

# Malaria Parasite Classification from Microscopic Images using EfficientNetV2B0 with Bayesian Optimization

Milda Safrila Oktiana<sup>1\*</sup>, Satria Harya Sulistyo<sup>2</sup>, Refina Nur Zahwa<sup>3</sup>, Luthfi Muhammad Chair<sup>4</sup>,  
Detty Purnamasari<sup>5</sup>

<sup>1, 3, 4</sup>Faculty of Computer Science and Information Technology, Universitas Gunadarma, Depok, 16424, Indonesia

<sup>2, 5</sup>Faculty of Industrial Technology, Universitas Gunadarma, Depok, 16424, Indonesia

(\* ) corresponding author

## ARTICLE INFO

### Article History:

Received April 6, 2025

Revised July 15, 2025

Accepted July 15, 2025

### Keywords:

CRISP-DM

Deep Learning

EfficientNetV2B0

Malaria Disease

TensorFlow

### Correspondence:

E-mail: mildaokt21@gmail.com

## ABSTRACT

*The Plasmodium parasite, which spreads through the bite of the Anopheles mosquito, causes malaria, a significant global health concern. Notwithstanding attempts to curtail its proliferation, malaria continues to be a predominant cause of mortality in tropical nations, especially in Sub-Saharan Africa and certain regions of Southeast Asia. Timely identification and precise diagnosis are essential for effective treatment. This research seeks to create a malaria classification model using deep learning based on the EfficientNetV2B0 architecture. The model is engineered to identify malaria parasite infections in microscopic images of erythrocytes. The dataset used is an open-source collection of photographs depicting red blood cells categorised as either infected or uninfected with malaria. The development method encompasses multiple critical stages, beginning with data collection, followed by preprocessing, data augmentation, and modelling using transfer learning with the EfficientNetV2B0 model. Bayesian optimisation is used to improve the model's accuracy by adjusting its hyperparameters. Assessment metrics, including accuracy, precision, recall, and F1-score, are used to evaluate the trained model's performance. The results show that the model has an accuracy of 96%, with equivalent precision, recall, and F1-scores for both the infected (under the heading "Parasitised") and uninfected (under the heading "Uninfected") groups. The model is extremely effective in diagnosing malaria, making it a valuable diagnostic tool for malaria control and prevention, especially in resource-constrained locations.*

## 1. Introduction

Malaria is an infectious disease affecting red blood cells caused by the *Plasmodium* parasite, which is transmitted to humans through the bite of an infected female *Anopheles* mosquito. The term "malaria" is derived from two Italian words, *mal* (meaning "bad") and *aria* (meaning "air"), implying "bad air," as it was formerly widespread in marshy locations that generated foul odours. This disease is also referred to as Roman fever, swamp fever, tropical fever, and paludism. Malaria is prevalent in almost all regions globally, especially in nations with tropical and subtropical climates [1].

Malaria is a significant public health concern in tropical regions such as Africa, Southeast Asia, and Central and South America, causing considerable morbidity and fatality rates. A WHO analysis indicates that the global incidence of malaria cases in 2022 is projected to have exceeded the levels recorded before the COVID-19 epidemic in 2019. The research underscores various vulnerabilities to the global malaria response, including climate change. In 2022, almost 249 million malaria cases were documented in 85 endemic nations, with an incidence rate of 58 cases per 1,000 at-risk adults. In 2019, there were approximately 233 million global cases, with an incidence rate of 57 instances per 1,000 at-risk adults. In 2022, case numbers exceeded the

objective established by the Global Technical Strategy for Malaria 2025 by 55%, which targeted for merely 26 cases per 1,000 individuals in the at-risk population by that year [2].

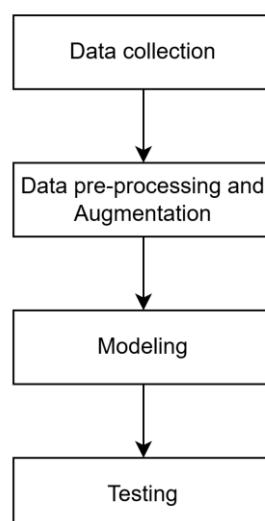
Malaria patients may be infected by one or more species of the Plasmodium parasite (mixed infection). The disease is generally marked by symptoms like fever, chills, headache, nausea, vomiting, and flu-like discomfort, with each malaria species potentially exhibiting distinct symptoms. Severe malaria can result in serious complications, including profound anaemia due to haemolysis, respiratory distress, hypoglycemia, altered consciousness, convulsions, coma, or neurological deficits. Efforts to manage the condition have persisted for an extended period. The existence of drug-resistant malaria parasites exacerbates efforts to eradicate the illness. Notwithstanding continuous prevention and treatment efforts, the early and precise identification of malaria cases continues to pose a problem, especially in resource-constrained regions. Current conventional diagnostic methods are inadequate, prompting the development of more efficient and precise detection techniques. Misidentification and misdiagnosis may contribute to a rise in malaria cases [3].

Recently, artificial intelligence technologies, especially deep learning, have demonstrated considerable potential in aiding malaria detection using microscopic blood cell imaging. A study by Rajaraman et al. revealed that Convolutional Neural Networks (CNNs) can successfully evaluate blood pictures and accurately detect the presence of malaria parasites [4]. Furthermore, the EfficientNet architecture model has proven to be superior in image classification with high efficiency, as shown in a study by Tan and Le [5].

This study seeks to establish a novel methodology for malaria diagnosis by analysing blood sample images via deep learning techniques with the expectation of enhancing accuracy in distinguishing positive from negative samples. We anticipate that this research will significantly improve malaria detection efforts and serve as a model for the development of AI-powered automated diagnostic systems [6] [7].

## 2. Method

The methodology used in this study to detect malaria-infected blood cells and classify them consists of four stages: data collection, data preprocessing and augmentation, modelling, and testing. The stages of this research are illustrated in Figure 1.



**Figure 1.** The stages of research

In the first stage, image data of blood cells to be used for detection are collected, which include two samples: healthy blood cells and infected blood cells. The collected data then proceeds to the preprocessing stage, which involves data splitting and augmentation to optimise the processing workflow.

Modelling is performed using the EfficientNetV2B0 architecture as the base model, followed by fine-tuning. EfficientNetV2B0, as one of the models in the EfficientNet series, has proven to be effective in various image recognition applications, including the detection of malaria parasites in red blood cell images. After the model is developed, the testing and evaluation stage is conducted to assess the model's accuracy. This research uses the organised CRISP-DM (Cross-Industry Standard Process for Data Mining) method to handle the classification problem, starting from gathering data all the way to evaluating the model.

This study also applies techniques such as hyperparameter tuning to optimise training parameters, Bayesian optimisation to accelerate the search for the best parameter combinations, and the use of callbacks to monitor and manage the model training process. Through these key stages, the study aims to ensure accurate and optimal classification results in malaria detection using microscopic blood cell images.

## 2.1 Data Collection and Dataset Description

Data collection is one of the key stages in building a machine learning (ML) model. Accurate and high-quality data collection is essential to ensure the accuracy and effectiveness of the model to be developed. The CRISP-DM methodology provides a systematic framework for this process [8]. At this stage, the primary focus is on gathering relevant data sources for classification.

The dataset used in this study is an open-source dataset available on the Kaggle platform titled "Cell Images for Detecting Malaria." This dataset is a collection of red blood cell images, both infected and non-infected with malaria, sourced from the National Library of Medicine, Lister Hill National Center for Biomedical Communication [9].

The dataset provides microscopic images of human blood cells, divided into two classes: infected (malaria-infected) and uninfected (non-infected). With a total of over 27,500 images, this dataset offers a variety of cell shapes and types, which is crucial for enhancing the model's ability to detect malaria. This variation allows the model to learn from different blood cell conditions, thus improving detection accuracy. The average resolution of the images is 64 x 64 pixels. The dataset is split into three main subsets with a 70:20:10 ratio, designed to ensure that the model is trained with sufficient data while still retaining enough data for validation and testing. The distribution is as follows:

- a. Training Set: 70% of the data is used for model training.
- b. Validation Set: 20% of the data is used for validation during model training.
- c. Test Set: 10% of the data is used for evaluation after model training.

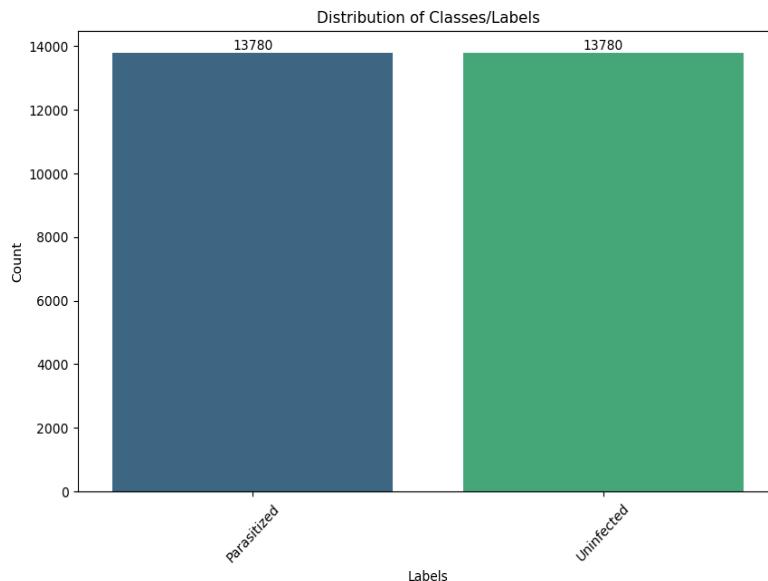
## 2.2 Data pre-processing and Augmentation

Pre-processing is a crucial stage that needs to be performed to enhance the quality of data before it is input into the model. This process involves several steps, including normalization, resizing, and augmentation, all of which contribute to improving the performance of the ML model.

Normalization is the process in which image pixels are transformed into a range of 0-1. The goal is to facilitate learning by the neural network. By performing normalization, the model can adapt and learn more quickly from the provided data, as a uniform input scale helps achieve faster convergence during training [5].

Resizing images is another important step in pre-processing. Images in the dataset often vary in size, so they need to be resized to a uniform dimension. In this context, the images will be resized to the same size, such as 64 x 64 pixels. This process not only reduces some details that may not be critical but also speeds up the learning process. According to Agustin (2023), resizing the dataset aims to accelerate the training and testing processes on convolutional neural networks (CNNs) [10].

The last step is data augmentation, a technique used to create variations in the data samples, which is essential for avoiding overfitting. By performing augmentation, the model can learn from a broader range of variations, thereby improving its ability to generalize to new data. Common augmentation techniques include zooming, random image rotation within a specified range, and flipping. This helps create a more diverse dataset from a limited original set, thus enhancing the model's robustness [11].



**Figure 2.** Class Distribution Analysis

Class distribution analysis is conducted to ensure the balance of the dataset, an essential factor in the development of machine learning models. The number of samples per class is calculated and visualized in a bar chart to detect potential data imbalance. Imbalance can lead the model to favor predicting the majority class, thus reducing performance on the minority class. If imbalance is detected, techniques such as oversampling, undersampling, or augmentation can be applied to enhance the representation of the minority class, ensuring that the model can learn patterns optimally and provide accurate predictions for all classes [12]. All labels are also processed in a categorical format, with class indices assigned as parasitized: 0 and uninfected: 1, ensuring consistency in the mapping of labels to numerical or numeric formats for multi-class classification.

The dataset is divided into three main parts: training, validation, and testing. Seventy percent of the data is allocated for training, while the remaining 30% is further split into validation data (20%) and test data (10%). This process uses the 'train\_test\_split' function with the 'shuffle=True'

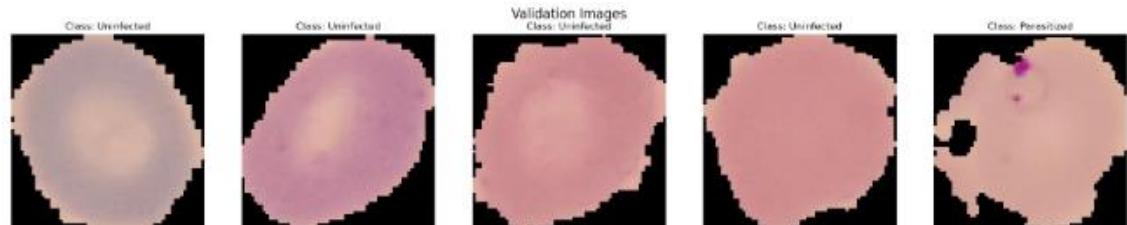
parameter to randomize the data before splitting and 'random\_state=123' to ensure consistent data division across each execution.

The use of 'ImageDataGenerator', which incorporates both preprocessing and data augmentation capabilities, is directly applied to the training, validation, and test data loaded through a data generator to increase the quantity and variability of the training data. Several transformations are applied, including pixel value normalization to a range of 0-1, random rotation up to 15 degrees, horizontal and vertical shifts, horizontal flipping, zoom adjustments up to 10%, and brightness adjustments within a range of 90%-110%. Any empty areas resulting from the transformations are filled using the nearest pixel values.



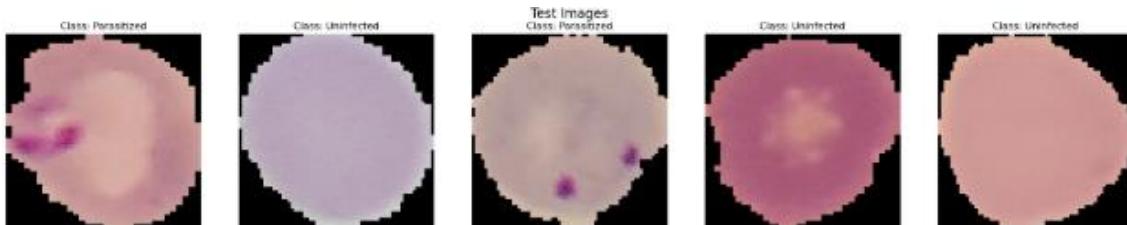
**Figure 3.** Training Images

The training data generator utilizes the prepared augmentations, with several parameters configured, such as resizing the target image to 64x64 pixels through the 'target\_size=(64,64)' parameter. This size is small enough to accelerate model training without losing too many important details of the images. Additionally, a batch size of 32 and RGB color format (Red, Green, and Blue) are used. The data is also shuffled to ensure that the model does not learn any specific order patterns.



**Figure 4.** Validation Images

The 'plot\_images' function is used to visualize 5 random samples from the images in each data subset using the generator, with the class labels displayed in Figure 4: Validation Images. In the validation data, random samples are shown to evaluate the model during testing. In contrast, for the validation data, only normalization is applied without augmentation, and the data is not shuffled to maintain the accuracy of the model evaluation, reflecting the model's true performance. This visualization is crucial to ensure that the validation data is well-distributed and properly represents the existing classes.



**Figure 5.** Test Images

The test data is used to measure the model's performance after training is completed and is used to evaluate the model's accuracy. In the test data, only normalization is applied without any augmentation, and the data is not shuffled to ensure that model evaluation is performed consistently. This visualization helps ensure that the test data is not biased, providing a reliable measure of the model's generalization ability.

### 2.3 Modeling

In the modeling phase for malaria disease detection, a transfer learning approach using the pre-trained EfficientNetV2B0 architecture was chosen as the primary method. The selection of this architecture is based on its lightweight nature and fast training time, while still maintaining optimal results [13]. Additionally, the base model of this architecture is capable of recognizing common features in images without the need for training from scratch.

After selecting EfficientNetV2B0 as the base model, the next step is fine-tuning to adapt the model to the pre-existing dataset. By performing appropriate fine-tuning, the model can be customized to the prepared dataset, thus improving accuracy and efficiency in detecting malaria infections. The fine-tuning process involves several key steps, including adding a classification layer after the base model, configuring and adding dense layers, incorporating a dropout layer, and adding the final dense layer, which serves as the output layer [14].

The process of searching for optimal hyperparameters is carried out using a Bayesian Optimization approach, implemented through Keras Tuner. The 'build\_model' function is used to construct the EfficientNetV2B0-based classification model architecture. This model uses pre-trained weights from ImageNet, with the top layers removed to allow for fine-tuning. The hyperparameters tested include the number of units in the Dense layers, dropout rate, and learning rate, with the search space predefined. The final layer of the model is designed for multi-class classification with a softmax activation function. The entire configuration is optimized to maximize validation accuracy using the categorical cross-entropy loss function [10].

The tuning process involves training the model for three epochs on each hyperparameter combination, using the training and validation data. Early stopping is applied to halt the training early if no improvement is observed in the validation loss, while ensuring the best weights are saved. After tuning is completed, the best hyperparameters and model are evaluated using the validation data to calculate accuracy and loss [15]. The optimal hyperparameters, such as the number of units, dropout rate, and learning rate, are saved in JSON format to facilitate replication and reuse in further training or testing. This approach ensures that the resulting model has an optimal configuration for the best classification performance.

After loading the best hyperparameters saved in the JSON file, the image classification model is built using the optimized parameters. The base model used is EfficientNetV2B0, pre-trained on the ImageNet dataset, with the classification top-layer removed to allow for fine-tuning. The loaded hyperparameters include the number of units in the Dense layers, dropout rate, and learning rate, which are applied to add custom classification layers. The first Dense layer is added with the number of units determined by the hyperparameter, followed by a Dropout layer to reduce overfitting. The output layer uses a softmax activation function to generate the probability distribution for each class. This structure ensures that the model can effectively learn from the training data and make accurate class predictions.

The model is then assembled by connecting the input from the base model and the output from the classification layers. The Adam optimizer is used with the predefined learning rate, and the categorical cross-entropy loss function is applied for multi-class classification. Accuracy is used

as the metric to evaluate the model's performance. Once the model is fully assembled and compiled, the complete model structure, along with the number of parameters, is displayed to provide an overview of the architecture. This helps in understanding the design and complexity of the model. The model is then ready for further training and evaluation on the relevant dataset.

Several callback strategies are used to optimise the model's training process. The first callback is EarlyStopping, which checks validation loss and stops training if no improvement is seen after a set number of epochs, returning the model weights to the best state discovered during training. The second callback is ReduceLROnPlateau, which cuts the learning rate when the validation loss stagnates in order to hasten the model's convergence. Additionally, ModelCheckpoint is used to save the best model throughout training, either as the full model or just the model weights, with monitoring of the validation loss to ensure that only the model with the best performance is saved. These callbacks help ensure efficient training and prevent overfitting by optimizing the model's learning process.

The model training process is conducted with the utilization of several callbacks previously described, and the training duration is calculated to assess the time required. The start time of training is recorded before the training process begins, and the end time is calculated after the process is completed. The total training time is then displayed in minutes and seconds to provide an overview of the training efficiency. The model is trained for 50 epochs, with continuous monitoring of the validation loss to optimize the model according to the prevailing conditions.

Next, the training history data, which contains information about the model's performance, including loss and accuracy for each epoch, is saved in JSON format. This storage allows for further analysis of the training process, with the data available for visualization or future model evaluation. By adopting this approach, all training-related information, including model performance and training duration, can be accessed and analyzed in greater detail once the training is completed.

## 2.4 Testing and Evaluation

After the model has passed through the modeling and training stages, it moves on to the testing and evaluation phase to assess its accuracy. Evaluation on the training data is essential to ensure that the trained model performs well and does not experience overfitting or underfitting. Through evaluation, the model's learning process can be monitored to determine how it learns from the data and makes accurate predictions. To measure the model's performance, commonly used evaluation metrics include accuracy, recall, precision, and F1-Score [16].

### 2.4.1 Accuracy

The previously saved JSON data can be used to analyse the model's performance. The accuracy result measures how often the model makes correct predictions. In general, it indicates the ratio of correct predictions to the total number of predictions. The formula for calculating accuracy is shown as (1).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \quad (1)$$

Where:

TP : True Positive (The model correctly predicts positive samples)

TN : True Negative (The model correctly predicts negative samples)

FP : False Positive (The model predicts positive samples that are actually negative)  
FN : False Negative (The model predicts negative samples that are actually positive)

#### 2.4.2 Recall

Recall is a metric that measures how often the model correctly identifies true positives from all ground truth positives. This metric is used to extract the ratio of correctly identified positive instances to all actual positives in the ground truth. The formula for calculating recall is shown as formula (2).

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

#### 2.4.3 Precision

Precision is an evaluation statistic that assesses a model's ability to correctly forecast positive samples. This measure is the ratio of true positives to all positive predictions, including false positives, using the following formula (3).

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

#### 2.4.4 F-1 Score

The F1-score is a statistic that combines precision and recall to give a more complete picture of the model's performance. This statistic is very valuable when dealing with class imbalance, as precision or recall alone may not suffice. The F1-score goes from 0 to 1, with values near 1 suggesting superior model performance in terms of precision and recall. The F1-score can be determined using the following formula (4).

$$F-1 Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

### 3. Results and Discussion

The results of the training and evaluation of the malaria classification model based on EfficientNetV2B0 will be presented in this section. Additionally, an analysis of the model's performance will be provided, along with the factors that influenced these results. The discussion section will include the interpretation of the results and the accuracy level of the model.

Overall, the research on the EfficientNetV2B0-based malaria classification model demonstrates excellent and satisfactory performance. For each class, 'Parasitized' and 'Uninfected', the model exhibits a good balance in detecting both classes, with a relatively low error rate. The model achieves an accuracy rate of 96%, indicating that it can classify images very effectively, whether they belong to the infected class ('Parasitized') or the non-infected class ('Uninfected').

The precision for the 'Parasitized' class is higher (0.97) compared to the 'Uninfected' class (0.95). The higher precision in the 'Parasitized' class indicates that the model is more accurate in identifying images as infected. This is likely due to the larger number of samples in the 'Parasitized' class (1,410 labels) compared to the 'Uninfected' class (1,345 labels). As a result, it can be concluded that the model interacted more frequently with data from the 'Parasitized' class during training, enhancing its ability to recognize 'Parasitized' images.

However, the recall for the 'Uninfected' class is higher (0.97), which means the model performs better in detecting 'Uninfected' images. This indicates that, despite the 'Uninfected' class having fewer sample images, the model is still able to accurately detect these images. This suggests that the model is effective in identifying the 'Uninfected' class, even with a smaller sample size.

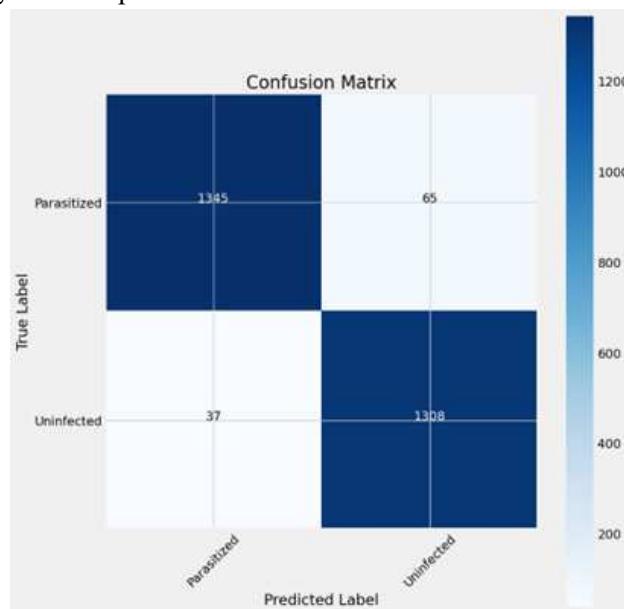
The F1-Score shows a balanced value for both classes (0.96), indicating that the model in this study is not biased towards the more dominant class and is able to classify both classes accurately. This suggests that the model performs well in achieving a balance between precision and recall for both the 'Parasitized' and 'Uninfected' classes.

**Table 1.** Model Performance Evaluation

Class Labels	Evaluation Metrics		
	Precision	Recall	F-1 Score
Parasitized	0.97	0.95	0.96
Uninfected	0.95	0.97	0.96
<b>Accuracy</b>			0.96
<b>Val Accuracy</b>			0.95

In Figure 6, the confusion matrix is presented to visualize the evaluation of recall and how the model classifies each label from the testing dataset. The results show that the model successfully predicted 95.39% of the parasitized samples correctly, demonstrating a high accuracy in detecting positive cases. However, there is a misclassification rate of 4.61%, where samples that were actually infected were incorrectly classified as uninfected. On the other hand, for the uninfected samples, the model performed excellently, correctly predicting 97.24% of those labels. Nonetheless, a misclassification rate of 2.76% occurred, where samples that were actually uninfected were mistakenly classified as parasitized.

Based on this analysis, it can be concluded that the model exhibits strong capabilities in classifying both infected and uninfected samples with a relatively high accuracy. This highlights the model's potential for use in real-world applications, where accuracy and reliability in diagnosis are crucial, especially under supervised conditions.



**Figure 6.** Confusion Matrix based on the testing dataset results

#### 4. Conclusion

This study demonstrates that the EfficientNetV2B0 model delivers excellent performance, achieving a high accuracy rating of 96% after hyperparameter tuning using Bayesian optimisation. These adjustments optimised an already well-trained model, improving its ability to detect malaria-infected blood cells. The approach is in line with recent research leveraging machine learning to enhance the efficiency, speed, and accuracy of disease detection. By utilising deep learning models such as EfficientNetV2B0, the identification of potential malaria infections can be performed much faster compared to traditional methods. This accelerates diagnostic time and improves accuracy, potentially making a significant impact on malaria control and prevention efforts. As a result, the research aims to contribute to the advancement of disease diagnosis technologies, making them more sophisticated and reliable, particularly in remote areas with limited medical resources.

#### Acknowledgment

The authors would like to express their sincere gratitude to Universitas Gunadarma for providing the resources and academic support necessary for the completion of this research. Special thanks are also due to the DGX Development Team for their assistance and support throughout the research process. Their collaboration and contributions have been essential in achieving the goals of this project.

#### References

- [1] N. P. Kar, A. Kumar, O. P. Singh, J. M. Carlton and N. Nanda, "A review of malaria transmission dynamics in forest ecosystems," *Parasites & Vectors*, no. 265, 2014.
- [2] P. Venkatesan, "The 2023 WHO World malaria report," vol. 5, no. 3, p. e214, March 2024.
- [3] A. Mahittikorn, F. R. Masangkay, K. U. Kotepui, G. D. J. Milanez and M. Kotepui, "Quantification of the misidentification of Plasmodium knowlesi as Plasmodium malariae by microscopy: an analysis of 1569 P. knowlesi cases," *Malaria Journal*, no. 179, 09 April 2021. doi: 10.1186/s12936-021-03714-1.
- [4] S. Rajaraman, S. K. Antani, M. Poostchi, K. Silamut, M. A. Hossain, R. J. Maude, S. Jaeger and G. R. Thoma, "Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images," *PeerJ*, vol. 6, p. e4568, 2018. doi: 10.7717/peerj.4568/supp-1.
- [5] M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proceedings of the 36th International Conference on Machine Learning*, 2019.
- [6] M. Mujahid, F. Rustam, R. Shafique, E. C. Montero, E. S. Alvarado, I. D. I. T. Diez and I. Ashraf, "Efficient deep learning-based approach for malaria detection using red blood cell smears," *Scientific Reports*, vol. 14, no. 1, p. 13249, 2024. doi: 10.1038/s41598-024-63831-0.
- [7] H. A. H. Chaudhry, M. S. Farid, A. Fiandrotti and M. Grangetto, "A lightweight deep learning architecture for malaria parasite-type classification and life cycle stage detection," *Neural Computing and Applications*, vol. 36, pp. 19795-19805, August 2024.
- [8] C. Schröer, F. Kruse and J. M. Gómez, "A Systematic Literature Review on Applying CRISP-DM Process Model," in *Procedia Computer Science*, 2021. doi: 10.1016/j.procs.2021.01.199.
- [9] Kaggle, "Cell Images for Detecting Malaria," Kaggle. [Online]. Available: <https://www.kaggle.com/datasets/iarunava/cell-images-for-detecting-malaria>.
- [10] A. F. Agustin, H. R. Albarki, R. S. H. Martin and J. A, "Identification of fish meal adulterated with rice bran by using an image analysis method," in *The 6th International Conference on Agriculture, Environment and Food Security*, 2023. doi: 10.1088/1755-1315/1241/1/012138.

- [11] P. Zeng, "Research on Similar Animal Classification Based on CNN Algorithm," in *International Conference on Computer Technology, Information Engineering and Electron Materials (CTIEEM 2021)*, 2021. doi: 10.1088/1742-6596/2132/1/012001.
- [12] W. Chen, K. Yang, Z. Yu, Y. Shi and C. L. P. Chen, "A survey on imbalanced learning: latest research, applications and future directions," *Artif Intell Rev* 57, no. 137, 2024. doi: 10.1007/s10462-024-10759-6.
- [13] H. Dastour and Q. K. Hassan, "A Comparison of Deep Transfer Learning Methods for Land Use and Land Cover Classification," *Sustainability*, vol. 15, no. 10, p. 7854, 2023. doi: 10.3390/su15107854.
- [14] A. Heikal, A. El-Ghamry, S. Elmougy and M. Z. Rashad, "Fine tuning deep learning models for breast tumor classification," *Sci Rep*, vol. 14, no. 1, 2024. doi: 10.1038/s41598-024-60245-w.
- [15] P. Mishra and D. Passos, "A tutorial on automatic hyperparameter tuning of deep spectral modelling for regression and classification tasks," *Chemometrics and Intelligent Laboratory Systems*, 2022. doi: 10.1016/j.chemolab.2022.104520.
- [16] S. A. Hicks, I. Strümke, V. Thambawita, M. Hammou, M. A. Riegler, P. Halvorsen and S. Parasa, "On evaluation metrics for medical applications of artificial intelligence," *Sci Rep*, vol. 12, no. 1, p. 5979, 8 April 2022. doi: 10.1038/s41598-022-09954-8.