CAUCHY – Jurnal Matematika Murni dan Aplikasi
Volume 11 (1) (2026), Pages 382-400
p-ISSN: 2086-0382; e-ISSN: 2477-3344

# Hybrid Otsu Morphological Pre-processing for EfficientNetB4 Based Acute Lymphoblastic Leukemia Classification

Maretta Mia Audina[1*], Sugiyarto Surono[1], Aris Thobirin[1], and Goh Khang Wen[2]

[1]*Departemen of Mathematics, Ahmad Dahlan University, Yogyakarta, indonesia*
[2]*INTI International University, Malaysia*

**Abstract**

Image quality plays a crucial role in improving the performance of image-based classification models, particularly when raw images exhibit noise, uneven illumination, and unclear object boundaries. This study proposes a hybrid segmentation approach to enhance object separation by reducing background interference and refining object contours. The method combines Otsu thresholding for initial object–background separation with elliptical morphological operations to improve region consistency and boundary definition. The segmented grayscale images are replicated into three channels and resized to 224×224 pixels before being used as input to an EfficientNetB4-based classification model optimized with the AdamW optimizer and fine-tuning. Experimental results under identical data splits, training settings, and fine-tuning protocols show that the proposed segmentation-based method achieves a final test accuracy of 97%, outperforming the baseline model trained on raw images (95% test accuracy) using the same EfficientNetB4-AdamW configuration. These results demonstrate that incorporating segmentation in the preprocessing stage effectively enhances discriminative feature learning and improves overall classification performance.

**Keywords:** Acute Lymphoblastic Leukemia; AdamW; EfficientNetB4; Morphological Operations; Otsu Thresholding



## 1. Introduction

In the era of Industry 4.0, deep learning has emerged as a key technology, originating from the broader fields of machine learning and artificial intelligence [1]. Deep learning is fundamentally based on linear algebra operations because the entire calculation uses vectors and matrices [2]. Deep learning trains models from examples so that they are able to recognize and classify text, sounds, and images [3]. One of the deep learning techniques that has proven effective for data classification is the Convolutional Neural Network (CNN) [4]. CNN is a neural network architecture that is widely used in image data recognition and classification tasks [5].

Although CNNs demonstrate strong performance in image classification, they primarily learn discriminative patterns from pixel intensities rather than explicitly understanding object boundaries or semantic structures [6]. As a result, irrelevant background regions or noise may still influence the learned features, particularly in medical images where objects of interest often occupy limited spatial regions. Therefore, image pre-processing is commonly employed as a

*Corresponding author. E-mail: 2200015010@webmail.uad.ac.id

Submitted: January 26, 2026 Reviewed: February 02, 2026 Accepted: February 19, 2026
DOI: https://doi.org/10.18860/cauchy.v11i1.40730

supportive strategy to improve input quality and emphasize relevant visual information prior to classification [7]. Variations in image context and quality may cause simple grayscale binarization to remove important regions or object details [8]. Consequently, image segmentation techniques can be utilized to further enhance feature focus by partitioning images into multiple segments based on pixel similarity [9]. Among various segmentation methods, Otsu Thresholding is widely used due to its simplicity and effectiveness in separating foreground objects from the background in grayscale images based on intensity distribution [10].

However, the Otsu Thresholding method has limitations when applied to images with non-uniform lighting or complex backgrounds [11]. To address this limitation, elliptical morphological operations can be applied to enhance segmentation by preserving the object's form and structural features while strengthening the distinction between its region and the background [12]. The integration of Otsu Thresholding with elliptical morphological operations is therefore expected to produce more accurate and informative input representations prior to the classification stage. From an optimization perspective, input images containing extensive background regions may introduce irrelevant information that affects the learning dynamics of CNN-based models. Preprocessing through segmentation helps suppress background interference, leading to more focused input representations and potentially more stable training behavior.

Following the preprocessing and segmentation stage, a classification model capable of efficiently learning discriminative visual features is required. EfficientNet is a CNN architecture designed to achieve high classification accuracy while maintaining computational efficiency [13]. It consists of a family of models ranging from B0 to B7, which employ compound scaling to systematically balance network depth, width, and input resolution for optimal performance with efficient parameter utilization [14], [15]. In addition, EfficientNet leverages MobileNetV2-based MBConv blocks and squeeze-and-excitation (SE) modules, enabling effective feature extraction with a relatively low computational burden compared to conventional CNN architectures [16].

The choice of optimizer is also a critical factor in improving image classification performance. AdamW is an optimization algorithm derived from Adam that introduces a decoupled weight decay mechanism, allowing regularization to be applied independently from the gradient-based parameter updates [17].

Due to its architectural efficiency and strong performance, EfficientNet has been widely adopted in studies on Acute Lymphoblastic Leukemia (ALL) image classification. ALL is a hematological malignancy characterized by the uncontrolled proliferation of lymphocytes in the bone marrow and surrounding tissues, affecting both children and adults [18]. In previous studies, EfficientNetV2M had a stable performance with an F1-Score of 0.891 [19], Meanwhile, and EfficientNetB3 achieved 96% accuracy which turned out to be superior to the VGG19 architecture [20].

Recent studies in hematology image classification have increasingly explored CNN-based approaches using either raw images or task-specific segmentation pipelines, many of which rely on complex architectures or data-intensive training procedures. Although these approaches have achieved high performance, the role of lightweight and deterministic preprocessing strategies in complementing CNN feature learning remains underexplored. Consequently, irrelevant background information and staining variability may still influence learned representations and limit classification robustness.

To address this limitation, this study proposes a hybrid image preprocessing framework that combines Otsu thresholding and elliptical morphological operations as a feature-enhancing preprocessing step prior to classification, rather than as a task-specific object delineation method. This strategy aims to emphasize relevant cellular regions while suppressing background interference, thereby improving the quality and consistency of input representations. The preprocessed images are subsequently classified using the EfficientNetB4 architecture optimized with the AdamW optimizer. This framework is expected to improve classification accuracy and robustness by providing more informative and structurally coherent input features.

The main contributions of this study are twofold. First, it introduces a lightweight and deterministic hybrid preprocessing strategy based on Otsu thresholding and elliptical morphological operations to enhance image quality while preserving essential structural characteristics in ALL images. Second, it evaluates the effectiveness of integrating this preprocessing strategy with the EfficientNetB4 architecture optimized using the AdamW optimizer for improving ALL image classification performance.

## 2. Methods

This section outlines the workflow of the proposed study, covering data preparation, hybrid segmentation (Otsu thresholding and elliptical morphology), model training with EfficientNetB4–AdamW, and performance evaluation. The overall research flow is shown below.

### 2.1. Research Flow

In this study, the data processing and model development were carried out through several successive stages.

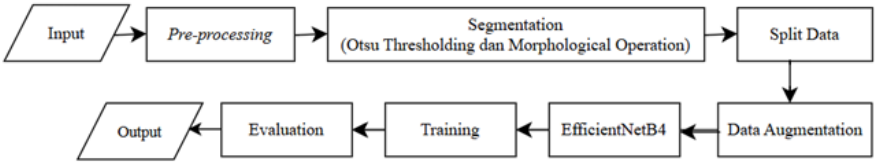


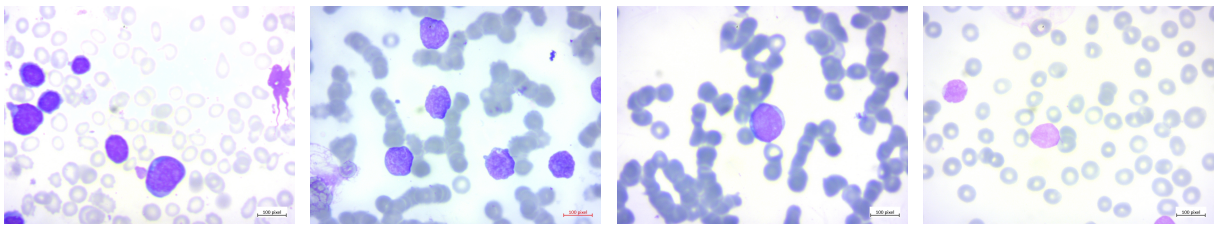

**Figure 1:** Research flow

Based on Fig. 1, this study began with image pre-processing, followed by segmentation using Otsu Thresholding and elliptical morphology to separate cells from the background. The segmentation results were divided into training and validation datasets and then were expanded through data augmentation. The final stage included modeling, training, and evaluation of classification performance.

### 2.2. Dataset Description

The dataset employed in this study was sourced from Kaggle, titled "Blood Cells Cancer (ALL) Dataset", which contains 3,242 peripheral blood smear (PBS) images for classifying acute lymphoblastic leukemia (ALL) according to white blood cell morphology. The images were acquired in the bone marrow laboratory of Taleqani Hospital (Tehran, Iran) using a Zeiss microscope at 100x magnification and labeled by specialists using flow cytometry [21]. The dataset was divided into four categories: Benign, Early Pre-B, Pre-B, and Pro-B.

To prevent data leakage, splitting was performed at the patient level, ensuring that all images from a single patient appear in only one subset (training, validation, or testing)



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

**Figure 2:** presents representative samples of lymphoblast cell conditions corresponding to each class in the dataset.

Fig. 2 Lymphoblast cell conditions: Fig. 2a normal cells with no indication of cancer (Benign, 512 images); Fig. 2b early morphological changes in lymphoblasts indicating the early stage of leukemia (Early Pre-B, 979 images); Fig. 2c characteristic malignancy in lymphoblast cells, including irregular shapes and thickened cytoplasm (Pre-B, 955 images); and Fig. 2d advanced stages of leukemia cells characterized by increased cell size and high chromatin density (Pro-B).

## 2.3. Pre-processing

The pre-processing stage was performed to standardize the input images before segmentation and classification. All images were resized to 224×224 pixels to reduce computational cost while still preserving sufficient cellular detail. Subsequently, for the purpose of segmentation, the images were temporarily converted from RGB to grayscale so that thresholding and morphological operations could be more easily applied to create cell masks. After segmentation was completed, the images were converted back to 3-channel RGB format, ensuring that the input to the EfficientNetB4 model remained compatible with the expected format and allowing the pretrained weights to be used without modifying the model architecture.

Data preparation was performed to improve input data quality and enhance model generalization, including image rescaling, data augmentation, target size adjustment, and class weighting to address class imbalance [22]. Class weights were computed using the following formula:

$$w_c = \frac{N}{n_c \times C} \tag{1}$$

where $w_c$ is the weight of class $c$, $N$ is the total number of samples, $n_c$ is the number of samples in class $c$, and $C$ denotes the total number of classes

## 2.4. Segmentation

### *2.4.1. Otsu Thresholding*

Otsu Thresholding was employed as the initial segmentation method by maximizing inter-class variance to determine the optimal threshold that separated foreground and background regions in grayscale images ranging from 0 to $L-1$ [23]. For an image with dimensions $M\ddot{O}N$, the probability of each gray-level intensity was calculated as:

$$B(i) = \frac{a(i)}{M \times N} \tag{2}$$

where $a(i)$ represented the number of pixels with intensity level $i$. The background and foreground probability were then computed as:

$$w_0(t) = \sum_{i=0}^{t} B(i) \tag{3}$$

$$w_1(t) = 1 - w_0(t) \tag{4}$$

The mean gray-level values for background and foreground were claculated as

$$\mu_0(t) = \frac{\sum_{i=0}^{t} iB(i)}{w_0(t)} \tag{5}$$

$$\mu_1(t) = \frac{\sum_{i=t+1}^{L-1} iB(i)}{w_1(t)} \tag{6}$$

Before calculating the variance between classes ($\sigma_B^2$) that is the basis for selcting the segmentation threshold, the average grayscale value ($T$) of the entire image must first be compute as.

$$T = \sum_{i=1}^{L-1} iB(i) \tag{7}$$

The between-class variance was then calculated as follows:

$$\sigma_B^2 = w_0(t)(\mu_0(t) - T)^2 + w_1(t)(\mu_1(t) - T)^2 \tag{8}$$

The optimal threshold $t^*$ was defined as the threshold value that resulted in the maximum between-class variance. Pixels whose intensity values exceeded $t^*$ were designated as foreground, while those with intensity values less than or equal to $t^*$ were considered background [24]. This Otsu Thresholding process produced a binary image $A$, where pixel value 1 represented the foreground and 0 represented the background.

Formally, the hybrid segmentation operator $I_{\text{hybrid}}$ is defined by directly processing binary mask $A$ with the elliptical structuring element $B$ through dilation ($\oplus$) and erosion ($\ominus$):

$$I_{\text{hybrid}} = (A \oplus B) \ominus B \tag{9}$$

This two step process thresholding followed by morphological refinement combining coarse object separation with structural enhancement.

### *2.4.2. Morphological Operation of the Ellips*

To refine the segmentation result, elliptical morphological operations were applied after Otsu Thresholding. These operations aimed to smooth object contours, close small holes, and remove noise while preserving cell structures [25]. The dilation and erosion of a binary image A with an elliptical structuring element B were defined as follows:

$$A \oplus B = \{z \mid (B)_z \cap A \neq \emptyset\} \tag{10}$$

$$A \ominus B = \{z \mid (B)_z \subseteq A\} \tag{11}$$

Dilation expanded foreground regions, while erosion reduced them depending on the structuring element shape [26], [27].

In this study, an elliptical structuring element of size $5 \times 5$ pixels was used, and each morphological operation was applied once to refine the segmentation. After these steps, the segmented image was converted back to 3-channel RGB format so that it could be used as input to EfficientNetB4 without modifying the pretrained model architecture.

## 2.5. Split Data

The dataset had been divided into 70% training, 15% validation, and 15% testing subsets after the segmentation stage, before entering the classification phase.

**Table 1:** Data distribution per class after splitting

| Class | Train | Validation | Test | Total |
|---|---|---|---|---|
| Benign | 358 | 77 | 77 | 512 |
| Early Pre-B | 658 | 147 | 147 | 979 |
| Pre-B | 668 | 143 | 144 | 955 |
| Pro-B | 557 | 120 | 120 | 796 |
| Total | 2.268 | 486 | 488 | 3.242 |

The distribution of the dataset per class after being split into training, validation, and test subsets is presented in Table 1. Each class was proportionally allocated, with 2.268 samples for training, 486 for validation, and 488 for testing. This approach ensured that the model received sufficient data for training while maintaining separate subsets for performance evaluation

## 2.6. Augmentation

Augmentation, specifically data augmentation, is a technique used to generate additional training data to enhances the model's generalization capability by creating new variations through synthesized data [28]. In this study, image augmentation was applied. Image manipulation was divided into two types, geometric and non-geometric manipulation [29]. Some of the geometric augmentation methods used are described below.

Rotation is a transformation that rotates an image by certain angle. Mathematically, it can be expressed as [30]:

$$\begin{bmatrix} f_x \\ f_y \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{12}$$

In Eq. (12), $f_x$ and $f_y$ are the new position of each pixel after rotation, whereas the coordinate pair $x$ and $y$ is the original image.

Translation moves objects to different positions in the image to prevent position bias [30]. Mathematically:

$$\begin{bmatrix} f_x \\ f_y \end{bmatrix} = \begin{bmatrix} x & Tx \\ y & Ty \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{13}$$

In Eq. (13), $f_x$ and $f_y$ represent the new coordinates of each pixel following translation, while $x$ and $y$ denotes original coordinate of the image and $T$ is the magnitude of the displacement to the axis $x$ or $y$.

Flipping creates new image variation by flipping the original image along horizontal and vertical axes, which were expressed in the following equations [31]:
Horizontal flipping

$$\begin{bmatrix} f_x \\ f_y \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{14}$$

Vertical flipping

$$\begin{bmatrix} f_x \\ f_y \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{15}$$

In Eq. (14) and Eq. (15), $f_x$ and $f_y$ are the new coordinates of each pixel after translation, whereas $x$ and $y$ is the original coordinate of the image and $T$ is the magnitude of the displacement to the axis $x$ or $y$.

Scaling, an augmentation technique that resizes an image by zooming in or out of an object's scale [32]. In this study, data augmentation was performed on-the-fly and applied only to the training set using ImageDataGenerator. The augmentations included a $15^o$ rotation, 10% horizontal and vertical translation, horizontal and vertical flipping, and scaling ratio through 10% zooming. No augmentation was applied to the validation and test sets to maintain consistent evaluation.

## 2.7. EfficientNetB4

The main building block of EfficientNet is MBConv to which blocks are added squeenze and excitation as well as the swish activation function [33]. EfficientNet uses MBConv (Mobile Inverted Bottleneck Convolution) as the building block previously introduced by MobileNetV2 [34]. Squeeze and Excitation (SE) is a CNN unit that strengthens feature representations by modeling relationships between channels through stages Squeeze (Global Average Pooling) dan Excitation (fully connected layer) [35].

### *2.7.1. MBConv*

MBConv reduces the channel dimension using depthwise separable convolution and a bottleneck, then expands it using an inverted residual block and pointwise convolution according to the expand ratio [36].

**1×1 Conv2D (Expand ratio).** The input to this stage is a feature tensor $A \in \mathbb{R}^{h \times w \times d_0}$,where $d_0$ denotes the number of input channels. Pointwise convolution (PWConv) applies a $1 \times 1$ kernel to linearly combine channel-wise information at each spatial location, producing an expanded feature representation [37]. Pointwise convolution is defined as follows:

$$Y(i, j, k') = \sum_{c=1}^{d_0} W(1, 1, c, k') \cdot A(i, j, c) + b_{k'} \tag{16}$$

In Eq. (16), $W$ is the weight for PWConv, $A$ is the input channel, $b$ is the bias and $Y$ is the output of the generated channel. Through teh expansion ratio $t$, the channel dimension is increas from $d_0$ to $t \cdot d_0$ [38], resulting in an output tensor $Y \in \mathbb{R}^{h \times w \times (td_0)}$.

**3×3 Depthwise Convolution.** Deptwise convolution (DWConv) extracts spatial features by applying a single filter to the input channel [39]. Given the expanded tensor $A \in \mathbb{R}^{h \times w \times td_0}$, DWConv is defined as follows.

$$Y_{p,q,k'} = \sum_{m=1}^{d} \sum_{n=1}^{d} W_{m,n,k'} \, A_{p+m-1,\, q+n-1,\, k'} \tag{17}$$

In Eq. (17) $A$ is the input, $W \in \mathbb{R}^{d \times d \times td_0}$ represent the depthwise kernel, and the resulting output tensor has dimensions $Y \in \mathbb{R}^{\frac{h}{s} \times \frac{w}{s} \times (td_o)}$, where $s$ denotes the stride.

**1×1 Conv2D (Projection).** In this process the channel is reduced again from $td_0$ to $d'$ , by the same method i.e. pointwise convolution. This operation produces the final MBConv output tensor $Y \in \mathbb{R}^{\frac{h}{s} \times \frac{w}{s} \times (d')}$

### *2.7.2. Squeeze and Excitation*

In depthwise convolution each channel is processed separately so that inter-channel information is not captured, so the SE block is used to analyze the entire channel output, amplify important features, and weaken less relevant features for subsequent transformations to be more effective [16].

Let the output feature map of the depthwise convolution be denoted as $U \in \mathbb{R}^{h \times w \times d_0}$, where $h$ and $w$ represent the spatioal dimensions and $d_0$ denotes the number of channels. The SE block begins with the Squeeze stage, which applies Global Average Pooling to aggregate spatial information across each channel [40]. This operation transforms the feature map $U$ into a channel descriptor $z \in \mathbb{R}^{d_0}$, where each element corresponds to the globlas representation of a single channel.

In the Excitation stage, the channel descriptor $z$ is passed through two fully connected layers that first reduce the dimensionality form $d_0$ to $\frac{d_0}{r}$ using ReLU and then restore it to $d_0$ with a sigmoid activation to generate channel wise attention weights $s \in [0, 1]^{d_0}$, which are applied to feature map via channel wise multiplication to adaptively recalibrate feature responses with minimal computational overhead [16].

### *2.7.3. Swish activation function*

The swish activation function, designed as an enhancement of the ReLU function, aims to improve deep learning model performance by ensuring smoother gradients are maintained throughout the training process [41]. Therefore, the swish activation function can be described as follows.

$$f(x) = x \cdot \frac{1}{1 + e^{-x}} \tag{18}$$

The swish activation function is capable of handling the problem vanishing gradient on sigmoid activation so that the learning process is not at the desired level and at the same time able to overcome problems negative region which also appears in the ReLU function [42].

### *2.7.4. Batch Normalization*

Batch Normalization is a method used to normalize the activation distributions within each batch, making the training process more stable, convergence is faster, and model performance as inference increases [43], [44]. Batch normalization is generally applied after all types of convolutions [45]. Mathematically, batch normalization is defined as follows.

$$Y_c = \frac{X_c - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}} \tag{19}$$

$$Z_c = \phi(\gamma_c Y_c + \beta_c) \tag{20}$$

In Eq. (19), $\mu_c$ and $\sigma_c$ successively descending is the mean and standard deviation of the channel $c$, $\epsilon$ is a small constant for numerical stability, and $X_c$ the input channel, so $Y_c$ as to produce an output. Once the output is obtained, a degree of freedom with the parameters is restored $\gamma_c$ and $\beta_c$ which allows the network to continue to learn optimal scale and shifts even though it has been normalized [46].

### *2.7.5. Global Average Pooling*

Global Pooling Summarize feature maps to reduce the parameters and computations that are commonly used Global Average Pooling (GAP) [47]. Mathematically, GAP is defined as follows:

$$\text{GAP} = \frac{1}{H \times W} \sum_{k=1}^{W} \sum_{l=1}^{H} X_{k,l} \tag{21}$$

In Eq. (21), $H$ and $W$ each of the width and height of the feature map, the GAP represents the average activation of the $i$ channel which indicates the level of importance of each channel [48].

### *2.7.6. Fully Connected*

In efficientnet, the fully connected part consists of only one dense layer which is used for the final classification process after Global Average Pooling [49]. Mathematically stated as follows.

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{b} \tag{22}$$

In Eq. (22), the input vector $\mathbf{x}$ is multiplied by the weight matrix $\mathbf{W}$, and the resulting product is added to the bias vector $\mathbf{b}$. The input $\mathbf{x}$ originates from the output of the Global Average Pooling process. The $\mathbf{y}$ is then passed to an activation in this operation will be passed to activation function, the softmax activation function is used to convert the input into a probability distribution of the prediction into each category [50]. The softmax function is commonly used in classification with more than two classes [51]. The softmax function is defined as follows:

$$P(y_i) = \frac{e^{y_i}}{\sum_{j=1}^{k} e^{y_j}} \tag{23}$$

where it $y_i$ is the result of the multiplication of inputs, weights, and the addition of biases. Where $y_i$ is the output of the $i$ neuron before activation, and $k$ is the total number of classes.

## 2.8. Training

During the training stage, the model employed Sparse Categorical Cross Entropy as the loss function for multi-class classification tasks [52]. The loss function worked with integer form labels that don't require a target label in the form one-hot-encoding [53].

$$L(y_{\text{true}}, y_{\text{pred}}) = -\sum_{i=1}^{C} y_{\text{true},i} \log(y_{\text{pred},i}) \tag{24}$$

where $C$ is the number of classes, $y_{\text{true}}$ is the vector of the true label in the form of an integer and $y_{\text{pred}}$ is the vector of the prediction probability distribution for the class $C$.

In this study, training was conducted using checkpoints and early stopping based on validation loss with a patience of 7. In the initial phase, the model was trained for 50 epochs with the EfficientNetB4 layers frozen, so that only the GAP, Dropout (rate=0.4), and Dense layers were updated, with class weighting applied to address class imbalance. Subsequently, fine-tuning was performed by unfreezing the last 60 layers for 30 epochs, while still applying early stopping and checkpoints to achieve optimal performance. Each training session used a train generator with on-the-fly augmentation, validation was performed using the original data, the batch size was kept at 8 for consistency, and all random operations used a fixed seed to ensure reproducibility. The learning rate was kept constant during each phase of training.

## 2.9. Optimizer AdamW

AdamW was a modification of the Adam optimizer in which weight decay is decoupled from the gradient computation, ensuring that the decay term did not interfere with the gradient itself and is instead applied directly during the parameter update step [17]. Mathematically, the parameter update in AdamW is expressed as follows.

$$\theta_{t+1} = \theta_t - \eta \left( \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} + \lambda \theta_t \right) \tag{25}$$

In Eq. (25), $\theta_t$ is the weight at iteration $t$, $\theta_{t+1}$ the weight after being updated in the iteration, $\eta$ is the learning rate, $\hat{m}_t$ and $\hat{v}_t$ is the first-order and the second-order moments, $\epsilon$ a small constant to prevent zero division, and $\lambda$ is the weight decay coefficient. In this study, an AdamW optimizer was used with a learning rate of magnitude $1 \times 10^{-4}$ and weight decay $1 \times 10^{-4}$ it was applied during both initial training and fine-tuning stages.

## 2.10. Evaluation

The evaluation in this study was conducted using a confusion matrix, which summarizes the relationship between the model's predicted labels and the ground-truth labels. For a multi-class classification problem with $C$ classes, the confusion matrix is an $n \times n$ table, where each element represents the number of samples predicted for a given class versus their true class.

For class-wise analysis, a one-vs-rest strategy was adopted to compute the following quantities for each class $i$ : true positives ($TP_i$), false positives ($FP_i$), false negatives ($FN_i$), and true negatives ($TN_i$), these values were used to derive class-level precision, recall and F1-score [54].

Accuracy measures the overall proportion of correctly classified samples and is defined for multi-class classification as:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{N} = \frac{\sum_{i=1}^{C} TP_i}{N} \tag{26}$$

where $N$ denotes the total number of test samples. Precision for class $i$ represents the proportion of correctly predicted samples among all samples predicted as class $i$:

$$\text{Precision}_i = \frac{TP_i}{TP_i + FP_i} \tag{27}$$

Recall also referred to as sensitivity or true positive rate, measures the model's ability to correctly identify samples belonging to class $i$:

$$\text{Recall}_i = \frac{TP_i}{TP_i + FN_i} \tag{28}$$

The F1-score for class $i$ is defined as the harmonic mean of precision and recall:

$$\text{F1}_i = \frac{2 \times \text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \tag{29}$$

Given the presence of class imbalance in the dataset (e.g., Benign: 512 samples, Early Pre-B: 979 samples), overall performance was further evaluates using micro-averaged, macro-averaged, and weighted F1-score strategies [55].

The Micro-averaged F1-score is computed by globally aggregating true positives, false positives, and false negatives across all classes:

$$\text{Precision}_{\text{micro}} = \frac{\sum_i TP_i}{\sum_i (TP_i + FP_i)}, \quad \text{Recall}_{\text{micro}} = \frac{\sum_i TP_i}{\sum_i (TP_i + FN_i)} \tag{30}$$

$$\text{F1}_{\text{micro}} = \frac{2 \times \text{Precision}_{\text{micro}} \times \text{Recall}_{\text{micro}}}{\text{Precision}_{\text{micro}} + \text{Recall}_{\text{micro}}} \tag{31}$$

The Macro-averaged F1-score is defined as the unweighted mean of the per-class F1-scores:

$$\text{F1}_{\text{macro}} = \frac{1}{C} \sum_{i=1}^{C} \text{F1}_i \tag{32}$$

The Weighted F1-Score accounts for class imbalance by weighting each class F1-score by its support:

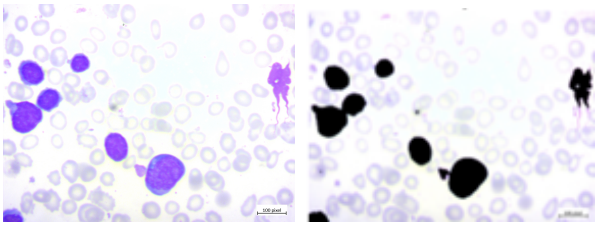
$$\text{F1}_{\text{weighted}} = \sum_{i=1}^{C} w_i \cdot \text{F1}_i, \quad w_i = \frac{n_i}{\sum_{j=1}^{C} n_j} \tag{33}$$

where $n_i$ denotes the number of true samples in class $i$. All reported evaluation metrics follow the standar definitions implemented in the scikitlearn library using the classification_report function with the corresponding averaging strategies.

## 3. Results and Discussion

This study employed the EfficientNetB4 architecture combined with a segmentation approach during the pre-processing stage to reduce non-cell regions using Otsu Thresholding and elliptical morphological operations, enabling the model to focus primarily on the cell nucleus. The training process utilized the AdamW optimizer with a learning rate of $1 \times 10^{-4}$ and a weight decay of $1 \times 10^{-4}$.

The pre-processing stage focused on image segmentation to enhance the model's attention to white blood cell regions (lymphoblasts) by minimizing background interference in microscopic images. A segmentation approach was implemented by integrating the Otsu Thresholding method with elliptical morphological operations. The segmentation results obtained from this process are presented in Fig. 3.



**(a)** Original Image **(b)** Segmentation Image

**Figure 3:** Comparison between the original image and the segmented image.

As shown Fig. 3a, the original image still contained all blood cell components with varying color intensities. After the segmentation process, as illustrated in Fig. 3b, the cell nucleus region became more prominent while background components were significantly reduced. These results indicate that the proposed segmentation method successfully separated the region of interest (cell nucleus) from irrelevant background areas. Consequently, the segmented images became more

informative for the subsequent model training stage. Following this process, the dataset was divided into training, validation, and testing subsets as described in the research methodology.

After the dataset of segmentation results is divided into three subsets, then data augmentation is carried out in the training data. Furthermore, the model was trained using EfficientNetB4 which has gone through segmentation, with pixel input 224 × 224. The initial weight uses pre-trained ImageNet, then added classification in the form of GAP, 0.4 dropout and softmax activated dense. The optimization process uses AdamW with learning rate $1 \times 10^{-4}$ and weight decay $1 \times 10^{-4}$.

### 3.1. Baseline Performance Using Raw Image

The baseline model in this study is defined as EfficientNetB4 trained on raw images using the AdamW optimizer, followed by a fine-tuning stage. All baseline results are evaluated on the same test set as the proposed method to ensure a fair comparison. During the initial training

**Table 2:** Accuracy and loss of initial training and fine-tuning stages using raw images

| Training Stage | Data | Accuracy | Loss |
|---|---|---|---|
| Initial training | Train | 84.66% | 0.4970 |
| | Validation | 94.44% | 0.3706 |
| Fine-tuning | Train | 98.32% | 0.0591 |
| | Validation | 97.12% | 0.0931 |

stage, the model achieved a training accuracy of 84.66% and a validation accuracy of 94.44%, with corresponding loss values of 0.4970 and 0.3706, respectively. After fine-tuning, performance improved substantially, reaching 98.32% training accuracy and 97.12% validation accuracy, accompanied by a significant reduction in loss. These results indicate that fine-tuning effectively enhances feature adaptation and model convergence.

To further illustrate the effect of fine-tuning, Figures 4 and 5 present comparisons of accuracy and loss before and after fine-tuning.

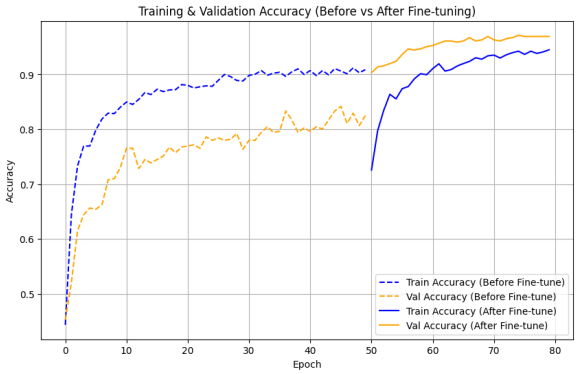



**Figure 4:** Accuracy comparison before and after fine-tuning for the baseline model

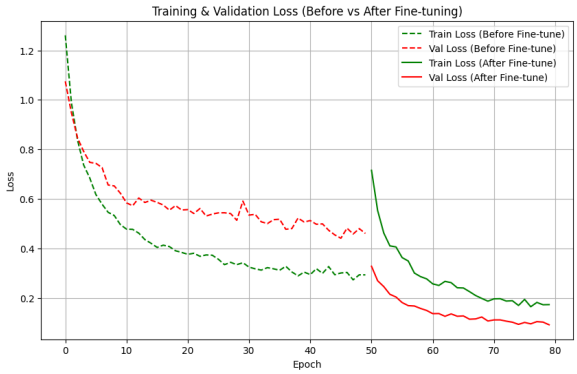



**Figure 5:** Loss comparison before and after fine-tuning for the baseline model

Figures Fig. 4 and Fig. 5 illustrate the evolution of accuracy and loss before and after fine-tuning. Prior to fine-tuning, a noticeable gap between training and validation curves is observed, along with relatively unstable validation loss, suggesting limited feature specialization. After fine-tuning, both accuracy curves converge at higher values, while the loss decreases sharply and stabilizes, indicating improved generalization and more stable learning behavior.

**Classification performance before fine-tuning.** Based on the test results obtained after the initial training stage, the model recorded a loss of 0.4724. The corresponding classification report is presented in Table 3

**Table 3:** Classification report for each class

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Benign | 84.29% | 76.62% | 80.27% | 77 |
| Pre-B | 68.27% | 98.61% | 80.68% | 144 |
| Pro-B | 100.00% | 61.67% | 76.29% | 120 |
| Early Pre-B | 86.76% | 80.27% | 83.39% | 147 |
| Accuracy | | | 80.53% | 488 |
| Macro Avg | 84.83% | 79.29% | 80.16% | 488 |
| Weighted Avg | 84.17% | 80.53% | 80.35% | 488 |

The classification report obtained before fine-tuning (Table 4) shows class-dependent performance variations, with high recall for the Pre-B class (98.61%) but relatively low recall for Pro-B (61.67%). The confusion matrix in Figure 6 further reveals notable misclassifications among morphologically similar classes, particularly between Pro-B and Pre-B, as well as between Benign and Early Pre-B. These results suggest that the frozen feature extractor has not yet learned sufficiently discriminative representations.

**Table 4:** Classification report after segmentation and fine-tuning

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Benign | 87% | 90% | 88% | 77 |
| Pre-B | 96% | 99% | 97% | 144 |
| Pro-B | 99% | 96% | 97% | 120 |
| Early Pre-B | 94% | 93% | 94% | 147 |
| Accuracy | | | 95% | 488 |
| Macro Avg | 94% | 94% | 94% | 488 |
| Weighted Avg | 95% | 95% | 95% | 488 |

**Classification performance after fine-tuning.** After fine-tuning, the baseline model achieved an overall accuracy of 95% and a weighted F1-score of 95%, as shown in Table 4. Improvements are observed across all classes, with more balanced precision and recall values. The corresponding confusion matrix in Figure 7 exhibits stronger diagonal dominance and reduced misclassification, confirming that fine-tuning improves class separability. However, residual confusion between certain leukemia subtypes remains, motivating the integration of segmentation in the proposed method.

## 3.2. Proposed Method Performance with Hybrid Segmentation

The proposed method integrates hybrid segmentation based on Otsu thresholding and elliptical morphological operations with EfficientNetB4 and the AdamW optimizer. The training process was conducted in two stages: initial training and fine-tuning. The initial training phase was configured for 50 epochs, with the best validation performance achieved at epoch 39, which was selected as the final model for this stage. Subsequently, a fine-tuning stage was carried out for 30 epochs to further optimize the model parameters

The accuracy and loss values obtained during both training stages are summarized in Table 5.

**Table 5:** Accuracy and loss of initial training and fine-tuning stages using raw images

| **Training Stage** | **Data** | **Accuracy** | **Loss** |
|---|---|---|---|
| Initial training | Train | 89.11% | 0.3318 |
| | Validation | 90.12% | 0.2943 |
| Fine-tuning | Train | 95.28% | 0.1512 |
| | Validation | 96.50% | 0.1059 |

During the initial training stage, the model achieved a training accuracy of 89.11% and a validation accuracy of 90.12%, with corresponding loss values of 0.3318 and 0.2943, respectively. Compared to the baseline model trained on raw images, these results indicate that hybrid segmentation facilitates more effective early feature extraction by reducing background interference and emphasizing relevant cellular regions.

In the fine-tuning stage, the proposed method reached a training accuracy of 95.28% and a validation accuracy of 96.50%, accompanied by a further reduction in loss. The consistent improvement across both stages suggests that the combination of segmentation and selective parameter updating enhances convergence and learning stability.

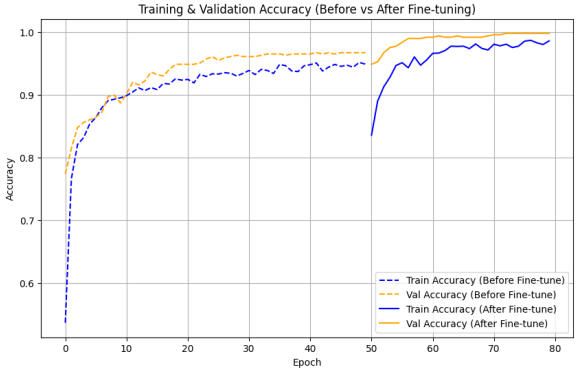



**Figure 6:** Accuracy comparison before and after fine-tuning using hybrid segmentation

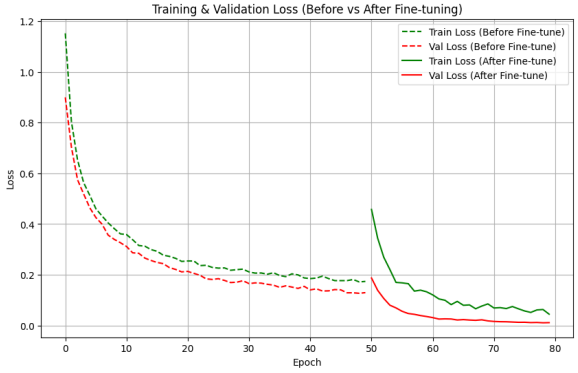



**Figure 7:** Loss comparison before and after fine-tuning using hybrid segmentation

Figures Fig. 6 and Fig. 7 present the accuracy and loss curves before and after fine-tuning. As shown, fine-tuning results in higher and more stable accuracy for both training and validation sets, along with a consistent decrease in loss values. These trends indicate improved feature discrimination while maintaining stable generalization behavior.

**Classification performance before fine-tuning.** Based on the test results obtained after the initial training stage, the model recorded a loss of 0.2915. The corresponding classification report is presented in Table 6.

**Table 6:** Classification report of EfficientNetB4 model

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Benign | 82.56% | 92.21% | 87.12% | 77 |
| Pre-B | 83.93% | 97.92% | 90.38% | 144 |
| Pro-B | 99.02% | 84.17% | 90.99% | 120 |
| Early Pre-B | 94.70% | 85.03% | 89.61% | 147 |
| Accuracy | | | 89.75% | 488 |
| Macro Avg | 90.05% | 89.83% | 89.52% | 488 |
| Weighted Avg | 90.67% | 89.75% | 89.78% | 488 |

Table 6 shows that the highest precision was achieved in the Early Pre-B class (94.70%), while the highest recall was observed in the Pre-B class (96.92%). These results indicate class-dependent performance variations, motivating a deeper analysis using a confusion matrix.

Prior to fine-tuning, the confusion matrix analysis revealed noticeable inter-class misclassification. Out of 77 Benign samples, 71 were correctly classified, while six were misclassified, primarily as Early Pre-B (3 samples), Pre-B (2 samples), and Pro-B (1 sample). For the Pre-B class, 141 out of 144 samples were correctly predicted, with three misclassified as Early Pre-B. The Pro-B class exhibited relatively higher confusion, with 101 out of 120 samples correctly identified; misclassifications mainly occurred toward Pre-B (14 samples) and Benign (4 samples).

Similarly, for the Early Pre-B class, 125 out of 147 samples were correctly classified, while 22 were misclassified, predominantly as Benign (11 samples) and Pre-B (11 samples). These results indicate considerable overlap among morphologically similar leukemia subtypes prior to fine-tuning.

**Classification performance after fine-tuning.** After fine-tuning, the model achieved a reduced test loss of 0.0974 The corresponding classification report is shown in Table 7.

**Table 7:** Classification report for the fine-tuning stage using hybrid segmentationl

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Benign | 90% | 99% | 94% | 77 |
| Pre-B | 97% | 97% | 97% | 144 |
| Pro-B | 99% | 99% | 99% | 120 |
| Early Pre-B | 98% | 93% | 95% | 147 |
| Accuracy | | | 97% | 488 |
| Macro Avg | 96% | 97% | 97% | 488 |
| Weighted Avg | 97% | 97% | 97% | 488 |

After fine-tuning, the proposed method achieved a test loss of 0.0974 and an overall accuracy of 97%, as shown in Table 7. The macro and weighted F1-scores both reached 97%, reflecting balanced performance across all classes. Precision, recall, and F1-score values range from 90% to 99%, indicating improved sensitivity to subtle morphological differences among leukemia subtypes.

Classification performance improved markedly across all classes. In the Benign class, 76 out of 77 samples were correctly classified, with only one misclassified as Early Pre-B. For the Pre-B class, 140 out of 144 samples were correctly identified, with minimal confusion distributed across Early Pre-B (2 samples), Pro-B (1 sample), and Benign (1 sample). The Pro-B class achieved near-perfect recognition, with 119 out of 120 samples correctly classified and only one misclassified as Pre-B. Likewise, Early Pre-B improved to 137 correct predictions out of 147 samples, reducing total misclassifications from 22 to 10. Overall, fine-tuning significantly reduced cross-class confusion, resulting in stronger diagonal dominance in the confusion matrix and improved class separability and model robustness.

## 3.3. Comparison of Performance Before and After Segmentation

The performance of the baseline model trained on raw images and the proposed method incorporating hybrid segmentation was compared using the same test set to ensure a fair and consistent evaluation. The baseline results reported in this section correspond to the fine-tuned EfficientNetB4 model without segmentation.

**Table 8:** Performance comparison

| Method | Segmentation | Model | Optimizer | Accuracy (%) | F1-Score (%) |
|---|---|---|---|---|---|
| Baseline | - | EfficientNetB4 | AdamW | 95% | 95% |
| Proposed | Otsu Thresholding and elliptical morphological operations | EfficientNetB4 | AdamW | 97% | 97% |

The proposed method achieved an accuracy and F1-score of 97%, improving upon the 95% obtained by the baseline model. Although the absolute gain is 2%, the improvement is consistent across both accuracy and F1-score, indicating a meaningful enhancement in classification performance. This improvement suggests that hybrid segmentation reduces background interference and emphasizes discriminative nuclear features, leading to fewer misclassifications between

morphologically similar leukemia subtypes. No statistical significance test was conducted; therefore, the results are reported as empirical performance gains rather than statistically validated improvements

## 4. Conclusion

This study aimed to develop a deep learning–based framework for classifying acute lymphoblastic leukemia from microscopic blood cell images. Experimental results show that the integration of hybrid segmentation using Otsu thresholding and elliptical morphological operations improves classification performance by emphasizing the cell nucleus as the primary region of interest and reducing background interference.

Data augmentation increased image variability and supported improved generalization without requiring additional data acquisition. In addition, EfficientNetB4 optimized with the AdamW optimizer demonstrated enhanced learning behavior after fine-tuning, leading to more discriminative feature representations and reduced misclassification across leukemia subtypes.

Although consistent performance gains were observed across accuracy and F1-score metrics, robustness claims are limited by the use of a single dataset and the absence of statistical uncertainty analysis or external validation. Future work will therefore focus on patient-level data splitting, evaluation on independent external cohorts, and robustness enhancement through domain adaptation and stain or illumination normalization techniques.

From a methodological perspective, the use of a fixed elliptical structuring element introduces mathematical constraints, particularly its sensitivity to highly irregular or elongated nuclear shapes, such as those observed in certain Pre-B cell instances. Adaptive or shape-aware morphological operators, as well as learnable segmentation modules, represent promising directions for further algorithmic refinement.

Overall, the proposed framework provides an empirically effective baseline for automated leukemia image classification, while also highlighting clear opportunities for methodological extension and broader clinical validation.

## CRediT Authorship Contribution Statement

**Maretta Mia Audina:** contributed to conceptualization, methodology, investigation, data curation, and writing of the original draft of the manuscript. **Sugiyarto Surono:** contributed to validation, methodological consultation, and review. **Aris Thobirin:** was responsible for validation, consultation, and review. **Goh Khang Wen:** contributed to validation, methodological consultation, and review.

## Declaration of Generative AI and AI-assisted technologies

Generative AI tools were used during the preparation of this manuscript to improve language quality and enhance clarity of writing. Specifically, ChatGPT was utilized to assist with grammar correction, paraphrasing, and refinement of academic language. The authors take full responsibility for the content, interpretation, and integrity of this manuscript.

## Declaration of Competing Interest

The authors declare no competing interests.

## Funding and Acknowledgments

This research received no external funding. The authors would like to acknowledge the providers of the publicly available dataset used in this study, as well as the Ahmad Dahlan University that provided computational facilities supporting the conduct of the research.

## Data and Code Availability

The dataset used in this study is publicly available through the Kaggle repository entitled "Blood Cells Cancer (ALL) Dataset." The code supporting the findings of this study is available from the corresponding author upon reasonable request.

## References

[1] I. H. Sarker, "Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions," *SN Computer Science*, vol. 2, no. 6, pp. 1–20, 2021. DOI: 10.1007/s42979-021-00815-1.

[2] S. G. A, R. Prabha, S. Sridevi, and A. Rohini, "Artificial Intelligence: Advanced Mathematical Constructs and Theoretical Framework in Machine Learning and Deep Learning Algorithms," *SSRN Electronic Journal*, 2025. DOI: 10.2139/ssrn.5066774.

[3] Z. Alshingiti, R. Alaqel, J. Al-Muhtadi, Q. E. U. Haq, K. Saleem, and M. H. Faheem, "A deep learning-based phishing detection system using cnn, lstm, and lstm-cnn," *Electronics (Switzerland)*, vol. 12, no. 1, pp. 1–18, 2023. DOI: 10.3390/electronics12010232.

[4] D. Kurniasari, A. Su, and F. R. Lumbanraja, "Evaluating User Satisfaction in the Halodoc Application Using a Hybrid CNN-BiLTSM Model for Sentiment Analysis," vol. 18, no. 2, pp. 209–225, 2025. DOI: 10.15408/jti.v18i2.42762.

[5] M. M. Taye, "Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions," *Computation*, vol. 11, no. 3, 2023. DOI: 10.3390/computation11030052.

[6] G. Rangel, J. C. Cuevas-Tello, J. Nunez-Varela, C. Puente, and A. G. Silva-Trujillo, "A survey on convolutional neural networks and their performance limitations in image recognition tasks," *Journal of Sensors*, vol. 2024, 2024. DOI: 10.1155/2024/2797320.

[7] S. Surono, M. Rivaldi, D. A. Dewi, and N. Irsalinda, "New approach to image segmentation: U-net convolutional network for multiresolution ct image lung segmentation," *Emerging Science Journal*, vol. 7, no. 2, pp. 498–506, 2023. DOI: 10.28991/ESJ-2023-07-02-014.

[8] G. S. Junior, J. Ferreira, C. Millán-arias, R. Daniel, and A. C. Junior, "Ceramic cracks segmentation with deep learning," *applied sciences*, vol. 11, no. 13, 2021. DOI: 10.3390/app11136017.

[9] V. Sravan, K. Swaraja, and M. Kollati, "Magnetic Resonance Images Based Brain Tumor Segmentation - A Critical survey," *Proceedings of the Fourth International Conference on Trends in Electronics and Informatics (ICOEI 2020)*, no. July 2021, 2020. DOI: 10.1109/ICOEI48184.2020.9143045.

[10] C. M. Wairo, S. Nugroho, and H. Suyuti, "Hotspot detection in photovoltaic module using otsu thresholding method," *Majalah Kesehatan*, 2020. DOI: 10.21776/ub.majalahkesehatan.006.01.3.

[11] S. Hong, Z. Jiang, L. Liu, J. Wang, L. Zhou, and J. Xu, "Improved Mask R-CNN Combined with Otsu Preprocessing for Rice Panicle Detection and Segmentation," *applied sciences*, vol. 12, no. 22, 2022. DOI: 10.3390/app122211701.

[12] A. Kulshreshtha and A. Nagpal, "Analysis of morphological operations on image segmentation techniques," *ICTACT Journal on Image and Video Processing*, vol. 9102, no. August, pp. 2555–2558, 2021. DOI: 10.21917/ijivp.2021.0362.

[13] R. Kumar et al., "Precision pest management in agriculture using Inception V3 and EfficientNet B4 : A deep learning approach for crop protection," *Information Processing in Agriculture*, no. September, 2025. DOI: 10.1016/j.inpa.2025.09.005.

[14] A. Zafar et al., "Convolutional Neural Networks: A Comprehensive Evaluation and Benchmarking of Pooling Layer Variants," *Symmetry*, vol. 16, no. 11, 2024. DOI: 10.3390/sym16111516.

[15] H. Farman, J. Ahmad, B. Jan, Y. Shahzad, M. Abdullah, and A. Ullah, "Efficientnet-based robust recognition of peach plant diseases in field images," *Computers, Materials and Continua*, vol. 71, no. 1, pp. 2073–2089, 2022. DOI: 10.32604/cmc.2022.018961.

[16] V. T. Hoang and K. H. Jo, "Practical Analysis on Architecture of EfficientNet," *International Conference on Human System Interaction, HSI*, vol. 2021-July, pp. 2–5, 2021. DOI: 10.1109/HSI52170.2021.9538782.

[17] P. Zhou, X. Xie, Z. Lin, and S. Yan, "Towards Understanding Convergence and Generalization of AdamW," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 9, pp. 6486–6493, 2024. DOI: 10.1109/TPAMI.2024.3382294.

[18] M. Sheykhhasan, H. Manoochehri, and P. Dama, "Use of CAR T-cell for acute lymphoblastic leukemia (ALL) treatment: a review study," *Cancer Gene Therapy*, vol. 29, no. 8-9, pp. 1080–1096, 2022. DOI: 10.1038/s41417-021-00418-1.

[19] A. Trubnikov and D. Savelyev, "Exploring Convolutional Neural Networks for the Classification of Acute Lymphoblast Leukemia Blood Cell Images," *Journal of Biomedical Photonics and Engineering*, vol. 10, no. 1, 2024. DOI: 10.18287/JBPE24.10.010302.

[20] A. A. Soladoye, D. B. Olawade, I. A. Adeyanju, T. Adereni, K. M. Olagunju, and A. C. Davidolawade, "International Journal of Medical Informatics Enhancing leukemia detection in medical imaging using deep transfer learning," *International Journal of Medical Informatics*, vol. 203, no. June, 2025.

[21] A. Hosseini et al., "A mobile application based on efficient lightweight CNN model for classification of B-ALL cancer from non-cancerous cells : A design and implementation study," *Informatics in Medicine Unlocked*, vol. 39, no. April, p. 101 244, 2023. DOI: 10.1016/j.imu.2023.101244.

[22] M. N. Razali, N. Arbaiy, and P.-c. Lin, "Optimizing Multiclass Classification Using Convolutional Neural Networks with Class Weights and Early Stopping for Imbalanced Datasets," vol. 14, no. 705, 2025. DOI: 10.3390/electronics14040705.

[23] J. Zheng, Y. Gao, H. Zhang, Y. Lei, and J. Zhang, "Otsu multi-threshold image segmentation based on improved particle swarm algorithm," *Applied Sciences*, vol. 12, no. 22, p. 11 514, 2022. DOI: 10.3390/app122211514.

[24] M. A. Javeed et al., "Lane Line Detection and Object Scene Segmentation Using Otsu Thresholding and the Fast Hough Transform for Intelligent Vehicles in Complex Road Conditions," *Electronics (Switzerland)*, vol. 12, 2023. DOI: 10.3390/electronics12051079.

[25] C. Jiarui et al., "Location and Detection Method of Ring-shaped Culture Carrier for Nucleic Acid Detection," *IEE 9th Data Driven Control and Learning Systems Conference*, no. 3182027, pp. 1257–1262, 2020. DOI: 10.1109/DDCLS49620.2020.9275178.

[26] D. A. Zebari, D. Q. Zeebaree, A. M. Abdulazeez, H. Haron, H. Nuzly, and A. Hamed, "Improved threshold based and trainable fully automated segmentation for breast cancer boundary and pectoral muscle in mammogram images," *IEEE Access*, vol. 8, pp. 203 097–203 116, 2020. DOI: 10.1109/ACCESS.2020.3036072.

[27] M. F. Abdullah, S. N. Sulaiman, and M. K. Osman, "Analysis of Image Processing Using Morphological Erosion and Dilation," *Journal of Physics: Conference Series*, vol. 2071, no. 1, 2021. DOI: 10.1088/1742-6596/2071/1/012033.

[28] C. Cao, F. Zhou, Y. Dai, J. Wang, and K. Zhang, "A Survey of Mix-based Data Augmentation: Taxonomy, Methods, Applications, and Explainability," *ACM Computing Surveys*, vol. 57, no. 2, pp. 1–41, 2024. DOI: 10.1145/3696206.

[29] T. Kumar, R. Brennan, A. Mileo, and M. Bendechache, "Image Data Augmentation Approaches: A Comprehensive Survey and Future Directions," *IEEE Access*, vol. 12, no. September, pp. 187 536–187 571, 2024. DOI: 10.1109/ACCESS.2024.3470122. eprint: 2301.02830.

[30] N. Eldeen, K. Mohamed, and L. Seyedali, "A comprehensive survey of recent trends in deep learning for digital images augmentation," *Artificial Intelligence Review*, vol. 55, no. 3, pp. 2351–2377, 2022. DOI: 10.1007/s10462-021-10066-4.

[31] S. Montaha and S. Azam, "MNet-10 : A robust shallow convolutional neural network model performing ablation study on medical images assessing the effectiveness of applying optimal data augmentation technique," no. August, 2022. DOI: 10.3389/fmed.2022.924979.

[32] A. Chaudhari, C. Bhatt, A. Krishna, and P. L. Mazzeo, "ViTFER : Facial Emotion Recognition with Vision Transformers," *applied sytem innovation*, vol. 5, no. 4, p. 80, 2022. DOI: 10.3390/asi5040080.

[33] Y. Guan et al., "A framework for efficient brain tumor classification using MRI images," *Mathematical Biosciences and Engineering*, vol. 18, no. 5, pp. 5790–5815, 2021. DOI: 10.3934/MBE.2021292.

[34] T. Ahmed and N. H. N. Sabab, "Classification and Understanding of Cloud Structures via Satellite Images with EfficientUNet," *SN Computer Science*, vol. 3, no. 1, 2022. DOI: 10.1007/s42979-021-00981-2. eprint: 2009.12931.

[35] Y. Peng, X. Li, and Y. Wang, "Quantum Squeeze-and-Excitation Networks," *Proceedings - IEEE Quantum Week 2024, QCE 2024*, vol. 2, pp. 39–43, 2024. DOI: 10.1109/QCE60285.2024.10249.

[36] M. Pendse, V. Thangarasa, V. Chiley, R. Holmdahl, J. Hestness, and D. DeCoste, "Memory Efficient 3D U-Net with Reversible Mobile Inverted Bottlenecks for Brain Tumor Segmentation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12659 LNCS, pp. 388–397, 2021. DOI: 10.1007/978-3-030-72087-2_34. arXiv: 2104.09648.

[37] W. Peng, B. Li, P. Wang, H. Huang, Y. Zou, and X. Qiao, "PVConv : Enhancing Depthwise Separable Convolution via Preference-Value Learning for Similar-Feature Discrimination," vol. 14, no. 4978, 2025. DOI: 10.3390/electronics14244978.

[38] Mark Sandler, A. Howard, M. Zhu, A. Zhmoginov, and Liang-Chieh Chen, *MobileNetV2: Inverted Residuals and Linear Bottlenecks Mark*. 2019, pp. 99–107. arXiv: arXiv:1801.04381v4.

[39] J.-G. Jang, C. Quan, H. D. Lee, and U. Kang, "FALCON: Lightweight and Accurate Convolution," no. 2017, pp. 1–18, 2019. arXiv: 1909.11321.

[40] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-Excitation Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 8, pp. 2011–2023, 2020. DOI: 10.1109/TPAMI.2019.2913372. arXiv: 1709.01507.

[41] A. Izang, "Swish activation and efficient net-based imaging with adaptive optics : Revolutionizing order picking and placing in robotic automation," *International Journal of Advance Research in Infromation Technology and Management Science*, vol. 1, no. 1, pp. 50–58, 2024.

[42] S. Kılıçarslan, K. Adem, and M. Çelik, "An overview of the activation functions used in deep learning algorithms," *Journal of New Results in Science*, vol. 10, no. 3, pp. 75–88, 2021. DOI: 10.54187/jnrs.1011739.

[43] H. Peng, Y. Yu, and S. Yu, "Re-Thinking the Effectiveness of Batch Normalization and Beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 46, no. 1, pp. 1–22, 2023. DOI: 10.1109/TPAMI.2023.3319005.

[44] A. Fariza, "Tooth and Supporting Tissue Anomalies Detection from Panoramic Radiography Using Integrating Convolution Neural Network with Batch Normalization," vol. 17, no. 2, 2024. DOI: 10.22266/ijies2024.0430.19.

[45] S. Batool et al., "Deploying efficient net batch normalizations (BNs) for grading diabetic retinopathy severity levels from fundus images," *Scientific Reports*, vol. 13, no. 1, pp. 1–13, 2023. DOI: 10.1038/s41598-023-41797-9.

[46] D. Masters, A. Labatie, Z. Eaton-Rosen, and C. Luschi, "Making EfficientNet More Efficient: Exploring Batch-Independent Normalization, Group Convolutions and Reduced Resolution Training," 2021. arXiv: 2106.03640.

[47] Y. Dogan, "A New Global Pooling Method for Deep Neural Networks: Global Average of Top-K Max- Pooling," *Traitement du Signal*, vol. 40, no. 2, pp. 577–587, 2023. DOI: 10.18280/ts.400216.

[48] K. Patel and G. Wang, "A discriminative channel diversification network for image classification," *Pattern Recognition Letters*, vol. 153, pp. 176–182, 2022. DOI: 10.1016/j.patrec.2021.12.004. arXiv: 2112.05861.

[49] K. Ali, Z. A. Shaikh, A. A. Khan, and A. A. Laghari, "Multiclass Skin Cancer Classification using EfficientNets -A First Step towards Preventing Skin Cancer," *Neuroscience Informatics*, no. December, p. 100 034, 2021. DOI: 10.1016/j.neuri.2021.100034.

[50] Q. Zhu, Z. He, T. Zhang, and W. Cui, "Improving Classification Performance of Softmax Loss Function Based on Scalable Batch-Normalization," *applied sciences*, vol. 10, no. 2950, pp. 1–8, 2020. DOI: 10.2290/app10082950.

[51] L. Nieradzik, G. Scheuermann, D. Saur, and C. Gillmann, "Effect of the output activation function on the probabilities and errors in medical image segmentation," 2021. arXiv: 2109.00903.

[52] Ö. Ezerceli and R. Dehkharghani, *Mental disorder and suicidal ideation detection from social media using deep neural networks*. Springer Nature Singapore, 2024, vol. 7, pp. 2277–2307. DOI: 10.1007/s42001-024-00307-1.

[53] I. Hossain, S. Jahan, R. Al, and K. Ahmed, "Smart Agricultural Technology Detecting tomato leaf diseases by image processing through deep convolutional neural networks," *Smart Agricultural Technology*, vol. 5, no. August, 2023. DOI: 10.1016/j.atech.2023.100301.

[54] O. Rainio, "Evaluation metrics and statistical tests for machine learning," *Scientific Reports*, pp. 1–14, 2024. DOI: 10.1038/s41598-024-56706-x.

[55] M. Cristina, H. Lee, J. Braet, and J. Springael, "Performance Metrics for Multilabel Emotion Classification : Comparing Micro , Macro , and Weighted F1-Scores," *applied sciences*, vol. 14, no. 9863, 2024. DOI: 10.3390/appl14219863.