Jurnal ELTIKOM :
Jurnal Teknik Elektro, Teknologi Informasi dan Komputer

Vol. 10, No. 1, June 2026, page. 11-23
ISSN 2598-3245 (Print), ISSN 2598-3288 (Online)
DOI: http://doi.org/10.31961/eltikom.v10i1.2089
Available online at http://eltikom.poliban.ac.id

HYBRID RESNET50 WITH CONVOLUTIONAL BLOCK ATTENTION MODULE (CBAM) FOR IMAGE CLASSIFICATION USING FINE-TUNING

Aulya Rachma Dewi*, Aris Thobirin, Sugiyarto Surono
Department of Mathematics, Ahmad Dahlan University, Yogyakarta, Indonesia
e-mail: 2200015022@webmail.uad.ac.id, aris.thobi@math.uad.ac.id, sugiyarto@math.uad.ac.id

Received: 6 December 2025 – Revised: 16 March 2026 – Accepted: 19 March 2026

ABSTRACT

Image classification is a crucial area in digital image processing that requires models capable of robust and stable feature representation. The main challenges in this study include variations between visual classes, diverse image quality, and limited labeled data, which often hinder the model's ability to generalize optimally. This study proposes a hybrid ResNet50-CBAM approach, which integrates the strengths of the ResNet50 architecture in deep feature extraction with the Convolutional Block Attention Module (CBAM) attention mechanism to improve the model's focus on the most informative areas of the image. The training process was carried out in two phases, namely transfer learning to utilize the initial representation from the ImageNet dataset, followed by fine-tuning to adjust the network weights to the image characteristics of the research dataset. The datasets were reorganized and split into 70% training, 15% validation, and 15% testing subsets to ensure a balanced distribution of samples. In addition, various augmentation techniques were applied to increase data diversity and improve the model's generalization capability. The evaluation results showed that this hybrid approach achieved an overall accuracy of 99%, indicating very high and consistent performance across the entire dataset. The integration of CBAM into the ResNet50 architecture was proven to strengthen the feature extraction process by highlighting the most relevant areas, resulting in a more accurate, stable, and effective image classification model for a wide range of artificial intelligence image processing applications.

Keywords*: attention mechanism, convolutional block attention module (CBAM), fine-tuning, image classification, ResNet50.*

I. INTRODUCTION

In recent years, deep learning has rapidly developed as a method capable of solving various complex problems, particularly in the field of image processing [1]. This advancement is driven by deep learning's ability to automatically learn and extract features from high-dimensional raw data [2], [3]. This approach is based on the working principles of biological neural networks and is implemented through artificial neural networks with interconnected node structures [4], [5]. One of the most commonly adopted architectures in deep learning is the Convolutional Neural Network (CNN) [6], [7], [8].

CNNs are designed to process grid-structured data such as images, with the ability to capture patterns and features hierarchically and adaptively through convolutional layers [9]. Through this mechanism, CNNs can build efficient feature representations, thereby improving accuracy in pattern recognition and image classification tasks [7], [10]. However, this architecture still has limitations, particularly the vanishing gradient problem. This problem arises in very deep networks when the backpropagation process produces increasingly smaller gradient values because of repeated multiplication of activation function derivatives, causing the gradients to approach zero as the network depth increases [11]. To address this problem, a CNN architecture called ResNet50 was developed [12].

The ResNet50 architecture consists of layers composed of convolutional layers, batch normalization, ReLU activation, and several residual blocks connected through skip connections [9], [13], [14]. This

architectural design allows data and gradient flow to pass through several layers at once, thereby overcoming the vanishing gradient problem and accelerating training convergence [12], [15]. Its ability to learn deep features efficiently makes ResNet50 one of the most widely used architectures in various image recognition studies and applications.

Although the ResNet50 architecture has demonstrated good performance in feature extraction, it is not fully capable of focusing attention on image regions that contain important information because of its limitations in capturing inter-channel and spatial relationships. The capabilities of ResNet50 can be improved by adding attention mechanisms such as the Convolutional Block Attention Module (CBAM) to strengthen the network's focus on important information in images. CBAM works by as-signing attention weights in two dimensions, namely channel and spatial, so that the network can focus more on important features and suppress background noise that may affect the analysis results [16], [17], [18]. The integration of this attention mechanism requires a proper training process so that the network weights can adjust to the characteristics of the data used.

One of the most widely used training strategies is fine-tuning pretrained models to adjust network weights so they better suit the characteristics of the images used in research [19]. This approach allows models to adapt to data with complex structures, such as medical images, which show variations in texture, color, and shape and therefore require deeper feature analysis to achieve accurate classification results [20]. One such application is white blood cell classification, which requires high precision in distinguishing morphological variations across cell types to support accurate medical diagnosis.

Previous research has successfully developed a deep learning-based method using a CNN architecture for leukocyte segmentation and classification, achieving an accuracy of 97.98% [21]. However, most of these models still have limitations in highlighting the most relevant morphological features, which calls for an approach that can improve feature selectivity and representation in blood images. Although CNNs have been widely used for white blood cell image classification, the results still vary because of the model's limitations in improving feature selectivity and handling the complexity of cell shapes effectively.

Based on this, this study aims to implement the ResNet-50 architecture integrated with the CBAM attention mechanism and fine-tuning for white blood cell image classification. This approach is designed to adjust network weights to the characteristics of medical images and significantly improve classification accuracy. Additionally, this study aims to evaluate the extent to which the CBAM attention mechanism, combined with fine-tuning, can strengthen the representation of important features in medical images. The results of this study are expected to produce a more accurate model for white blood cell image classification to support the early diagnosis of hematological disorders.

II. RESEARCH METHOD

In this study, the data processing and model development processes were carried out through several systematically arranged phases. Based on Figure 1, the research began with dataset reorganization, preprocessing, data augmentation, and the training process, which included transfer learning and fine-tuning.

A. Dataset Description

The dataset used in this study is *Blood Cell Images* (https://www.kaggle.com/datasets/paultimothymooney/blood-cells) obtained from the Kaggle platform. The original dataset contains 12,500 microscopic images categorized into four classes: Eosinophil, Lymphocyte, Monocyte, and Neutrophil. All images are represented in RGB format with varying resolutions and have been categorized according to their respective class labels. In this study, only images from the train and test subsets were used, resulting

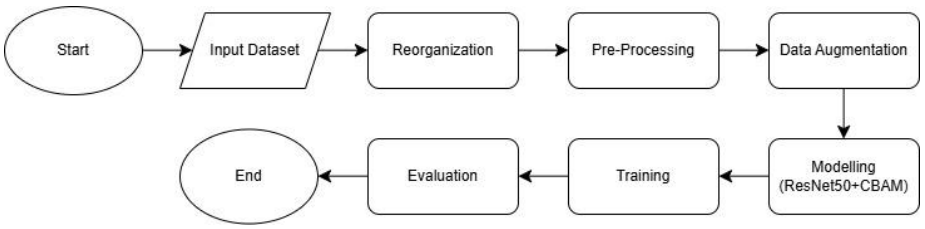
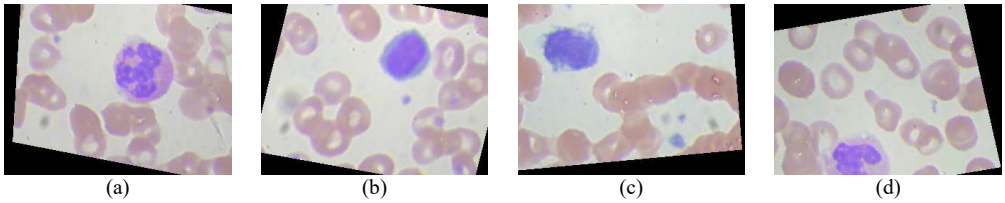


Figure 1. Research Flow Diagram.

TABLE 1
DATA SPLIT RESULTS.

Class	Train	Validation	Test	Total
Eosinophil	2.184	468	468	3.120
Lymphocyte	2.172	465	466	3.103
Monocyte	2.168	464	466	3.098
Neutrophil	2.186	468	469	3.123
Total	8.710	1,865	1,869	12.444



(a) (b) (c) (d)

Figure 2. (a) Eosinophil; (b) Lymphocyte; (c) Monocyte; (d) Neutrophil

in a total of 12,444 images utilized in the experiments. Files located outside these subsets were excluded and therefore were not involved in the model training or evaluation process.

Figure 2(a) shows an image of an eosinophil, characterized by a bilobed nucleus and reddish-orange cytoplasmic granules. Figure 2(b) shows an image of a lymphocyte, characterized by a large, dark purple nucleus that occupies most of the cell, accompanied by a minimal amount of light blue cytoplasm. Figure 2(c) shows an image of a monocyte, which is larger in size, with a horseshoe-shaped nucleus and extensive grayish-blue cytoplasm. Figure 2(d) shows an image of a neutrophil, which has a nucleus with three to five lobes and fine light purple granules in its cytoplasm.

B. Reorganization

In this stage, the dataset was reorganized by combining images from the original train and test folders. The combined dataset was then randomly shuffled and randomly split into 70% training, 15% validation, and 15% testing subsets to ensure a representative distribution of samples across the subsets. Each image was assigned to only one subset to avoid duplicate or overlapping samples. The distribution of images for each class in the three subsets is presented in Table 1, resulting in a total of 12,444 images, consisting of 8,710 training images, 1,865 validation images, and 1,869 testing images.

C. Pre-processing

Pre-processing is the first step in model development and helps determine accuracy and appropriate representation. Good preprocessing can reduce training time and produce more reliable predictions. The pre-processing phases in this study included converting RGB to BGR, resizing images to 224×224 pixels, and Z-Score normalization. The equation used for Z-Score normalization is (1).

$$Z = \frac{X - \mu}{\sigma} \tag{1}$$

In (1), Z is the standardized value, X is the original value of the image data, μ is the mean of the feature or variable, andσ is the standard deviation of the feature or variable [22], [23], [24], [25].

D. Augmentation

Augmentation is an important technique for improving model robustness against input variations, thereby preventing overfitting [26]. Data augmentation has been proven effective in improving generalization capability. This study applies geometric augmentation as the main strategy to enrich the training dataset and improve model robustness against data variations. The augmentation process is applied only to the training subset, while the validation of the model performance. The following augmentation methods were used in this study.

1) Rotation is a transformation method that rotates an image by a certain angle [27]. Mathematically, the equation used is as (2).

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2}$$

In (2), x' and y' are the new coordinates of a point after it is rotated by a certain angle θ [28], [29], [30].

2) Translation is the process of shifting the position of a point or object in an image by adding a certain shift value to its original coordinates. The equations are as (3) and (4).

$$x' = x + t_x \tag{3}$$

$$y' = y + t_y \tag{4}$$

In (3) and (4), t_x represents the shift on the horizontal axis, and t_y represents the shift on the vertical axis [30].

3) Flipping is the process of reversing an image or object based on a specific axis, such as the horizontal or vertical axis.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{5}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{6}$$

In (5) and (6), x and y are the pixel coordinates in the original image, while the transformation matrix determines the reversal of the pixel position relative to the horizontal or vertical axis, so that x' and y' are the new coordinates produced after flipping [29], [30], [31].

4) Nearest is a simple interpolation method in which new pixel values are determined by taking values from the nearest original pixels. In this process, the output pixels take values from the input pixels that are closest in distance without performing additional calculations. The equation is as (7).

$$I'(x, y) = I\big(round(x), round(y)\big) \tag{7}$$

In (7) [32], the mapping of coordinates (x', y') to position (x, y) in the original image is performed by rounding the transformed coordinate values using the operation $round(\cdot)$ [32].

E. Model

The present research employs a hybrid architecture combining ResNet50 with the Convolutional Block Attention Module (CBAM). This combination aims to enhance the quality of feature representations by leveraging ResNet50's feature extraction strengths and CBAM's adaptive attention mechanisms, allowing the model to produce more informative features in the classification process.

1) *ResNet-50*

ResNet50 is a deep convolutional neural network architecture that uses residual learning to address the vanishing gradient issue. This approach makes ResNet50 more stable and efficient in extracting visual features in image classification tasks. Each layer processes visual information to produce higher-level feature representations, enabling the network to recognize increasingly complex pattern details.

a. Convolution Layer

The convolutional layer extracts local features from images, such as patterns or textures. This process is performed by shifting the filter over a specific area of the image to produce a feature map.

$$c_{i,j} = \left(\sum_{u=0}^{n-1} \sum_{v=0}^{n-1} (a_{u+i,v+j} \times k_{i+1,j+1}) \right) + b_q \tag{8}$$

In (8), a represents the values in the input feature map, k represents the kernel elements, and bq is the bias [33].

b. Batch Normalization

Batch normalization improves training stability by normalizing the inputs of each layer within every mini-batch, thereby maintaining zero mean and unit variance [20].

$$\hat{x}_{i,j} = \frac{x_{i,j} - \mu_j}{\sqrt{\sigma_j^2 + \varepsilon}} \tag{9}$$

In (9), $x_{i,j}$ is the input value at position (i, j), μ_j and σ_j^2 are the mean and variance of feature map j, and ε is a small constant used to prevent division by zero [33].

c. ReLU Function

ReLU as an activation layer converts negative values to zero, thereby accelerating the computation process.

$$f(x) = max(x, 0) = \begin{cases} x, & if\ x \geq 0 \\ 0, & if\ x < 0 \end{cases} \tag{10}$$

In (10), for each input value $x \in (-\infty, +\infty)$, the function $f(x)$ represents the ReLU activation [9], [34].

d. Global Average Pooling

Global Average Pooling (GAP) calculates the average of each feature map to prepare the classification layer, reducing overfitting and computation time without adding trainable parameters [35]. The equation is as (11).

$$f_{avg}(X) = \frac{1}{N * M} \sum_{i=1}^{N} \sum_{i=1}^{M} X_{i,j} \tag{11}$$

e. Residual Block

A residual block is a learning method in neural networks designed to simplify the optimization process in very deep network models. In (12), $F(x, W)$ is a transformation function. This function maps the input x to a new representation with parameters W [36], [37].

$$y = F(x, W) + x \tag{12}$$

f. Dense Layer

This layer converts the feature extraction results into a numerical representation that can be used for classification decisions. In (13), W is the weight matrix in the dense layer neurons, X is the input matrix from the previous layer, and b is the added bias value.

$$Z = W^T \times X + b \tag{13}$$

g. Softmax Layer

Softmax converts the dense layer output into probabilities for each class, making it easier to determine the most appropriate class [38]. Equation (14) is the Softmax function, which converts scores z_i into probabilities p_i by normalizing the exponential values of each class so that the total becomes 1 [39].

$$p_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{14}$$

2) *Convolutional Block Attention Module (CBAM)*

The Convolutional Block Attention Module, or CBAM, is an attention mechanism designed to improve how well CNNs extract features by focusing on the most important details during training. CBAM comprises two primary components: the Channel Attention Module and the Spatial Attention Module.

a. The Channel Attention Module (CAM) highlights important elements in the channel dimension by applying global average pooling and maximum pooling, which are then processed through a multi-layer perceptron. Equation (15) produces the attention weight for each channel, while (16) applies these weights to strengthen the most relevant channels in the features [40], [41].

$$Mc(X) = \sigma(f_{MLP}(GAP(X)) + f_{MLP}(GMP(X))) \tag{15}$$

$$F' = Mc(F) \times F \tag{16}$$

b. The Spatial Attention Module (SAM) highlights the main spatial regions within the feature maps by performing both average pooling and maximum pooling across the channel dimension. Equation (17) produces a spatial attention map that highlights important locations in the feature, while (18) applies this map to the input feature to strengthen the most spatially relevant areas [40], [41].

$$Ms(X) = \sigma(f_{conv}([GAP(X)); GMP(X)])) \tag{17}$$

$$F^{"} = Mc(F') \times F' \tag{18}$$

3) *Hybrid ResNet50-CBAM*

In this study, a hybrid model combining ResNet50 and CBAM. ResNet50 served as the main part of the model to extract strong deep features. Its residual structure helps prevent the vanishing gradient problem, making the model more effective. After the backbone, the CBAM module was added to improve feature representation. It uses both channel and spatial to help the model focus more on the most important channels and regions in the image. The output from the attention module is then fed into the classification head, which includes Global Average Pooling, normalization layers, multiple dense layers, and dropout for regularization. This integration produces a model that remains efficient in basic feature extraction while becoming more adaptive in highlighting important patterns in images, thereby supporting improved classification performance.

F. Training

In this study, the model training process included two main stages: transfer learning and fine-tuning. During the transfer learning phase, the ResNet50 model equipped with CBAM used pre-trained weights obtained from initial training on large-scale datasets such as ImageNet. The use of these initial weights allows the model to retain the basic feature representations that have already been learned, so the adaptation process to new data can take place more efficiently. Next, the fine-tuning phase was carried out by readjusting some of the model layers so that they could learn more specific patterns from the research dataset. The application of these two phases was intended to progressively improve the model's capacity to achieve more accurate and stable classification.

1) Transfer Learning

In the transfer learning phase, all base layers of ResNet50 with pre-trained weights from ImageNet are frozen to retain the feature representations learned during previous training. Above the base model, a CBAM module and a classification head consisting of pooling, normalization, and several dense layers are added to adapt the model to the target classes. Only these additional layers are trained, while the training process is controlled through callbacks such as Early Stopping, ReduceLROnPlateau, and ModelCheckpoint to maintain stability and obtain the best weights.

2) Fine-tuning

During the fine-tuning phase, most of the initial layers of ResNet50 remain frozen, while the last 80 layers are reactivated to allow the model to learn more specific patterns from the dataset. Unfreezing only the deeper layers enables the network to adapt high-level feature representations that are more task-specific, while preserving the low-level features learned during the transfer learning phase. This strategy helps maintain the general visual representations obtained from large-scale datasets such as ImageNet while allowing the model to better adjust to the characteristics of the target dataset. The training process then continues using a lower learning rate so that weight updates occur gradually and steadily, reducing the risk of large parameter changes during fine-tuning.

3) Loss Function

The loss function shows how much the model's predictions differ from the actual target labels, helping adjust the model during training to make it more accurate. In this study, Categorical Cross-Entropy (CCE) loss is used for multi-class classification problems [42]. In (19), C is the number of classes, y_{true} is the original label, and y_{pred} is the predicted probability.

$$\mathcal{L}_{(y_{true}, y_{pred})} = -\sum_{i=1}^{C} y_{true,i} log\left(y_{pred,i}\right) \tag{19}$$

4) Adam Optimizer

Adam is an adaptive optimization method that updates each parameter using a learning rate that is dynamically adjusted according to the mean gradient (momentum) and the mean squared gradient. In (20), θ_t is the model parameter value at iteration t after the update process, θ_{t-1} is the parameter value from the previous iteration, α is the learning rate, $\widehat{m_t}$ is the first moment estimate (gradient average), $\widehat{v_t}$ is the second momentum estimate, and ε is a small constant is included to ensure stability during numerical computations [43].

$$\theta_t = \theta_{t-1} - \alpha \frac{\widehat{m_t}}{\sqrt{\widehat{v_t}} + \varepsilon} \tag{20}$$

In this study, the training employed the Adam optimizer, using a learning rate of 1×10^{-3} for the transfer learning phase and 1×10^{-5} for the fine-tuning phase. A higher learning rate was used during transfer learning to allow the newly added classification layers to learn more quickly, while a smaller learning rate during fine-tuning helps adjust pretrained weights gradually and maintain the learned feature representations.

G. Evaluation

In the evaluation phase, analysis of the model's performance was analyzed using a confusion matrix that includes the values of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) [44]. Based on these four components, the values of accuracy, precision, recall, and F1-score can be calculated as follows.

1) Accuracy computes the proportion of correct predictions from the entire test dataset [45], [46], [47], [48] as shown in (21).
2) Precision indicates how accurate the model is when giving positive predictions [45], [46], [47], [48] as shown in (22).
3) Recall (Sensitivity) assesses the extent to which the model can identify all positive instances in the data [45], [46], [47], [48] as shown in (23).
4) F1-Score is a single measure that combines both precision and recall to show how well a model performs overall [45], [46], [47], [48] as shown in (24).

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{21}$$

$$Precision = \frac{TP}{TP + FP} \tag{22}$$

$$Recall = \frac{TP}{TP + FN} \tag{23}$$

$$F_1 = \frac{2TP}{2TP + FP + FN} \tag{24}$$

III. Result and Discussion

This study uses the ResNet50 architecture with CBAM, forming a Hybrid ResNet50-CBAM architecture. The addition of CBAM aims to improve the model's capability to focus on important features at both the channel and spatial levels. After the hybrid architecture was built, a fine-tuning process was carried out to adjust the model weights to the characteristics of the dataset, which had a different distribution and feature pattern from ImageNet, the source of the pre-trained model. This fine-tuning process aims to optimize the model's generalization capabilities.

A. Model Training Result Analysis

The training process in this study consisted of two phases, namely transfer learning and fine-tuning. Both phases were carried out to ensure that the model could learn the initial representation stably before the weights in the deep layers were opened for adjustment.

As shown in Table 2, there is a noticeable improvement between the transfer learning phase (phase 1) and the fine-tuning phase (phase 2). In phase 1, training accuracy reached only 78.86% and validation accuracy was 81.76%, indicating that white blood cell image features were beginning to form but were not yet fully optimized. After the deeper layers were opened in Phase 2, training accuracy increased to

TABLE 2
TRANSFER LEARNING AND FINE-TUNING ACCURACY.

Training	Train (%)	Validation (%)
Transfer Learning (Phase 1)	78.86	81.76
Fine-Tuning (Phase 2)	99.63	99.95

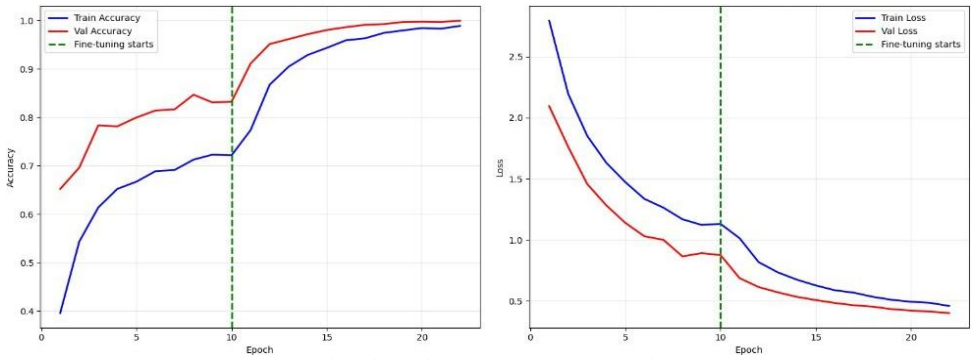


Figure 6. Combined Graph of Transfer Learning and Fine-Tuning.

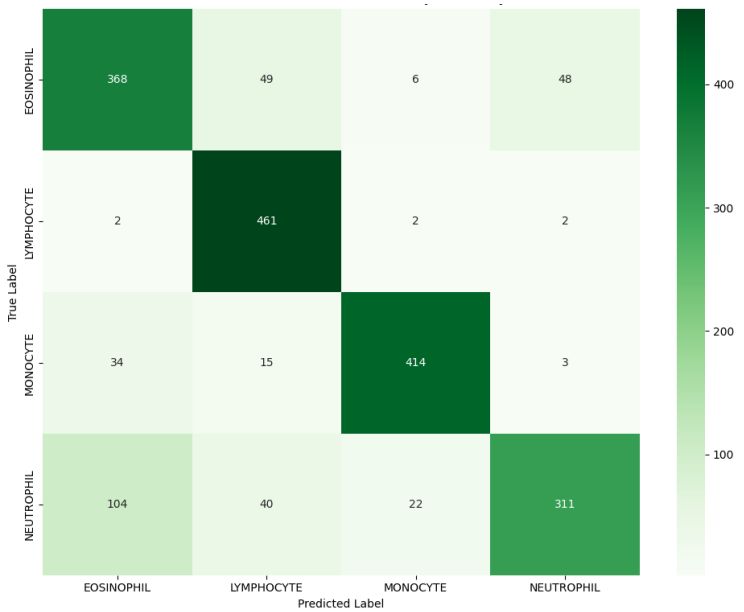


Figure 7. Confusion Matrix of the ResNet50-CBAM Model in the Transfer Learning Phase.

99.63%, and validation accuracy reached 99.95%. Overall, the fine-tuning process contributed to improved model performance compared to the initial transfer learning phase.

Based on Figure 6, the model's performance can be observed across two training phases, namely transfer learning and fine-tuning. In the transfer learning phase (phase 1), where all ResNet50 convolutional layers are frozen, the accuracy improves gradually. This indicates that the newly added classification layers successfully learned the initial representations of white blood cell characteristics, even though the basic features still relied entirely on pretrained weights. At this phase, the accuracy and loss curves still show fluctuations, indicating that the model's adaptation capacity remains limited.

The change in performance becomes more apparent after the fine-tuning phase (phase 2) begins, as indicated by the green dotted line on the training graph. In this phase, several residual layers with higher levels of abstraction are unfrozen and updated using a smaller learning rate. As training progresses, the accuracy increases more noticeably, while the loss values gradually decrease for both the training and validation datasets. This behavior indicates that the fine-tuning process enables the model to capture more task-specific features and improve feature representation. In addition, the relatively consistent trends between the training and validation curves suggest stable learning behavior throughout the training process.

B. Evaluation

Based on the test data results, classification performance is analyzed using a classification report. The following table presents the evaluation results at the transfer learning phase. Table 3 shows that the model achieved an overall accuracy of 82.00%, which means it performed quite well in identifying and classifying four different types of white blood cells. The Lymphocyte and Monocyte classes show the best model performance, with F1-scores of 89.00% and 90.00%, respectively, indicating that the model

TABLE 3
CLASSIFICATION REPORT FOR THE RESNET50-CBAM IN THE TRANSFER LEARNING PHASE.

	Precision (%)	*Recall* (%)	*F1-Score* (%)	*Support*
Eosinophil	71.00	73.00	72.00	469
Lymphocyte	82.00	97.00	89.00	466
Monocyte	91.00	89.00	90.00	465
Neutrophil	84.00	68.00	75.00	475
accuracy			82.00	1875
macro avg	82.00	82.00	82.00	1875
Weighted avg	82.00	82.00	82.00	1875

TABLE 4.
CLASSIFICATION REPORT OF THE RESNET50-CBAM MODEL IN THE FINE-TUNING PHASE.

	Precision (%)	*Recall* (%)	*F1-Score* (%)	*Support*
Eosinophil	99.58	100.00	99.79	471
Lymphocyte	100.00	100.00	100.00	467
Monocyte	100.00	100.00	100.00	466
Neutrophil	100.00	99.58	99.79	477
accuracy			99.89	1881
macro avg	99.89	99.90	99.89	1881
Weighted avg	99.89	99.89	99.89	1881

TABLE 5.
PERFORMANCE COMPARISON OF DIFFERENT MODELS

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score
Deep Feature Map Extraction of Segmented Leukocyte	97.98	97.97	97.00	97.00
ResNet50	99.88 ± 0.04	99.84	99.84	99.84
ResNet50 + CBAM	99.95 ± 0.06	99.89	99.90	99.89

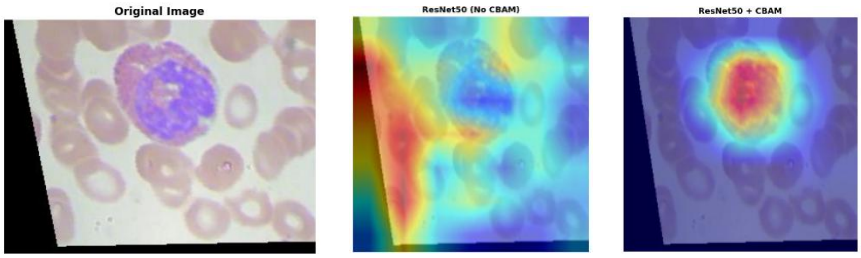
can recognize both classes consistently and accurately. Based on these results, the model's performance in the transfer learning phase can be further analyzed through a confusion matrix to identify the pattern of prediction errors in each class. This visualization offers a clearer understanding of how effectively the model differentiates the morphological characteristics of white blood cells during the early phases of training.

Based on Figure 7, the confusion matrix from the test data in the early part of the evaluation shows that the model performs well in distinguishing among various types of white blood cells. Specifically, the model achieves high accuracy in identifying Lymphocyte and Monocyte cells, as shown by the substantial number of correctly classified samples and the minimal misclassification errors within these categories. Nonetheless, the model shows notable confusion between the Eosinophil and Neutrophil classes. Misclassifications occur in both directions, with 48 Eosinophil samples incorrectly predicted as Neutrophils and 104 Neutrophil samples misclassified as Eosinophils. This pattern highlights the difficulty the model faces when distinguishing between these two morphologically similar cell types. Such challenges are common in medical image classification, where subtle differences in cell structure can lead to overlap in feature representation. Despite these issues, the model's performance remains robust, although the results also reveal opportunities for further refinement to enhance its discriminatory capability.

Based on Table 4, the evaluation results at the fine-tuning phase show that the model achieves near-perfect performance across all classes. The precision, recall, and F1-score values for all classes, namely Eosinophil, Lymphocyte, Monocyte, and Neutrophil, are in the range of 99.00% to 100.00%, indicating that the prediction error rate is very low. The overall accuracy value of 99% on the 1,881 samples further confirms that the model has excellent discriminatory capability. The macro average and weighted average also show nearly identical values, which means that performance across classes is balanced and no class is neglected. To provide a more specific picture of the model's prediction patterns in each class, the following confusion matrix is presented.

Figure 8 shows the results of the confusion matrix visualization illustrating the model's performance in classifying four types of white blood cells. From these results, it is clear that the model achieves a very high level of accuracy in identifying each class, as demonstrated by the strong prediction values along the main diagonal of the confusion matrix. A total of 471 Eosinophil images and 475 Neutrophil images were classified correctly, with only one and two prediction errors, respectively. Meanwhile, all Lymphocyte and Monocyte images were classified correctly without any classification errors.

After evaluating the model performance using the classification report and confusion matrix, a further analysis was conducted by comparing the proposed approach with previous methods to provide a clearer



(a) (b) (c)

Figure 9. (a) Original Image; (b) ResNet50 (No CBAM) (c) Resnet50-CBAM

understanding of the contribution of the attention mechanism to classification performance. The comparison results for the previous method, the baseline model, and the proposed ResNet50-CBAM model are presented in Table 5.

As shown in Table 5, the Deep Feature Map Extraction of Segmented Leukocyte method achieves an accuracy of 97.98%, with a precision of 97.97%, and recall and F1-score values of 97.00%. The baseline ResNet50 model shows a substantial improvement, achieving an accuracy of 99.88 ± 0.04%, with precision, recall, and F1-score of 99.84%. This indicates that deep convolutional architectures are capable of extracting more discriminative morphological features from leukocyte images than earlier feature extraction approaches.

Furthermore, integrating the Convolutional Block Attention Module (CBAM) into the ResNet50 architecture results in an additional improvement, achieving an accuracy of 99.95 $\pm$ 0.06%, with precision of 99.89%, recall of 99.90%, and F1-score of 99.89%. Although the quantitative improvement over the baseline model is relatively small, the results suggest that the attention mechanism introduced by CBAM helps guide the model to focus on more relevant morphological regions. This observation is further supported by the attention visualization analysis presented in Figure 9, which provides qualitative evidence that the CBAM module improves feature localization during the classification process.

These results show that the model has an excellent ability to distinguish morphological characteristics among white blood cell types. The high accuracy shown by the confusion matrix is also consistent with the earlier classification report results, indicating that the model has generalized very well to the test data. Based on the evaluation at the fine-tuning phase, further analysis visualizes the model's attention through CBAM. A comparison of the original image and the image with the attention map is presented to show the morphological areas that become the model's main focus in decision-making.

Figure 9(a) shows the original white blood cell image used as the object of analysis, where one leukocyte with a dark purple nucleus is surrounded by paler erythrocytes. The complex morphology of the cell nucleus in this image is the main component used to identify the leukocyte type. Figure 9(b) presents the Grad-CAM visualization generated by the ResNet50 model without the CBAM module. The highlighted regions appear more broadly distributed, indicating that the model focuses not only on the leukocyte nucleus but also on surrounding areas such as erythrocytes and background region. This suggests that the baseline model tends to capture less specific features during the classification process. In contrast, Figure 9(c) shows the Grad-CAM visualization results from the ResNet50 model equipped with the CBAM module. The visualization shows that the area with red-to-yellow color intensity is predominantly focused on the leukocyte nucleus, indicating the highest contribution to the model's classification decision. Conversely, the area around the main cells, including the erythrocytes, appears blue and makes only a minimal contribution. This pattern shows that the integration of CBAM can direct the model's attention more selectively to relevant morphological features, thereby improving interpretability and ensuring that the model's predictions are based on cell structures that are truly significant in the classification process.

IV. CONCLUSION

This study developed a hybrid model that integrates ResNet50 with the Convolutional Block Attention Module (CBAM) for white blood cell image classification. The proposed approach employs two training phases, namely transfer learning and fine-tuning, to improve the model's ability to extract morphological features of blood cells. In the transfer learning phase, the main layers of ResNet50 were frozen and the classification head was retrained, resulting in an initial accuracy of approximately 82.00%. Performance

improved further during the fine-tuning phase by unfreezing several deeper layers, increasing the accuracy to 99.89%. The training and validation curves indicate stable learning behavior, suggesting that the model does not show significant signs of overfitting during training.

Evaluation results based on the confusion matrix and classification report show strong performance across all classes, including Eosinophil, Lymphocyte, Monocyte, and Neutrophil, with precision, recall, and F1-score values approaching 1.00. In addition, the CBAM module provides qualitative interpretability through attention visualization, which helps illustrate how the model focuses on relevant regions of the cell images, such as the nucleus, during the classification process.

The main contribution of this study lies in the integration of CBAM with the ResNet50 architecture combined with a two-stage training strategy using transfer learning and fine-tuning to improve feature representation for white blood cell classification. However, this study was evaluated using a dataset that was divided into training, validation, and testing subsets derived from the same source dataset. Therefore, future studies may evaluate the proposed model using larger and more diverse datasets to further assess the robustness and general applicability of the model for medical image classification tasks.

DECLARATION OF AI AND AI ASSISTED TECHNOLOGIES IN THE WRITING PROCESS

During the preparation of this work, the authors used ChatGPT (OpenAI) in order to assist with language improvement, grammar correction, and sentence structuring. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

CREDIT AUTHORSHIP CONTRIBUTION STATEMENT

Aulya Rachma Dewi: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Aris Thobirin**: Supervision, Validation, Methodology, Writing – review & editing. **Sugiyarto Surono**: Supervision, Validation, Resources, Writing – review & editing.

DECLARATION OF COMPETING INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

ACKNOWLEDGMENT

We sincerely thank the Mathematics Laboratory and the Faculty of Applied Science and Technology at Ahmad Dahlan University for their support and the facilities provided during the completion of this research.

REFERENCES

[1] A. Özcan, M. Ünver, and A. Ergüzen, "Deep learning applications in medical image processing," *Int. J. Trend Sci. Res. Dev.*, vol. 5, no. 2, pp. 459–474, 2022.

[2] L. Alzubaidi *et al.*, "A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications," *J. Big Data*, vol. 10, no. 1, 2023.

[3] F. Ben Nasr Barber and A. Elloumi Oueslati, "Human exons and introns classification using pre-trained Resnet-50 and GoogleNet models and 13-layers CNN model," *J. Genet. Eng. Biotechnol.*, vol. 22, no. 1, p. 100359, 2024.

[4] O. Elharrouss, Y. Akbari, N. Almadeed, and S. Al-Maadeed, "Backbones-review: Feature extractor networks for deep learning and deep reinforcement learning approaches in computer vision," *Comput. Sci. Rev.*, vol. 53, 2024.

[5] S. Shivadekar, B. Kataria, S. Hundekari, K. Wanjale, V. P. Balpande, and R. Suryawanshi, "Deep Learning Based Image Classification of Lungs Radiography for Detecting COVID-19 using a Deep CNN and ResNet 50," *Int. J. Intell. Syst. Appl. Eng.*, vol. 11, no. 1s, pp. 241–250, 2023.

[6] A. Younesi, M. Ansari, M. Fazli, A. Ejlali, M. Shafique, and J. Henkel, "A Comprehensive Survey of Convolutions in Deep Learning: Applications, Challenges, and Future Trends," *IEEE Access*, vol. 12, pp. 41180–41218, 2024.

[7] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar, *A review of convolutional neural networks in computer vision*, vol. 57, no. 4. Springer Netherlands, 2024.

[8] L. Zewen, L. Fan, Y. Wenjie, P. Shouheng, and Z. Jun, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE Trans. neural networks Learn. Syst.*, vol. 33, no. 12, pp. 6999–7019, 2021.

[9] M. Krichen, "Convolutional Neural Networks: A Survey," *Computers*, vol. 12, no. 8, pp. 1–41, 2023.

[10] K. W. Goh *et al.*, "Comparison of Activation Functions in Convolutional Neural Network for Poisson Noisy Image Classification," *Emerg. Sci. J.*, vol. 8, no. 2, pp. 592–602, 2024.

[11] L. Alzubaidi *et al.*, *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*, vol. 8, no. 1. Springer International Publishing, 2021.

[12] M. Feng, Y. Cai, and S. Yan, "Enhanced ResNet50 for Diabetic Retinopathy Classification: External Attention and Modified Residual Branch," *Mathematics*, vol. 13, no. 10, 2025.

[13] S. S. Sumit, S. Anavatti, M. Tahtali, S. Mirjalili, and U. Turhan, "ResNet-Lite: On Improving Image Classification with a Lightweight Network," *Procedia Comput. Sci.*, vol. 246, no. C, pp. 1488–1497, 2024, doi: 10.1016/j.procs.2024.09.597.

[14] L. Zhang, Y. Bian, P. Jiang, and F. Zhang, "A Transfer Residual Neural Network Based on ResNet-50 for Detection of Steel Surface Defects," *Appl. Sci.*, vol. 13, no. 9, 2023.

[15] L. E. MacDonald, J. Valmadre, H. Saratchandran, and S. Lucey, "On skip connections and normalisation layers in deep optimisation," *Adv. Neural Inf. Process. Syst.*, vol. 36, no. NeurIPS, 2023.

[16] J. Zhu, Y. Zhang, C. Ma, J. Wu, X. Wang, and D. Kong, "GM-CBAM-ResNet: A Lightweight Deep Learning Network for Diagnosis of COVID-19," *J. Imaging*, vol. 11, no. 3, pp. 1–27, 2025.

[17] Y. Zhang *et al.*, "Deep-Learning Model of ResNet Combined with CBAM for Malignant–Benign Pulmonary Nodules Classification on Computed Tomography Images," *Med.*, vol. 59, no. 6, 2023.

[18] B. Guo *et al.*, "WaveAttention-ResNet: a deep learning-based intelligent diagnostic model for the auxiliary diagnosis of multiple retinal diseases," *Front. Radiol.*, vol. 5, no. July, pp. 1–18, 2025.

[19] H. E. Kim, A. Cosa-Linan, N. Santhanam, M. Jannesari, M. E. Maros, and T. Ganslandt, "Transfer learning for medical image classification: a literature review," *BMC Med. Imaging*, vol. 22, no. 1, pp. 1–13, 2022.

[20] I. D. Mienye, T. G. Swart, G. Obaido, M. Jordan, and P. Ilono, "Deep Convolutional Neural Networks in Medical Image Analysis: A Review," *Inf.*, vol. 16, no. 3, pp. 1–28, 2025.

[21] V. Anand, S. Gupta, D. Koundal, W. Y. Alghamdi, and B. M. Alsharbi, "Deep learning-based image annotation for leukocyte segmentation and classification of blood cell morphology," *BMC Med. Imaging*, vol. 24, no. 1, pp. 1–15, 2024.

[22] N. Sapkota, K. B. Khattri, and D. Aryal, "Modeling Precipitation: A Statistical and Machine Learning Approach," *Int. J. Eng. Technol.*, vol. 2, no. 2, pp. 188–203, 2025.

[23] R. K. Hapsari, A. H. Salim, B. D. Meilani, T. Indriyani, and A. Rachman, *Comparison of the Normalization Method of Data in Classifying Brain Tumors with the k-NN Algorithm*, vol. 1. Atlantis Press International BV, 2023.

[24] H. Henderi, "Comparison of Min-Max normalization and Z-Score Normalization in the K-nearest neighbor (kNN) Algorithm to Test the Accuracy of Types of Breast Cancer," *IJIIS Int. J. Informatics Inf. Syst.*, vol. 4, no. 1, pp. 13–20, 2021.

[25] A. Tawakuli, B. Havers, V. Gulisano, D. Kaiser, and T. Engel, "Survey:Time-series data preprocessing: A survey and an empirical analysis," *J. Eng. Res.*, vol. 13, no. 2, pp. 674–711, 2025.

[26] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Glob. Transitions Proc.*, vol. 3, no. 1, pp. 91–99, 2022.

[27] E. W. Ghindawi, "Advanced Computer Vision Alignment Technique Using Preprocessing Filters and Deep Learning," *Ing. des Syst. d'Information*, vol. 29, no. 4, pp. 1493–1499, 2024.

[28] Y. Fu, M. Nguyen, and W. Q. Yan, "Grading Methods for Fruit Freshness Based on Deep Learning," *SN Comput. Sci.*, vol. 3, no. 4, pp. 1–13, 2022.

[29] C. Hahne *et al.*, "Isometric Transformations for Image Augmentation in Mueller Matrix Polarimetry," *IEEE Trans. Image Process.*, vol. 14, no. 8, pp. 1–9, 2025.

[30] B. A. Awaluddin, C. T. Chao, and J. S. Chiou, "Investigating Effective Geometric Transformation for Image Augmentation to Improve Static Hand Gestures with a Pre-Trained Convolutional Neural Network," *Mathematics*, vol. 11, no. 23, 2023.

[31] F. A. Damayanti, R. Andri Asmara, and G. B. Prasetyo, "Residual Network Deep Learning Model with Data Augmentation Effects in the Implementation of Iris Recognition," *Int. J. Front. Technol. Eng.*, vol. 2, no. 2, pp. 79–86, 2024.

[32] A. Kunlerd, A. Ritthiron, B. Nabumroong, S. Luangmaneerote, A. Chaiwachirakhampon, and J. Kaewyotha, "A New Data Preprocessing Framework to Enhance the Accuracy of Herbal Plants Classification Using Deep Learning," *J. Appl. Data Sci.*, vol. 6, no. 3, pp. 1723–1740, 2025.

[33] L. I. Kesuma and R. Rudiansyah, "Classification of Covid-19 Diseases Through Lung CT-Scan Image Using the ResNet-50 Architecture," *Comput. Eng. Appl. J.*, vol. 12, no. 1, pp. 11–30, 2023.

[34] T. Perumal, N. Mustapha, R. Mohamed, and F. M. Shiri, "A Comprehensive Overview and Comparative Analysis on Deep Learning Models," *J. Artif. Intell.*, vol. 6, no. 1, pp. 301–360, 2024.

[35] G. Habib and S. Qureshi, "GAPCNN with HyPar: Global Average Pooling convolutional neural network with novel NNLU activation function and HYBRID parallelism," *Front. Comput. Neurosci.*, vol. 16, 2022.

[36] B. Mandal, A. Okeukwu, and Y. Theis, "Masked Face Recognition using ResNet-50," 2021, [Online]. Available: http://arxiv.org/abs/2104.08997.

[37] Ş. Kılıç, "Deep feature engineering for accurate sperm morphology classification using CBAM-enhanced ResNet50," *PLoS One*, vol. 20, no. 9 September, pp. 1–26, 2025.

[38] A. Çı, M. Yıldırı, and Y. Eroğlu, "Traitement du Signal Classification of Pneumonia Cell Images Using Improved ResNet50 Model," *Int. Inf. Eng. Technol. Assoc.*, vol. 38, no. 1, pp. 165–173, 2021.

[39] A. Mujhid, S. Surono, N. Irsalinda, and A. Thobirin, "Comparison and Combination of Leaky ReLU and ReLU Activation Function and Three Optimizers on Deep CNN for COVID-19 Detection," *Front. Artif. Intell. Appl.*, vol. 358, pp. 50–57, 2022.

[40] R. T. Wahyunigrum, I. A. Siradjuddin, T. D. Farawati, I. S. Faradisa, A. Bauravindah, and D. A. Dewi, "Implementation of an EfficientNet-B4 Model Architecture with a Convolutional Block Attention Module (CBAM) for Betel Leaf Disease Classification," *Eng. Technol. Appl. Sci. Res.*, vol. 15, no. 5, pp. 26722–26730, 2025.

[41] W. Islam, M. Jones, R. Faiz, N. Sadeghipour, Y. Qiu, and B. Zheng, "Improving Performance of Breast Lesion Classification Using a ResNet50 Model Optimized with a Novel Attention Mechanism," *Tomography*, vol. 8, no. 5, pp. 2411–2425, 2022.

[42] Q. Wang, Y. Ma, K. Zhao, and Y. Tian, "A Comprehensive Survey of Loss Functions in Machine Learning," *Ann. Data Sci.*, vol. 9, no. 2, pp. 187–212, 2022.

[43] H. Sun *et al.*, "An Improved Medical Image Classification Algorithm Based on Adam Optimizer," *Mathematics*, vol. 12, no. 16, pp. 1–14, 2024.

[44] S. Hira, A. Bai, and S. Hira, "An automatic approach based on CNN architecture to detect Covid-19 disease from chest X-ray images," *Appl. Intell.*, vol. 51, no. 5, pp. 2864–2889, 2021.

[45] Jude Chukwura Obi, "A comparative study of several classification metrics and their performances on data," *World J. Adv. Eng. Technol. Sci.*, vol. 8, no. 1, pp. 308–314, 2023.

[46] A. Victor Ikechukwu, S. Murali, R. Deepu, and R. C. Shivamurthy, "ResNet-50 vs VGG-19 vs training from scratch: A comparative analysis of the segmentation and classification of Pneumonia from chest X-ray images," *Glob. Transitions Proc.*, vol. 2, no. 2, pp. 375–381, 2021.

[47] K. Kansal, T. B. Chandra, and A. Singh, "ResNet-50 vs. EfficientNet-B0: Multi-Centric Classification of Various Lung Abnormalities Using Deep Learning 'session id: ICMLDsE.004,'" *Procedia Comput. Sci.*, vol. 235, pp. 70–80, 2024.

[48] A. Shabbir *et al.*, "Satellite and Scene Image Classification Based on Transfer Learning and Fine Tuning of ResNet50," *Math. Probl. Eng.*, vol. 2021, 2021.