International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Acute Lymphoblastic Leukemia Detection and Classification Using an Ensemble of Classifiers and Pre-Trained Convolutional Neural Networks

Avinash Bhute[1], Harsha Bhute[2], Sandeep Pande[3], Amol Dhumane[4], Shwetambari Chiwhane[5], Shalini Wankhade[6]

Submitted: 25/08/2023 **Revised**: 16/10/2023 **Accepted**: 28/10/2023

Abstract: Leukemia affects a significant portion of the global population. It is one of the most prevalent types of cancer among adults and children, according to the World Health Organization (WHO). The global incidence of leukemia has been increasing over the past few decades, due in part to an aging population and improved survival rates for other types of cancer. Early-stage leukemia detection at the lowest possible cost is a serious challenge in the field of leukemia disease diagnosis. Traditional methods such as blood tests, bone marrow tests, and spinal fluid tests are very time-consuming and have limited ability to analyze large amounts of data. Comparatively, the use of ensemble learning and pre-trained convolutional neural network (CNN) algorithms provide more accurate and efficient methods to detect and analyze the disease in less time. To improve the accuracy and optimize training time, we are proposing ensemble learning-based models for the detection of types of leukemia based on blood microscopic images, rather than traditional techniques. Ensemble learning can scan enormous amounts of data, including images, laboratory results, and patient information, to find patterns and predict the presence of the disease. This is one of the key benefits of adopting ensemble learning for leukemia identification. This can be especially helpful for examining small and complex changes in blood cell images, which are frequently challenging to spot using conventional techniques. Different pre-trained models are used in this work to identify different forms of leukemia. Pretrained networks such as ResNet50, VGG16, and InceptionV3 are relatively simple approaches for applying ensemble learning to image analysis. The three models—ResNet50, VGG16, and InceptionV3 —have been improved for feature extraction and classification. Experiments were carried out on the dataset, and an accuracy of about 90% was achieved.

Keywords*: *Leukemia, Convolutional Neural Networks, Ensemble Learning, pre-trained models, Deep Learning, Cancer Detection

1. Introduction

Leukemia is a potentially fatal condition of the blood and/or bone marrow, brought on by the development of abnormal blood cells. The body's immune system is affected by the generation of these abnormal cells. Powerful infection-fighting agents - the white blood cells (WBCs), generally multiply and divide in an orderly manner as the body requires them. But in leukemia patients, there is a rapid growth in the production of aberrant WBCs in the bone marrow. These abnormal WBCs in the bloodstream further spread to the other organs like the brain, spleen, kidneys, and liver. Acute lymphoblastic leukemia (ALL), acute myeloid leukemia (AML), chronic lymphoblastic leukemia (CLL) and chronic myeloid leukemia (CML).

This work was supported in part by the Department of Computer Engineering, PCCOE, Pune under the scheme of NIDHI, India. Implemented by students along with Mentors.

lymphoblastic leukemia (CLL) and chronic myeloid leukemia (CML) are the four main kinds of leukemia according to the makeup of the cytoplasm.

Blood cancer is a surging concern and medical tests are time-consuming and sensitive to find any blast cells. With the use of traditional systems, it takes time to predict the result. To minimize medical risks, haematologists must detect the presence of leukemia and its specific type and establish the best treatment. Detecting leukemia through optical blood smear testing under the supervision of a specialist is a critical and time-consuming stage. Spinal fluid tests, blood tests, bone marrow tests, etc. are all examples of manual testing. Modern techniques can help to detect diseases more quickly and improve the cure rate. This can be done using various deep learning techniques. Many previous studies have proposed machine learning and deep learning techniques for detecting leukemia and its types. Our work mainly focuses on using ensemble learning and pre-trained CNN models to solve this problem.

[1]*Dept. of Computer Engg., Pimpri Chinchwad College of Engineering, Pune, MH, India., ORCID ID : 0000-0003-3236-686X*
[2]*Dept. of Information Tech., Pimpri Chinchwad College of Engineering, Pune, MH, India, Pune, MH, India., ORCID ID 0000-0002-1522-4053*
[3]*School of Computer Engg. & Tech., MIT Academy of Engineering, Alandi(D), Pune, MH, India., ORCID ID : 0000-0001-6969-0423*
[4]*Dept. of Computer Engg., Symbiosis Institute of Technology, Pune, MH, India., ORCID ID : 0000-0002-7529-4125*
[5]*Dept. of Computer Engg., Symbiosis Institute of Technology, Pune, MH, India. ,ORCID ID : 0000-0002-3534-9654*
[6]*Dept. of Information Tech.,Vishwakarma Inst. of Information Techlology Pune, MH, India., ORCID ID : 0009-0006-7048-9675*
** Corresponding Author Email: avinash.bhute@pccoepune.org*

Image processing is used to improve the quality of a picture or to extract information from an image and feed it to an algorithm for further detection. In image processing, data preparation and image preprocessing are done where we have rescaled the image and adjusted the brightness to improve the image quality. Data augmentation is performed to increase the image count by flipping it horizontally and vertically. To simplify the matrix calculation, images are transformed to grayscale.

Ensemble learning is a general conceptual approach to machine learning that aims to improve predictive performance by integrating the predictions from various models. In comparison to using a single model, this strategy facilitates the generation of greater prediction performance by using multiple models. In this paper, pre-trained CNN models such as ResNet-50, VGG16 and InceptionV3 have been used for ensemble learning.

We have solved the following problems.

a) Firstly, to improve the quality of blood microscopic images, image processing techniques such as image rescaling, RGB to grayscale conversion, brightness adaptation, and discrimination are used. Image rescaling resizes all the images to a fixed size before classification.

b) The number of images required in deep learning models is very large. In the dataset, AML has 935 images, ALL has 858, CML has 623 and CLL has 510. Comparing the four datasets, CLL has fewer images as compared to others. This causes an imbalance in the dataset. To create a balanced dataset, data augmentation is used. It increases the number of images for the particular class having fewer images thus balancing the dataset.

c) For feature extraction and classification, various pre-trained CNN models like ResNet-50, VGG16, and InceptionV3 are used. These models have previously been trained on various datasets and fine-tuned with weights and biases. These models can be quickly and simply retrained using our dataset.

d) We have implemented some machine learning algorithms such as Logistic Regression, Naive Bayes, Decision Tree, and Random Forest as well as transfer learning models like ResNet50, VGG16 and InceptionV3 for the detection of types of leukemia. The machine learning algorithms and transfer learning models are compared among themselves and with each other. This is to find out which ones are providing better results in terms of accuracy.

This paper is structured as follows: Section 1 introduced the problem statement of leukemia detection, the rigorous literature survey and have presented it in Section 2. Section 3 explains the dataset used along with the proposed model. Section 4 contains the final results and discussions. The paper is concluded in Section 5, which also gives a brief overview of future works.

2. Related Work

Aftab M O, et al [1] had given two methodologies. First, image processing had been performed, in which RGB to grayscale conversion, resizing, and normalization of microscopic images had been done. They had applied transfer learning utilizing the Apache Spark framework and the BigDL library which had given 97.33% training and 94.78% testing accuracy, respectively in the first methodology. In the second methodology, they had imported the Keras library and had created the CNN model which had given 96.42% training and 92.69% testing accuracy, respectively.

Nizar Ahmed, et al [2] had proposed an approach to detect the four subtypes of leukemia in which they had first performed image transformation and data augmentation. After image processing, CNN architecture had been used for feature extraction and classification. They had also explored various machine algorithms. For evaluation, they had used a 5-fold cross-validation technique. CNN had achieved 88% accuracy for binary classification and an 81% for multiclass classification of leukemia subtypes. Maryam Bukhari, et al [3], has suggested using a squeeze based on CNN architecture to identify leukemia. Firstly, they had performed image acquisition. After that, data augmentation was done. Lastly, the diagnosis of leukemia had been made by using the proposed CNN model. Squeeze and excitation learning had helped to extract robust features from leukemia-affected and normal blood cells. 98.3% accuracy had been achieved.

K K Anilkumar, et al [4], had given a pre-trained CNN model for leukemia detection. First, image acquisition had been performed. After that, flipping, rotating, and scaling of the images had been performed. Then, for classification, pre-trained CNNs along with transfer learning had been used. Pretrained networks such as AlexNet, VGG16, GoogleNet, VGG19, MobileNet-v2, Inceptionv3, Resnet101, DenseNet-201, ResNet-18, ResNet-50, Xception, and Inception-ResNet-v2 had been used. 100% classification accuracy had been obtained with all CNNs for the ALL_IDB1 dataset, and 100% accuracy had been obtained with all CNNs for the ALL_IDB2 except AlexNet and VGG-16.Luis H S Vogado, et al [5] had been proposed a technique for locating leukemic cells using transfer learning and images of blood smears. AlexNet, Caffenet, and Vgg-f are a few examples of pre-trained CNN models that have been utilized for feature extraction and image processing. The gain ratio had been chosen over other criteria for feature selection. SVM was then employed for additional classification. This gives 99.76% accuracy.

Mohamed Loey, et al [6] had been proposed two automated classification systems using transfer learning. In the first model, features were derived from blood microscopic images using a pre-trained deep convolutional neural network dubbed AlexNet. A variety of classifiers had been used for classification. Nighat Bibi, et al [7] had been proposed a system to improve and give a fast and safe detection of leukemia subtypes using IoMT based framework. First, data augmentation had been performed, and then ResNet-34 and DenseNet-121 had been used for classification. ResNet-34 had given 99.56% accuracy while DenseNet-121 had given 99.91% accuracy.

Raheel Baig, et al [8] had proposed a hybrid block of CNN 1 and CNN 2 for detecting ALL, AML, and multiple myeloma (MM). Here, image processing and segmentation had been done using adjusting brightness, and contrast, sharpening the images, and cropping the images, using the adaptive histogram equalization (AHE) method for enhancing contrast. Feature extraction had been done using CNN. For classification, SVM, fine KNN, Boosting, etc algorithms had been used for classification. They had reported an accuracy of 95.62% and a sensitivity of 95.20% for detecting leukemia cells, outperforming several of the most recent techniques. Chayan Mondal, et al [9] has proposed an approach that uses pre-trained CNN models and ensemble learning for leukemia classification. They had used different augmentation and pre-processing techniques to generalize a model. The dataset used had been C-NMC 2019. There has been use of several pre-trained models. An 89.7% F1 score, almost 88% accuracy, and an AUC of 0.948 were provided with the help of ensemble model.

Sultana S, et al [10], For feature extraction and classification, they had used a VGG-16 model. To boost the dataset's size, they had used techniques including rotation, magnification, flipping, and shifting. For the ALL-IDB dataset, the suggested method had obtained accuracy of 98.31%, sensitivity of 97.97%, and specificity of 98.64%. Vogado L H S, et al [11] had been developed an ensemble approach. Pre-trained CNN models like AlexNet and Vgg16 were used to extract the features from the ROIs during the feature extraction stage. A group of some machine learning classifiers were employed in the classification stage to categorize the photos based on the retrieved features. For ALL_IDB1 and ALL_IDB2, the ensemble approach has a success rate of 97.5% and 96.5%, respectively.

Gupta R, et al [12] had comprised feature extraction using transfer learning from pre-trained CNNs for extracting features from the blood cells. The collected characteristics were then fed into a group of machine learning classifiers. The proposed method's findings revealed a classification accuracy of 97.83%. Kumar A, et al [13] had used an ensemble learning strategy to suggest a strategy for automatically leukemia disease diagnosing. According to suggested approach, characteristics from the blood cell images were extracted using pre-trained CNN models. A classifier ensemble made up of SVMs, k-NNs, and random forests had been fed the extracted features after that. A 96.58% classification accuracy was demonstrated by the proposed method's findings.

Wang J, et al [14], had suggested utilizing pre-trained CNNs and an ensemble of classifiers. The suggested technique used transfer learning from CNNs that had already been trained to extract characteristics. After that, ensemble classifiers made up of SVMs, k-NNs, and decision trees had been fed the extracted features. The proposed method's findings revealed a classification accuracy of 97.12%. Wang S, et al [15], The suggested approach had comprised feature extraction using transfer learning from pre-trained CNNs. After feeding a group of pre-trained CNNs, the final classification by averaging the individual classifiers was determined. A classification accuracy of 98.30% is achieved. Gunda S, et al [16], Pre-processing and classification were the two key phases of the proposed system. To acquire the region of interest (ROI) containing the cells, picture enhancement and segmentation had been used during the pre-processing step. The classification is done using a pre-trained CNN model (VGG-16). The suggested model successfully identified leukemia with classification accuracy of 99.12%.

Alkhateeb H., et al [17], had suggested employing an ensemble learning method to detect acute myeloid leukemia (AML). Four pre-trained CNN models were used in the proposed ensemble learning strategy for the feature extraction and categorization of AML pictures. The final categorization outcome was created by combining the outputs of the four models using a majority vote. The suggested ensemble learning strategy attained a high classification accuracy of 98.9%. Akter S, et al [18], had used transfer learning and convolutional neural networks (CNN) that had been previously trained. To improve the contrast of the photos, the pre-processing stage had included normalization and histogram equalization. According to the results, ResNet-50 had the highest accuracy (96.6%), while Inception-v3 had the highest sensitivity (97.9%).

Biswas K, et al [19], had proposed a variety of CNN architectures, and their performance was assessed. Comparing the ensemble model to individual CNN models and other current approaches, it had better accuracy and sensitivity. S Ramaneswaran, et al [20] had suggested using microscopic blood cell pictures to classify acute lymphoblastic leukemia (ALL) using a hybrid Inception v3 XGBoost model. This technique classified images using the XGBoost model and extracted image features using Inception v3. The proposed model had achieved an F1 score 0.986.

3. Proposed Work

Detecting leukemia using Convolutional Neural Networks (CNNs) is an exciting and potentially life-changing application of deep learning in the medical field. Leukemia is a type of cancer that affects the blood and bone marrow and can be challenging to diagnose accurately. CNNs, being powerful image processing models, can be leveraged to analyse and detect abnormal blood cells, aiding in the early detection and diagnosis of leukemia. The proposed architecture for detecting the leukemia is shown in fig. 1.

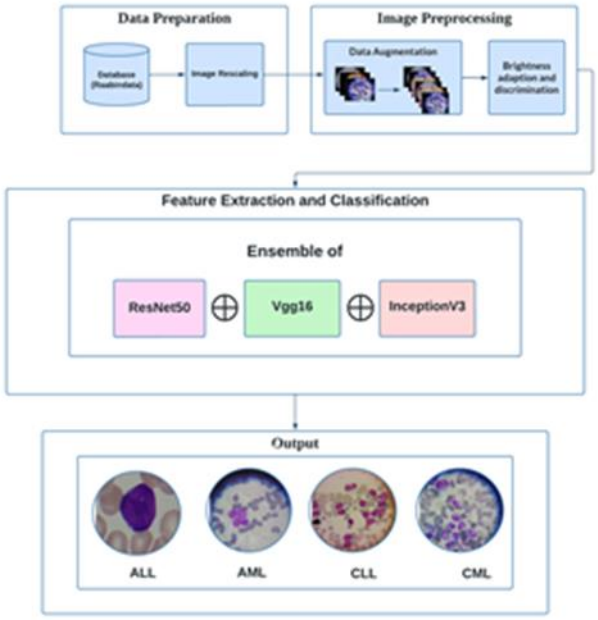


Fig. 1. Proposed architecture for detection of types of leukemia

The first phase is data preparation in which images are gathered from the source and then rescaled the image to match the input size required by different models. Later, the image preprocessing module preprocesses the images and is then fed to the feature extraction module. In feature extraction, different pre-trained models like VGG16, ResNet50, and InceptionV3 are used. Then at the classification module, it classifies different leukemia types feeding the image dataset to the fully connected network, at last, the softmax activation function is activated and then the images are classified as ALL, AML, CLL, and CML.

3.1. Data Analysis

All image samples are taken from the Raabin Leukemia dataset from patients who were referred to the medical laboratory, working with Takht-e Tavous Laboratory, Iran. Its imaging is done using a Zeiss microscope and an LG J3 smartphone camera. This is supervised data. There are a total of 1800 microscopic images of all the classes. From a total of 1800 images, we have allocated 320 training images, 40 testing images, and another 40 validation images. This distribution follows an 80:20 ratio for the train and test sets. The graph in figure 2 represents the distribution of images across different classes in the dataset. Figure 3 shows some sample images of different types of leukemia taken from this Raabin Leukemia dataset

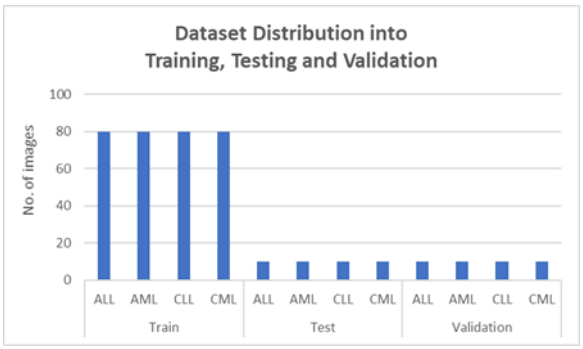


Fig. 2. Dataset Distribution

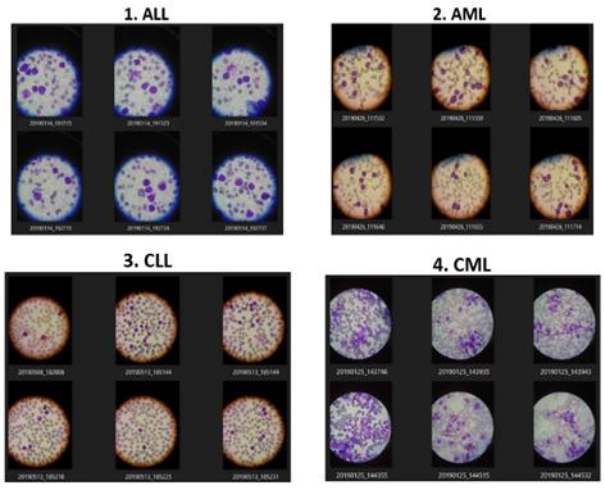


Fig. 3. Sample from dataset

3.2. Image Processing

Pre-processing aims to improve the image data by reducing unwanted distortions. In image preprocessing, rescaling, data augmentation, brightness adaptation, and discrimination are done by using Keras ImageDataGenerator class. Resizing a digital image is known as image scaling. An image becomes smaller as it is scaled down, and more prominent when scaled up. Scaling can be done on both raster and vector graphics, but the outcomes are different. The original images have RGB coefficients ranging from 0 to 255, which could be problematic for our models when using a normal learning rate. To address this issue, we need to adjust the values to a more manageable range. Therefore, we scale down the original images by dividing them by 255, resulting in values between 0 and 1.Increasing the number of images by rotating them vertically or horizontally is called data augmentation. Here, images are increased in number to equalize the image count in all four classes which improves the functionality of models being used in deep learning.

3.3. Feature Extraction

The process of turning an image's raw pixel values into more

insightful and practical information is called feature extraction. In feature extraction, InceptionV3, VGG16, and ResNet50 pre-trained models are used. These models are specifically used to extract nucleus features like their size and color. Particular features are extracted like in ALL, the nucleus is round or oval [21] whereas CML has the intended nucleus structure [22]. AML contains a multilobed nucleus [23] and CLL has more atypical lymphocytes with reduced nucleus size [24]. These features are being considered in this feature extraction phase.

3.3.1. Inception V3

InceptionV3 consists of a structure with 48 layers, representing an improvement over the earlier versions InceptionV1 and InceptionV2. This network should feed with an image size of 299 x 299 pixels. To utilize the InceptionV3 model, the 'imagenet' dataset weights are loaded. The top layer, which was originally used for classification, is excluded by setting the parameter include_top to False. In order to maintain the pre-trained weights, the layers are frozen by setting their trainable attribute to False. To adapt the model for the specific classification task, a new output layer is added. This is accomplished by applying the Flatten() function to convert the output of the InceptionV3 model into a one-dimensional tensor. Subsequently, a softmax activation is appended to produce class probabilities for the 4 classes present in the dataset. For compilation, the model employs the loss function which is sparse categorical loss, the optimizer which is Adam, and the accuracy metric to measure performance. To prevent overfitting, EarlyStopping is employed as a callback, which monitors the validation loss during training and stops the process if the loss does not improve for a consecutive 10 epochs. The model compiles with training and validation data. Training is done for 50 epochs with 32 batch sizes and shuffling the data. Following training, a classification report specific to the InceptionV3 model is generated to evaluate its performance. InceptionV3 addresses the vanishing gradient problem by utilizing an auxiliary classifier activation function, and it produces final predictions using the Softmax Activation Function. The mathematical model for InceptionV3 can be described in this manner

$$\text{inception}_{\text{model(x)}} = \text{softmax}\left(\text{Dense}\left(\text{Flatten}\left(\text{INCEPTIONV3}(x)\right)\right), 4\right) \quad (1)$$

where:

- x is the input tensor of shape (batch_size, 224, 224, 3) representing an RGB image with a height and width of 224 pixels and 3 color channels.
- Inception_model is the output tensor of shape (batch_size, 4) representing the predicted class probabilities for each of the 4 possible classes.

INCEPTIONV3(x) is the feature extraction convolutional neural network applied to the input tensor x.

- Flatten() is a layer that flattens the output tensor of INCEPTIONV3(x) to a one-dimensional tensor.
- Dense() is a fully connected layer that takes the flattened tensor as input and outputs a tensor of shape (batch_size, 4) using the softmax activation function

3.3.2. VGG16

The VGG16 architecture is named after its 16 weighted layers. When using the VGG16 function from the Keras applications module, a pre-trained VGG16 model is loaded with Imagenet dataset weights. To exclude the final layer of fully connected layers, which was originally designed for ImageNet's 1000 classes, the include_top parameter is set to False. The input_shape parameter specifies that the input images passed to the model should have a shape of (224, 224, 3). A new Flatten layer is added after the pre-trained VGG16 model's output to give the specific output. The softmax activation function is used for the particular 4 classes i.e. ALL, AML, CLL, and CML to give the probabilities for each class individually. For compilation, the model employs the loss function which is sparse categorical loss, the optimizer which is Adam, and the accuracy metric to measure performance. To monitor the validation loss during training and potentially stop early if it doesn't improve for 10 consecutive epochs, an EarlyStopping callback is created. The model compiles with training and validation data. Training is done for 50 epochs with 32 batch sizes and shuffling the data. After training, a classification report specific to the VGG16 model is generated to assess its performance. The overall structure of VGG16 can be summarized in these steps:

$$\text{VGG\_Model}(x) = \text{softmax}\left(\text{Dense}\left(\text{Flatten}\left(\text{VGG16}(x)\right)\right), 4\right) \quad (2)$$

where

- x is the input tensor of shape (batch_size, 224, 224, 3) representing an RGB image with a height and width of 224 pixels and 3 color channels.
- vgg_model is the output tensor of shape (batch_size, 4) representing the predicted class probabilities for each of the 4 possible classes.

VGG16(x) is the feature extraction convolutional neural network applied to the input tensor x.

- Flatten() is a layer that flattens the output tensor of VGG16(x) to a one-dimensional tensor.

- Dense() is a fully connected layer that takes the flattened tensor as input and outputs a tensor of shape (batch_size, 4) using the softmax activation function.

3.3.3. ResNet50

The 50 layers that makeup ResNet50's deep neural network design include 48 Convolution levels, 1 MaxPool layer, and 1 Average Pool layer. The model is pretrained on the ImageNet dataset and can be loaded using the ResNet50() function from the Keras library. By setting the weights parameter to 'imagenet', the pretrained weights are loaded. The top layer, which was originally used for classification, is excluded by setting the parameter include_top to False. The input_shape parameter is specified as (224, 224, 3). A for loop is used to set the trainable attribute of each layer to False to freeze the weights. A new fully connected output layer is then added to the ResNet50 model. The previous layer's output is flattened into a 1D array using the Flatten() layer. For multi-class classification with 4 classes, a Dense() layer with softmax activation function is used. The new model is defined using the Model() function, with inputs and outputs parameters set accordingly. For compilation, the model employs the loss function which is sparse categorical loss, the optimizer which is Adam, and the accuracy metric to measure performance. An EarlyStopping callback is implemented to monitor the validation loss and stop training if it doesn't improve for 10 consecutive epochs. The model compiles with training and validation data. Training is done for 50 epochs with 32 batch sizes and shuffling the data. ResNet50 uses skip connections to add the output of one layer to the next, which lessens the complexity of time. For the same output feature map size, it is a 34-layer network with the same number of filters, and when the feature map size is cut in half, the number of filters increases by two. The structure of ResNet50 can be described as follows:

$$\text{ResNet\_Model}(x) = \text{softmax}\left(\text{Dense}\left(\text{Flatten}\left(\text{RESNET50}(x)\right)\right), 4\right) \quad \textbf{(3)}$$

where:

- x is the input tensor of shape (batch_size, 224, 224, 3) representing an RGB image with a height and width of 224 pixels and 3 color channels.
- resnet_model is the output tensor of shape (batch_size, 4) representing the predicted class probabilities for each of the 4 possible classes.

RESNET50(x) is the feature extraction backbone of RESNET50 applied to input tensor x.

- Flatten() is a layer that flattens the output tensor of RESNET50(x) to a one-dimensional tensor.
- Dense() is a fully connected layer that takes the flattened tensor as input and outputs a tensor of shape (batch_size, 4) using the softmax activation function.

3.4. Classification

Distinguish between different types of leukemia using a fully connected network that uses the softmax activation function. The Dense layer receives input from the previous layers, which is obtained after flattening a two-dimensional matrix into a one-dimensional matrix. We utilize the Softmax Activation Function to obtain probabilities for the four input classes: ALL, AML, CLL, and CML. The category to which the image belongs will be determined based on which class has the highest likelihood.

3.5. Proposed Algorithm

The algorithm - pseudocode written below gives the overall idea of the proposed work explaining the control abstraction for our work. The algorithm contains some functions explaining the exact working of the procedure in detail. It takes the image dataset as input, providing us with the results having a particular class label assigned to it. The algorithm is discussed as below:

Algorithm LeukemiaDetection(ImageDataset){

// Description: Prediction of Leukemia Disease Types as ALL, AML, CLL, CML

// Input ← ImageDataset and output = NULL

Import libraries()

for i ← 1 to dataset_size do

// Split Dataset into training, testing and validation

train_data ← splitted train data (70%)

test_data ← splitted test data (10%)

validation_data ← splitted validation data (20%)

end

function DataPreparation(training_Images){

Image rescaling()

}

function ImageProcessing(Preprocessed_Image){

//Perform Data Augmentation for balancing the number of images in each classes

Data Augmentation()

}

function Feature_Extraction(Processed_Images){

Fit_model ← Model_name(weight = 'imagenet', top_layer ← False, Shape)

Features ← (input_shape == conv_base_layer)

Labels ← ('ALL', 'AML', 'CLL', 'CML')

```
Generator ← (Images, class_mode = 'categorical')

        i ← 0
        for Inputs,Labels in Generator do
                Inputs = conv_base.predict()
                Features = Input + Labels
                i ← i + 1
                feature_vector = Features + Labels
                return feature_vector
                end
        }
function classification(feature_vector){
 // Build the model
        model.add()
// Compile and Train the model
        model.compile()
         model.fit()
        Output ← Model.predict(Features)
        return Output
        }
end
```

4. Results and Discussion

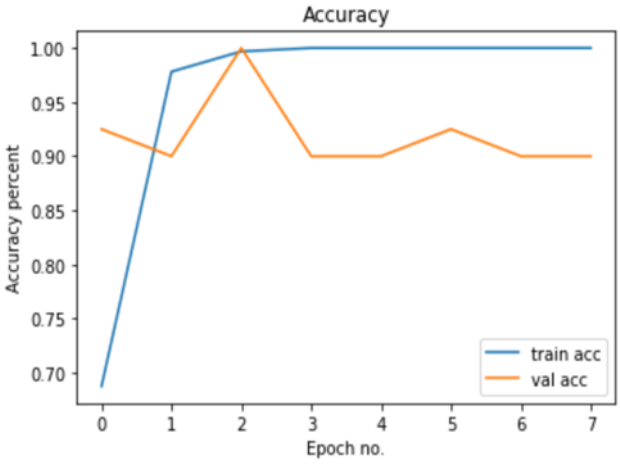
Ensemble learning involves the utilization of multiple models in machine learning to collectively enhance the performance of a predictive model. Instead of relying on a single model to make predictions, ensemble learning generates more accurate and reliable predictions. Ensemble learning has been shown to improve the accuracy, stability, and generalization of predictive models, especially in complex and noisy datasets. An ensemble model is built by stacking the individual model in the list, then averaging the outputs from each model. Table 1 depicts the overall analysis of each model with its accuracy

Table 1 Model with its Accuracy

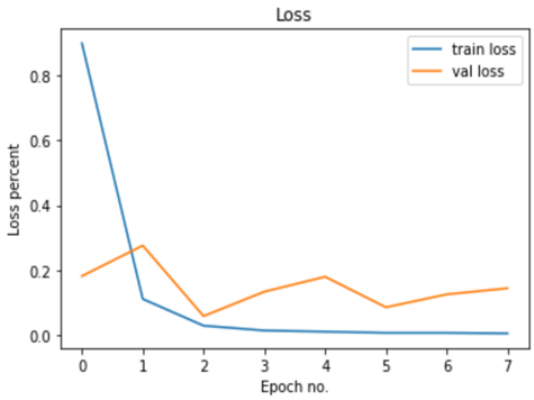
Model Name	**MODEL**	**Accuracy**
E1	VGG16	98%
E2	ResNet50	90%
E3	InceptionV3	92.5%
E4	InceptionV3+VGG16+Resnet50	99.8%
E5	VGG16+ResNet50	25%
E6	InceptionV3+VGG16	93%
E7	Resnet50+InceptionV3	91%

From table 1, it has been clear that the ensemble of VGG16, Resnet50, and InceptionV3 gives the highest 99.8 % accuracy. Whereas VGG16 and Resnet50 give the lowest 25% accuracy. Among all the individual models, VGG16 gives 98% highest accuracy. It has been noticed that the Ensemble of three models gives the highest accuracy and takes the maximum time for execution whereas individual models comparatively give less accuracy in faster execution time.

VGG16: The VGG16 model trained for 8 epochs calling early checkpoints and it is giving training and validation accuracy between 90 to 100%. Validation loss is higher than the training loss. The overall test accuracy is 98%. This is shown in figure 4.



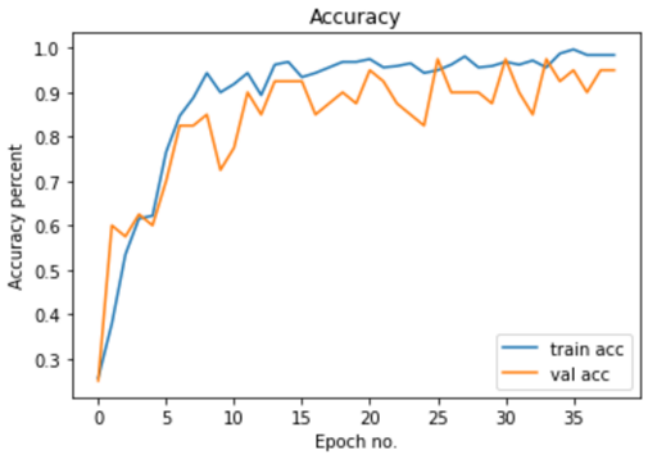
(a)



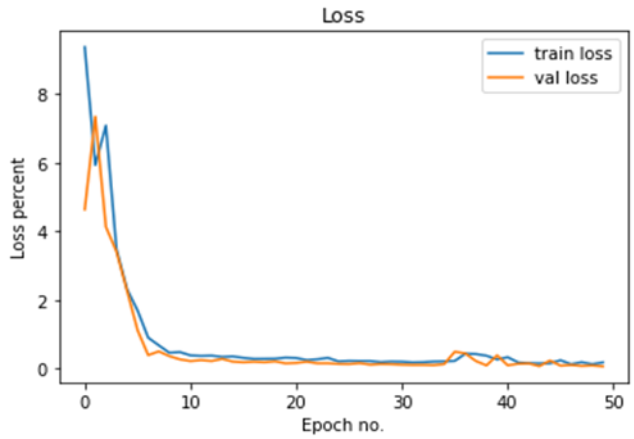
(b)

Fig 4 (a) Accuracy (b) Loss Graph of VGG16

ResNet50: The Resnet50 model trained for calling early checkpoints. It is giving training and validation accuracy between 70 to 95%. Training loss is slightly higher than the validation loss and is between 0 to 2%. The overall testing accuracy is 90%. Figure 5 shows the graph of the accuracy and loss of Resnet50



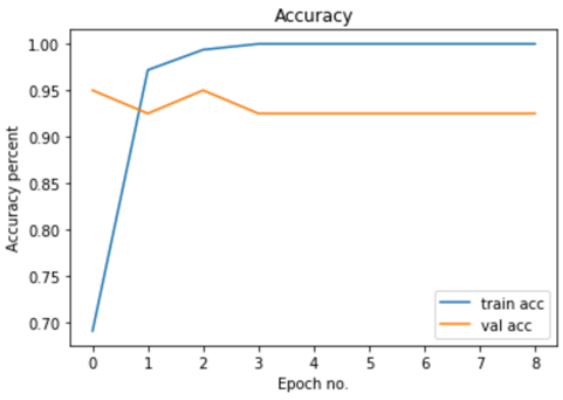
(a)



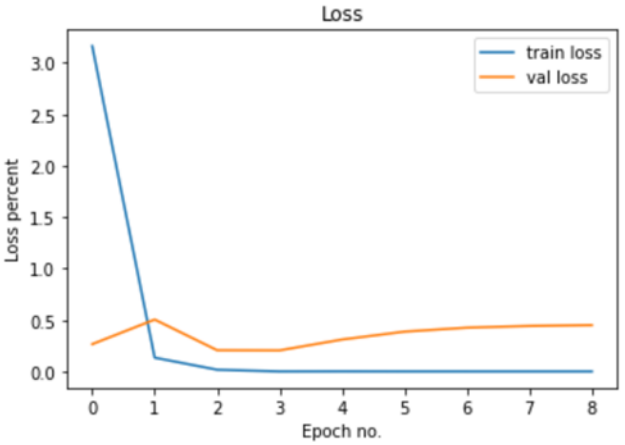
(b)

Fig. 5. a) Accuracy (b) Loss Graph of ResNet50

InceptionV3: The InceptionV3 model is trained for 6 epochs calling early checkpoint. It is giving training and validation accuracy between 95 to 100%. Validation loss is slightly higher than training loss. The overall testing accuracy is 92.5%. Figure 6 depicts the accuracy and loss graph.



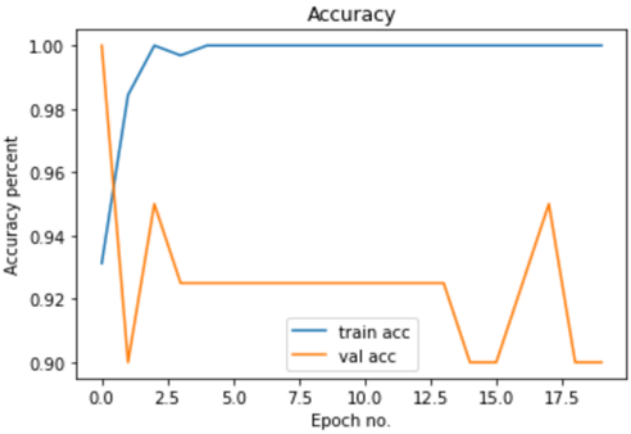
(a)



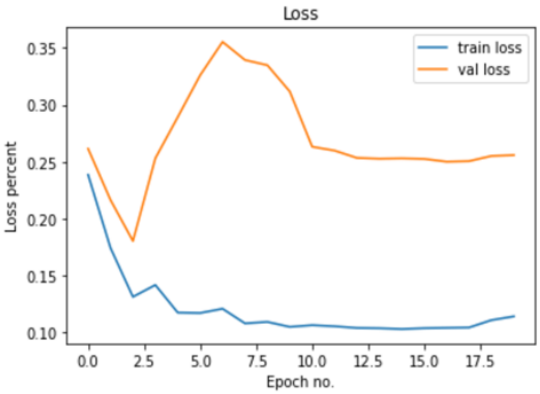
(b)

Fig. 6. (a) Accuracy (b) Loss Graph of InceptionV3

The ensemble model: InceptionV3, VGG16, and Resnet50 gives train and test accuracy of about 92 to 100%. The train and validation loss are minimum and lies between 1 to 3%. The accuracy and loss graph of the Ensemble model is shown in



(a)

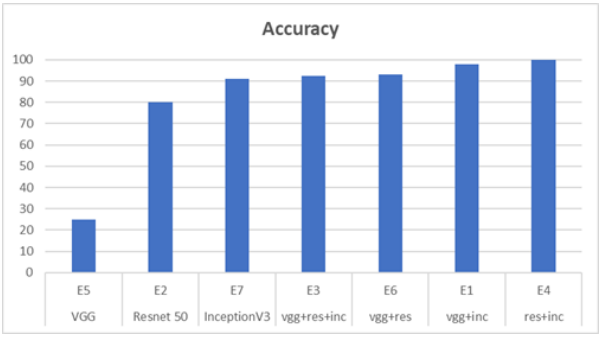


(b)

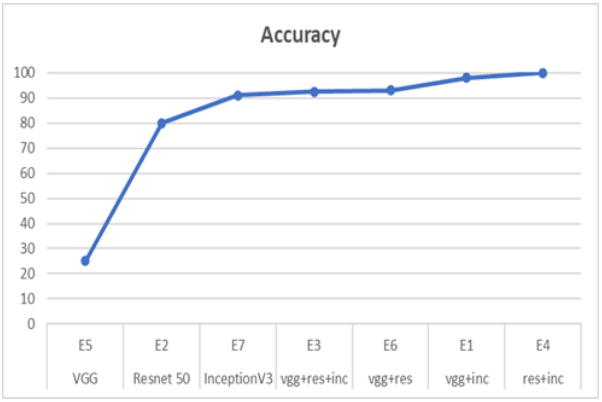
Fig. 7. (a) Accuracy (b) Loss Graph of Ensemble Model

From Table 1 it has been cleared that the E4 model gives the highest accuracy and E5 gives the lowest accuracy. Figure

8 shows the simple graphical representation for the accuracies of each model.



(a)



(b)

Fig. 8. Graphical Representation for Model Accuracies (a) and (b).

5. Conclusion and Future Work

Using pre-learned models like Resnet50, VGG16, and InceptionV3, as well as an ensemble of these models, we have completed leukemia kinds detection. The accuracies are discussed with the accuracy and loss graph for each model. Considering this, we have performed some performance analysis and used the ensemble model gives us the most appropriate results with higher 99.98% accuracy. In this model, to speed up training, it employs a method known as 'batch normalization'. Batch normalization is a technique that speeds up network training by normalizing the inputs to each layer. It has been observed that VGG16 and Resnet50 give a minimum of 25% accuracy. The ensemble models take quite a lot of execution time but have more accuracy as compared with the individual models.

Further in the future, we can work with a big and live dataset collected directly from hospitals. The accuracy of the models might be increased by providing them with a huge dataset.

Conflicts of interest

There won't have any financial or personal relationships that may influence the research or interpretation of data in this manuscript. This is article based on sample data collected from different sources. If there are no conflicts of interest, I am declaring that no conflict of interest in any financial and personal"

References

[1] Aftab M O, Javed Awan M, Khalid S, Javed R, Shabir H. Executing Spark BigDL for Leukemia Detection from Microscopic Images using Transfer Learning. 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA). 2021 (pp. 216-220)

[2] Ahmed, Yigit, Isik, Alpkoçak. Identification of Leukemia Subtypes from Microscopic Images Using Convolutional Neural Network. Diagnostics. 2019; 9(3):104.

[3] Maryam Bukhari, Sadaf Yasmin, Saima Sammad, Ahmed A. Abd El-Latif. A Deep Learning Framework for Leukemia Cancer Detection in Microscopic Blood Samples Using Squeeze and Excitation Learning. Mathematical Problems in Engineering. 2022; 1-18.

[4] K K Anilkumar, V J Manoj, T M Sagi. Automated detection of leukemia by pre-trained deep neural networks and transfer learning: A comparison. Medical Engineering & Physics. 2021; 98(3):8-19.

[5] Luis H S Vogado, Rodrigo M S Veras, Flavio H D Araujo, Romuere R V Silva, Kelson R T Aires. Leukemia diagnosis in blood slides using transfer learning in CNNs and SVM for classification. Engineering Applications of Artificial Intelligence. 2018; 72: 415-422.

[6] Mohamed Loey, Mukdad Naman, Hala Zayed. Deep Transfer Learning in Diagnosing Leukemia in Blood Cells. Computers. 2020; 9(2):29.

[7] Nighat Bibi, Misba Sikandar, Ikram Ud Din, Ahmad Almogren, Sikandar Ali. IoMT-Based Automated Detection and Classification of Leukemia Using Deep Learning. Journal of Healthcare Engineering. 2020; 2020(1):1-12.

[8] Raheel Baig, Abdur Rehman, Abdullah Almuhaimeed, Abdulkareem Alzahrani, Hafiz Tayyab Rauf. Detecting Malignant Leukemia Cells Using Microscopic Blood Smear Images: A Deep Learning Approach. Applied Sciences. 2022; 12(13):6317.

[9] Chayan Mondal, Md Kamrul Hasan, Mohiuddin Ahmad, Md Abdul Awal, Md Tasnim Jawad, Aishwariya Dutta, Md Rabiul Islam, Mohammad Ali Moni. Ensemble of Convolutional Neural Networks to diagnose Acute Lymphoblastic Leukemia from microscopic images. Informatics in Medicine Unlocked. 2021; 27(12):100794.

[10] Sultana, S., Sultana, S., & Tariqul Islam, A. F. M.

(2021). Detection of Leukemia in Microscopic Blood Images Using a Pre-Trained Convolutional Neural Network. In 2021 IEEE International Conference on Engineering Research & Practice (ICERP) (pp. 1-5). IEEE.

[11] Vogado, L. H. S., Veras, R. D. M. S., Andrade, A. R., Araujo, F. H. D. D., Silva, R. R. V. e, & Aires, K. R. T. (2017). Diagnosing Leukemia in Blood Smear Images Using an Ensemble of Classifiers and Pre-Trained Convolutional Neural Networks. 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). doi:10.1109/sibgrapi.2017.55

[12] Gupta, R., Singh, R., & Kaur, H. (2018). Automated detection of leukemia using pre-trained convolutional neural networks and an ensemble of classifiers. Computer methods and programs in biomedicine, 166, 99-108.

[13] Kumar, A., Mishra, S., & Mishra, A. (2019). Automated detection of leukemia using an ensemble of classifiers and pre-trained CNN. In 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU) (pp. 1-4). IEEE.

[14] Wang, J., Li, Y., Li, H., & Cheng, J. (2020). Automated detection of leukemia in blood smear images using pre-trained convolutional neural networks and ensemble learning. Journal of medical systems, 44(3), 1-9.

[15] Wang, S., Peng, C., Wang, C., Huang, Q., Liu, X., Zhang, L., & Chen, J. (2021). Automated detection of leukemia in blood smear images using ensemble of pre-trained convolutional neural networks. Computer Methods and Programs in Biomedicine, 205, 106093.

[16] Gunda, S., Nagisetty, V. K., & Rao, P. V. (2019). Deep Learning Framework for Detection of Leukemia Using Blood Smear Images. In 2019 IEEE 5th International Conference on Computational Intelligence and Applications (ICCIA) (pp. 1-6). IEEE.

[17] Alkhateeb, H., & Al-Betar, M. A. (2020). Acute Myeloid Leukemia Detection Using Ensemble of Pre-Trained Deep Convolutional Neural Networks. Journal of Medical Systems, 44(5), 98.

[18] Akter, S., Hossain, M. S., & Rahman, A. (2019). Leukemia Detection Using Pre-Trained Convolutional Neural Networks and Transfer Learning. In 2019 IEEE International Conference on Imaging Systems and Techniques (IST) (pp. 1-6). IEEE.

[19] Biswas, K., Mukherjee, S., & Chakraborty, C. (2021). Ensemble of Convolutional Neural Networks for Leukemia Detection in Blood Microscopic Images. International Journal of Image, Graphics and Signal Processing (IJIGSP), 13(4), 1-16.

[20] S Ramaneswaran, Kathiravan Srinivasan, P M Durai Raj Vincent, Chuan-Yu Chang. Hybrid Inception v3 XGBoost Model for Acute Lymphoblastic Leukemia Classification. Computational and Mathematical Methods in Medicine. 2021; 2021:2577375.

[21] Chiaretti S, Zini G, Bassan R. Diagnosis and Subclassification of Acute Lymphoblastic Leukemia. Mediterranean Journal of Hematology and Infectious Diseases. 2014; 6(1):e2014073.

[22] Narjes Ghane, Alireza Vard, Ardeshir Talebi, Pardis Nematollahy. Classification of chronic myeloid leukemia cell subtypes based on microscopic image analysis. EXCLI Journal. 2019; 18:382-404.

[23] Vakiti A, Mewawalla P. Acute Myeloid Leukemia. Treasure Island (FL): StatPearls Publishing. 2021.

[24] V A Ostapenko, N G Kruchinskiĭ, L A Smirnova, A B Cherednik, V N Nesterov, A I Tepliakov. Morphometric analysis of lymphocyte nuclei in chronic lymphocytic leukemia. Gematol Transfuziol. 1994; 39(4):15-7.

[25] Özcan, Şeyma Nur, Tansel Uyar, and Gökay Karayeğen. "Comprehensive Data Analysis of White Blood Cells with Classification and Segmentation by Using Deep Learning Approaches." Available at SSRN 4604383.Raabin Leukemia Data from https://raabindata.com/free-data/

[26] Bhute, Avinash, Rakesh S. Patil, Shreyas Sanghai, Prajakta Yadav, and Sanketkumar Sathe. "Functional Electrical Stimulation for Hand Movement of Person." In Proceedings of International Conference on Machine Intelligence and Data Science Applications: MIDAS 2020, pp. 465-481. Springer Singapore, 2021.

[27] Choudhary, Manisha, Monika Lokhande, Rushikesh Borse, and Avinash Bhute. "A machine learning approach to aid paralysis patients using EMG signals." In Advanced Data Mining Tools and Methods for Social Computing, pp. 107-125. Academic Press, 2022.

[28] Thorat, Adhiksha, and Avinash Bhute. "SquashCord: Video Conferencing Application Using WebRTC." In High Performance Computing and Networking: Select Proceedings of CHSN 2021, pp. 425-436. Singapore: Springer Singapore, 2022.

[29] Sangle, Priyanka S., R. M. Goudar, and A. N. Bhute. "Methodologies and Techniques for Heart Disease Classification and Prediction." In 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1-6. IEEE, 2020.