ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

White Blood Cell Classification Using SMOTE-SVM Method with Hybrid Feature Extraction and Image Segmentation Using Gaussian Mixture Model

Tata Sutabri[1] and Celvine Adi Putra[2]

ABSTRACT

White blood cells are crucial to the immune system. The irregular structure of white blood cells, along with the fact that each type has its unique structure, makes manual identification challenging. Manual identification is prone to errors due to medical personnel's subjectivity and fatigue from time and effort demands. A fast and accurate method for classifying white blood cells is needed, but challenges remain regarding the quality and quantity of samples for each cell type. This study proposes the use of SMOTE and SVMSMOTE to address the issue of data imbalance, as well as a combination of shape features (size, circularity, convexity, solidity) and convolutional autoencoder (CAE) for feature extraction, along with a Gaussian mixture model for nucleus segmentation. The study finds that, without using SMOTE or SVMSMOTE for data balancing, the proposed features are already sufficient to represent each cell type except eosinophils, achieving an accuracy of 92.4%, precision of 91.9%, recall of 92.3%, F1-Score of 92%, MCC of 0.862, and CEN of 0.1376 using a polynomial kernel. The worst results were obtained with the sigmoid kernel.

The combined feature extraction (shape and CAE) outperformed individual methods. Shape alone achieved 86.8% accuracy, CAE alone 87.8%. Recall for eosinophil cells improved using SMOTE and SVMSMOTE.

Article information:

Keywords: Convolutional Autoencoder, Gaussian Mixture Model, SMOTE, SVMSMOTE, Support Vector Machine, White Blood Cells

Article history:
Received: June 14, 2024
Revised: September 19, 2024
Accepted: November 21, 2024
Published: December 21, 2024 (Online)

DOI: 10.37936/ecti-cit.2025191.257148

1. INTRODUCTION

The word "leukocytes" is derived from the Greek words for "leuko", meaning white, and "cyte", meaning cell. White blood cells are an integral part of the immune system and fight off many pathogens invading the body they are major cellular elements of the inflammatory and immune response that protect against infection and neoplasia and help repair damaged tissue [1]. Identifying types of white blood cells can assist medical professionals in diagnosing various kinds of diseases related to white blood cells, such as human immunodeficiency virus (HIV), AIDS, hepatitis, immune disorders, and leukaemia [2], [3]. White blood cells can be categorized into granular leukocytes and agranular leukocytes. Granular leukocytes contain granules in their cytoplasm and consist of basophils, eosinophils, and neutrophils. Agranular leukocytes do not have granules and include monocytes and lymphocytes [4]. Each type of white blood cell can be identified by its nucleus, size, and the colour of its granules [5]. Figure 1 shows a diagram of WBC structure (example using monocyte cell).

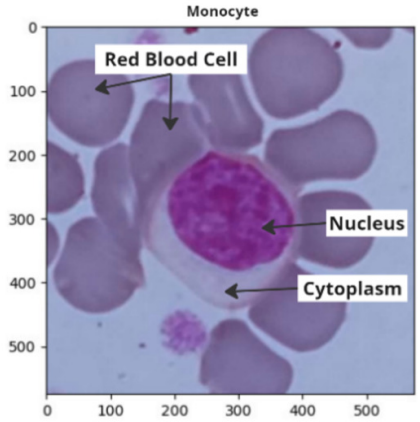

Fig. 1: *Diagram of WBC structure (monocyte cell example).*

[1,2] The authors are with the Faculty of Computer Science, Masters in Informatics Engineering, Universitas Bina Darma, Indonesia, Email: tata.sutabri@gmail.com and celvineadiputra@gmail.com

Therefore, manually identifying white blood cell types becomes complex and time-consuming for medical staff, and it is also subject to their interpretation [2]. In addition to manual methods, a haematology analyzer machine can be used. However, using this device requires regular maintenance, calibration control, and trained medical personnel, these analyzers often suffer from poor resolution and are limited to certain classes of leukocyte types, and the machine itself is expensive for developing countries [6], [7], making it unavailable in all regions, especially those with limited resources [8].

Designing a model for identifying each WBC type is essential, as this would benefit the field of medicine by helping medical teams diagnose diseases quickly and at low costs. For this, obtaining information regarding the type of white blood cells from the patients in a relatively short time would mean serving as data for diagnosing the disease being experienced [5]. This highlights the weakness of previous methods, which did not account for this imbalance [6], [9]. In the work, class imbalance will be a problem for the data used in this study.

Machine learning is one method that can be used for prediction processes. However, classifying white blood cells presents challenges, especially regarding the quality and quantity of the available data [6]. One issue with data quantity is data imbalance, where one or more classes have significantly more data than others, known as majority and minority classes. Imbalanced data can lead to misclassification, especially for the minority class, making it more difficult to predict due to its limited representation, which causes bias toward the majority class [10]. There are two main approaches to addressing data imbalance: algorithm-level approaches and data-level approaches. The data-level approach balances the class distribution, whether majority or minority, using techniques like undersampling, oversampling, or both. On the other hand, the algorithm-level approach involves modifying or optimizing the algorithm itself [10]. The data-level approach is considered more effective for handling data imbalance issues because it is more flexible and not dependent on the algorithm used [11].

2. RELATED WORKS

Previous research related to the classification of white blood cell types was widely implemented using different methods, both traditional and deep learning techniques [12]. In traditional techniques, the first step involved manual feature extraction. Meanwhile, deep learning techniques did not require manual feature extraction but needed many parameters and a large amount of training data. The dataset had to be sufficiently large to ensure the model was trained accurately [13]. Using traditional techniques, researchers examined which feature extraction methods were most suitable for white blood cell classification and how cell image segmentation could distinguish cells from the background or other cells.

In 2023, Lin *et al.* conducted research using imbalanced datasets such as "new thyroid1", Ecoli2, Wisconsin, and lung cancer. The study used SVM and various data-balancing techniques. The best results were achieved using SMOTE for the "new thyroid1" dataset, with an accuracy of 96.5%. Similarly, SMOTE achieved the best results for the Ecoli2 dataset, with an accuracy of 88.8%. For the Wisconsin and lung cancer datasets, the best results were obtained using MMTD-ELM, achieving accuracies of 90.5% and 99.8%, respectively [14].

In 2021, Tavakoli *et al.* implemented shape feature extraction (solidity, convexity, circularity) and colour feature extraction (RGB, HSV, LAB, YCrCb). Using a Support Vector Machine (SVM) for classification, they achieved an accuracy of 94.65% [13].

In 2021, Devella *et al.* implemented saliency to segment white blood cells and proposed Speeded-Up Robust Features (SURF) for feature extraction. This study achieved an accuracy of 78.60% using SVM [15].

In 2021, Yohannes *et al.* implemented saliency to segment white blood cells and proposed Scale Invariant Feature Transform (SIFT) for feature extraction. This study achieved an accuracy of 77.08% using SVM [16].

n 2020, Riaz *et al.* used the Gaussian Mixture Model for medical image segmentation. The segmentation was applied to three distinct imaging modalities: MRI, dermoscopy, and chromoendoscopy. This study showed that the proposed method provided better qualitative and quantitative results than existing medical image segmentation methods [17].

In 2019, Wang *et al.* conducted a study using a combination of colour, texture, and parameter values such as area, solidity, eccentricity, and perimeter. They used the Gray Level Co-occurrence Matrix (GLCM) to extract texture features and used Support Vector Machines (SVM) for classification. The study used 200 images, with 40 images for each cell type. The method achieved a best accuracy of 88.5% [18].

In 2019, Mery *et al.* implemented a Convolutional Autoencoder (CAE) for feature extraction in plant leaf classification. The features extracted by the CAE were classified using SVM, achieving an accuracy of 94.74% [19].

3. THE PROPOSED METHODS

3.1 Dataset

The dataset used in this paper was provided by previous research [5]. There are three files: train.rar, testA.rar, and testB.zip. Only train.rar and testA.rar have been labelled manually by two medical professionals. Therefore, this study will only use these two files. Each file contains jpg format images with a resolution of 575 × 575 pixels. These types are

basophils, eosinophils, neutrophils, monocytes, and lymphocytes. The data distribution is as follows: For the test file, there are 212 basophils, 744 eosinophils, 6231 neutrophils, 561 monocytes, and 2327 lymphocytes. For the training file, there are 89 basophils, 332 eosinophils, 2660 neutrophils, 231 monocytes, and 1034 lymphocytes. Figure 1 provides an example of the dataset used.

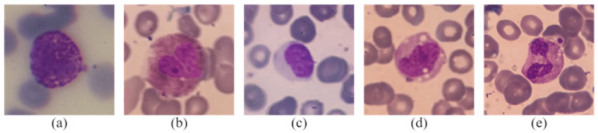

Fig.2: *Example of (a) Basophil, (b) Eosinophil, (c)Lymphocyte, (d) Monocyte, (e) Neutrophil.*

The dataset will undergo further manual selection to remove damaged or incomplete cell images and images containing more than one cell type. The resulting data distribution will be as follows: 179 basophils, 655 eosinophils, 5633 neutrophils, 545 monocytes, and 2375 lymphocytes for the train.rar file; and 77 basophils, 282 eosinophils, 2581 neutrophils, 227 monocytes, and 1028 lymphocytes for the testA.rar file. Thus, this data will be used for the next step.

3.2 Methodology

Figure 3 demonstrates the overall research method applied for this study. There will be some main stages involved in this study. First, the process will be converting the RGB images to HSV. After obtaining the HSV images, the next process is the segmentation of WBC nuclei. This segmentation process will produce images of the same size as the input images. The next step in the process is feature extraction, where two kinds of features will be used: shape features, such as size, solidity, convexity, and circularity, and the features obtained from a convolutional autoencoder. These extracted features are combined for the next stage, the testing scenarios. There will be three testing scenarios: the first without data balancing, the second using SMOTE, and the third using SVMSMOTE. Each scenario will be classified using SVM. The details of the steps will be mentioned in the next sections.
The explanation of Figure 3 is as follows:

3.2.1 RGB to HSV

In this study, an RGB image of 575 × 575 pixels will be used as input. The image will be converted to HSV to facilitate image processing, ensuring the subsequent segmentation process achieves more accurate clustering results. The following are examples of the results from the conversion to the S channel for each cell type.

Fig.3: *The Proposed System Flow.*

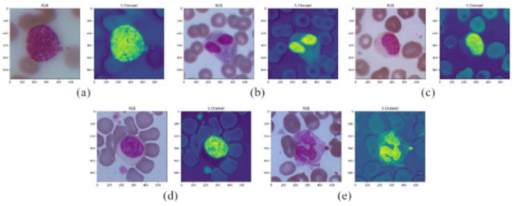

Fig.4: *Examples of the conversion results from RGB to the S channel for each cell type (a) Basophil, (b) Eosinophil, (c)Lymphocyte, (d) Monocyte, (e) Neutrophil.*

3.2.2 Segmentation

Four distinct steps were carefully implemented in this study's proposed nucleus segmentation phase. The initial step utilizes a Gaussian Mixture Model (GMM).

GMM is a probabilistic model used to represent a dataset as a combination of multiple Gaussian distributions. Each cluster in the dataset is modelled as a separate Gaussian distribution, characterized by its mean and variance. GMM helps estimate the parameters of these distributions, specifically the mean, variance, and weight (π) for each cluster. The probability density function of Gaussian distribution, de-

fined at 1 [20]:

$$f(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \quad (1)$$

The calculation steps for GMM are below [20]:
1. Initialise mean, variance, and weight for all clusters

$$\mu_k = \frac{k * \max(j)+1}{N+1} = \frac{\max(j)+1+\pi(c_j)}{\sigma} = \frac{1}{N} \quad (2)$$

2. Suppose the probability x_i of belonging to any class c_j

$$p(c_j|x_i) = \frac{p(x_i|c_j)\cdot p(c_j)}{p(x_i)} \quad (3)$$

$$p(x_i|c_j) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x_i-\mu_j}{\sigma_j}\right)^2\right) \quad (4)$$

$$p(x_i) = \sum\nolimits_j p(x_i|c_j)\cdot p(c_j) \quad (5)$$

3. Re-estimate the parameter based on the calculated probabilities

$$\mu_k = \frac{\sum_i p(C_i|x_i)\cdot x_i}{\sum_i p(C_i|x_i)} \quad \sigma_k^2 = \frac{\sum_i p(C_i|x_i)\cdot (x_i-\mu_k)^2}{\sum_i p(C_i|x_i)} \quad (6)$$

$$p(C_j) = \frac{\sum_i p(C_j|x_i)}{n} \quad (7)$$

4. Iterate until convergence

Unlike other clustering algorithms, such as K-Nearest Neighbours (KNN), the Gaussian Mixture Model (GMM) provides several advantages. First, GMM employs a probabilistic approach, which allows it to model clusters by combining multiple probability distributions, leading to a more flexible and robust identification of cluster boundaries. Additionally, GMM excels in parameter estimation, as it can accurately estimate the mean and variance of each cluster's probability distribution. Another key advantage is that GMM performs soft clustering, meaning that each data point can belong to more than one cluster with a certain probability, which provides a more nuanced and realistic clustering result compared to the hard clustering approach of KNN. These features make GMM particularly effective in handling complex and overlapping clusters.

In this study, we apply GMM with three components (n_components=3), a parameter optimized through extensive trial and error. This clustering step is crucial for effectively distinguishing the nuclei from other structures within the cellular images of white blood cells.

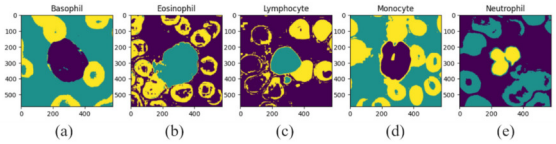

Fig.5: *Example of the clustering result using a Gaussian mixture model.*

As shown in Figure 5, colours do not correspond to any specific object. This paper introduces an approach to address this issue: the centroid method enhances segmentation by employing clustering. It generates centroids and identifies additional points around the initial centroid to detect more nuclei. Figure 6 presents the centroid method in red. Inspired by the human visual system, it focuses on objects of interest to obtain a clear perception of them.

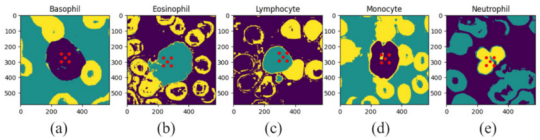

Fig.6: *Example image centre point.*

The next step involves applying morphological operations (opening and closing) and masking. These operations are intended to refine the clustering results before proceeding with the masking step.

Morphological operations, such as opening and closing, are essential techniques in image processing that improve the structure of objects in binary images. The opening operation removes small objects or noise from the foreground while preserving the shape and size of larger objects, making it particularly useful for separating connected components. Conversely, the closing operation fills small holes and gaps within the foreground objects, resulting in a more solid representation.

The masking step produces an RGB nucleus image the same size as the input, which is 575 × 575 pixels. Figure 7 provides an example of the result of this process.

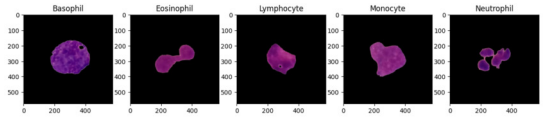

Fig.7: *Example of the final result of the segmentation process.*

3.2.3 Feature Extraction

In image processing, feature extraction refers to identifying and isolating key attributes from raw data to enhance the accuracy of subsequent analysis and classification.

In this study, we combine the results of shape feature extraction (solidity, convexity, circularity, and size) with the results from convolutional autoencoder feature extraction.

Shape Feature Extraction: In this work, two methods of feature extraction will be proposed: the first one is related to shape features and includes extraction by size, circularity, convexity, and solidity, based on the statement in the journal [13], [21] which mentions that each cell type can be distinguished based on size, shape, and colour of granules. Shape feature extraction for size provides information about the size of an object, circularity indicates how close an object is to a perfect circle with a maximum value of one (indicating perfect roundness) [22], convexity describes how close an object is to its convex hull [23], and solidity indicates the convexity or concavity of an object [24]. The definitions for circularity, convexity, and solidity are provided as follows:

$$circularity = 4\pi \times \frac{area}{(perimeter)^2} \tag{8}$$

$$\text{Solidity} = \frac{\text{ObjectArea}}{\text{ConvexArea}} \tag{9}$$

$$\text{convexity} = \frac{\text{Convex perimeter}}{\text{ObjectPerimeter}} \tag{10}$$

Convolutional Autoencoder as Feature Extraction: An autoencoder, an unsupervised learning technique, consists of an encoder and a decoder [25]. The former compresses the input data into a lower-dimensional representation, called latent space, while the latter reconstructs data from this latent space. Training for the autoencoder involves reconstruction errors between the original input and the output created by the decoder. One of the many varieties of autoencoders is the Convolutional Autoencoder, which aims to decrease data dimensionality and extract fundamental characteristics from images using Convolutional Neural Networks. Convolutional autoencoders have also been applied in image denoising [26]. The autoencoder is applied after segmentation to reduce residual noise and irrelevant features. While segmentation isolates the region of interest, there might still be imperfections, such as noise or unimportant data in the segmented region. The convolutional autoencoder thus helps refine the features by learning a compact, low-dimensional representation, which aids in more accurate classification.

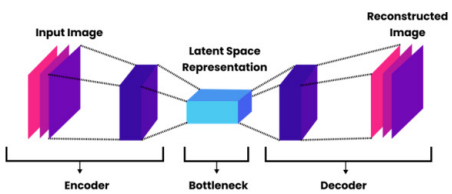

Fig.8: *Structure from Convolutional Autoencoder.*

The process begins with the input image at the encoder layer, where the encoder translates the input and compresses it into a more straightforward representation. The output from the encoder is then passed to the decoder to regenerate the input [27]. Using a Convolutional Autoencoder (CAE) for feature extraction involves the entire process from the encoder to the latent space, or it may include using *max_pooling2d_2*, as shown in Table 1. Table 1 presents the model summary of the CAE used in this study.

Table 1: *Model Summary Convolutional Autoencoder.*

Layers (Type)	Output Shape	Parameter
InputLayer	[64, 64, 3]	0
conv2d	[64, 64, 16]	448
max_pooling2d	[32, 32, 16]	0
conv2d_1	[32, 32, 8]	1160
max_pooling2d_1	[16, 16, 8]	0
conv2d_2	[16, 16, 8]	584
max_pooling2d_2	[8, 8, 8]	0
conv2d_3	[8, 8, 8]	584
up_sampling2d	[16, 16, 8]	0
conv2d_4	[16, 16, 8]	584
up_sampling2d_1	[32, 32, 8]	0
conv2d_5	[32, 32, 16]	1168
up_sampling2d_2	[64, 64, 16]	0
conv2d_6	[64, 64, 3]	435

The optimizer used in this study is Adam, with 20 epochs and a batch size of 100. Figure 9 presents the model's loss results, with the loss values ranging from 0.0379 to a validation loss (val_loss) of 0.0396.

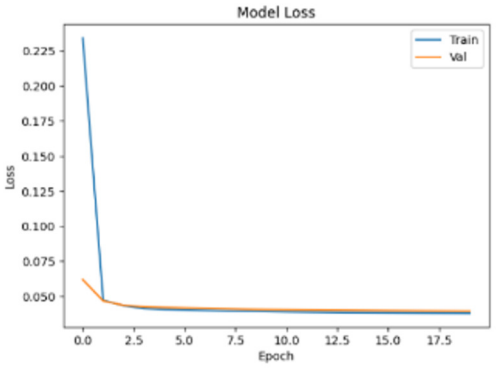

Fig.9: *Model Loss Convolutional Autoencoder.*

The results of shape feature extraction and the Convolutional Autoencoder (CAE) will be combined, resulting in the number of features displayed in Table 2. This combination results in 516 features, which will be used in the next stage.

3.2.4 K-Fold Cross Validation

K-Fold cross-validation is a technique used to split data into training and testing sets. With the k-fold

Table 2: *Sum of Shape Features and CAE Features.*

Feature Extraction	Sum of Feature
Shape Feature Extraction	4
Convolutional Autoencoder	512

method, data is randomly divided into K-equals sizes. One subset is used as the testing data, while the remaining subsets are used as training data. This classification process is repeated K times[28]. In this study, K=5 will be considered for every type of test to be more fair.

3.2.5 Classification by Support Vector Machine

This section will explain the classification process using SVM with the kernel trick, based on the combined feature extraction performed in the previous stage. This process will include three types of testing: the first without class balancing, the second with SMOTE, and the third with SVMSMOTE.

The primary challenge in this research is the significant class imbalance in the white blood cell classification dataset. Specifically, certain types of white blood cells are much less represented compared to others, resulting in a bias in the model toward the majority classes, which hampers the accurate identification of minority classes. To address this, data balancing algorithms such as SMOTE (Synthetic Minority Over-sampling Technique) and its variation, SVMSMOTE, have been utilized.

SMOTE is applied to handle the class imbalance by generating synthetic examples for the minority class [29], thereby increasing its representation in the training dataset. The synthetic data is generated by finding the nearest neighbours of existing minority samples and creating interpolations, which helps prevent overfitting [30]. This synthetic oversampling ensures that the training set is more balanced, allowing the model to effectively learn the characteristics of the minority classes rather than being overwhelmed by the majority class. For this study, SMOTE was essential to ensure adequate representation of the less frequent white blood cell types, ultimately improving classification performance across all cell types.

However, SMOTE has a limitation known as overgeneralization, which refers to its potential ineffectiveness in generating meaningful synthetic samples when the minority class distribution is complex [31]. To address this limitation, SVMSMOTE, a variant of SMOTE, was introduced. SVMSMOTE focuses on samples near the decision boundary between the minority and majority classes. By leveraging SVM to identify borderline instances, SVMSMOTE aims to generate more informative synthetic samples, which are crucial for distinguishing between classes in challenging areas. This approach is particularly beneficial in this research, as the decision boundaries between different types of white blood cells are often difficult to define clearly due to overlapping characteristics [32].

Using SMOTE or SVMSMOTE enables us to build a more robust classifier by effectively tackling the issues associated with class imbalance. By applying oversampling only to the training subset and leaving the test subset unchanged, the model is trained on balanced data while still being evaluated on realistic, imbalanced data that represents actual conditions. This approach ensures that the model can learn features from balanced classes while being tested in a way that simulates real-world scenarios. Consequently, the model trained with SMOTE or SVMSMOTE shows improved accuracy in identifying minority cell types, as evidenced by reduced misclassification rates and increased recall for underrepresented classes.

Support Vector Machine (SVM): SVM belongs to the supervised learning category, meaning the data must be labelled beforehand before training. The goal of SVM is to obtain the best possible hyperplane to separate classes by optimizing the margin or decision boundary, which can separate classes linearly [33]. Initially, SVM could only be used on linear data, but with its development, it can now be applied to non-linear data using kernel functions. These kernel functions map from low to higher dimensions, with several commonly used types such as radial basis function (RBF), linear, polynomial, and sigmoid. The RBF kernel can be defined by equation 11, the polynomial kernel by equation 12, the linear kernel by equation 13, and the sigmoid kernel by equation 14.

$$K(x_i, x_j) = x_i^T x_j \tag{11}$$

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x\|^2, \gamma > 0 \tag{12}$$

$$K(x_i, x_j) = (y \cdot x_i^T x + r)^2 \tag{13}$$

$$K(x_i, x_j) = \tanh(\sigma(x_i, x_j) + c) \tag{14}$$

A key parameter in Support Vector Machines (SVM) is C, a regularization term that balances the trade-off between maximizing the margin and minimizing classification errors. A smaller value of C allows for a wider margin while permitting more classification errors, making the model more resilient to overfitting by allowing some misclassifications. In contrast, a larger value of C forces the model to classify all data points correctly, resulting in a narrower margin and a more complex model, which can increase the risk of overfitting [34].

Selecting an appropriate C value is essential for optimal SVM performance. Cross-validation typically determines this value, a technique to find the best balance between model complexity and prediction ac-

curacy.

Initially, SVM was designed solely for binary classification problems. However, various strategies have been developed to address multi-class classification using SVM. One such strategy is the one-against-all approach. In this method, one class is evaluated against all other classes. For example, in a three-class scenario, SVM would differentiate class 1 from classes 2 and 3, class 2 from classes 1 and 3, and class 3 from classes 1 and 2. [35].

3.3 Evaluation Matrix

The evaluation metrics used to assess the performance of the models for white blood cell classification in this study are as follows.

1. Accuracy

Accuracy is determined by dividing the number of correctly predicted data points by the total number of data points [36]. The formula is as follows:

$$Accuracy = \frac{TP + TN}{TP + Tn + FP + FN} \tag{15}$$

2. Precision

Precision is calculated by dividing the number of true positive (TP) predictions by the total number of positive predictions [37]. The formula is as follows:

$$Precision = \frac{TP}{TP + FP} \tag{16}$$

3. Recall

Recall is calculated by dividing the number of true positive (TP) predictions by the total number of positive data [37]. The formula is as follows:

$$Recall = \frac{TP + TN}{TP + FN} \tag{17}$$

4. F1-Score

F1-Score is the harmonic mean of precision and recall, resulting in a single value that ranges from 0 to 1. A value of 1 represents the best possible precision and recall, while a value of 0 indicates the lowest precision and recall. [37]. The formula is as follows:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{18}$$

5. Matthews correlation coefficient (MCC)

The Matthews correlation coefficient (MCC) will still be calculated using a confusion matrix. The MCC value ranges from -1 to +1, where +1 indicates the best performance and -1 indicates the worst. A value of 0 indicates a useless result or a result equivalent to random guessing [38]. The MCC equation for multi-class [39] is defined below.

$$\text{MCC} = \frac{\sum_{k,l,m=1}^{N} C_{kk} C_{ml} - C_{lk} C_{km}}{\sqrt{\sum_{K=1}^{N} \left[\left(\sum_{l=1}^{N} C_{lk} \right) \left(\sum_{f,g=1 f \neq k}^{N} C_{gf} \right) \right]} \sqrt{\sum_{k=1}^{N} \left[\left(\sum_{l=1}^{N} C_{kl} \right) \left(\sum_{f,g=1 f \neq kC_{fg}}^{N} \right) \right]}} \tag{19}$$

6. Confusion entropy (CEN)

Confusion entropy (CEN) is a metric used to evaluate classification performance in machine learning. CEN can be interpreted as the "average error" in classification. A low CEN value indicates that the classification produces few errors, while a high CEN value indicates that the classification produces many errors. In addition, CEN becomes one of the evaluation metrics in the imbalanced multi-class case [40], CEN equation is defined as follows [18]:

$$CEN = \sum_{i=1}^{M} P_i \cdot CEN_i \tag{20}$$

$$P_i = \frac{\sum_{K=1}^{C} mat_{i,k} + mat_{k,i}}{2 * \sum_{k,l=1}^{C} mat_{k,l}} \tag{21}$$

$$\text{CEN}_i = -\sum_{K=1,k\neq i}^{C} (P_{j,k}^{j} \log_{2(C-1)} (P_{k,k}^{i} \log_{2(C-1)} (P_{k,i}^{i}))) \tag{22}$$

4. RESULTS AND DISCUSSION

Based on the results of an experiment using four types of SVM kernels: linear, radial basis function (RBF), sigmoid, and polynomial. As previously mentioned, the experiment involves three types of testing: without data balancing, with SMOTE, and with SVMSMOTE. Each type of testing uses k-fold cross-validation with K=5.

4.1 Without data balancing

Table 3 presents the performance of the classification model using four different kernel functions: Linear, RBF, Sigmoid, and Polynomial. Various metrics such as Accuracy, Precision, Recall, F1-Score, MCC, and CEN are evaluated for each kernel. The parameters used for each kernel are as follows: Linear kernel (C=10), RBF kernel (C=100, gamma=Scale), Sigmoid kernel (C=100, gamma=Scale), and Polynomial kernel (C=1, gamma=Scale, degree=5).

Table 3: *Best Results for Each Kernel Type In Tests Without Data Balancing.*

	Linear	RBF	Sigmoid	Poly
Accuracy	91.5%	92.2%	85.8%	92.4%
Precision	91.5%	91.8%	84.2%	91.9%
Recall	91.5%	92.2%	85.9%	92.3%
F1-Score	91.1%	91.9%	84.9%	92%
MCC	0.848	0.860	0.746	0.862
CEN	0.1475	0.1396	0.2036	0.1376

Accuracy: The Polynomial kernel achieved the highest accuracy (92.4%), followed closely by the RBF kernel (92.2%) and the Linear kernel (91.5%). The Sigmoid kernel resulted in the lowest accuracy, at 85.8%.

Precision: Similar to accuracy, the Polynomial kernel showed superior precision (91.9%), slightly

outperforming the RBF (91.8%) and Linear kernels (91.5%). The Sigmoid kernel also underperformed here, with a precision of 84.2%.

Recall: The best recall was achieved by the Polynomial kernel (92.3%), closely followed by the RBF kernel (92.2%), while the Linear and Sigmoid kernels had lower recall scores at 91.5% and 85.9%, respectively.

F1-Score: The Polynomial kernel again demonstrated the best F1-Score (92.0%), while the RBF (91.9%) and Linear (91.1%) kernels performed similarly. The Sigmoid kernel scored the lowest (84.9%).

MCC: The highest MCC value was achieved by the Polynomial kernel (0.862), followed by the RBF (0.860) and Linear (0.848) kernels. The Sigmoid kernel had the lowest MCC (0.746), indicating weaker performance.

CEN: The lowest CEN value (indicating superior performance) was observed with the Polynomial kernel (0.1376), closely followed by the RBF kernel (0.1396). The Sigmoid kernel exhibited the highest CEN (0.2036), indicating a less effective classifier.

Based on these results, the Polynomial kernel performed the best overall, with the highest accuracy (92.4%), MCC (0.862), and lowest CEN (0.1376). Although the Polynomial kernel demonstrates overall superior performance, it is important to note that it struggles to identify certain cell types, particularly eosinophil cells, with the recall dropping to 52.8%, as shown in Figure 10. However, the recall for other cell types remains above 80%.

Fig.10: *Comparison of precision, recall, and F1-score using the polynomial kernel without data balancing, (B) Basophils, (E) Eosinophils, (L) Lymphocytes, (M) Monocytes, (N) Neutrophils.*

The test results indicate an improvement when combining shape features and convolutional autoencoders, compared to using shape features alone or convolutional autoencoders alone. The approach using the combined features achieves an accuracy of 92%. Detailed metrics for precision, recall, and F1-score are shown in Figure 10.

In contrast, using only shape features results in an accuracy of 85.2%, with detailed metrics presented in Figure 11, using a polynomial kernel (C = 0.1, gamma = 10, degree = 5). Meanwhile, using convolutional autoencoders alone yields an accuracy of 87%, with detailed metrics shown in Figure 12, using a polynomial kernel (C = 0.1, gamma = 5, degree = 5).

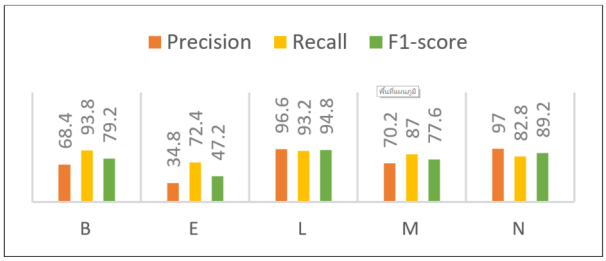

Fig.11: *Comparison of precision, recall, and F1-score using the polynomial kernel without data balancing (shape feature extraction), (B) Basophils, (E) Eosinophils, (L) Lymphocytes, (M) Monocytes, (N) Neutrophils.*

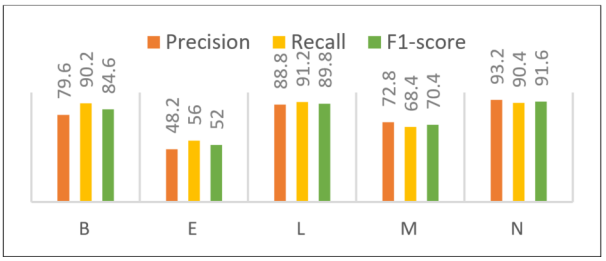

Fig.12: *Comparison of precision, recall, and F1-score using the polynomial kernel without data balancing (CAE feature extraction), (B) Basophils, (E) Eosinophils, (L) Lymphocytes, (M) Monocytes, (N) Neutrophils.*

4.2 SMOTE

Table 4 presents the performance of the classification model using four different kernel functions: Linear, RBF, Sigmoid, and Polynomial. Various metrics such as Accuracy, Precision, Recall, F1-Score, MCC, and CEN are evaluated for each kernel. The parameters used for each kernel are as follows: Linear kernel (C=1.25), RBF kernel (C=100, gamma=Scale), Sigmoid kernel (C=10, gamma=Scale), and Polynomial kernel (C=4, gamma=Scale, degree=5).

Table 4: *Best Results for Each Kernel Type In Tests Using SMOTE.*

	Linear	RBF	Sigmoid	Poly
Accuracy	88%	89.4%	80.4%	89.4%
Precision	91.1%	91%	86.7%	90.9%
Recall	87.8%	89.3%	80.3%	89.3%
F1-Score	88.9%	90%	82.3%	89.9%
MCC	0.803	0.820	0.700	0.819
CEN	0.1794	0.1710	0.2491	0.1729

Accuracy: The highest accuracy was achieved by the RBF and Polynomial kernels, both at (89.4%), while the Linear kernel was close behind with 88%. The Sigmoid kernel had the lowest accuracy (80.4%).

Precision: The Linear kernel had the highest precision (91.1%), followed closely by the Polynomial (90.9%) and RBF (91%) kernels. The Sigmoid kernel trailed behind with a precision of 86.7%.

Recall: The RBF and Polynomial kernels showed the highest recall (89.3%), with the Linear kernel slightly lower (87.8%). The Sigmoid kernel once again underperformed with a recall of 80.3%.

F1-Score: The RBF kernel produced the best F1-Score (90%), with the Polynomial kernel closely following at 89.9%. The Linear kernel had an F1-Score of 88.9%, while the Sigmoid kernel had the lowest (82.3%).

MCC: The highest MCC was achieved by the RBF kernel (0.820), followed closely by the Polynomial kernel (0.819) and the Linear kernel (0.803). The Sigmoid kernel had a significantly lower MCC (0.700), indicating weaker classification performance.

CEN: The lowest CEN (indicating better classification performance) was observed with the RBF kernel (0.1710), followed by the Polynomial (0.1729) and Linear (0.1794) kernels. The Sigmoid kernel had the highest CEN value (0.2491), reflecting its poorer performance.

In summary, the RBF and Polynomial kernels performed best overall, particularly with identical accuracy (89.4%) and similar F1-Scores, MCC, and CEN values. However, the RBF kernel slightly outperforms the Polynomial kernel regarding MCC (0.820 vs. 0.819) and CEN (0.1710 vs. 0.1729), indicating that the RBF kernel is marginally more effective in classification. Additionally, the RBF kernel has a slightly better F1 Score (90% vs. 89.9%). Based on these subtle differences, the RBF kernel can be considered the best overall performing kernel in this scenario. The Linear kernel also demonstrated strong performance, particularly in precision (91.1%), although its overall results were slightly lower than those of the RBF and Polynomial kernels. The Sigmoid kernel, by contrast, consistently yielded the weakest results across all metrics, with its performance noticeably lagging behind the others.

Based on these results, using SMOTE with the RBF kernel achieved the best accuracy of 89.4%, an MCC of 0.820, and the lowest CEN value of 0.1710. The comparison between precision, recall, and F1-Score for each type of cell type with RBF kernel is presented in Figure 13. These results clearly show that applying SMOTE improved the recall from 52.8% without data balancing to 70.4% with SMOTE. However, after applying SMOTE, the precision for eosinophil cells decreased to 48%, indicating that the model made more errors in predicting eosinophil cells.

The test results indicate an improvement when combining shape features and convolutional autoencoders, compared to using shape features alone or convolutional autoencoders alone. The approach using the combined features achieves an accuracy of 89.4%. Detailed metrics for precision, recall, and F1-score are shown in Figure 13.

Fig.13: *Comparison of precision, recall, and F1-score using the polynomial kernel using SMOTE, (B) Basophils, (E) Eosinophils, (L) Lymphocytes, (M) Monocytes, (N) Neutrophils.*

In contrast, using only shape features results in an accuracy of 85%, with detailed metrics presented in Figure 14, using an RBF kernel (C = 100, gamma = 10). Meanwhile, using convolutional autoencoders alone yields an accuracy of 86.8%, with detailed metrics shown in Figure 15, using an RBF kernel (C = 0.1, gamma = 5).

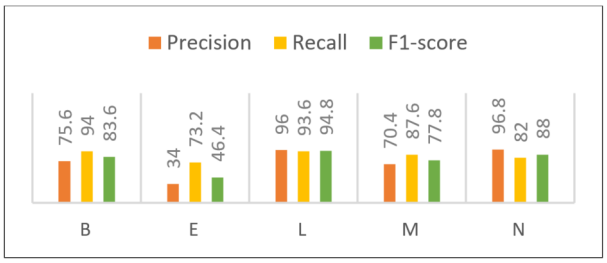

Fig.14: *Comparison of precision, recall, and F1-score using the RBF kernel SMOTE (shape feature extraction), (B) Basophils, (E) Eosinophils, (L) Lymphocytes, (M) Monocytes, (N) Neutrophils.*

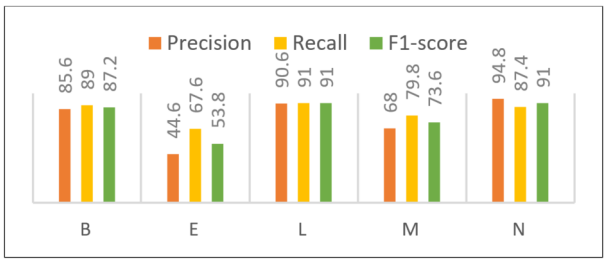

Fig.15: *Comparison of precision, recall, and F1-score using the RBF kernel SMOTE (CAE), (B) Basophils, (E) Eosinophils, (L) Lymphocytes, (M) Monocytes, (N) Neutrophils.*

4.3 SVMSMOTE

Table 5 presents the performance of the classification model using four different kernel functions: Linear, RBF, Sigmoid, and Polynomial. Various metrics

such as Accuracy, Precision, Recall, F1-Score, MCC, and CEN are evaluated for each kernel. The parameters used for each kernel are as follows: Linear kernel (C=2), RBF kernel (C=100, gamma=Scale), Sigmoid kernel (C=5, gamma=Scale), and Polynomial kernel (C=10, gamma=Scale, degree=5).

Table 5: *Best Results for Each Kernel Type In Tests Using SVMSMOTE.*

	Linear	RBF	Sigmoid	Poly
Accuracy	89.2%	90%	78.5%	90.4%
Precision	90.5%	90.7%	83.8%	90.7%
Recall	89%	90.1%	78.5%	90.3%
F1-Score	89.5%	90.3%	79.7%	90.4%
MCC	0.817	0.832	0.680	0.836
CEN	0.1779	0.1689	0.2680	0.1625

Accuracy: The Polynomial kernel achieved the highest accuracy (90.4%), slightly outperforming the RBF kernel (90%) and the Linear kernel (89.2%). The Sigmoid kernel yielded the lowest accuracy at 78.5%.

Precision: Both the RBF and Polynomial kernels produced the highest precision (90.7%), followed closely by the Linear kernel (90.5%). The Sigmoid kernel trailed behind with a precision of 83.8%.

Recall: The Polynomial kernel demonstrated the highest recall (90.3%), followed closely by the RBF kernel (90.1%). The Linear kernel had a slightly lower recall at 89%, while the Sigmoid kernel showed the weakest recall performance at 78.5%.

F1-Score: The Polynomial kernel again displayed the highest F1-Score (90.4%), closely followed by the RBF kernel (90.3%) and the Linear kernel (89.5%). The Sigmoid kernel had the lowest F1-Score at 79.7%.

MCC: The highest MCC value was achieved by the Polynomial kernel (0.836), closely followed by the RBF kernel (0.832). The Linear kernel performed well with an MCC of 0.817, while the Sigmoid kernel showed the weakest MCC at 0.680, indicating weaker classification performance.

CEN: The lowest CEN (indicating better performance) was achieved by the Polynomial kernel (0.1625), followed closely by the RBF kernel (0.1689). The Linear kernel showed a CEN value of 0.1779, while the Sigmoid kernel had the highest CEN (0.2680), indicating less effective classification.

In summary, while both the RBF and Polynomial kernels performed strongly across most metrics, the Polynomial kernel slightly outperformed the RBF kernel, particularly in terms of accuracy (90.4% vs 90%), recall (90.3% vs 90.1%), and MCC (0.836 vs 0.832). The Polynomial kernel also demonstrated the lowest CEN (0.1625), making it the best overall kernel for this dataset. Despite the RBF kernel being a close competitor, the Polynomial kernel can be considered the best-performing kernel based on these results.

Based on these results, using SVMSMOTE with a polynomial kernel yielded the best accuracy at 90.4%, MCC of 0.836, and the lowest CEN value of 0.1625. A comparison of precision, recall, and F1-Score for each cell type with a polynomial kernel is shown in Figure 16. The results indicate that the model generated after using SVMSMOTE still obtained the lowest score for eosinophil cells. However, there was no significant difference in precision and recall with SVMSMOTE, with precision at 63.8% and recall at 56.8%.

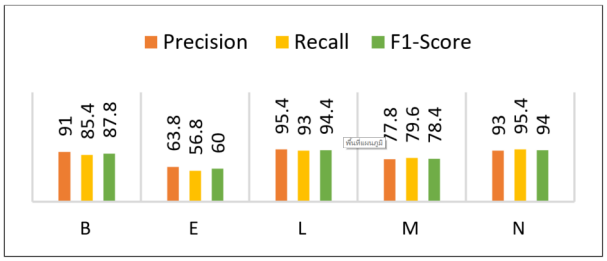

Fig.16: *Comparison of precision, recall, and F1-score using the polynomial kernel using SVMSMOTE, (B) Basophils, (E) Eosinophils, (L) Lymphocytes, (M) Monocytes, (N) Neutrophils.*

The test results indicate an improvement when combining shape features and convolutional autoencoders, compared to using shape features alone or convolutional autoencoders alone. The approach using the combined features achieves an accuracy of 90.4%. Detailed metrics for precision, recall, and F1-score are shown in Figure 16.

In contrast, using only shape features results in an accuracy of 86.8%, with detailed metrics presented in Figure 17, using a polynomial kernel (C = 10, gamma = Scale, degree=5). Meanwhile, using convolutional autoencoders alone yields an accuracy of 87.8%, with detailed metrics shown in Figure 15, using a polynomial kernel (C = 10, gamma = Scale, degree = 5).

Fig.17: *Comparison of precision, recall, and F1-score using the polynomial kernel SVMSMOTE (shape feature extraction), (B) Basophils, (E) Eosinophils, (L) Lymphocytes, (M) Monocytes, (N) Neutrophils.*

4.4 Comparison of the best result in each scenario

Figure 19 compares the best results across all scenarios. The highest accuracy, precision, recall, and F1-score were achieved using the polynomial kernel

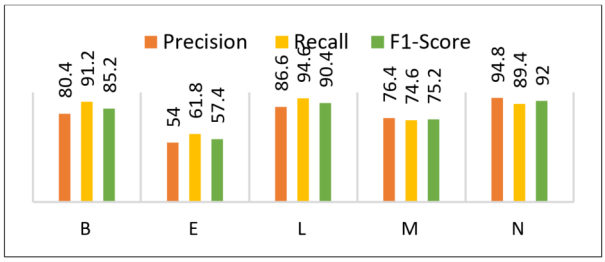

Fig.18: *Comparison of precision, recall, and F1-score using the polynomial kernel SVMSMOTE (CAE), (B) Basophils, (E) Eosinophils, (L) Lymphocytes, (M) Monocytes, (N) Neutrophils.*

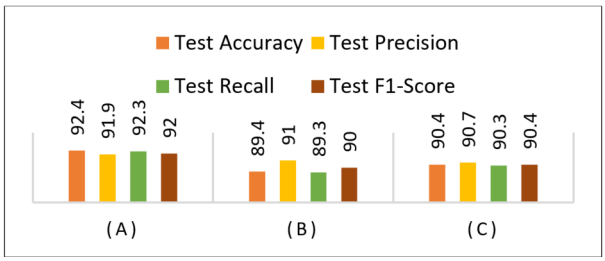

Fig.19: *Comparison of the best result in each scenario, (A) Without data balancing using polynomial kernel, (B) SMOTE using RBF kernel, (C) SVMSMOTE using a polynomial kernel.*

without data balancing, with an accuracy of 92.4%, precision of 91.9%, recall of 92.3%, and an F1-score of 92%. Conversely, the lowest scores were recorded when using SMOTE, with an accuracy of 89.4%, precision of 91%, recall of 89.3%, and an F1-score of 90%.

In this experiment, when comparing the performance of SMOTE and SVMSMOTE, better results were achieved using SVMSMOTE, with an accuracy of 90.4%, precision of 90.7%, recall of 90.3%, and an F1-score of 90.4%.

White blood cell classification using different machine learning techniques has significant potential to enhance healthcare services by enabling early detection and reducing the risks associated with manual identification errors. This capability will be crucial in providing medical professionals with information regarding increases or decreases in the counts of specific cell types. With further development, accurate cell count estimation can be achieved. This would greatly assist in clinical decision-making by enabling the rapid and precise diagnosis of various haematological diseases, including leukaemia, AIDS, HIV, immune disorders, and other blood-related conditions.

The experimental results indicate that the developed model successfully detects all cell types except eosinophils. This limitation suggests that the model faces challenges in providing accurate information for diseases associated with elevated eosinophil counts, such as asthma, allergic rhinitis, and dermatitis. Despite this limitation, the model remains reliable for detecting other cell types.

Machine learning-based analysis of white blood cell images not only improves efficiency but also achieves high accuracy in identifying WBC types. This advancement opens opportunities for future research to build upon these findings, contributing significantly to haematology and medicine.

5. CONCLUSION

Based on the results obtained in this study for white blood cell classification using the Gaussian Mixture Model and the proposed image centre point method for segmentation, combined with shape feature extraction and a convolutional autoencoder for feature extraction, and comparing the performance of SVM with no data balancing, SMOTE, and SVMSMOTE, this research achieved several key outcomes.

The proposed feature extraction method effectively represented each type of white blood cell (WBC), including minority class data such as basophils, which consisted of only 179 images. The best results were achieved without data balancing, using a polynomial kernel, which yielded the highest accuracy of 92.4%, the highest Matthews correlation coefficient (MCC) of 0.862, and the lowest cross-entropy loss (CEN) of 0.1376. However, eosinophils exhibited the lowest recall, although the application of SMOTE increased the recall to 70%.

The lowest overall results were observed when using SMOTE with the RBF kernel, yielding an accuracy of 89.4%, an MCC of 0.820, and a CEN of 0.1710. Among all tested kernels, the sigmoid kernel performed the poorest, resulting in the lowest scores across all evaluated metrics.

When comparing the performance of SMOTE and SVMSMOTE, SVMSMOTE performed better, achieving an accuracy of 90.4%, an MCC of 0.836, and a CEN of 0.1625. Additionally, both SVMSMOTE and SMOTE improved the average recall for eosinophils. SMOTE demonstrated a more significant improvement in recall for eosinophils when the focus is on recall over precision. In contrast, SVMSMOTE provided a better balance between precision and recall, ensuring that the increase in recall did not significantly reduce precision.

Future Enhancement:

1. Implement feature selection, e.g. Chi-Square, etc.
2. Experiment with more parameters to get other results.
3. Implement another algorithm for classification.

ACKNOWLEDGEMENT

The results of this study would not have been possible without the help of Almighty God and the support of all those involved.

AUTHOR CONTRIBUTIONS

Conceptualization: Tata Sutabri and Celvine Adi Putra; Investigation, Methodology, and Analysis: Tata Sutabri and Celvine Adi Putra; Software: Celvine Adi Putra; Validation, Review, and Supervision: Tata Sutabri; Data Curation and Editing: Celvine Adi Putra. All authors have read and approved the final version of the manuscript for publication.

References

[1] J. Goretzko *et al.*, "P-selectin-dependent leukocyte adhesion is governed by endolysosomal two-pore channel 2," *Cell Rep*, vol. 42, no. 12, p. 113501, 2023.

[2] M. Zhu, W. Chen, Y. Sun and Z. Li, "Improved U-net-based leukocyte segmentation method," *J Biomed Opt*, vol. 28, no. 04, Apr. 2023.

[3] B. S. S. Rao and B. S. Rao, "An Effective WBC Segmentation and Classification Using MobilenetV3-ShufflenetV2 Based Deep Learning Framework," *IEEE Access*, vol. 11, pp. 27739–27748, 2023.

[4] O. Katar and O. Yildirim, "An Explainable Vision Transformer Model Based White Blood Cells Classification and Localization," *Diagnostics*, vol. 13, no. 14, Jul. 2023.

[5] Z. M. Kouzehkanan et al., "A large dataset of white blood cells containing cell locations and types, along with segmented nuclei and cytoplasm," *Sci Rep*, vol. 12, no. 1, Dec. 2022.

[6] H. Chen *et al.*, "Accurate classification of white blood cells by coupling pre-trained ResNet and DenseNet with SCAM mechanism," *BMC Bioinformatics*, vol. 23, no. 1, Dec. 2022.

[7] S. Rashid, M. Raza, M. Sharif, F. Azam, S. Kadry and J. Kim, "White blood cell image analysis for infection detection based on virtual hexagonal trellis (VHT) by using deep learning," *Sci Rep*, vol. 13, no. 1, Dec. 2023.

[8] L. Zucchini *et al.*, "Characterization of a Novel Approach for Neonatal Hematocrit Screening Based on Penetration Velocity in Lateral Flow Test Strip," *Sensors*, vol. 23, no. 5, Mar. 2023.

[9] I. Lin, O. Loyola-González, R. Monroy and M. A. Medina-Pérez, "A review of fuzzy and pattern-based approaches for class imbalance problems," *Appl. Sci.*, vol. 11, no. 14:6310, 2021.

[10] L. Wang, M. Han, X. Li, N. Zhang and H. Cheng, "Review of Classification Methods on Unbalanced Data Sets," *IEEE Access*, vol. 9, pp. 64606–64628, 2021.

[11] M. Koziarski, "Potential Anchoring for imbalanced data classification," *Pattern Recognition*, vol. 120, Dec. 2021.

[12] S. Khan, M. Sajjad, T. Hussain, A. Ullah and A. S. Imran, "A review on traditional machine learning and deep learning models for WBCs classification in blood smear images," *IEEE Access*, vol. 9, pp. 10657–10673, 2021.

[13] S. Tavakoli, A. Ghaffari, Z. M. Kouzehkanan and R. Hosseini, "New segmentation and feature extraction algorithm for classification of white blood cells in peripheral smear images," *Sci Rep*, vol. 11, no. 1, Dec. 2021.

[14] L. S. Lin, C. H. Kao, Y. J. Li, H. H. Chen and H. Y. Chen, "Improved support vector machine classification for imbalanced medical datasets by novel hybrid sampling combining modified mega-trend-diffusion and bagging extreme learning machine model," *Mathematical Biosciences and Engineering*, vol. 20, no. 10, pp. 17672–17701, 2023.

[15] S. Devella, Y. Yohannes and C. Adi Putra, "Penggunaan Fitur Saliency-SURF Untuk Klasifikasi Citra Sel Darah Putih Dengan Metode SVM," vol. 8, no. 4, 2021.

[16] Y. Yohannes, S. Devella and W. Hadisaputra, "Pemanfaatan Scale Invariant Feature Transform Berbasis Saliency untuk Klasifikasi Sel Darah Putih," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 7, no. 2, Aug. 2021.

[17] F. Riaz *et al.*, "Gaussian Mixture Model Based Probabilistic Modeling of Images for Medical Image Segmentation," *IEEE Access*, vol. 8, pp. 16846–16856, 2020.

[18] P. Wang, E. Fan and P. Wang, "Comparative analysis of image classification algorithms based on traditional machine learning and deep learning," *Pattern Recognit Lett*, vol. 141, pp. 61–67, Jan. 2021.

[19] M. M. Paco Ramos, V. M. Paco Ramos, A. L. Fabian, and E. F. Osco Mamani, "A Feature Extraction Method Based on Convolutional Autoencoder for Plant Leaves Classification," in *Communications in Computer and Information Science*, Springer, pp. 143–154, 2019.

[20] H. Mahmood, T. Mehmood and L. A. Al-Essa, "Optimizing Clustering Algorithms for Anti-Microbial Evaluation Data: A Majority Score-Based Evaluation of K-Means, Gaussian Mixture Model, and Multivariate T-Distribution Mixtures," *IEEE Access*, vol. 11, pp. 79793–79800, 2023.

[21] K. Al-Dulaimi, J. Banks, K. Nguyen, A. Al-Sabaawi, I. Tomeo-Reyes and V. Chandran, "Segmentation of White Blood Cell, Nucleus and Cytoplasm in Digital Haematology Microscope Images: A Review-Challenges, Current and Future Potential Techniques," *IEEE Rev Biomed Eng*, vol. 14, pp. 290–306, 2021.

[22] W. F. Lamberti, "Blood cell classification using interpretable shape features: A Comparative Study of SVM models and CNN-Based approaches," *Computer Methods and Programs in Biomedicine Update*, vol. 1, Jan. 2021.

[23] N. Louanjli *et al.*, "Infiltration of Leukocytes into the Human Ejaculate and its Association with Semen Quality and Oxidative Stress with Sperm Function, and Leukocytospermia Management," 2021.

[24] S. Mahajan, A. Raina, X.-Z. Gao and A. K. Pandit, "Plant Recognition Using Morphological Feature Extraction and Transfer Learning over SVM and AdaBoost," *Symmetry*, vol. 13, no. 2:356, 2021.

[25] M. Irfan, Z. Jiangbin, M. Iqbal, Z. Masood and M. H. Arif, "Knowledge extraction and retention based continual learning by using convolutional autoencoder-based learning classifier system," *Inf Sci (N Y)*, vol. 591, pp. 287–305, 2022.

[26] E. Pintelas, I. E. Livieris and P. E. Pintelas, "A convolutional autoencoder topology for classification in high-dimensional noisy image datasets," *Sensors*, vol. 21, no. 22, Nov. 2021.

[27] J. Sonawane, M. Patil and G. Birajdar, "A novel feature extraction and mapping using convolutional autoencoder for enhancement of Underwater image/video," *ITM Web of Conferences*, vol. 44, p. 03066, 2022.

[28] M. Asrol, P. Papilo, and F. E. Gunawan, "Support Vector Machine with K-fold Validation to Improve the Industry's Sustainability Performance Classification," in *Procedia Computer Science*, Elsevier B.V., pp. 854–862, 2021.

[29] S. Wang, Y. Dai, J. Shen and J. Xuan, "Research on expansion and classification of imbalanced data based on SMOTE algorithm," *Sci Rep*, vol. 11, no. 1, Dec. 2021.

[30] J. H. Joloudari, A. Marefat, M. A. Nematollahi, S. S. Oyelere and S. Hussain, "Effective Class-Imbalance Learning Based on SMOTE and Convolutional Neural Networks," *Applied Sciences (Switzerland)*, vol. 13, no. 6, Mar. 2023.

[31] A. Kim and I. Jung, "Optimal selection of resampling methods for imbalanced data with high complexity," *PLoS One*, vol. 18, Jul. 2023.

[32] M. Khushi *et al.*, "A Comparative Performance Analysis of Data Resampling Methods on Imbalance Medical Data," *IEEE Access*, vol. 9, pp. 109960–109975, 2021.

[33] D. Mustafa Abdullah and A. Mohsin Abdulazeez, "Machine Learning Applications based on SVM Classification A Review," *Qubahan Academic Journal*, vol. 1, Nov. 2021.

[34] A. C. Kemila, W. Fawwaz and A. Maki, "Parameter Optimization of Support Vector Machine using River Formation Dynamic on Brain Tumor Classification," *Open Access Journal*, vol. 5, no. 3, pp. 177–184, 2023.

[35] T. Ke *et al.*, "A general maximal margin hypersphere SVM for multi-class classification," *Expert Syst Appl*, vol. 237, p. 121647, 2024.

[36] R. Yuranda, T. Sutabri and D. Wahyuningsih, "Machine Learning Approach in Evaluating News Labels Based on Titles: Online Media Case Study," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 12, no. 3, pp. 434–439, Nov. 2023.

[37] S. Pertiwi, D. Handoko Wibowo and S. Widodo, "Deep Learning Model for Identification of Diseases on Strawberry (Fragaria sp.) Plants," vol. 13, no. 4, 2023.

[38] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, Jan. 2020.

[39] K. Y. Foo *et al.*, "Multi-class classification of breast tissue using optical coherence tomography and attenuation imaging combined via deep learning," *Biomed Opt Express*, vol. 13, no. 6, pp. 3380–3400, 2022.

[40] B. Krawczyk, C. Bellinger, R. Corizzo and N. Japkowicz, "Undersampling with Support Vectors for Multi-Class Imbalanced Data Classification," in *Proceedings of the International Joint Conference on Neural Networks*, Institute of Electrical and Electronics Engineers Inc., Jul. 2021.

Tata Sutabri is a dedicated researcher with expertise in various domains of computer science. He earned his Master's degree from Gunadarma University and completed his PhD at the same institution in 2018. Since 2024, he has also been an accomplished author, publishing books in the fields of information systems and technology. His research interests include Information Systems, Artificial Intelligence, Blockchain, the Internet of Things (IoT), and Smart Systems, reflecting his commitment to advancing these cutting-edge areas of study.

Celvine Adi Putra received his Bachelor's degree in Information Technology from Multi Data Palembang University, Indonesia, in 2021, and his Master's degree in Information Technology Engineering from Bina Darma University, Indonesia, in 2024. Since completing his undergraduate studies, he has been working as a Software Developer, specializing in software solutions development and system design. His research interests encompass Machine Learning, Image Processing, IT Management, and Software Security. He is actively engaged in various projects, including the application of machine learning models for image analysis and the development of secure software systems.