

# Residual Block-Driven CNN for Accurate White Blood Cell Image Analysis and Classification

Antu Roy Chowdhury <sup>1</sup>, Sadman Hasib Emon <sup>2</sup> Md. Abu Ismail Siddique <sup>3</sup>, and Saraf Anzum Shreya <sup>4</sup>

<sup>1,2,3,4</sup>Electronics & Telecommunication Engineering

Rajshahi University of Engineering & Technology

Rajshahi, Bangladesh

Email: anturoychowdhury3@gmail.com, sadmanhasib21@gmail.com, saif101303@gmail.com, sasshreya2001@gmail.com

**Abstract**—White blood cells are crucial for our immune system because they offer protection against the outer body substances and other hostile organisms. WBC serves different purposes in the body's overall immune response. All sorts of precise WBC counts are frequently used in diagnostics to check the proper proportions of different WBC types and help identify possible health problems. Proper diagnosis of WBCs is essential for diagnosing various diseases. This present study shows an optimized road map using residual block-driven convolutional neural network(CNN) architecture to improve WBC classification. Our method of detection of WBC improves the efficiency of pre-train models to make the training fast while proper extraction of detailed features. This proposed method exceeds traditional ways by fine-tuning the network and implementing residual blocks. This amended framework suggests the potential to enhance the diagnostic accuracy of 99.20% for ResNet50 and efficiency in WBC analysis, giving a robust tool for medical diagnostics.

**Index Terms**—WBC, Residual Block enhanced CNN, Xception-Net, InceptionNetV3, MobileNet, ResNet50, Medical Imaging..

## I. INTRODUCTION

A significant issue in the world of health is an accurate assessment of white blood cell (WBC) detection; this affects everything from monitoring a person's immune system to recognizing early signs of illness. WBC levels are critical for diagnosing health issues. Proper WBC numbers can be linked to autoimmune and inflammatory illnesses, infections, cancers including leukaemia and Hodgkin's disease, and allergic reactions [1]. Counting the number of WBCs can help identify infections, inflammation, or autoimmune diseases. These infections may include fever, chills, body aches, headaches, and red incisions. Doctors can detect and treat underlying health issues earlier, leading to more effective medication. If a WBC count is found to be high, which is known as leukocytosis, more testing may be required to determine the underlying cause. This may demand blood tests to detect infections, inflammation, or other conditions. In some cases, further imaging studies or biopsies may be necessary. If a low WBC (leukopenia), it is critical to investigate the possible causes. This might include tests to check immune system activity, screening for infections, or looking for bone marrow abnormalities.

Conventional white blood cell categorization methods are sometimes liable to mistakes. Computer vision algorithms enable doctors to monitor patients by interpreting data from smart devices and sensors that detect vital indicators. This

helps doctors to discover trends and abnormalities and take relevant action fast. In WBC categorisation, Deep learning-based systems have handled more accurately and [2] significantly with improved accuracy and efficiency. This has minimised the amount of manual labour and time spent on medical image classification jobs. This paper introduces an efficient methodology for Optimizing White Blood Cell Diagnostics using Residual Block-Driven CNN Architecture [3].

In 2021, a work was published by Tavakoli et. al in Raabin-WBC [4] where ShapeColorClassifier was used to classify white blood cells. This model achieved 94.65% accuracy. However, this model needed to identify the proper order of convolution. After that, in 2022, another improvement work on the Raabin-WBC [4] dataset was done by Mr. Lei Jiang and his fellow researcher with the DRFA-Net [5] model. They had achieved an accuracy of about 95.17%, but his DRFA-Net needs to rely on a pre-train model like ResNeXt for network initialization. In 2023, Mr. Rufus Rubin and his team published another work on Raabin-WBC [4] that was able to get 97% accuracy on Raabin-WBC [4] with the Deep-ViT [6] model. However, the model they used - Deep ViT is a transformer-based architecture requiring significant computational resources for training, particularly in terms of memory and time. The ability to interpret the model's decisions is limited, which can be a concern in medical applications where understanding the cause behind predictions is tough.

Deep learning capabilities specify residual blocks and a convolutional neural network (CNN) architecture. The Raabin-WBC [4] dataset has been added and specially constructed for WBC image analysis. As for the limitation of previous research on WBC classification, we propose a more efficient and low computational cost solution with our work: residual block enhanced CNN model and transfer learning on MobileNet, InceptionNet, XceptionNet, and ResNet50 pre-trained models, which achieve an outstanding accuracy of nearly 99% on the Raabin-WBC [4] dataset.

In this research, we employed a residual blocked derived custom-based CNN model. This provides a decent accuracy with fewer numbers of parameters and consumes limited memory. We fine-tuned and trained our model using the training dataset and assessed its performance on the testing dataset. The results demonstrated that our model outperformed existing methods for this task.

## II. MATERIALS & METHODS

### A. Research Approach

The entire research workflow is showcased within the following flow chart 1.

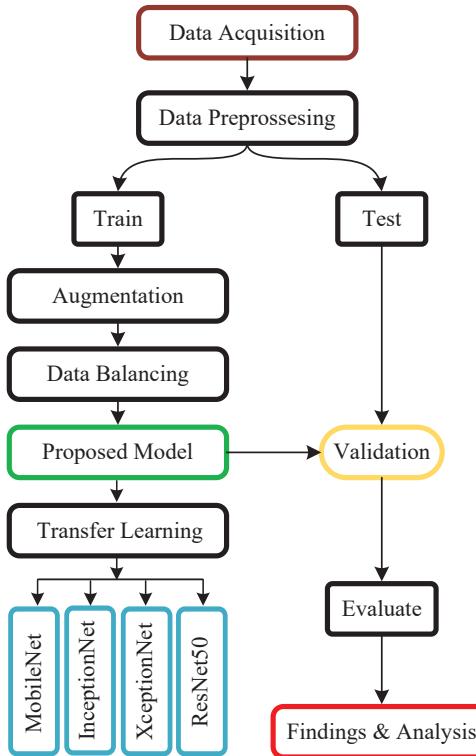


Fig. 1: Workflow of WBC classification

### B. Data Acquisition

The well-known Raabin-WBC [4] dataset, which is openly accessible and comprises five different classes— Lymphocytes, Monocytes, Neutrophils, Eosinophils, and Basophils, was utilized in our study. While additional white blood cell datasets, such as BCCD [7], exist, they only consist of four classes and do not contain basophils. This was a major factor in our decision to use the Raabin-WBC [4] dataset for our research. A synopsis of the dataset and the number of photos in each class are shown in the table I.

TABLE I: Raabin-WBC [4] Train and Test Data Distribution

Label	Total Data (No. of Images)	Test Data (No. of Images)	Augmented Train Data (No. of Images)
Neutrophil	10862	100	4000
Lymphocyte	3609	100	4000
Eosinophil	1066	100	4000
Monocyte	795	100	4000
Basophil	301	100	4000
<b>Total</b>	<b>17965</b>	<b>500</b>	<b>20000</b>

### C. Data Preprocessing

Table I demonstrates that the Raabin-WBC [4] dataset, obtained from Raabin's website, exhibits an imbalanced dis-

tribution of classes. Every class was allocated a test dataset consisting of 100 randomly selected images, while the other images were utilized to construct the training dataset. Hence, augmentation is essential for enhancing performance. The test dataset doesn't change; only the train dataset does.

For augmentation purposes, images were randomly zoomed by 5 to 20 % to introduce variations in the dataset. Following this phase, the images were cropped and resized to keep their original aspect ratio. Images are randomly rotated at a 5°, 10°, 15°, or 90° angle both clockwise and counterclockwise, followed by cropping to remove any extra white space. In addition, horizontal and vertical flips were performed to boost the variability of the training dataset. The number of times a picture must be augmented in order to achieve a balanced distribution in the training dataset is determined as the ratio of the anticipated total image count per class to the total image count for each class. This augmentation step is conducted using OpenCV in Python, and the augmented images are utilized to generate the training dataset. This allows you to see the augmented images, confirm the augmentation quality, and modify the scaling, resizing, rotation, and extension parameters of the Python augmentation function to create better-augmented images. 4000 images per class have been made utilizing augmentation from the original dataset. Later, the train dataset has been split into a train set and a validation set at a ratio of 80:20. An image sample of each class and images after augmentation is shown in figure 2.

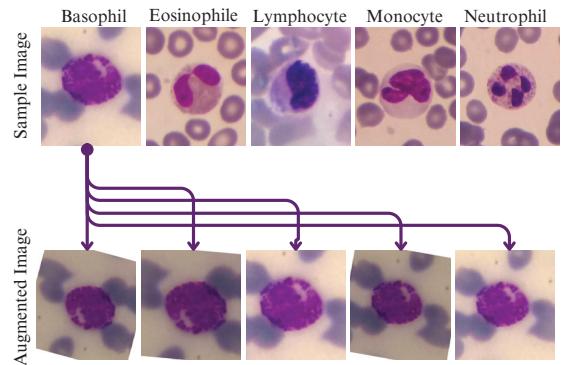


Fig. 2: Overview of dataset and augmented image

### D. Residual Block Enhanced Model

Residual blocks for deep learning models were first introduced by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun in their work, "Deep Residual Learning for Image Recognition" [8]. In a residual block, Layers not only connect sequentially but also skip connections over 2–3 Layers. This extra parallel path for gradient flow improves the model's performance and stability, allowing neural networks to become much deeper without experiencing issues like vanishing gradients or performance degradation.

In a residual block, the network seeks to learn the residual or difference between the desired output, let it be  $S(x)$  and the

input  $x$ ; therefore, the residual,  $R(x)$ , can be defined as

$$R(x) = S(x) - x$$

Rearranging the equation, it can be derive that

$$S(x) = R(x) + x$$

The identity connection from  $x$  allows the Layers Name to focus on learning this residual  $R(x)$ , which simplifies the learning effort. Compared to typical networks that learn the complete output, the residual networks focus on learning this smaller and more controllable residual  $R(x)$ .

The [9]residual block network [3b] we used for our residual block enhanced CNN is shown in figure 3.

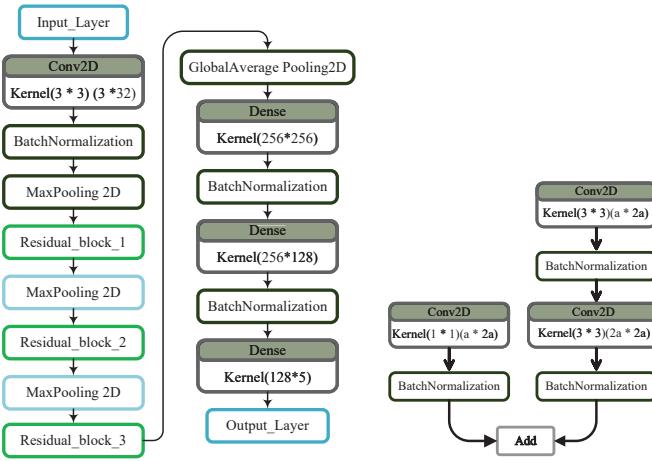


Fig. 3: Residual Block Enhanced CNN Architecture

We utilized three residual blocks with filter sizes of 64, 128, and 256, respectively. Additionally, we incorporated three dense layers with sizes of 256, 128, and 5.

#### E. Transfer Learning

Transfer learning [10] is a way of transferring knowledge from a source to a targeted task. It's used to adapt and improve the learning rate. Making a well-trained weight file causes huge computational cost, so we use a pre-trained model. It gives us the extra benefit of learning knowledge. We have checked the pre-trained model and filtered out which parameters would be best for us to use to improve the learning.

To enhance the classification task we also used pre-trained models like InceptionV3 [11], MobileNetV2 [12], Xception-Net [13] and ResNet [8] for transfer learning and They showed excellent result on Raabin-WBC [4] dataset. The model summary of these models is described in the section below.

1) *MobileNetV2*: It is a lightweight and efficient deep-learning model, intended for mobile and edge devices. It improves upon its predecessor by introducing inverted residual blocks and linear bottlenecks, which minimize computational

complexity while maintaining high accuracy in applications like picture categorization and object detection.

TABLE II: MobileNetV2 Model Architecture Summary

Layers Name	Output Shape	Parameters
InputLayer	(None, 224, 224, 3)	0
MobileNet	(None, 7, 7, 1280)	2,258,015
GlobalAveragePooling2D	(None, 1280)	0
Dense (64 units)	(None, 64)	81,984
BatchNormalization	(None, 64)	225
ReLU	(None, 128)	0
Dropout	(None, 64)	0
Dense (128 units)	(None, 128)	8,320
BatchNormalization	(None, 128)	512
ReLU	(None, 128)	0
Dropout	(None, 128)	0
Dense (128 units)	(None, 128)	16,512
BatchNormalization	(None, 128)	512
ReLU	(None, 128)	0
Dropout	(None, 128)	0
Dense (5 units)	(None, 5)	645
<b>Total Params</b>		2,366,725 (9.03 MB)
<b>Trainable Params</b>		1,963,205 (7.49 MB)
<b>Non-trainable Params</b>		403,520 (1.54 MB)

2) *InceptionV3*: It is a complex deep-learning model known for its efficiency and accuracy in image classification. It extends the previous Inception architecture with techniques such as factorized convolutions, label smoothing, and auxiliary classifiers that improve performance while reducing computational overhead. This makes InceptionNetV3 a popular choice for high-precision and resource-efficient applications.

TABLE III: InceptionV3 Model Architecture and Parameter Summary

Layers Name	Output Shape	Parameters
InputLayer	(None, 224, 224, 3)	0
InceptionV3	(None, 5, 5, 2048)	21,802,784
GlobalAveragePooling2D	(None, 2048)	0
Dense (64 units)	(None, 64)	131,136
Dense (128 units)	(None, 128)	8,320
BatchNormalization	(None, 128)	512
ReLU	(None, 128)	0
Dropout	(None, 128)	0
Dense (128 units)	(None, 128)	16,512
BatchNormalization	(None, 128)	512
ReLU	(None, 128)	0
Dropout	(None, 128)	0
Dense (5 units)	(None, 5)	645
<b>Total Params</b>		21,960,421 (83.77 MB)
<b>Trainable Params</b>		19,783,493 (75.47 MB)
<b>Non-trainable Params</b>		2,176,928 (8.30 MB)

3) *XceptionNet*: XceptionNet, short for Extreme Inception, is a deep learning model that extends the Inception architecture by replacing regular Inception modules with depthwise separable convolutions. This reduces parameters and computational complexity, improving efficiency and model performance. XceptionNet is particularly effective in image classification and other image-processing tasks, providing a balance between speed and accuracy.

TABLE IV: XceptionNet Model Architecture and Parameter Summary

Layers Name	Output Shape	Parameters
InputLayer	(None, 224, 224, 3)	0
InceptionV3	(None, 7, 7, 2048)	20,861,480
GlobalAveragePooling2D	(None, 2048)	0
Dense (64 units)	(None, 64)	131,136
Dense (128 units)	(None, 128)	8,320
BatchNormalization	(None, 128)	512
ReLU	(None, 128)	0
Dropout	(None, 128)	0
Dense (128 units)	(None, 128)	16,512
BatchNormalization	(None, 128)	512
ReLU	(None, 128)	0
Dropout	(None, 128)	0
Dense (5 units)	(None, 5)	645
<b>Total Params</b>	21,019,117 (80.18 MB)	
<b>Trainable Params</b>	20,964,077 (79.97 MB)	
<b>Non-trainable Params</b>	55,040 (215.00 KB)	

4) *ResNet50*: It is a deep learning model famous for inventing the notion of residual learning, which helps train very deep networks by solving the vanishing gradient problem. With 50 Layers Name, ResNet50 exploits shortcut connections to bypass some levels, enabling more quick training and improved accuracy in applications like image classification, object detection, and segmentation. It is commonly deployed due to its outstanding performance and scalability.

TABLE V: ResNet50 Model Architecture Summary

Layers Name	Output Shape	Parameters
InputLayer	(None, 224, 224, 3)	0
ResNet50	(None, 7, 7, 2048)	23,587,712
GlobalAveragePooling2D	(None, 2048)	0
Dense (64 units)	(None, 64)	131,136
Dense (128 units)	(None, 128)	8,320
BatchNormalization	(None, 128)	512
ReLU	(None, 128)	0
Dropout	(None, 128)	0
Dense (128 units)	(None, 128)	16,512
BatchNormalization	(None, 128)	512
ReLU	(None, 128)	0
Dropout	(None, 128)	0
Dense (5 units)	(None, 5)	645
<b>Total Params</b>	23,745,349 (90.58 MB)	
<b>Trainable Params</b>	23,691,717 (90.38 MB)	
<b>Non-trainable Params</b>	53,632 (209.50 KB)	

#### F. Evaluation Metrics

After completing the training of our model, we evaluate it with accuracy (1), precision (eqref:precision) (2), recall (3), and F1 score (4). The equation for these metrics is shown below.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (1)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2)$$

$$\text{Recall (Sensitivity or TPR)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3)$$

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

Where, TP: True Positive, FP: False Positive, TN: True Negative, FN: False Negative.

### III. RESULT & ANALYSIS

#### A. Accuracy and Loss Curves

Both curves, accuracy and loss, are of importance in the training phase of observing how well a machine learning model is behaving. The accuracy curve shows how well the model can classify (the higher, the better) samples by matching corresponding true and predicted labels. The loss curve describes optimization for the model: small values mean good prediction with respect to ground truth. Problems of overfitting or underfitting can be revealed by these curves and also show whether the model has converged—important pieces of information for further optimization efforts.

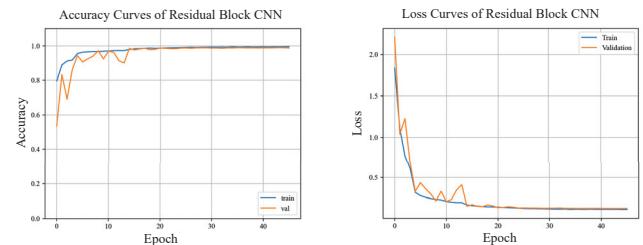


Fig. 4: Accuracy and Loss Curves of Residual Block CNN

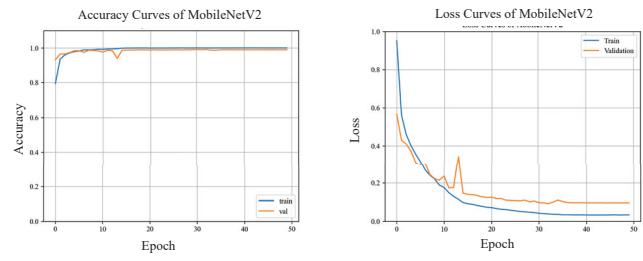


Fig. 5: Accuracy and Loss Curves of MobileNetV2

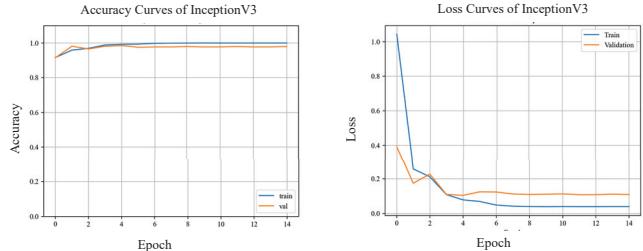


Fig. 6: Accuracy and Loss Curves of InceptionNetV3

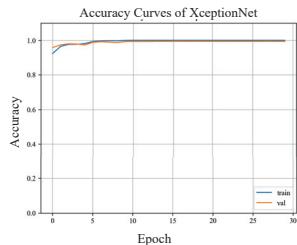


Fig. 7: Accuracy and Loss Curves of XceptionNet

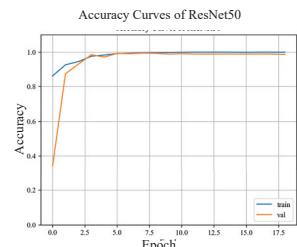
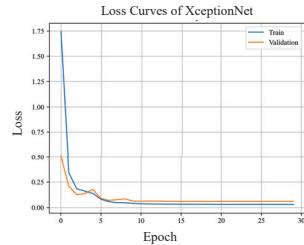
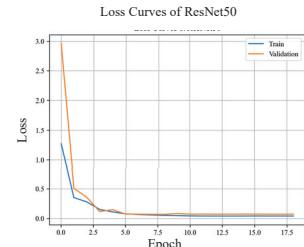


Fig. 8: Accuracy and Loss Curves of ResNet50



### B. Confusion Matrix and ROC Curves

The ROC curve and confusion matrix are fundamental tools for classification model evaluation.

1) *confusion matrix*: explicitly displays the model performance by showing true positives, true negatives, false positives, and false negatives. It facilitates the estimation of critical metrics like precision, recall, and F1-score, providing additional insight into the model's prediction skills.

2) *ROC curve*: An ROC curve (Receiver Operating Characteristic) plots sensitivity against 1-specificity across several threshold values, offering a visual picture of a model's discriminatory performance. The bigger the region Under the AUC (AUC), the better the group separability.

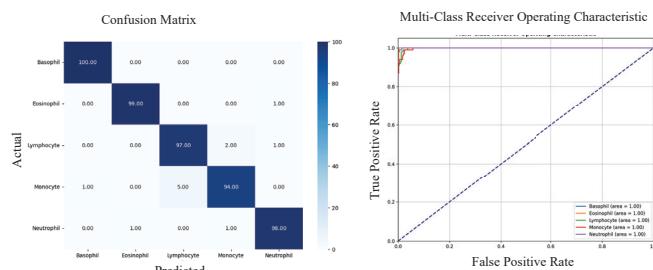


Fig. 9: Confusion Matrix and ROC Curve of CNN

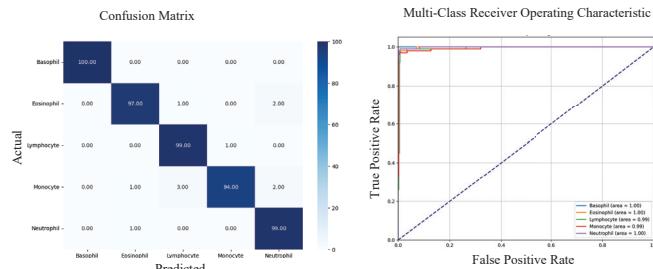


Fig. 10: Confusion Matrix and ROC Curve of MobileNetV2

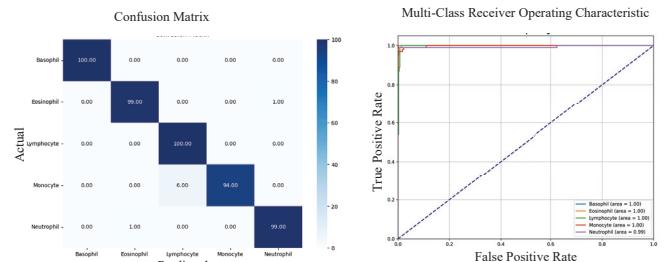


Fig. 11: Confusion Matrix and ROC Curve of InceptionNetV3

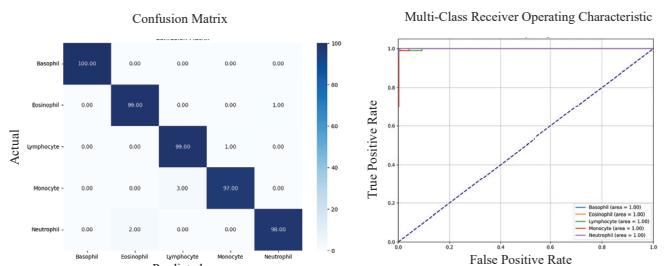


Fig. 12: Confusion Matrix and ROC Curve of XceptionNet

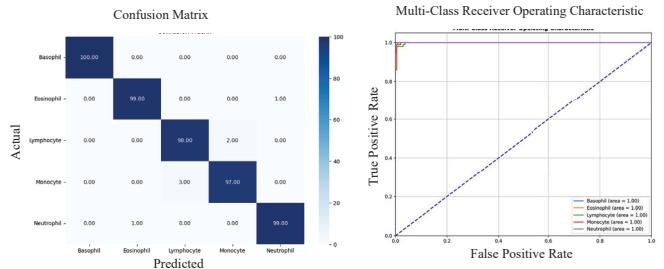


Fig. 13: Confusion Matrix and ROC Curve of ResNet50

### C. Performance Metrics Evaluation

In Table VI, we compare our proposed work with each other in terms of precision, recall, and F1 score metrics.

TABLE VI: Performance Comparison of Our Work

Metric	Model Name	B	E	L	M	N	Total
Precision	Residual Block CNN	0.99	0.99	0.95	0.97	0.98	0.976
	MobileNet	1.00	0.98	0.96	0.99	0.96	0.978
	InceptionNetV3	1.00	0.99	0.94	1.00	0.99	0.984
	XceptionNet	1.00	1.00	0.95	0.99	0.99	0.986
	<b>ResNet50</b>	<b>1.00</b>	<b>1.00</b>	<b>1.00</b>	0.97	<b>1.00</b>	<b>0.994</b>
Recall	Residual Block CNN	1.00	0.99	0.97	0.94	0.98	0.976
	MobileNet	1.00	0.97	0.99	0.94	0.96	0.972
	InceptionNetV3	1.00	0.99	<b>1.00</b>	0.94	0.99	0.984
	XceptionNet	1.00	0.98	0.99	0.96	1.00	0.986
	<b>ResNet50</b>	<b>1.00</b>	<b>1.00</b>	0.97	<b>0.99</b>	<b>1.00</b>	<b>0.992</b>
F1-score	Residual Block CNN	1.00	0.99	0.96	0.95	0.98	0.976
	MobileNet	1.00	0.97	0.98	0.96	0.98	0.978
	InceptionNetV3	1.00	0.99	0.97	0.97	0.99	0.984
	XceptionNet	1.00	0.99	0.97	0.97	1.00	0.986
	<b>ResNet50</b>	<b>1.00</b>	<b>1.00</b>	<b>0.98</b>	<b>0.98</b>	<b>1.00</b>	<b>0.992</b>

Where, B: Basophil, E: Eosinophile, L: Lymphocyte, M: Monocyte, N: Neutrophile

### D. Comparison with Recent Studies

Table VII summarizes our achievements in White Blood Cell classification, comparing our results to existing methods.

TABLE VII: White Blood Cell Classification Models

Ref	Year	Model	Dataset	Acc (%)
[14]	2021	ShapeColorClassifier		94.65
[5]	2022	DRFA-Net		95.17
[6]	2023	Deep ViT		97.00
Our Work	2024	Residual Block CNN	Raabin-WBC [4]	97.65
		MobileNet		97.85
		InceptionNet		98.40
		XceptionNet		98.60
		ResNet50		99.20

## IV. CONCLUSION &amp; DISCUSSION

Our findings show that using residual blocks is a very effective approach for detecting and classifying white blood cells (WBCs). As residual block offers lossless data restoration, We built a custom CNN model with residual blocks and achieved an accuracy of around 97.65%. By comparison, ResNet50, a top performer in ImageNet competition, reached 99.20% accuracy. ResNet50, which is likewise based on deep residual blocks, has over 2 million trainable parameters. Despite that, the accuracy difference is only 1.55% compared to our custom-build model, which has fewer than 400,000 trainable parameters. This indicates that our model achieves roughly the same accuracy with a considerably simpler design.

We also tested our dataset (Raabin-WBC [4]) on other models that use depthwise separable convolutions, like MobileNet, InceptionNet, and XceptionNet. After fine-tuning, these models achieved accuracies of 97.85%, 98.40%, and 98%, respectively.

What's intriguing is that our custom model's performance is very close to ResNet50 across all WBC classes. The comparison table clearly shows that ResNet50 scores almost 99% on precision, recall, and F1 across every category, but the gap between the two models is really small, even though our custom model is much simpler.

## REFERENCES

- [1] K. Walkovich and J. A. Connolly, "Disorders of white blood cells," in *Lanzkowsky's Manual of Pediatric Hematology and Oncology*, pp. 207–235, Elsevier, 2022.
- [2] S. Khan, M. Sajjad, T. Hussain, A. Ullah, and A. S. Imran, "A review on traditional machine learning and deep learning models for wbc classification in blood smear images," *Ieee Access*, vol. 9, pp. 10657–10673, 2020.
- [3] M. A. I. Siddique, A. Z. B. Aziz, and A. Matin, "An improved deep learning based classification of human white blood cell images," in *2020 11th International Conference on Electrical and Computer Engineering (ICECE)*, pp. 149–152, IEEE, 2020.
- [4] Z. M. Kouzehkanan, S. Saghari, E. Tavakoli, P. Rostami, M. Abaszadeh, F. Mirzadeh, E. S. Satlsar, M. Gheidishahran, F. Gorgi, S. Mohammadi, *et al.*, "Raabin-wbc: a large free access dataset of white blood cells from normal peripheral blood," *bioRxiv*, pp. 2021–05, 2021.
- [5] H. Z. Lei Jiang, Chang Tang, "White blood cell classification via a discriminative region detection assisted feature aggregation network," *Scientific data*, vol. 13, no. 3, pp. 1–12, Sept 12, 2022.
- [6] W. M. Rufus Rubin; S M Anzar; Alavikunhu Panthakkhan, "Transforming healthcare: Raabin white blood cell classification with deep vision transformer," *Scientific data*, Nov 08, 2023.
- [7] shenggan, N. Chen, cosmicad, and akshaylamba, "Bccd: Blood cell count and detection," 2018.
- [8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

- [9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- [10] J. Hao, "Deep learning-based medical image analysis with explainable transfer learning," *Scientific data*, June 29, 2023.
- [11] T. Emara, H. M. Afify, F. H. Ismail, and A. E. Hassani, "A modified inception-v4 for imbalanced skin cancer classification dataset," in *2019 14th International Conference on Computer Engineering and Systems (ICCES)*, pp. 28–33, IEEE, 2019.
- [12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.
- [13] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.
- [14] Z. M. K. . R. H. Sajad Tavakoli, Ali Ghaffari, "New segmentation and feature extraction algorithm for classification of white blood cells in peripheral smear images," *Scientific data*, vol. 13, no. 1, pp. 1–12, Sept 30, 2021.