International Journal of Computational Intelligence in Control

CLASSIFICATION OF LEUKEMIA WITH OPTIMIZED DEEP LEARNING ARCHITECTURE

A. Premnath[1]and V. S. Meenakshi[2]

[1]Assistant Professor in Computer Science, Bishop Thorp College, Dharapuram, Tamil Nadu, India.

[2]Assistant Professor in Computer Science, Chikkanna Govt. Arts College, Tiruppur, Tamil Nadu, India.

ABSTRACT

Acute Leukemia is a life-threatening disease that affects both children and adults and can result in death if not treated. Acute Lymphocytic Leukemia (ALL) spreads quickly in children's bodies and kills them in a matter of weeks. The hematologists examine the blood and bone marrow to determine ALL. Manual blood testing techniques, which have been around for a long time, are often slow and produce less precise results. The intricate architecture of histopathological structures makes automatic diagnosis of leukemic B-lymphoblast cancer in microscopic images extremely difficult. To address this problem, an automatic and reliable diagnostic system for early identification and treatment is necessary. A novel Optimized Convolutional Neural Network (OCNN) is proposed in this paper, which adjusts the elemental type of the fundamentally based plan of the convolutional network in order to recognize and classify leukemia images into two groups (Yes and No). In the proposed CNN design, a Leaky Rectified Linear Unit (Leaky RELU) is consolidated, with the component of zero slopes negatively. The Kernel Percolate is then displayed, and it recognizes image pixels with a value of 1 along the image's diagonal.

KEYWORDS: Convolutional Neural Network, Leukemia, Rectified Linear Unit, Classification

1. INTRODUCTION

Blood is the most vital component of any human body since it keeps us alive. It has a variety of critical activities, including transferring oxygen, carbon dioxide, minerals, and other substances throughout the body in order to maintain metabolism. RBC, WBC, and Platelets are the three primary components of blood [1].

If not treated promptly, a lack of blood can have a significant impact on metabolism, which can be quite harmful. Leukemia is one of the most frequent blood illnesses. The most frequent type of cancer in youngsters is leukemia. All malignancies start in the cells of the body, and leukemia is a cancer that starts in the cells of the blood. Cells, in general, expand and proliferate to produce new cells when the body requires them. Cells die as they age and are replaced by new cells. This cycle does not always work properly. New cells are generated in cancer when the body doesn't require them, and old cells aren't killed when they should be [1].

Leukemia is a type of blood cancer that affects blood cells. Because leukemia spreads quickly, it must be discovered as soon as possible. When the bone marrow generates a significant quantity of white blood cells or aberrant white blood cells, it causes this condition. Acute Lymphocytic Leukemia (ALL), Acute Myelogenous Leukemia (AML), Chronic Lymphocytic Leukemia (CLL), and Chronic Myelogenous Leukemia (CML) are the four main kinds of leukemia [2][3]. Experts use a microscope to

manually detect leukemia. This manual assessment method is expensive, time-consuming, and completely reliant on the knowledge and skills of the operator. The following issues are solved through detection via digital image processing. Images are the major input for digital image processing, which does not necessitate the use of expensive lab equipment. Many image processing methods have been developed to automate the process of leukemia detection. The input for this system is tiny blood smear images. Various image processing techniques are employed to achieve the desired result depending on the type and quality of the image. To detect and identify leukemia kinds, various techniques such as image enhancement, segmentation, feature extraction, and classification are applied.

2. RELATED WORKS

Bibi, Nighat, et al [1] provided a system based on the Internet of Medical Things (IoMT) to improve and deliver speedy and safe leukemia detection Clinical devices are linked to network resources in the proposed IoMT system using cloud computing. The technology allows patients and healthcare providers to coordinate testing, diagnosis, and treatment of leukemia in real time, potentially saving time and effort for both patients and clinicians. Furthermore, in pandemics such as COVID-19, the given framework is useful for resolving the concerns of patients in critical condition. Dense Convolutional Neural Network (DenseNet-121) and Residual Convolutional Neural Network (RCNN) are the approaches employed in the proposed framework to identify leukemia subtypes (ResNet-34).

Anwar, Shamama, and Afrin Alam [2] presenteda convolutional neural network (CNN) model, an automated diagnostic system to detect acute lymphoblastic leukemia (ALL). The model detects malignant leukaemia cells using tagged microscopic blood smear images. The current study employed data from the Acute Lymphoblastic Leukaemia Image DataBase (ALL IDB) and applied several data augmentation approaches to enhance the number of training data, hence reducing the problem of over-training.

Jha, Krishna Kumar, and Himadri Sekhar Dutta [3] developed Leukemia detection module based on deep learning from blood smear images. Pre-processing, segmentation, feature extraction, and classification are all performed by the detection scheme. The proposed Mutual Information (MI) based hybrid model, which combines the segmentation results of the active contour model and the fuzzy C means algorithm, is used for segmentation. The statistical and Local Directional Pattern (LDP) features are then retrieved from the segmented images and fed into the suggested Chronological Sine Cosine Algorithm (SCA) based Deep CNN classifier for classification.

Di Ruberto, Cecilia, Andrea Loddo, and Giovanni Puglisi [4] proposeda new method for identifying white blood cells in microscopic blood images and categorizing them as healthy or leukemia-affected The SMC-IDB, IUMS-IDB, and ALL-IDB public datasets for leukemia identification are used to test the described approach.

Ahmed, Nizar, et al [5] proposed employing convolutional neural networks (CNN) to diagnose all subtypes of leukemia from microscopic blood cell images, which requires a big training data set as a result, the authors conducted synthetic research into the effects of data augmentation for an increasing number of training samples. ALL-IDB and ASH Image Bank were two publicly available leukemia data sources used by the authors. As data augmentation, the authors used seven distinct image alteration techniques. We created a CNN architecture that can recognize all forms of leukemia subtypes. The authors also looked at naive Bayes, support vector machines, k-nearest neighbor, and decision trees, among other well-known machine learning techniques.

Kassani, Sara Hosseinzadeh, et al [6] proposed a deep learning-based automated technique for distinguishing immature leukemic blasts from normal cells The proposed deep learning-based hybrid method can extract high-level characteristics from input images and is enriched by several data augmentation techniques.

Ayyappan, Vinay, et al [7] developed a suite of machine learning classifiers that distinguish healthy B cells from lymphoblasts and identify phases of B cell acute lymphoblastic leukemia, including convolutional neural networks The authors demonstrated that normal B cells have a lower average dry mass and volume than malignant cells, and that these morphologic characteristics increase as disease progresses.

Wang, Qiwei, et al [8] took Single Shot Multibox Detector and An Incremental Improvement Version of You Only Look Once are two amazing object detection techniques that are used to solve the task of leukocyte recognition. Some essential parameters affecting various object detection methodologies are investigated, and detection models are created using the train set of 14,700 annotated images, in order to improve recognition performance.

Kumar, Deepika, et al [9] utilizing deep learning techniques, such as convolutional neural networks, the model eliminates the possibility of errors in the manual process. The model, which was trained on images of cells, pre-processes the images before extracting the best attributes. The model is then trained using an improved Dense Convolutional Neural Network architecture (referred to as DCNN here) before being used to predict the type of cancer present in the cells.

Yildirim, Muhammed, and Ahmet CevahirCinar [10] selected to distinguish between distinct types of white blood cells, such as eosinophils, lymphocytes, monocytes, and neutrophils, a convolutional neural network (CNN) was used. The CNN was trained with the Kaggle Dataset after being linked with Alexnet, Resnet50, Densenet201, and GoogleNet. The images in the database were then subjected to independent Gaussian and median filters. CNN classified the fresh imagegraphs with each of the four networks once more.

Loey, Mohamed, Mukdad Naman, and Hala Zayed [11] propose dusing standard methodologies that have various drawbacks, researchers used transfer learning to create two automated classification models based on blood microscopic images to diagnose leukemia. Blood microscopic images are pre-processed in the first model, and then features are retrieved using a pre-trained deep convolutional neural network termed AlexNet, which produces classifications using a variety of well-known classifiers. After pre-processing the images, AlexNet is fine-tuned for feature extraction and classification in the second model.

Pansombut, Tatdow, et al [12] implementeda CNN classifier to investigate the feasibility of a deep learning strategy for identifying lymphocytes and acute lymphoblastic leukaemia (ALL) subtypes, and this approach is compared to a widely used approach of support vector machines (SVMs) using handmade feature engineering. In addition, the comparison uses two classic machine learning classifiers: multilayer perceptron (MLP) and random forest.

Shah, Salman, et al [13] employeda deep learning model created specifically for the categorization of immature lymphoblasts and normal cells An ensemble of convolutional and recurrent neural networks is used in the proposed model. The authors also used a discrete cosine transform in conjunction with an RNN to take use of the cells' spectral properties.

Kasani, Payam Hosseinzadeh, Sang-Won Park, and Jae-Won Jang [14] aimed to build an aggregated deep learning model for the classification of leukemic B-lymphoblasts To create a trustworthy and accurate deep learner, data augmentation techniques were used to deal with the small dataset size, and a transfer learning strategy was used to speed up the learning process and increase the proposed network's performance. The results reveal that our suggested approach was able to combine data extracted through the best deep learning models and outperformed individual networks in Leukemic B-lymphoblast diagnosis, with a test accuracy of 96.58 percent.

Janaki, R [15] investigated the cancer of White Blood Cells (WBC) image recognition is difficulty. A Gaussian Feature Convolutional Visual Recognition system is used to classify different types of white blood cells (GFCVR). The most relevant properties or parts of these blood cells are fed into the neural network as tiny images. The key outcomes of this study include the detection of blood cancer-affected regions, as well as feature extraction for specific white blood cell images. Pre-processing the image and using multi-level clustering to determine the impacted area in blood cells is the proposed method.

3. PROPOSED CLASSIFICATION METHOD FOR LEUKEMIA PREDICTION

Convolutional Neural Networks (CNN) are the most successful image classification models of the present period, as they are based on deep learning models. CNN's multi-story designs are inspired by biology science [17]. We decided to employ a deep learning strategy to classify the leukemia image into a category because of the high accuracy and rapid expansion of CNN's research. The proposed Optimized Convolutional Neural Network (OCNN) adjusts the fundamental form of the coevolutionary network's structured-based design in such a way that it recognizes and classifies leukemia images and delivers considerably better results than previous leukemia classification approaches. Instead of imposing layer limits, the OCNN model maximizes the number of layers. For intermediate coevolutionary OCNN layers, a variety of filter sizes were also used. An input leukemiaimage is a greyscale image with a dimension of 64x64x1 that is used for OCNN preprocessing [18]. In addition to the activation function, the size of the convolutional layer is improved here. A stomp of two is performed in max pooling to reduce the size of the output image created by the convolutional layers. In the next stage, the image will be flattened and transmitted to the Multi-Layer Perceptron as well as a completely linked layer. The softmax classifier is then used to return the image class probability for each class ranging from 0 to 25.

Multi-layered perceptron stimulated CNNs are a type of neural system in which specific layers, such as convents with mandatory accessibility models, are determined. Without the input of anyone else, Neural Networks do not scale effectively for a whole image. Over-fitting is a common problem with traditional neural systems. The optimized convolutional neural network, on the other hand, takes advantage of the locally based properties of the given leukemia images, such as treating close and inaccessible leukemia image pixels in an unexpected way. As a result, the recommended technique is used to classify numerous leukemiaimages based on their characteristics.

3.1 Convolutional Layer

The OCNN's input is a leukemia grayscale image x(w,h,d), where w determines the width, d determines the depth (d=1), and h determines the height of the leukemia images. Invariant features of leukemia images are cultivated hierarchically and automatically through this layer. To begin with, the convolution layer recognizes an image's two-level features in order to save the spatial relationship between image pixels by depleting tiny portions of the image. Local image attributes from one layer are

recognized and then mapped to the featuring maps. A 2-Dimensional representation of the convolutionally layers is also determined, where the input source image is a 64x64x1 matrix (width, height, and depth) and a 3x3x1 filter is applied on it to transform the supplied image to the feature map.

50 filters of size 5x5x1 were used in the first layer of convolution. The OCNN model takes a 64x64x1 leukemia image with a zero-padding p=2 as input. The dot product calculation, together with the input leukemiaimage, is used to decide filter solicitation. The output volumes are 64x64x50 in size, with w=h=64 =((64-5)+2*2)/1+1 and d=50. This layer has a total of 64x64x50=204800 neurons. Furthermore, every 204800 neurons is linked to a kernel region that is based locally. We strided 70 3x3x1 convolutional filters amongst an input image using a strider 1 on the next two Convolutional layers. These filters are given random values at first, and then they are applied to the training dataset. The activation function is the next stage.

3.2 Leaky ReLU

For bringing non-linearity into the weights contained in intermediate layers of CNN, traditional CNN architectures use the Rectified Linear Unit (ReLU) activation function [14][15]. While the architecture we propose uses the concept of Leaky ReLU to solve the dying ReLU problem, which states that when a negative gradient is present, the ReLU activation function completely reduces to zero, the leaky ReLU does not do so, as shown by the following equation, because it does have zero slopes present in a negative direction (2). This function is used to indicate a unique identification of potential features. As a result, Leaky ReLU produces output images of 64x64x50 pixels. To decompress into its original size, this productivity image must be sampled. As a result, the image is sent to the pooling layer to be sampled.

3.3 Pooling Layer

The input leukemia image is turned to an image stack after the convolution layer. The implementation of the pooling layer, which is how we shrink the image stack, is the next stage, and it's very straightforward. Pooling is used to reduce the spatial size of the input. By decreasing image feature dimensions and calculations, pooling prevents over-fitting. Pooling creates a network that is immune to minor biases and changes. It also ensures that a scale-invariant depiction of the leukemia image is obtained. Max, Min, and Average Pooling are three different types of poling. The max-pooling algorithm is used in a suggested OCNN model. We'll start with a 2x2 pixel window size. Choose a stride of 2 and traverse your window in stride through a filtered image that is the result of an earlier convolutional layer, ensuring that each window takes the maximum value. As a result, the pooling layer provides a 32x32x50 output image.

The pooling layer's output images are sent to the second convolutionally based layer as an input image. 70 kernels of 3x3x1 size have been added to the second layer. The remaining parameters are identical to those used in the first convolutional layer. After applying the Leaky ReLU activation function to the second layer, the result is 64x64x70, followed by max pooling for resampling the leukemia image. For the third convolutional layer, repeat the technique with 70 kernels of size 3x3x1 and output volumes of size 64x64x70 images. All kernels are given a randomized value at the start. For obtaining the leukemia image features, the kernel's value is altered at each epoch. Following image sampling, the image is transmitted to a fully connected layer for neuron overlapping reduction and error minimization.

3.4 Fully Connected Layer

Neurons in this layer are linked to all activation from a previous layer. Simple matrix multiplication is performed here, coupled with a bias offset. Furthermore, when two or more connected neurons detect the same leukemia often characteristics, neurons are whispered to ensure established codependency for co-adaption on all neurons, which raises the problem of overfitting. In this type of case, a dropout function is used in our work to eliminate such complexity by neglecting neurons random sets during the training phase. The DropOut function disables nodes at random, preventing learning or pattern-based updates. After that, the proposed OCNN uses 256 dense layers and softmax classifiers to classify the leukemia images.

3.5 Soft Max Classifier

The likelihood of each class label is provided by the Softmax classifier, while the loss of the hinge provides the margin. It is considerably easier for us to evaluate probabilities than it is to interpret marginal scores as humans (such as loss of hinge and loss of squared hinge). We also look at the rank-5 precision of Convolutional Neural Networks for datasets like Image Net (where we check to see if the ground-truth label for a specified input image is in the top-5 expected labels returned by a network). The softmax classifier has chosen the appropriate category for the classification of the leukemiaimages.

There must also be a change in the weights due to the fundamental randomness in weights. The OCNN model is applied in a systematic manner. Remember that each of the k output neurons, which connects to all of the neurons in the previous levels, is related to leukemia in some way. The number of output neurons must be noted and equal to the number of leukemia patients in each category.

3.6 Proposed OCNN Learning Algorithm

Step 1: Leukemiaimages with discrete width and height were included in the dataset. We needed a 64x64x1 leukemia image for the planned OCNN. As a result, pre-process the leukemia image to obtain the desired input image size. A grayscale image that is an executable represented as x(w,h,d), where h is the height, w is the width, and d is the depth (d=1) is one of the model's inputs. Before feeding an input image to the OCNN system for smoothing and noise removal, it is also necessary to perform image initialization.

Step 2: The input image for the convolutional layer was I(wxh), with a (f1xf2) kernel filter F and constant bias b. The convolutional layer output is as follows:

$$(I * F)_{ij} = \sum_{m=0}^{f_1-1} \sum_{n=0}^{f_2-1} F_{m,n}.I_{i+m,j+n} + b \qquad (1)$$

Where $0 \leq i \leq w - f_1$ and $0 \leq j \leq h - f_2$

Step 3: The equation (2) gives the leaky ReLU activation function, which is used on convolution maps to map a specific output to a specific input.

$$f(x) = \{0.01x\ for\ x < 0x\ for\ x => 0\} \qquad (2)$$

x – The actual value of a pixel.

Step 4: Pooling Layer: Choose a stride, commonly 2 or 3, and walk your window in stride across the filtered image, taking the maximum value for each window, a process known as down sampling. Pool

Layer generates a volume [w1xh1xd1], where w1, h1, and d1 are calculated using equations (3), (4), and (5):

$$w1 = \frac{w - f}{s} + 1 \qquad (3)$$

$$h1 = \frac{h - f}{s} + 1 \qquad (4)$$

$$d1 = d \qquad (5)$$

Step 5: Fully connected layer: At the end of the process, all of the features are concatenated into a single vector derived from previous layers.

Consider: The input to unit a_i^l (ith unit in layer l). The output of unita_i^l.

f(x): Activation function.

w_{ij}^l: Weight from some unit

$a_i^{l'}s$: Output to some other unit $a_i^{l=1}$

L: Output layer, *l*: the hidden layer.

E: The squared error function.

3.6.1 Forward Propagation

Step 1: The outcome of step 1 is utilized to compute the inputs for the following layer.:

$$P_i^l = \sum_j w_{ji}^{l-1} Q_j^{l-1} \qquad (6)$$

Step 2: Calculate the activations for the layer whose input is known:

$$Q_j^l = f(x) + Q_j^{l-1} \qquad (7)$$

Where f(x) is calculated by equation (2).

Step 3: For the productivity layer, repeat steps 1 and 2 of the forward propagation process.

3.6.2 Backward Propagation

Step 1: Computing error at the output layer L:

$$\frac{\partial E}{\partial Q_j^L} = \frac{dE}{dQ_j^L} Q^L \qquad (8)$$

Step 2: Calculate the partial derivative of the error concerning the image input of a neuron at the first layer l for which mistakes are known to update bias.

$$\frac{\partial E}{\partial P_j^l} = f\left(P_j^l\right)\left(\frac{dE}{dQ_j^l}\right) \qquad (9)$$

$$b_j = b_j - b_{j-1}\frac{\partial E}{\partial P_j^l} \quad (10)$$

Step 3: Error computation at a lower layer using a partial derivative of error (as in step 1):

$$\frac{\partial E}{\partial Q_j^l} = \sum w_{ij}^l \left(\frac{dE}{dP_j^{l+1}}\right) (11)$$

Step 4: Backpropagation steps 2 and 3 are performed until partial derivatives of errors are identified at all layers accepting an input layer.

Step 5: Calculation of the error's gradient:

$$\frac{\partial E}{\partial w_{ij}^l} = Q_j^l \left(\frac{\partial E}{\partial P_j^{l+1}}\right) (12)$$

Step 6: Repeat steps 2–5 until the error is less than or equal to the minimum error (chosen by the programmer)

Step 7: After step 6, the trained model is ready. For the classification of the testing leukemiaimage, use this model.

Convolution operation extracts various features after the kernel for those features is known. The OCNN has no prior knowledge of an individual's nature or the whole amount of traits. The convolution layer will no longer be able to extract any single feature. In forward propagation, the convolution layer predicted the features, then backpropagation attempted to refine those predictions successively. After training, numerous filters are bypassed in favor of error minimization, and eventually, convolution retrieves features that were previously undetectable. The proposed OCNN model extracts anonymous characteristics from leukemiaimages using three convolution layers, the first with 50 filters and the other two with 70 filters each. Furthermore, once the training is completed, the precise nature of each feature can be determined.

4. RESULT AND DISCUSSION

4.1 Dataset Description

Leukemia dataset is considered from the Kaggle repository [16]. This dataset contains two Database with cancer images and without cancer images. The first database ALL_DB1 have 108 images (49 Cancer images, and 59 healthy images), and the second database ALL_DB2 have 260 images of the cell is placed in the center of the image is blast or not. Among 260 images, 130 images are taken from healthy individuals, and 130 are cancer patient images.

4.2 Performance Metrics

Table 1 represents the performance metrics considered in this research paper to evaluate the performance of the proposed OCNN classifier.

Table 1: Performance Metrics

Performance Metrics	Equation
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Sensitivity	$\frac{TP}{TP+FN}$
Specificity	$\frac{TN}{TN+FP}$
Precision	$\frac{TP}{TP+FP}$
False Positive Rate (FPR)	1- Specificity
Miss Rate	1-Sensitivity
False Discovery Rate (FDR)	1-Precision

True Positive (TP) = The number of images correctly identified as healthy

True Negative (TN) = The number of images correctly identified as cancer

False Positive (FP) = The number of images incorrectly identified as healthy

False Negative (FN) = The number of images incorrectly identified as cancer.

The performance of the proposed OCNN classifier is compared with existing classification techniques like Artificial Neural Network (ANN) and Convolutional Neural Network (CNN) using existing feature extraction techniques like Principal Component Analysis (PCA) and optimization-based Feature Extraction techniques like Particle Swarm Optimization (PSO), and Animal Migration Optimization (AMO).

Table 2 gives the accuracy obtained by the Proposed OCNN, ANN, and CNN classification techniques with PCA, PSO and AMO based feature extraction techniques. From the table 2, it is clear that the Proposed OCNN gives more accuracy than other classification techniques.

Table 2: Classification Accuracy (in %) obtained by Proposed OCNN, ANN, and CNN classification techniques with PCA, PSO and AMO based feature extraction techniques

Feature Extraction Techniques	Accuracy (in %) Classification Techniques		
	Proposed OCNN	ANN	CNN
PCA	92.14	79.24	82.43
AMO	91.35	80.66	83.78
PSO	68.41	60.45	64.73

Table 3 gives the sensitivity obtained by the Proposed OCNN, ANN, and CNN classification techniques with PCA, PSO and AMO based feature extraction techniques. From the table 3, it is clear that the Proposed OCNN gives more sensitivity than other classification techniques.

Table 3: Sensitivity (in %) obtained by Proposed OCNN, ANN, and CNN classification techniques with PCA, PSO and AMO based feature extraction techniques

Feature Extraction Techniques	Sensitivity (in %) by Classification Techniques		
	Proposed OCNN	ANN	CNN
PCA	90.71	66.84	80.47
AMO	91.78	67.75	81.81
PSO	69.72	54.57	62.26

Table 4 gives the specificity obtained by the Proposed OCNN, ANN, and CNN classification techniques with PCA, PSO and AMO based feature extraction techniques. From the table 4, it is clear that the Proposed OCNN gives more specificity than other classification techniques.

Table 4: Specificity (in %) obtained by Proposed OCNN, ANN, and CNN classification techniques with PCA, PSO and AMO based feature extraction techniques

Feature Extraction Techniques	Specificity (in %) by Classification Techniques		
	Proposed OCNN	ANN	CNN
PCA	89.75	65.53	78.74
AMO	90.13	64.81	80.27
PSO	69.13	53.86	60.58

Table 5 gives the precision obtained by the Proposed OCNN, ANN, and CNN classification techniques with PCA, PSO and AMO based feature extraction techniques. From the table 5, it is clear that the Proposed OCNN gives more precision than other classification techniques.

Table 5: Precision (in %) obtained by Proposed OCNN, ANN, and CNN classification techniques with PCA, PSO and AMO based feature extraction techniques

Feature Extraction Techniques	Precision (in %) by Classification Techniques		
	Proposed OCNN	ANN	CNN
PCA	92.75	66.43	80.53
AMO	93.67	68.46	82.85
PSO	69.65	54.72	61.49

Table 6 gives the false positive rate obtained by the Proposed OCNN, ANN, and CNN classification techniques with PCA, PSO and AMO based feature extraction techniques. From the table 6, it is clear that the Proposed OCNN gives reduced false positive rate than other classification techniques.

Table 6: False Positive Rate (in %) obtained by Proposed OCNN, ANN, and CNN classification techniques with PCA, PSO and AMO based feature extraction techniques

Feature Extraction Techniques	False Positive Rate (in %) by Classification Techniques		
	Proposed OCNN	ANN	CNN
PCA	10.25	34.47	21.26
AMO	9.87	35.19	19.73
PSO	30.87	46.14	39.42

Table 7 gives the miss rate obtained by the Proposed OCNN, ANN, and CNN classification techniques with PCA, PSO and AMO based feature extraction techniques. From the table 7, it is clear that the Proposed OCNN gives reduced miss rate than other classification techniques.

Table 7: Miss Rate (in %) obtained by Proposed OCNN, ANN, and CNN classification techniques with PCA, PSO and AMO based feature extraction techniques

Feature Extraction Techniques	Miss Rate (in %) by Classification Techniques		
	Proposed OCNN	ANN	CNN
PCA	9.29	33.16	19.53
AMO	8.22	32.25	18.19
PSO	30.28	45.43	37.74

Table 8 gives the false discovery rate obtained by the Proposed OCNN, ANN, and CNN classification techniques with PCA, PSO and AMO based feature extraction techniques. From the table 8, it is clear that the Proposed OCNN gives reduced false discovery rate than other classification techniques.

Table 8: False Discovery Rate (in %) obtained by Proposed OCNN, ANN, and CNN classification techniques with PCA, PSO and AMO based feature extraction techniques

Feature Extraction Techniques	False Discovery Rate (in %) by Classification Techniques		
	Proposed OCNN	ANN	CNN
PCA	7.25	33.57	19.47
AMO	6.33	31.54	17.15
PSO	30.35	45.28	38.51

5. CONCLUSION

Computer Aided Diagnosis (CAD) systems are increasingly utilizing image analysis and Deep Learning (DL) techniques, due to their high accuracy in several medical imaging fields, including the detection of Acute Lymphoblastic (or Lymphocytic) Leukemia (ALL) from peripheral blood samples. In this paper, an Optimized Convolutional Neural Network (OCNN) classifier is proposed to classify the ALL from healthy cells. Leaky ReLUactivation function in this proposed OCNN is used to correctly classify the images. From the results obtained, it is clear that the proposed OCNN gives more accuracy, specificity, sensitivity and precision and also it reduced the miss rate, false positive rate and false discovery rate when it is compared against other classifiers like CNN, and ANN with different feature extraction techniques.

REFERENCES

[1] Bibi, Nighat, et al. "IoMT-based automated detection and classification of leukemia using deep learning." *Journal of Healthcare Engineering* 2020 (2020).

[2] Anwar, Shamama, and Afrin Alam. "A convolutional neural network–based learning approach to acute lymphoblastic leukaemia detection with automated feature extraction." *Medical & Biological Engineering & Computing* 58.12 (2020): 3113- 3121.

[3] Jha, Krishna Kumar, and Himadri Sekhar Dutta. "Mutual information based hybrid model and deep learning for acute lymphocytic leukemia detection in single cell blood smear images." *Computer methods and programs in biomedicine* 179 (2019): 104987.

[4] Di Ruberto, Cecilia, Andrea Loddo, and Giovanni Puglisi. "Blob detection and deep learning for leukemic blood image analysis." *Applied Sciences* 10.3 (2020): 1176.

[5] Ahmed, Nizar, et al. "Identification of leukemia subtypes from microscopic images using convolutional neural network." *Diagnostics* 9.3 (2019): 104.

[6] Kassani, Sara Hosseinzadeh, et al. "A hybrid deep learning architecture for leukemic B-lymphoblast classification." *2019 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2019.

[7] Ayyappan, Vinay, et al. "Identification and staging of B-cell acute lymphoblastic leukemiausing quantitative phase imaging and machine learning." *ACS sensors* 5.10 (2020): 3281-3289.

[8] Wang, Qiwei, et al. "Deep learning approach to peripheral leukocyte recognition." *Plo Sone* 14.6 (2019): e0218808.

[9] Kumar, Deepika, et al. "Automatic detection of white blood cancer from bone marrow microscopic images using convolutional neural networks." *IEEE Access* 8 (2020): 142521-142531.

[10] Yildirim, Muhammed, and Ahmet CevahirCinar. "Classification of White Blood Cells by Deep Learning Methods for Diagnosing Disease." *Rev. d' Intelligence Artif.* 33.5 (2019): 335-340.

[11] Loey, Mohamed, Mukdad Naman, and Hala Zayed. "Deep transfer learning in diagnosing leukemia in blood cells." *Computers* 9.2 (2020): 29.

[12] Pansombut, Tatdow, et al. "Convolutional neural networks for recognition of lymphoblast cell images." *Computational Intelligence and Neuroscience* 2019 (2019).

[13] Shah, Salman, et al. "Classification of normal and leukemic blast cells in B-ALL cancer using a combination of convolutional and recurrent neural networks." *ISBI 2019 C- NMC Challenge: Classification in Cancer Cell Imaging*. Springer, Singapore, 2019. 23-31.

[14] Kasani, Payam Hosseinzadeh, Sang-Won Park, and Jae-Won Jang. "An Aggregated-Based Deep Learning Method for Leukemic B-lymphoblast Classification." *Diagnostics* 10.12 (2020): 1064.

[15] Janaki, R. "Detection of leukemia in microscopic white blood cell images using Gaussian feature convolutional visual recognition algorithm." *J Critic Rev* 7.3 (2020): 173-180.

[16] https://www.kaggle.com/nikhilsharma00/leukemia-dataset

[17] Sakthiraj, Fredric Samson Kirubakaran. "Autonomous Leukemia Detection Scheme Based on Hybrid Convolutional Neural Network Model Using Learning Algorithm." *Wireless Personal Communications* (2021): 1-16.

[18] Anwar, Shamama, and Afrin Alam. "A convolutional neural network–based learning approach to acute lymphoblastic leukaemia detection with automated feature extraction." *Medical & Biological Engineering & Computing* 58.12 (2020): 3113-3121.