-687.
16. Krishnan, Rayan, Pranav Rajpurkar, and Eric J. Topol. "Self-supervised learning in medicine and healthcare." *Nature Biomedical Engineering* 6.12 (2022): 1346-1352.
17. Alanazi, Abdullah. "Using machine learning for healthcare challenges and opportunities." *Informatics in Medicine Unlocked* 30 (2022): 100924.
18. Morales, Eduardo F., and Hugo Jair Escalante. "A brief introduction to supervised, unsupervised, and reinforcement learning." *Biosignal processing and classification using computational learning and intelligence*. Academic Press, 2022. 111-129.
19. Dhanaraj, Rajesh Kumar, K. Rajkumar, and U. Hariharan. "Enterprise IoT modeling: supervised, unsupervised, and reinforcement learning." *Business Intelligence for Enterprise Internet of Things* (2020): 55-79.
20. O'Shea, K. "An introduction to convolutional neural networks." *arXiv preprint arXiv:1511.08458* (2015).
21. Li, Zewen, et al. "A survey of convolutional neural networks: analysis, applications, and prospects." *IEEE transactions on neural networks and learning systems* 33.12 (2021): 6999-7019.

22. Pešić, Ivana. "Segmentation and Classification of Leucocyte Images for Detection of Acute Lymphoblastic Leukemia." *2020 7th ETRAN&IcETRAN international conference [Internet]. Belgrade: IcETRAN*. 2020.
23. Ketkar, Nikhil, et al. "Convolutional neural networks." *Deep learning with Python: learn best practices of deep learning models with PyTorch* (2021): 197-242.
24. Pinaya, Walter Hugo Lopez, et al. "Convolutional neural networks." *Machine learning*. Academic Press, 2020. 173-191.
25. Teuwen, Jonas, and Nikita Moriakov. "Convolutional neural networks." *Handbook of medical image computing and computer assisted intervention*. Academic Press, 2020. 481-501.
26. Kamath, Uday, et al. "Convolutional neural networks." *Deep learning for NLP and speech recognition* (2019): 263-314.
27. Qu, Yili, Yaobin Ke, and Wei Yu. "A solution for input limit in CNN due to fully-connected layer." *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*. IEEE, 2018.
28. Das, Sumit & Sanyal, Manas. (2019). Application of AI and Soft Computing in Healthcare: A review and Speculation. International Journal of Scientific & Technology Research. VOLUME 8. 1786-1806.
29. Mishra, Ashish. (2020). Soft Computing Applications and Techniques in Healthcare. 10.1201/9781003003496.
30. Zhang, Wengang, et al. "State-of-the-art review of soft computing applications in underground excavations." *Geoscience Frontiers* 11.4 (2020): 1095-1106.
31. Shiruru, Kuldeep. (2016). AN INTRODUCTION TO ARTIFICIAL NEURAL NETWORK. International Journal of Advance Research and Innovative Ideas in Education. 1. 27-30.
32. Guillod, Thomas, Panteleimon Papamanolis, and Johann W. Kolar. "Artificial neural network (ANN) based fast and accurate inductor modeling and design." *IEEE Open Journal of Power Electronics* 1 (2020): 284-299.
33. Yen, J., and Langari, R., (1999), Fuzzy logic: intelligence, control, and Information. Prentice-Hall.

34. Zadeh, L. A., (1965), "Fuzzy sets," Information and control, vol. 8(3), pp. 338–353.
35. Gautam, Ritu, Prableen Kaur, and Manik Sharma. "A comprehensive review on nature inspired computing algorithms for the diagnosis of chronic disorders in human beings." *Progress in Artificial Intelligence* 8.4 (2019): 401-424.
36. Katoch, Sourabh, Sumit Singh Chauhan, and Vijay Kumar. "A review on genetic algorithm: past, present, and future." *Multimedia tools and applications* 80 (2021): 8091-8126.
37. Pham, Xuan-Lam, and Thi Thu Le. "Bibliometric Analysis and Systematic Review of Research on Expert Finding: A PRISMA-guided Approach." *The International Arab Journal of Information Technology* 21.4 (2024).
38. Greener, Sue. "Evaluating literature with bibliometrics." *Interactive Learning Environments* 30.7 (2022): 1168-1169.
39. Shams, Usman Ali, et al. "Bio-net dataset: AI-based diagnostic solutions using peripheral blood smear images." *Blood Cells, Molecules, and Diseases* 105 (2024): 102823.
40. Percival, Mary-Elizabeth, et al. "Bone marrow evaluation for diagnosis and monitoring of acute myeloid leukemia." *Blood reviews* 31.4 (2017): 185-192.
41. Paessler, Michele E., Marybeth Helfrich, and Gerald BW Wertheim. "Cytochemical staining." *Acute Myeloid Leukemia: Methods and Protocols* (2017): 19-32.
42. Grimwade, Lizz F., Kathryn A. Fuller, and Wendy N. Erber. "Applications of imaging flow cytometry in the diagnostic assessment of acute leukaemia." *Methods* 112 (2017): 39-45.
43. Baliakas, Panagiotis, et al. "Cytogenetic complexity in chronic lymphocytic leukemia: definitions, associations, and clinical impact." *Blood, The Journal of the American Society of Hematology* 133.11 (2019): 1205-1216.
44. Sokolenko, Anna P., and Evgeny N. Imyanitov. "Molecular diagnostics in clinical oncology." *Frontiers in molecular biosciences* 5 (2018): 76.

45. Rajaraman, Ajay & Kiran L, Shashi. (2021). An Automated Detection of Leukemia. International Journal for Research in Applied Science and Engineering Technology. 9. 10.22214/ijraset.2021.37670.
46. Kaur, Navpreet, and Amar Singh. "Leukemia detection: An overview of different artificial intelligence-based approaches." *Recent advances in computing sciences* (2023): 45-50.
47. Aby, Aswathy Elma, et al. "A review on leukemia detection and classification using Artificial Intelligence-based techniques." *Computers and Electrical Engineering* 118 (2024): 109446.
48. Al Mamun, Abdullah, et al. "Acute lymphoblastic leukemia detection approach from peripheral blood smear using color threshold and morphological techniques." *International Journal of Electrical and Computer Engineering* 12.4 (2022): 3692.
49. Liu, Ke, and Jie Hu. "Classification of acute myeloid leukemia M1 and M2 subtypes using machine learning." *Computers in Biology and Medicine* 147 (2022): 105741.
50. Gupta, Ritu, Shiv Gehlot, and Anubha Gupta. "C-NMC: B-lineage acute lymphoblastic leukaemia: A blood cancer dataset." *Medical Engineering & Physics* 103 (2022): 103793.
51. Das, Biplab Kanti, and Himadri Sekhar Dutta. "GFNB: Gini index–based Fuzzy Naive Bayes and blast cell segmentation for leukemia detection using multi-cell blood smear images." *Medical & Biological Engineering & Computing* 58 (2020): 2789-2803.
52. Saleem, Saba, et al. "Leukemia segmentation and classification: A comprehensive survey." *Computers in Biology and Medicine* (2022): 106028.
53. Maurya, Sonam, et al. "A review on recent developments in cancer detection using Machine Learning and Deep Learning models." *Biomedical Signal Processing and Control* 80 (2023): 104398.
54. Bharat, Nikhil, Avinash Kumar, and P. S. C. Bose. "A study on soft computing optimizing techniques." *Materials Today: Proceedings* 50 (2022): 1193-1198.

55. Bouchet, Agustina, et al. "Intuitionistic fuzzy set and fuzzy mathematical morphology applied to color leukocytes segmentation." *Signal, Image and Video Processing* 14 (2020): 557-564.
56. Acharya, Vasundhara, et al. "Peripheral blood smear analysis using automated computer-aided diagnosis system to identify acute myeloid leukemia." *IEEE Transactions on Engineering Management* 70.8 (2021): 2760-2773.
57. Chen, Pingjun, et al. "Chronic lymphocytic leukemia progression diagnosis with intrinsic cellular patterns via unsupervised clustering." *Cancers* 14.10 (2022): 2398.
58. Shukla, Alok Kumar, Pradeep Singh, and Manu Vardhan. "Gene selection for cancer types classification using novel hybrid metaheuristics approach." *Swarm and Evolutionary Computation* 54 (2020): 100661.
59. Sharma, Anuj, et al. "Automatic leukaemia segmentation approach for blood cancer classification using microscopic images." *Computers, Materials & Continua* 73.2 (2022): 3629-3648.
60. Abhishek, Arjun, et al. "Automated classification of acute leukemia on a heterogeneous dataset using machine learning and deep learning techniques." *Biomedical Signal Processing and Control* 72 (2022): 103341.
61. Anagha, V., et al. "Detection of leukemia using convolutional neural network." *Emerging Research in Computing, Information, Communication and Applications: ERCICA 2020, Volume 1*. Springer Singapore, 2022.
62. Anilkumar, K. K., V. J. Manoj, and T. M. Sagi. "Automated detection of b cell and t cell acute lymphoblastic leukaemia using deep learning." *Irbm* 43.5 (2022): 405-413.
63. Abas, Shakir Mahmood, Adnan Mohsin Abdulazeez, and Diyar Qader Zeebaree. "A YOLO and convolutional neural network for the detection and classification of leukocytes in leukemia." *Indonesian Journal of Electrical Engineering and Computer Science* 25.1 (2022): 200-213.
64. Vieira, Guilherme, and Marcos Eduardo Valle. "Acute lymphoblastic leukemia detection using hypercomplex-valued convolutional neural networks." *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022.

65. Rastogi, Priyanka, Kavita Khanna, and Vijendra Singh. "LeuFeatx: Deep learning–based feature extractor for the diagnosis of acute leukemia from microscopic images of peripheral blood smear." *Computers in Biology and Medicine* 142 (2022): 105236.
66. Mallick, Pradeep Kumar, et al. "Convergent learning–based model for leukemia classification from gene expression." *Personal and Ubiquitous Computing* 27.3 (2023): 1103-1110.
67. Ansari, Sanam, et al. "A customized efficient deep learning model for the diagnosis of acute leukemia cells based on lymphocyte and monocyte images." *Electronics* 12.2 (2023): 322.
68. Ananthu, K. S., et al. "Acute lymphoblastic leukemia detection using transfer learning techniques." *Intelligent Sustainable Systems: Proceedings of ICISS 2021*. Springer Singapore, 2022.
69. Elhassan, Tusneem A., et al. "Classification of atypical white blood cells in acute myeloid leukemia using a two-stage hybrid model based on deep convolutional autoencoder and deep convolutional neural network." *Diagnostics* 13.2 (2023): 196.
70. Sridhar, K., et al. "Enhanced Machine learning algorithms Lightweight Ensemble Classification of Normal versus Leukemic Cel." *Journal of Pharmaceutical Negative Results* (2022): 496-505.
71. Chand, Sunita, and Virendra P. Vishwakarma. "A novel deep learning framework (DLF) for classification of acute lymphoblastic leukemia." *Multimedia Tools and Applications* 81.26 (2022): 37243-37262.
72. Muhamad, Hersh A., Shahab Wahhab Kareem, and Amin Salih Mohammed. "A deep learning method for detecting leukemia in real images." *NeuroQuantology* 20.7 (2022): 2358.
73. Mohsin, Saba, et al. "Leukemia Cells Classification using Deep Learning Approaches." *2023 25th International Multitopic Conference (INMIC)*. IEEE, 2023.

74. Dibouliya, Ashish, and Varsha Jotwani. "Detection and Estimation of Leukemia Using Artificial Intelligence and Fuzzy Logic." *2023 2nd International Conference on Ambient Intelligence in Health Care (ICAIHC)*. IEEE, 2023.

75. Boreiri, Zahra, Alireza Norouzi Azad, and Amin Ghodousian. "A Convolutional Neuro-Fuzzy Network Using Fuzzy Image Segmentation for Acute Leukemia Classification." *2022 27th International Computer Conference, Computer Society of Iran (CSICC)*. IEEE, 2022.

76. Aftab, Muhammad Omer, et al. "Executing spark BigDL for leukemia detection from microscopic images using transfer learning." *2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA)*. IEEE, 2021.

77. Akalin, Fatma, and Nejat Yumuşak. "Early Detection of ALL Disease Using YOLOv4 Algorithm on Peripheral Blood Smear Images." *2022 Innovations in Intelligent Systems and Applications Conference (ASYU)*. IEEE, 2022.

78. Chand, Sunita, and Virendra P. Vishwakarma. "A novel deep learning framework (DLF) for classification of acute lymphoblastic leukemia." *Multimedia Tools and Applications* 81.26 (2022): 37243-37262.

79. Mondal, Chayan, et al. "Acute lymphoblastic leukemia detection from microscopic images using weighted ensemble of convolutional neural networks." arXiv preprint arXiv:2105.03995 (2021).

80. Ghaderzadeh, Mustafa, et al. "A fast and efficient CNN model for B-ALL diagnosis and its subtypes classification using peripheral blood smear images." *International Journal of Intelligent Systems* 37.8 (2022): 5113-5133.

81. Vogado, Luis, et al. "Diagnosis of leukaemia in blood slides based on a fine-tuned and highly generalisable deep learning model." *Sensors* 21.9 (2021): 2989.

82. Chadaga, Krishnaraj, et al. "A machine learning and explainable artificial intelligence approach for predicting the efficacy of hematopoietic stem cell transplant in pediatric patients." *Healthcare Analytics* 3 (2023): 100170.

83. Vishwaraj, Naik Parrikar, et al. "Machine learning assisted cancer cell detection using strip waveguide Bragg gratings." *Optik* 284 (2023): 170947.

84. Wu, Hao, et al. "A soft-computing based approach to overlapped cells analysis in histopathology images with genetic algorithm." *Applied Soft Computing* 126 (2022): 109279.
85. Priyadharshini, N., et al. "A novel hybrid Extreme Learning Machine and Teaching–Learning-Based Optimization algorithm for skin cancer detection." *Healthcare Analytics* 3 (2023): 100161.
86. Nematzadeh, Sajjad, et al. "Tuning hyperparameters of machine learning algorithms and deep neural networks using metaheuristics: A bioinformatics study on biomedical and biological cases." *Computational biology and chemistry* 97 (2022): 107619.
87. Baby, Diana, Sujitha Juliet, and M. M. Anishin Raj. "An efficient lymphocytic leukemia detection based on EfficientNets and ensemble voting classifier." *International Journal of Imaging Systems and Technology* 33.1 (2023): 419-426.
88. Devi, Tulasi Gayatri, et al. "Segmentation and classification of white blood cancer cells from bone marrow microscopic images using duplet-convolutional neural network design." *Multimedia Tools and Applications* 82.23 (2023): 35277-35299.
89. Rejula, M. Anline, S. Amutha, and G. M. Shilpa. "Classification of acute lymphoblastic leukemia using improved ANFIS." *Multimedia Tools and Applications* 82.23 (2023): 35475-35491.
90. Fauzi, I. R., Z. Rustam, and A. Wibowo. "Multiclass classification of leukemia cancer data using Fuzzy Support Vector Machine (FSVM) with feature selection using Principal Component Analysis (PCA)." *Journal of Physics: Conference Series*. Vol. 1725. No. 1. IOP Publishing, 2021.
91. Ramya, V. Jeya, and S. Lakshmi. "Acute myelogenous leukemia detection using optimal neural network based on fractional black-widow model." *Signal, image and video processing* 16.1 (2022): 229-238.
92. Bai, G. Mercy, and P. Venkadesh. "Optimized Deep Neuro-Fuzzy Network with MapReduce Architecture for Acute Lymphoblastic Leukemia Classification

and Severity Analysis." *International Journal of Image and Graphics* 24.03 (2024): 2450028.

93. Narayanan, K., et al. "A Hybrid Machine Learning Technique for Acute Lymphoblastic Leukemia Classification." (2023).
94. Dhal, Krishna Gopal, et al. "Illumination-free clustering using improved slime mould algorithm for acute lymphoblastic leukemia image segmentation." *Journal of Bionic Engineering* 20.6 (2023): 2916-2934.
95. Jha, Krishna Kumar, and Himadri Sekhar Dutta. "Nucleus and cytoplasm–based segmentation and actor-critic neural network for acute lymphocytic leukaemia detection in single cell blood smear images." *Medical & Biological Engineering & Computing* 58 (2020): 171-186.
96. Zakir Ullah, Muhammad, et al. "An attention-based convolutional neural network for acute lymphoblastic leukemia classification." *Applied Sciences* 11.22 (2021): 10662.
97. Mounika, Bakkanarappa Gari, et al. "A robust hybrid deep learning model for acute lymphoblastic leukemia diagnosis." *Advances in Networks, Intelligence and Computing*. CRC Press 679-688.
98. Khatter, Harsh, et al. "Analysis of the S-ANFIS algorithm for the detection of blood infections using hybrid computing." *Electronics* 11.22 (2022): 3733.
99. Rodrigues, Larissa Ferreira, et al. "Optimizing a deep residual neural network with genetic algorithm for acute lymphoblastic leukemia classification." *Journal of Digital Imaging* 35.3 (2022): 623-637.
100. Mahesh, T. R., et al. "Hybrid ant lion mutated ant colony optimizer technique with particle swarm optimization for leukemia prediction using microarray gene data." *IEEE Access* (2024).
101. Sallam, Nada M., et al. "An efficient strategy for blood diseases detection based on grey wolf optimization as feature selection and machine learning techniques." *Applied Sciences* 12.21 (2022): 10760.

102. Hosseinzadeh, M., et al. "A Diagnostic Model for Acute Lymphoblastic Leukemia Using Metaheuristics and Deep Learning Methods." *arXiv preprint arXiv:2406.18568* (2024).

103. Agustin, Rosi Indah, Agus Arif, and Usi Sukorini. "Classification of immature white blood cells in acute lymphoblastic leukemia L1 using neural networks particle swarm optimization." *Neural Computing and Applications* 33.17 (2021): 10869-10880.

104. Sallam, Nada M., et al. "An efficient EGWO algorithm as feature selection for B-ALL diagnoses and its subtypes classification using peripheral blood smear images." *Alexandria Engineering Journal* 68 (2023): 39-66.

105. Veeraiah, Neenavath, Youseef Alotaibi, and Ahmad F. Subahi. "MayGAN: Mayfly Optimization with Generative Adversarial Network-Based Deep Learning Method to Classify Leukemia Form Blood Smear Images." *Comput. Syst. Sci. Eng.* 46.2 (2023): 2039-2058.

106. Balasubramanian, Kishore, N. P. Ananthamoorthy, and K. Ramya. "An approach to classify white blood cells using convolutional neural network optimized by particle swarm optimization algorithm." Neural Computing and Applications 34.18 (2022): 16089-16101.

107. Ahmad, Riaz, et al. "White blood cells classification using entropy-controlled deep features optimization." Diagnostics 13.3 (2023): 352.

108. Kumar, Rajiv, Shivani Joshi, and Avinash Dwivedi. "CNN-SSPSO: a hybrid and optimized CNN approach for peripheral blood cell image recognition and classification." *International Journal of Pattern Recognition and Artificial Intelligence* 35.05 (2021): 2157004.

109. Alrefai, Nashat, and Othman Ibrahim. "Optimized feature selection method using particle swarm intelligence with ensemble learning for cancer classification based on microarray datasets." *Neural Computing and Applications* 34.16 (2022): 13513-13528.

110. Alabdulqader, Ebtisam Abdullah, et al. "Improving prediction of blood cancer using leukemia microarray gene data and Chi2 features with weighted convolutional neural network." *Scientific Reports* 14.1 (2024): 15625.\

111. Amin, Javaria, et al. "An integrated design based on dual thresholding and features optimization for white blood cells detection." *IEEE Access* 9 (2021): 151421-151433.

112. Ay, Şevket, Ekin Ekinci, and Zeynep Garip. "A comparative analysis of meta-heuristic optimization algorithms for feature selection on ML-based classification of heart-related diseases." *The Journal of Supercomputing* 79.11 (2023): 11797-11826.

113. Shahzad, Asim, et al. "Categorizing white blood cells by utilizing deep features of proposed 4B-AdditionNet-based CNN network with ant colony optimization." *Complex & Intelligent Systems* 8.4 (2022): 3143-3159.

114. Haznedar, Bulent, Mustafa Turan Arslan, and Adem Kalinli. "Optimizing ANFIS using simulated annealing algorithm for classification of microarray gene expression cancer data." Medical & Biological Engineering & Computing 59 (2021): 497-509.

115. Sampathkumar, A., et al. "An efficient hybrid methodology for detection of cancer-causing gene using CSC for micro array data." *Journal of Ambient Intelligence and Humanized Computing* 11 (2020): 4743-4751.

116. Li, Gengluo, and Giorgos Jimenez. "Optimal diagnosis of the skin cancer using a hybrid deep neural network and grasshopper optimization algorithm." *Open Medicine* 17.1 (2022): 508-517.

117. Meenakshi, A., et al. "Automatic classification of white blood cells using deep features based convolutional neural network." *Multimedia Tools and Applications* 81.21 (2022): 30121-30142.

118. Chen, Yao-Mei, et al. "Classifying microscopic images as acute lymphoblastic leukemia by Resnet ensemble model and Taguchi method." *BMC bioinformatics* 22.Suppl 5 (2021): 615.

119. Salama, Mohamed E., et al. "Artificial intelligence enhances diagnostic flow cytometry workflow in the detection of minimal residual disease of chronic lymphocytic leukemia." *Cancers* 14.10 (2022): 2537.

120. Taino, Daniela F., et al. "Analysis of cancer in histological images: employing an approach based on genetic algorithm." *Pattern Analysis and Applications* 24 (2021): 483-496

121. Alsuliman, Maraheb, and Heyam H. Al-Baity. "Efficient diagnosis of autism with optimized machine learning models: An experimental analysis on genetic and personal characteristic datasets." *Applied Sciences* 12.8 (2022): 3812.

122. Oyelade, Olaide N., Jeffrey O. Agushaka, and Absalom E. Ezugwu. "Evolutionary binary feature selection using adaptive ebola optimization search algorithm for high-dimensional datasets." *Plos one* 18.3 (2023): e0282812.

123. Aher, Chetan Nimba, and Ajay Kumar Jena. "Rider-chicken optimization dependent recurrent neural network for cancer detection and classification using gene expression data." *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization* 9.2 (2021): 174-191.

124. Venkatesh, B., and J. Anuradha. "A review of feature selection and its methods." *Cybernetics and information technologies* 19.1 (2019): 3-26.

125. Veni N, Manjula J (2023) VGG-16 architecture for MRI brain tumor image classifcation. In: Futuristic Communication and Network Technologies: Select Proceedings of VICFCNT 2021, vol 1. Springer, Singapore. pp 319–328

126. Devi, G. Meena, and V. Neelambary. "Computer-aided diagnosis of white blood cell leukemia using VGG16 convolution neural network." *2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA)*. IEEE, 2022.

127. Bharadiya JP (2023) A tutorial on principal component analysis for dimensionality reduction in machine learning. Int J Innov Sci Res Technol 8(5):2028–2032.

128. Makem, Mimosette, and Alain Tiedeu. "An efficient algorithm for detection of white blood cell nuclei using adaptive three stage PCA-based fusion." *Informatics in Medicine Unlocked* 20 (2020): 100416.

129. Singh, Amar, et al. "Face Recognition: A Combined Parallel BB-BC & PCA Approach to Feature Selection." *International Journal of Computer Science & Information Technology* 2.2 (2015): 1-5.
130. Sharma R, Singh A (2022) Big bang–big crunch-CNN: an optimized approach towards rice crop protection and disease detection. Arch Phytopathol Plant Prot 55(2):143–161
131. Kumar S, Singh A, Walia S (2018) Parallel big bang-big crunch global optimization algorithm: performance and its applications to routing in WMNs. Wirel Pers Commun 100:1601–1618
132. C-NMC (2019) The Cancer Imaging Archive, 1, https://doi.org/ 10.7937/tcia.2019.dc64i46r
133. Dey, Nilanjan, et al. "Customized VGG19 architecture for pneumonia detection in chest X-rays." *Pattern Recognition Letters* 143 (2021): 67-74.
134. Mascarenhas, Sheldon, and Mukul Agarwal. "A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification." *2021 International conference on disruptive technologies for multi-disciplinary research and applications (CENTCON)*. Vol. 1. IEEE, 2021.
135. Bagaskara, Abitya, and Muhammad Suryanegara. "Evaluation of VGG-16 and VGG-19 deep learning architecture for classifying dementia people." *2021 4th International Conference of Computer and Informatics Engineering (IC2IE)*. IEEE, 2021.
136. Sharma, Rahul, and Amar Singh. "Big bang–big crunch-CNN: an optimized approach towards rice crop protection and disease detection." *Archives of Phytopathology and Plant Protection* 55.2 (2022): 143-161.
137. Ghosh, Sukanta, Amar Singh, and Shakti Kumar. "BBBC-U-Net: Optimizing U-Net for automated plant phenotyping using big bang big crunch global optimization algorithm." *International Journal of Information Technology* 15.8 (2023): 4375-4387.
138. Singh, Hakam, and Yugal Kumar. "Hybrid big bang-big crunch algorithm for cluster analysis." *Futuristic Trends in Networks and Computing Technologies:*

Second International Conference, FTNCT 2019, Chandigarh, India, November 22–23, 2019, Revised Selected Papers 2. Springer Singapore, 2020.

139. Rezaee Jordehi, A. "A chaotic-based big bang–big crunch algorithm for solving global optimisation problems." *Neural Computing and Applications* 25 (2014): 1329-1335.
140. Mirjalili, Seyedali, and Seyedali Mirjalili. "Genetic algorithm." *Evolutionary algorithms and neural networks: theory and applications* (2019): 43-55.
141. Singh, Amar, et al. "Three-parent GA: A Global Optimization Algorithm." *Journal of Multiple-Valued Logic & Soft Computing* 32 (2019).
142. Singh, Amar, et al. "Parallel 3-parent genetic algorithm with application to routing in wireless mesh networks." *Implementations and applications of machine learning* (2020): 1-28.
143. Zhang, Xin, and Qinglian Zhang. "A parallel genetic algorithm with three-parent crossover for real parameter optimization." *Communications, Signal Processing, and Systems: Proceedings of the 2017 International Conference on Communications, Signal Processing, and Systems*. Springer Singapore, 2019.
144. Singh, Amar, Sukhbir Singh Walia, and Shakti Kumar. "P3PGA: Multi-Population 3 Parent Genetic Algorithm and its Application to Routing in WMNs." *International Journal of Advanced Research in Computer Science* 8.5 (2017).
145. Mańdziuk, Jacek, and Hussein Abbass. "Conference Report on 2021 IEEE Congress on Evolutionary Computation (IEEE CEC 2021)[Conference Reports]." *IEEE Computational Intelligence Magazine* 16.4 (2021): 5-8.
146. Aszemi, Nurshazlyn Mohd, and P. D. D. Dominic. "Hyperparameter optimization in convolutional neural network using genetic algorithms." *International Journal of Advanced Computer Science and Applications* 10.6 (2019).

LIST OF PUBLICATIONS

Research Papers and Conference Proceedings

[1] Kaur, Navpreet, and Amar Singh. "VGG16-PCA-PB3C: A hybrid PB3C and deep neural network-based approach for leukemia detection." *International Journal of Information Technology* (2024): 1-11.

Paper Communicated

[1] Kaur, Navpreet, and Amar Singh. " An Intelligent Framework for Leukemia Detection Using VGG19-PCA-Big Bang Big Crunch." *International Journal of Engineering Trends and Technology (Accepted)*.

LIST OF CONFERENCES

[1] Kaur, Navpreet, and Amar Singh. "Leukemia detection: An overview of different artificial intelligence-based approaches." *Recent advances in computing sciences* (2023)", School of Computer Application, Lovely Professional University.

[2] Kaur, Navpreet, and Amar Singh. "Leukemia detection: Classical vs Soft Computing based Leukemia diagnosis approaches" *1st International Conference on futuristic Computation Techniques: Approaches, Implementation and Applications (ICFCT-2022)*, Department of Computer Applications, Panipat Institute of Engineering & Technology.

[3] "A novel advanced Transfer Learning based Approach for Leukemia Detection" *International Conference on Cybernation & Computation (CYBERCOM 2024) (IEEE),* Graphic Era Hill University.