IJCRT.ORG ISSN : 2320-2882
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

REVOLUTIONIZING BLOOD CELL ANALYSIS USING MACHINE LEARNING

R. Prasanna Kumari
Assistant Professor, Information Technology, GVPCEW
Visakhapatnam, Andhra Pradesh, India
Prassuna1@gmail.com

J. Pramitha Sri
Information Technology, GVPCEW
Visakhapatnam, Andhra Pradesh, India
pramithasri.jetti@gmail.com

G. Vaishtnavi
Information Technology, GVPCEW
Visakhapatnam, Andhra Pradesh, India
vaishtnavigunapu@gmail.com

R. Gowthami
Information Technology, GVPCEW
Visakhapatnam, Andhra Pradesh, India
21jg5a1205.gowthami@gvpcew.ac.in

G. Keerthana
Information Technology, GVPCEW
Visakhapatnam, Andhra Pradesh, India
keerthanakeerthy2222@gmail.com

Abstract- In this Research Blood cells help us to fight infections by attacking bacteria, viruses, and germs that invade the body. White blood cells originate in the bone marrow but circulate throughout the bloodstream, while red blood cells help transport oxygen to our body. Accurate counting of those may require laboratory testing procedure that is not usual to everyone. Generating codes that will help counting of blood cells that produce accurate response via images gives a relief on this problem. The suggested method includes preprocessing blood cell pictures, feature extraction and training machine learning models for cell detection and counting. A variety of advanced classification techniques are used they are Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN) are used to evaluate the shape of cells and differentiate between various cell types. To guarantee accurate results in a wide range of applications the model is trained on a Blood Cell Images dataset which includes different blood cell types and conditions.

Keywords: RBC, WBC, Blood cell Detection, Blood cell counting, Machine Learning.

I.INTRODUCTION

Blood Cells are also known as Hematopoietic cells, Hemocytes, or Hematocytes. They are formed through the process of Hematopoiesis. Red Blood Cells(erythrocytes), White Blood Cells(leukocytes), and platelets(thrombocytes) are the three main types of Blood Cells. These three types of blood cells together make up 45% of the volume of blood tissue plasma, the liquid component of blood, makes up the remaining 55% of the volume. By mainly focusing on the White Blood cells, there are of different types such as Neutrophils(Kill bacteria, fungi and foreign debris), Monocytes(Clean up damaged cells), Eosinophils(Kill parasites, cancer cells and involved in allergic response), Lymphocytes(Help fight viruses and make antibodies).Thus in this model, we want to classify and count the blood cells to which class they belongs to .The dataset named "Blood Cell Images" Dataset is used which contains the 4 classes of different types of White Blood cells. The "Blood Cell Images" Dataset is collected and trained on model for classifying and counting the Blood Cells. The proposed CNN model is trained and classifies into 4 classes (Eosinophils, Monocyte, Lymphocyte, Neutrophils). A Microscopic Image is given to the model to classify the class and count the number of cells in each class. The CNN model extracts important features from the input test images to classify into 4 classes.

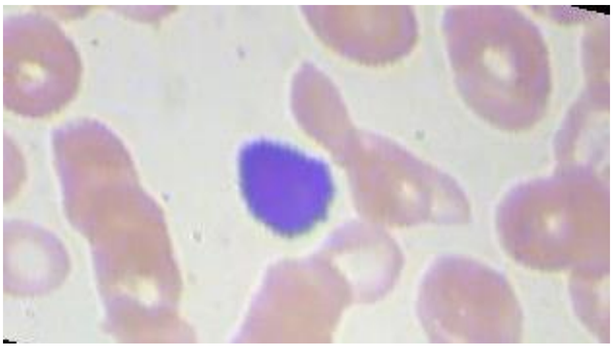

Fig 1: Microscopic Image of Blood Cell

II.LITERATURE SURVEY

In medical diagnostics and research, blood cell identification and counting are essential procedures that are necessary for determining a person's overall health and making a variety of disease diagnoses. It takes a lot of time, work, and human error to analyze blood cells manually using traditional methods therefore, automated solutions have to be developed for accurate and efficient analysis. The development of automated blood cell detection and counting systems has been made possible in recent years by developments in digital imaging technology, as well as the quick development of machine learning and computer vision techniques.

[1] In the paper "Blood Cell Detection Method Based on Improved YOLOv5" written by Yecai Guo, Andmengyaozhang. The authors worked on multiple techniques YOLO, YOLOv5.They introduced the attention mechanism in the feature channel, modifying SPP module in YOLOv5, in their existing system.

[2] In the paper "A Machine Learning Approach of Automatic Identification and Counting of Blood Cells" written by Mohammad Mahmudul Alam1 and Mohammad Tariqul Islam. The authors worked on machine learning approach for automatic identification and counting of three types of blood cells using 'You only look once'(YOLO) object detection and classification algorithm.

[3] In the paper "Complete Blood Cell Detection and Counting Based on Deep Neural Networks" written by Shin-Jye Lee, Pei-Yun Chen, Jeng-Wei Lin. The authors worked on deep neural network-based architecture to accurately detect and count blood cells on blood smear images.

[4] In the paper "An efficient algorithm for segregation of white and red blood cells based on modified hough transform" written by S. Seth and K. Palodhi. The authors worked on modified hough transform known as hough transform filters. Using these filters, simultaneous shape recognition of circles of different radii becomes much easier without having to change image parameters using thresholding or contrast enhancement etc. for each step.

[5] In the paper "Automatic identifying and counting blood cells in smear images by using single shot detector and Taguchi method" written by Yao-Mei Chen, Jinn-Tsong Tsai and Wen-Hsien Ho. The authors worked on algorithm Resnet50. It is the backbone network of the SSD with its optimized algorithm hyperparameters, which is called the Resnet50-SSD model.

III.METHODOLOGY

The dataset we used for this problem statement is "Blood Cell Images". Then we preprocessed the images like rotation, rescaling and gray conversion. This dataset moves into model for training and then tested on test data. We also assess the performance of the model by using evaluation metrics.

The fig2. represents the process

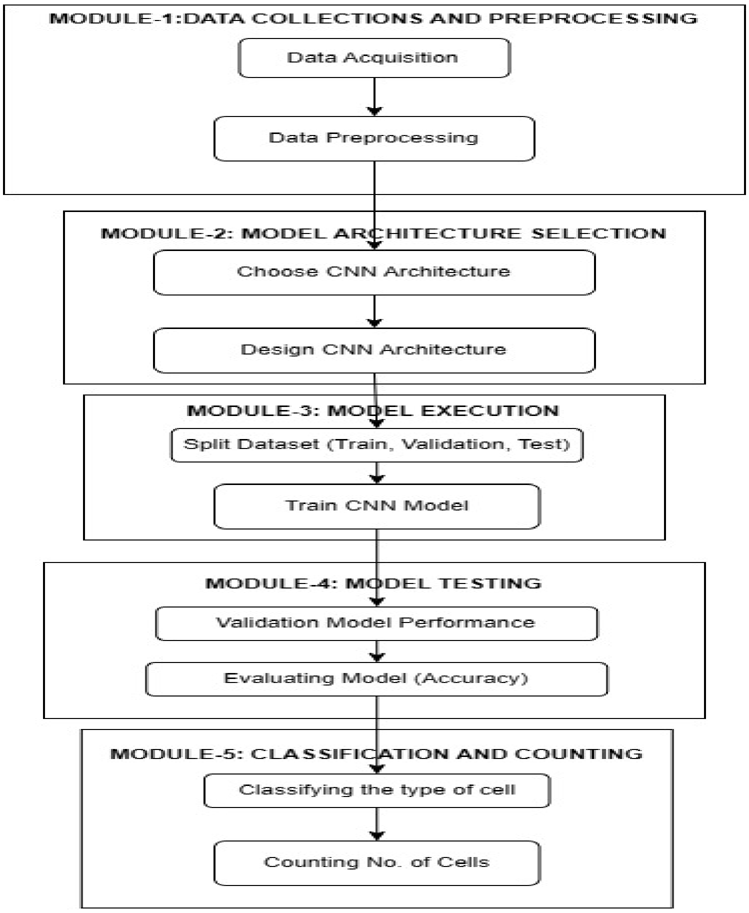

Fig 2:Data Flow Of Proposed system

1.Data Acquisition:

We have used the dataset named as "Blood Cell Images". The dataset is made up of blood cell images divided into 4 classes they are Eosinophils, Monocyte, Lymphocyte, Neutrophil containing 12500 images, having approximately 3000 images from each class. The images are represented in JPEG format.

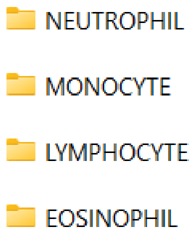

Fig 3: Dataset classes

2.Data Preprocessing:

Data Preprocessing is the process of converting raw data into format suitable for machine learning algorithms. It is done to improve the quality and enhance the performance of data. In the context of preprocessing, resize refers to converting all the images into same size. By normalization, the categorical data is transformed into numerical format and grayscale turns the image into gray color. This process prepares the input data for training.

3.Model Architecture Selection:

Selection architecture for our model plays a vital role. Depending on the complexity of the problem and size of the dataset architecture's will be considered. Design the architecture to accept microscopic images as input and classify the classes and their count as output. We choose CNN as our architecture. Convolutional Neural Networks are good at image classification tasks as feature extraction is done through series of convolutional layers with activation functions.

4.Model Building using CNN:

To build a Convolutional Neural Network (CNN), first import libraries like TensorFlow or PyTorch. Then load and preprocess your image dataset. Define the CNN architecture by stacking convolutional layers with activation functions (like ReLU), pooling layers for down-sampling and possibly dropout layers for regularization. Compile the model with a loss function and optimizer, then train it on the training data, monitoring performance with validation data. Evaluate the trained model on test data and finally, deploy the model for inference on new data.

A. Importing library – "**TensorFlow**"

TensorFlow is a free and open-source machine learning framework developed by Google Brain Team. It is used to build and train deep learning models. TensorFlow offers flexibility, scalability for performing tasks like image classification, natural language processing and reinforcement learning. To import TensorFlow library, simply write a command "import tensor flow as tf".

B. Input Layer

Input Layer is the first Layer of CNN model. This layer serves as entry point for the dataset. In case of images they turn out into pixel values and enters. The input layer comprises of images dimensions, such as height, width, and number of color channels. Fixed range of 120x120 RGB pixels are taken as inputs.

C. Convolutional Layer

The Conv2D layer is a fundamental building block in Convolutional Neural Networks (CNNs) for processing two-dimensional inputs, such as images. It performs convolutional operations on the input data, applying a set of filters to extract features. This layer performs a convolution operation on the input data, using a set of learnable filters to extract features. As the filters slide over the input image, they compute the dot product with overlapping region of the input at each position, generating a feature map.

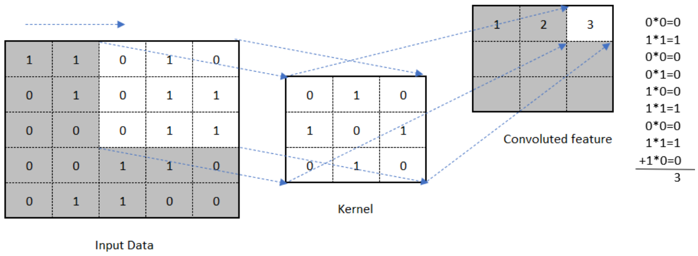

Fig 4: Convolutional Operation

D. Pooling layer

Pooling layer is responsible for decreasing the dimensions of feature maps. It extracts the important features from numerous features maps where it performs dimensionality reduction. Moving forward, the operations will be held on summed-up features rather than precisely positioned features from convolutional Layer. We considered max pooling to extract maximum value from feature maps. So it reduces the dimension of feature maps.

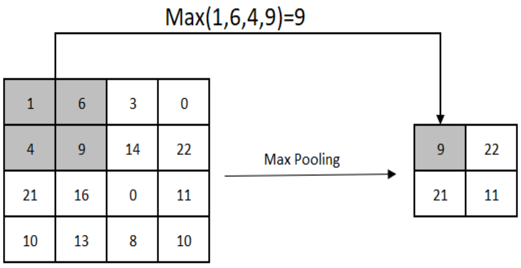

Fig 5: Max Pooling Operation

E. Fully Connected Layer

Fully connected layer connects every neuron to other neuron present in the layer allowing the network to learn complex patterns. Fully connected layers works in conjunction with convolutional layer for image classification tasks, where they combine the features learned by preceding layers.

F. Activation Function

Activation Functions are the mathematical operations applied over the neurons in the neural networks including CNN's, introducing non-linearity to the model for extracting complex patterns from the images. ReLU is commonly used but also many other depending upon the problem.

G. ReLU -Activation Function

ReLU, short for Rectified Linear Unit, is one of the most commonly used activation functions in neural networks, including Convolutional Neural Networks (CNNs). It's defined as f(x)=max(0,x), which means it outputs the input value if it's positive and zero otherwise.

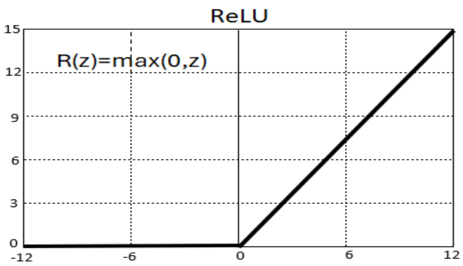

Fig 6: ReLU- Activation Function

H. Sigmoid - Activation Function

Sigmoid is an activation function commonly used in neural networks, particularly in binary classification tasks. It squashes the output of a neuron to the range [0, 1], making it suitable for tasks where the output needs to be interpreted as a probability. Mathematically, the sigmoid function is defined as $f(x)=1/(1+e^{-x})$,where x represents the input to function.

:

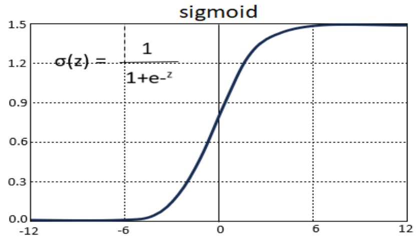

Fig7: Sigmoid-activation function

I.CNN Architecture:

We have designed the CNN model in a way that is comprised of convolutional layers followed up by pooling layers which is sequential that gradually decrease the spatial dimension of feature maps. It begins with a max-pooling 2x2 window and a 3x3 convolutional layer with 16 filters using ReLU activation. Max-pooling layers similarly come after convolutional layers 32,64, with increasing filter counts. For dense layers, the tensor is reshaped by a flatten layer after the convolutional stack. Dropout layers are combined with dense layers that have 128 and 4 units, respectively, and use ReLU activation for regularization. Lastly, sigmoid-activated neurons are used for classification.

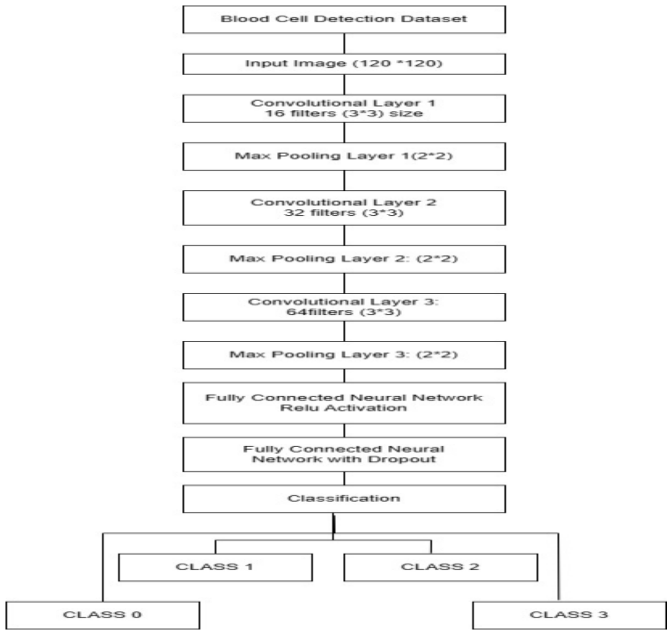

Fig 8: CNN Architecture

5.Model Execution:

This phase involves in testing of the trained CNN model for classification of classes and counting. After CNN trained and parameters are optimized, it is ready for deployment. Then it classifies the classes based on the characteristics of images. The model execution phase comprises feeding new data samples into the trained CNN and reviewing the output predictions to evaluate the accuracy, sensitivity, specificity, and other performance metrics of the model.

6.Model testing and Evaluation:

When evaluating the effectiveness of a convolutional neural network (CNN) intended for blood cell classification and counting, model testing is a crucial stage. This stage involves measuring the accuracy, sensitivity, specificity, precision, and F1-score of the trained CNN using a “Blood Cell Images” dataset. Preprocessing is done on input data before it is entered into the prediction model. The model testing and evaluation is done separately.

Formulas of evaluation metrics:

$$Accuracy: \quad \frac{TP + TN}{TP + TN + FP + FN} \qquad (1)$$

$$Precision: \quad \frac{TP}{TP + FP} \qquad (2)$$

$$F1\text{-}score: \quad \frac{2TP}{2TP + FP + FN} \qquad (3)$$

$$Recall: \quad \frac{TP}{TP+FN} \qquad (4)$$

	precision	recall	f1-score	support
EOSINOPHIL (Class 0)	0.70	0.71	0.70	304
LYMPHOCYTE (Class 1)	0.99	0.97	0.98	320
MONOCYTE (Class 2)	0.95	0.93	0.94	314
NEUTROPHIL (Class 3)	0.73	0.75	0.74	306
accuracy			0.84	1244
macro avg	0.84	0.84	0.84	1244
weighted avg	0.84	0.84	0.84	1244

IV. SYSTEM ARCHITECTURE

The suggested system design enhances gathering images data relevant to Blood cells, preprocessing it via multiple convolutional layers, and then categorizing the classification of classes types based on features like colour, shape, size. Users provide input data, such as blood cell images to classify them. Various techniques are utilized to improve the quality and diversity of the data, such as resizing, rescaling, flipping, and rotation. The input data travels through multiple layers of CNN model for feature extraction i.e patterns, edges etc. This CNN model classifies the input image into designated class which it belongs to.

Fig 9 represents how the proposed system works and flow of the input image through layers for classification and counting of blood cells.

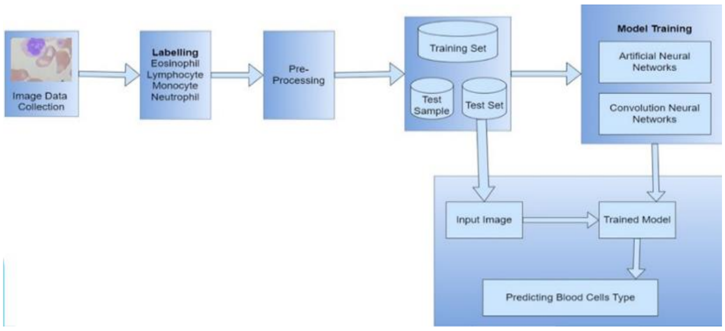

Fig 9: System Architecture

V.IMPLEMENTATION DETAILS

A. Python

We have developed this system using python. To work with ML models, python is efficient. Python offers wide range of libraries like numpy, keras and pandas. The platform we used is Visual Studio.

B. Visual Studio

Visual Studio is a powerful developer tool that you can use to complete the entire development cycle in one place. It is a comprehensive integrated development environment (IDE) that you can use to write, edit, debug, and build code, and then deploy your app.

VI.RESULTS

Convolutional Neural Networks:

We gained a accuracy of 89% by training the model with 25 epochs using the images from Blood Cell Images dataset.

The fig.10 represents the visualisation of train and validation set accuracy by plotting over accuracy and epochs.

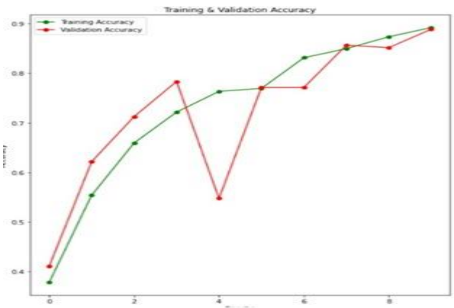

Fig 10. Accuracy of CNN

The below Fig.11 represents the confusion matrix where the predictions are done on test set. The no. of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions that the model made is displayed in the confusion matrix for classes.

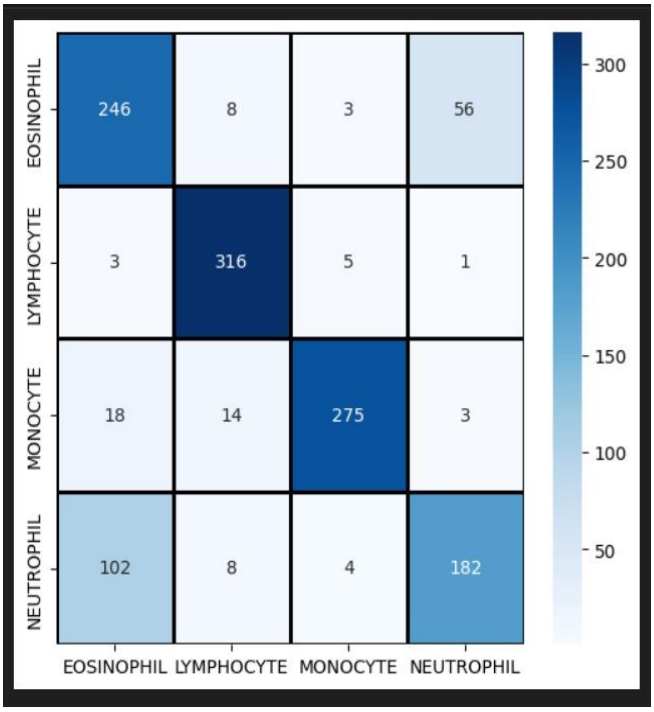

Fig.11 Confusion Matrix for classes

VII. CONCLUSION

The project aims to develop a robust method for

Machine learning techniques, primarily Convolutional Neural Networks (CNNs) and Artificial Neural Networks (ANNs). By using preprocessing blood cell images and extracting features, the models are trained on a diverse dataset to accurately differentiate between various cell types. The implementation involves a multi-step process including preprocessing, feature extraction, model training, deployment, validation, and evaluation. Results demonstrate improved accuracy in identifying blood cells, offering promising prospects for applications in medical diagnosis and healthcare. The project underscores the potential of machine learning in enhancing blood cell analysis, contributing to advancements in medical technology and patient care.

VIII .REFERENCES

[1] YECAI GUO, ANDMENGYAOZHANG - Blood Cell Detection Method Based on Improved YOLOv5 National Natural Science Foundation of China, (VOLUME 11, 2023 issued: 6 July 2023)

[2] Mohammad Mahmudul Alam1 and Mohammad Tariqul Islam- A Machine Learning Approach of Automatic Identification and Counting of Blood Cells, The Institution of Engineering and Technology (Volume: 11:54:32 issued: 2018/10/20)

[3] Shin-Jye Lee, Pei-Yun Chen, Jeng-Wei Lin- Complete Blood Cell Detection and Counting Based on Deep Neural Networks, Institute of Management of Technology, National Yang Ming Chiao Tung University (Volume: 12(16) issued: 14 August 2022)

[4] Gligoroska J Pluncevic, S Gontarev, B Dejanova, L Todor-ovska, D Shukova Stojmanova and S Manchevska, "Red Blood Cell Variables in Children and Adolescents regarding the Age and Sex", Iran J Public Health (vol. 48, no. 4, pp. 704-712, Apr 2019.)

[5] S. Seth and K. Palodhi, "An efficient algorithm for segregation of white and red blood cells based on modified hough transform", 2017 IEEE Calcutta Conference (CAL-CON), pp. 465-468, 2017.