



INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv



Classification of Malaria Cell Image Using Inception-V3 Architecture

Agus Eko Minarno^a, Laofin Aripa^a, Yufis Azhar^a, Yuda Munarko^{a,*}

Universitas Muhammadiyah Malang, Jl. Tlogomas No. 246, Malang, 65144, Indonesia

**Corresponding author: yuda@umm.ac.id*

Abstract— Malaria is a severe global public health problem caused by the bite of infected mosquitoes. It can be cured, but only with early detection and effective, quick treatment. It can cause severe conditions if not properly diagnosed and treated at an early stage. In the worst scenario, it can cause death. This study aims at focusing on classifying malaria cell images. Malaria is classified as a dangerous disease caused by the bite of the female *Anopheles mosquito*. As such, it leads to mortality when immediate action and treatment fails to be administered. In particular, this study aims to classify malaria cell images by utilizing the Inception-V3 architecture. In this study, training was conducted on 27,558 malaria cell image data through Inception-V3 architecture by proposing 3 scenarios. The proposed scenario 1 model applies the SGD optimizer to generate a loss value of 0.13 and an accuracy value of 0.95; scenario 2 model applies the Adam optimizer to generate a loss value of 0.09 and an accuracy value of 0.96; and lastly scenario 3 implements the RMSprop optimizer to generate a loss value of 0.08 and an accuracy value of 0.97. Applying the three scenarios, the results of the study apparently indicate that the Inception-V3 model using the RMSprop optimizer is capable of providing the best accuracy results with an accuracy of 97% with the lowest loss value, compared to scenario 1 and scenario 2. Further, the test results confirms that the proposed model in this study is capable of classifying malaria cells effectively.

Keywords—Convolutional neural networks; Malaria; inception-v3; classification.

Manuscript received 18 Oct. 2022; revised 14 Jan. 2023; accepted 25 Jan. 2023. Date of publication 30 Jun. 2023. International Journal on Informatics Visualization is licensed under a Creative Commons Attribution-Share Alike 4.0 International License.



I. INTRODUCTION

Malaria is a serious and potentially life-threatening disease caused by the Plasmodium parasite, and it is transmitted to humans through the bite of an infected Anopheles mosquito [1]. The disease affects millions of people worldwide, particularly in sub-Saharan Africa, where it is a leading cause of death among children under five years old [2]. The accurate and timely diagnosis of malaria is critical for the effective treatment and control of the disease [3]. However, traditional diagnostic methods, such as microscopy and rapid diagnostic tests (RDTs), have limitations and can be time-consuming and labor-intensive [4].

Deep learning, a branch of machine learning, has emerged as a powerful tool for classifying medical images [5]. In recent years, deep learning-based models have been applied to a wide range of medical imaging tasks, including the classification of skin lesions, brain tumors, and lung cancer [6]. With the availability of large datasets of microscopy images of malaria-infected blood cells, deep learning-based models have also been applied to classify malaria [7].

Deep learning models, such as convolutional neural networks (CNNs) and deep belief networks (DBNs), have

been trained to classify malaria-infected blood cells with high accuracy [8]. These models have been trained on a large dataset of microscopy images of blood cells, and have been shown to be able to accurately classify the images as infected or not infected with the Plasmodium parasite [9], [10].

One of the key advantages of deep learning-based models is their ability to learn features from the data, which can be used to classify the images. CNNs, in particular, have been shown to be effective in this task, as they are able to automatically extract features from the images, such as the shape and texture of the cells, which are important for the classification of malaria [11].

In addition to classification, deep learning-based models have also been applied to other tasks related to diagnosing malaria, such as the segmentation of infected cells in microscopy images and detecting the Plasmodium parasite in blood smears. These models have been trained on large datasets of images and have been shown to be able to accurately segment and detect the parasite, which can aid in the diagnosis of malaria [12].

A deep learning-based model is a powerful tool for classifying malaria-infected blood cells. This model has been trained on large datasets of microscopy images and shows

accuracy in classifying the images as infected or not infected with the Plasmodium parasite [13]. With the continuing advancement of deep learning-based models, it is expected that these models will play an increasingly important role in the future diagnosis and treatment of malaria [14].

Malaria is classified as a disease generally experienced by humans due to the bite of a female *Anophles mosquito* carrying the plasmodium parasites, such as *Plasmodium falciparum*, *Plasmodium vivax*, *Plasmodium malariae*, and *Plasmodium ovale*. Malaria is considered a serious infection caused by peripheral blood parasites of the genus *Plasmodium* [15].

Malaria is commonly acknowledged as a disease that often occurs in coastal areas. Malaria sufferers typically experience flu-like symptoms, high fever, chills, and headaches, attacking all demographic features (regardless of age and gender). Symptoms of malaria typically appear after ten days to 4 weeks in the form of fever, headache, vomiting, chills, anemia, and enlarged spleen [16].

Hence, early detection of malaria-infected blood cells is deemed vital by monitoring blood cell counts annually by experts trained to detect malaria infection [17], [18]. Malaria detection involves manually counting parasites and infected red blood cells, which is highly dependent on the experience and skills of the microscope expert. As such, this expertise is highly prioritized; however, limited resources and systems could certainly affect the quality of diagnostics, leading to inappropriate diagnostic decisions [19].

Prior studies have been devoted to discussing malaria cells, especially in the case regarding the classification of blood cells infected with malaria as conducted by Reddy and Juliet [8] in 2019, aiming to obtain appropriate diagnostic results against microscopic malaria cells by employing the Convolutional Neural Network (CNN) method and the Resnet-50 architecture. In 2020, malaria-related research was also conducted by Sayyed et al. [20] by observing the comparison of the effectiveness between convolutional combinations and neural networks to detect malaria parasites.

In particular, the CNN method has been considered a popular method in two-dimensional processing data, with a similar grid topology utilizing convolution as a substitute for matrix application, which applies at least one convolution in each layer [21]. CNN is considered a type of deep learning, a machine learning component to teach computers to conduct an activity such as the training process. Therefore, this method is effectively implemented for datasets that have a two-dimensional structure [22]. In addition, this study aims to improve the performance of the proposed model, which is the Inception-V3 model, to obtain better accuracy performance results than those in previous studies.

II. MATERIAL AND METHOD

This section discusses the implementation of the CNN method in classifying malaria cell images with the Inception-V3 architecture.

A. Dataset

The applied dataset includes a malaria cell image dataset [8], [23], containing 27,558 image data, divided into two classifications: 13,775 image data for the parasitized class and

13,813 image data for the uninfected class. The sample infected and uninfected images depict in Fig. 1 and Fig. 2.

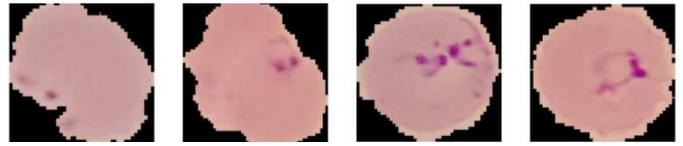


Fig. 1 Sample images infected cell of malaria

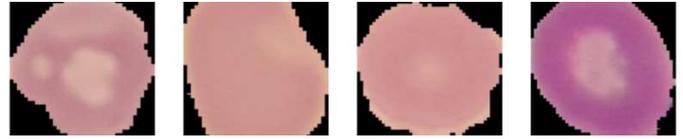


Fig. 2 Sample image uninfected cell of malaria

Parasitized class refers to image data of blood cells infected with malaria, while uninfected refers to image data that is not infected with malaria. The data is sourced from kaggle.com [24], and the original data is generated from the official NIH website [25], which is further uploaded on kaggle.com for open access. From all existing data, the dataset is later divided into 70, 15, and 15 (percent into train data, test data, and validation data, respectively).

B. Pre-processing

Image augmentation is a technique used to artificially increase the size of a dataset by applying various modifications to the existing images. These modifications include transformations such as rotation, flipping, and cropping, as well as changes in brightness, contrast, and color. The goal of image augmentation is to create new, diverse images from the original ones, which can be used to improve the performance and robustness of machine learning models that are trained on image data.

One of the main benefits of image augmentation is that it can help reduce overfitting, a common problem in deep-learning models. Overfitting occurs when a model is trained on a small dataset and becomes too specialized to the training data, leading to poor performance on new, unseen data. By augmenting the training dataset with diverse, modified versions of the original images, the model is exposed to a wider range of variations in the data, making it more robust to new, unseen data.

Another benefit of image augmentation is that it can help to improve the generalization of models trained on image data. In other words, it can make the models more accurate in recognizing objects in images that were not part of the training dataset. This is especially important when the available dataset is small or has certain biases, and image augmentation can help to mitigate these issues. Data processing is performed by augmenting data, employing a hard library (the Image Data Generator), and augmentation is performed to overcome the problem of overfitting; thus, the model is capable of appropriately predicting the image of malaria cells [26]. Pre-processing in this study includes setting the image size by converting it to the intended size, and the image size will adjust user input in the CNN architecture [27].

Augmentation is conducted for train data and validation data during the model learning process because the validation data requested by the model to predict contains original data and data obtained from the augmentation process. Hence, it is

apparent that the predicted object or image from the validation data and test data is likely similar, but if obtained from a different process, the model is further tasked to predict the results, which will be different from those in the validation test data.

The following data augmentation parameters with the image data generator are illustrated in Table I.

TABLE I
IMAGE DATA GENERATOR PARAMETERS

No	Parameter	Value
1	Horizontal Flip	True
2	Vertical Flip	True
3	Rotation Range	90
4	Height Shift Range	0.2
5	Width Shift Range	0.2
6	Zoom Range	0.2

The Table mentions the data augmentation parameters with the Image data generator, which is implemented into the data train and validation data, including: `rescale = 1./255`, `horizontal_flip = True`, `vertical_flip = true`, `rotation_range = 90`, `height_shift_range = 0.2`, `width_shift_range = 0.2`, and `zoom_range = 0.2`. Furthermore, the image data generator that is implemented into the test data includes: `rescale = 1./255`. These parameters are implemented to obtain the expected results from the proposed model training process.

C. Inception-V3

A convolutional Neural Network is defined as a method to classify data in images [28], specializing in image recognition problems [8]. The advantage of the CNN model lies in the hierarchical structure of the learning layer that can be trained intensely after the model topology matches the input features, and the model can work efficiently by utilizing the spatial relationship of visual patterns to reduce the number of parameters which improves performance accuracy [29].

Inception V3 is a convolutional neural network (CNN) architecture that Google developed for image classification tasks. It is the third version of the Inception architecture and was introduced in 2015 by Szegedy et al. [30]. Inception V3 is built upon the concepts of the previous Inception architectures, and it is designed to improve the performance and efficiency of image classification tasks. The architecture of Inception V3 is a deep and complex one, consisting of a stack of inception modules that are interconnected with each other. Each inception module combines different types of convolutional and pooling layers designed to extract different features from the input image.

One of the key features of Inception V3 is its use of factorized convolutions, which reduce the number of parameters in the network while maintaining a high accuracy level. Factorized convolution is a technique that allows the network to learn both local and global features of an image, improving the model's accuracy. The Inception V3 network uses a combination of 1x1, 3x3, and 5x5 convolutional filters to extract features from the input image. The 1x1 convolutional filters are used to reduce the dimensionality of the input, while the 3x3 and 5x5 filters are used to extract more complex features from the input image.

Another important feature of Inception V3 is the use of batch normalization, which is a technique that is used to

normalize the inputs to the network. Batch normalization helps to stabilize the training process and to reduce the internal covariate shift, which is the change in the distribution of the network's inputs during training.

Inception V3 is pre-trained on a large dataset called ImageNet, which contains over 14 million images and 1000 different classes. This allows the model to be used for transfer learning on other image classification tasks with a smaller dataset, where it can be fine-tuned for a specific task. Transfer learning is a technique that allows a pre-trained model to be reused on a different task by fine-tuning the model on a new dataset. This can greatly reduce the amount of data and computational resources required to train a new model from scratch.

Inception V3 has been widely used for image classification tasks and has been shown to perform very well on a variety of benchmarks. It has been used in various applications such as object detection, image segmentation, and video classification. Its use of factorized convolutions and Inception modules makes it a highly efficient and accurate model for image classification tasks. Moreover, its pre-training on ImageNet dataset makes it a versatile model that can be used in a variety of applications with a good amount of accuracy.

Inception V3 is a powerful CNN architecture that has been widely used for image classification tasks and has been shown to perform very well on a variety of benchmarks. Its use of factorized convolutions and Inception modules makes it a highly efficient and accurate model for image classification tasks. Its pre-training on the ImageNet dataset allows it to be used for transfer learning on other image classification tasks with a smaller dataset, where it can be fine-tuned for a specific task, thus reducing the amount of data and computational resources required to train a new model from scratch.

The Inception-V3 architectural model has the advantage due to its more complex architecture and more efficient computation, containing approximately 4 million parameters, significantly smaller compared to VGG, more complex architecture, and this model does not apply a fully-connected layer replacing it with a pooling layer only. These fewer parameters result in a smaller model size, enabling faster model calculation [31].

The initial layer of Inception-V3 consists of 3 standard convolution layers, followed by a max-pooling layer, 2 convolution layers, and a maxpooling layer. The next stage in the network includes inception convolution, which simultaneously convolutes the input by utilizing a different filter size for each convolution, further combining or stacking the results together and passing them across the network. Subsequent sections of the network re-start and rest a number of times, in which some sections repeat 10 or 20 times towards the end. The network implements the stop-learning layer to drop weights randomly (making the filter value equal to 0) to prevent overfitting. Furthermore, the second last layer is fully connected [32].

III. RESULTS AND DISCUSSION

A. Data Sample Collection

The initial stage of the data sample is performed by splitting the data and dividing each class into several parts, consisting of train data, validation data, and test data. Train

data is employed to train the model, from which the results of the learning are tested for validation data, and then test data is employed in the evaluation process to determine the model's performance.

TABLE II
SAMPLE DATA SHARING

No	Data Split	Number of Parasitized data	Number of Uninfected data
1	Train Data	9,646	9,646
2	Data Validation	4.134	4.134
3	Test Data	4.134	4.134

Validation data is additionally employed in test data, but for the purposes of model training, validation data is required, which is the test data itself. Each of these data is divided into 70% for train data, 15% for validation data, and 15% for test data after the data splitting process, illustrated in Table II.

B. Result

The first scenario is to employ similar optimizer as the reference journal, such as the SGD (Stochastic gradient descent) optimizer[8]. In this study, the SGD optimizer is implemented into the Inception-V3 model, with a learning rate of 0.001 and a data sharing of 70:15:15. In scenario 1, the set data training time is 3 hours, 18 minutes and 58 seconds. The evaluation of scenario 1 by utilizing the plot is described in Fig.3, covering Training and Validation of Accuracy, and Fig.4 describes the plot results from training and validation loss. The evaluation in the form of a classification report for the Inception-V3 model is illustrated in Table III which explains the results of the classification report values of scenario 1.

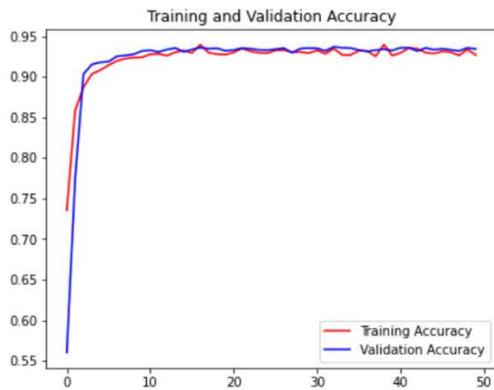


Fig. 3 Plot Training and Validation Accuracy of Scenario 1

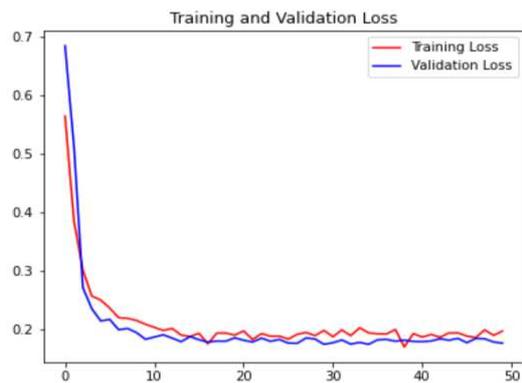


Fig. 4 Plot Training and Validation Loss of Scenario 1

TABLE III
CLASSIFICATION REPORT OF SCENARIO 1

Class	Precision	Recall	F1-Score	Support
Parasitized	0.97	0.95	0.96	4134
Uninfected	0.95	0.97	0.97	4134
Macro Avg	0.96	0.96	0.96	8268
Weight Avg	0.96	0.96	0.96	8268

Scenario 2 shows the adam optimizer (Adaptive Moment Estimation) is implemented with the Inception-V3 model. The adam optimizer has individual learning rates for different parameters. Thus, the adam optimizer is expected to provide better accuracy. In addition, the adam optimizer does not require a lot of storage space, and this optimizer is also lighter during the epoch training process [23]. The results of accuracy and loss of scenario 2 are depicted in in Figs. 5 and 6.

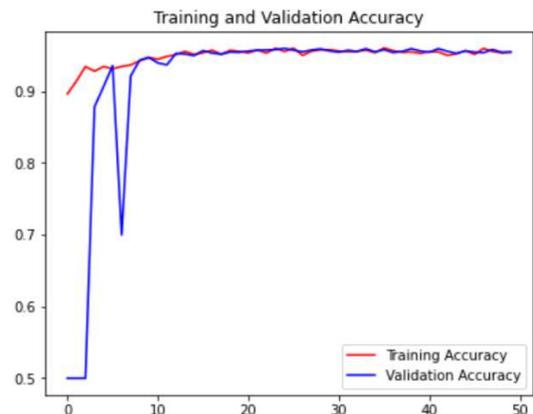


Fig. 5 Plot Training and Validation Accuracy of Scenario 2

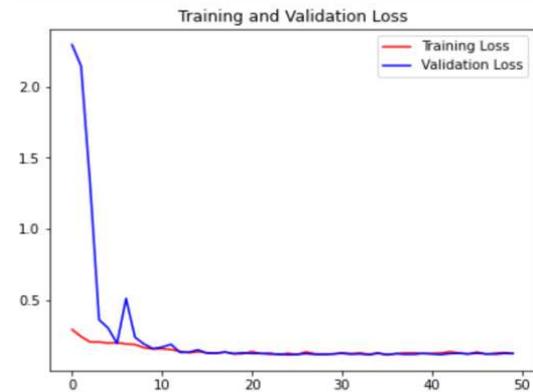


Fig. 6 Plot Training and Validation Loss of Scenario 2

Following are the results of the evaluation of scenario 2, implementing the adam optimizer and data sharing of 70:15:15. In this study, the adam optimizer is implemented into the Inception-V3 model by using a learning rate of 0.001 iterations of 50 epochs, and the training process time is set at 3 hours, 20 minutes and 18 seconds. The evaluation results of scenario 2 with the adam optimizer implemented into the Inception-V3 model are presented in Table IV.

Scenario 3 compares the best optimizer to generate a higher accuracy value. In scenario 3, the optimizer RMSprop (Root-Mean Square Propagation) is applied, indicating a good performance in classifying the model. Scenario 3 applies a learning rate of 0.0001, implemented into the Inception-V3 model with 50 epoch iterations. Optimizer RMSprop implemented into scenario 3 with 70:15:15 data sharing is applied. In scenario

3, the epoch training time is set at 3 hours, 8 minutes and 8 seconds. The results of accuracy and loss of scenario 2 are depicted in Figs. 7 and 8. The evaluation employs a scenario 3 classification report to straightforwardly determine the values of the implemented model results as depicted in Table V.

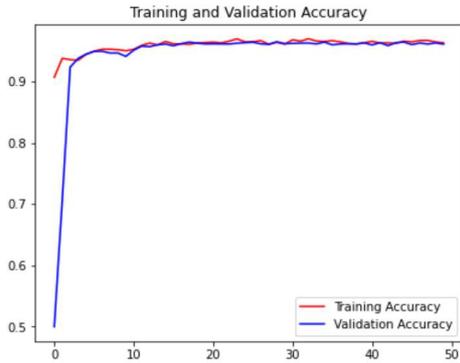


Fig. 7 Plot Training and Validation Accuracy of Scenario 3

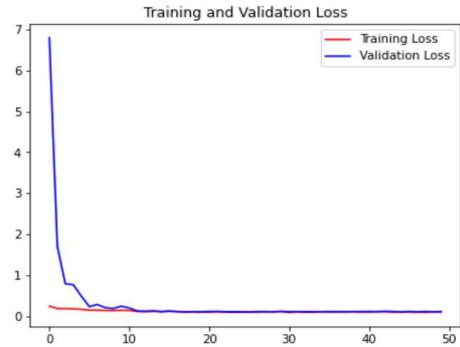


Fig. 8 Plot Training and Validation Loss of Scenario 3

TABLE IV
CLASSIFICATION REPORT OF SCENARIO 2

Class	Precision	Recall	F1-Score	Support
Parasitized	0.98	0.96	0.97	4134
Uninfected	0.96	0.98	0.97	4134
Macro Avg	0.97	0.97	0.97	8268
Weight Avg	0.97	0.97	0.97	8268

TABLE V
CLASSIFICATION REPORT OF SCENARIO 3

Class	Precision	Recall	F1-Score	Support
Parasitized	0.98	0.97	0.97	4134
Uninfected	0.97	0.98	0.97	4134
Macro Avg	0.97	0.97	0.97	8268
Weight Avg	0.97	0.97	0.97	8268

C. Comparison between Testing Scenarios

A comparison of each test scenario is conducted with a training process that implements iterations of 50 epochs and 193 steps per epoch in each scenario, which provides different performance results. The comparison table for each scenario is presented in Table VI.

TABLE VI
PERFORMANCE OF THE TRAINING PROCESS FOR EACH SCENARIO

Scenario	Training loss	Training Ac	Validation loss	Validation acc
Scenario 1	0.17	0.93	0.18	0.93
Scenario 2	0.11	0.95	0.11	0.95
Scenario 3	0.09	0.96	0.10	0.96

TABLE VII
PERFORMANCE RESULTS OF OVERALL PROPOSED SCENARIO

Scenario	Precision	Recall	F-1 Score	Accuracy
Scenario 1	0.95	0.97	0.96	0.96
Scenario 2	0.96	0.98	0.97	0.97
Scenario 3	0.97	0.98	0.97	0.97

Referring from the performed tests, particularly as presented in Table VI and Table VII, it shows the best model performance in scenario 3. The scenario employed the RMSprop optimizer with a learning rate of 0.0001, passing through a training process of 50 epochs iteration, resulting in better performance than other models. The Adam optimizer also indicates similar accuracy in scenario 2 but scenario 3 has a lower probability of prediction failure or value loss and has high accuracy training value among other scenarios.

D. Comparison of Architectural models with Previous Studies

The model that generates the best performance is further compared to the previous study models, including Resnet50 model with the SGD optimizer [8] and the Convolutional Neural Network model [33], as depicted in Table VIII.

TABLE VIII
MODEL PERFORMANCE COMPARISON

Model	Pre-processing	Train Acc	Train loss	Val Acc	Val Loss
ResNet50[34]	No aug	0.95	0.11	0.95	0.13
CNN (3 convolutional layers) [33]	Aug	0.95	0.12	0.96	0.02
Proposed Model	Aug	0.96	0.09	0.96	0.10

TABLE IX
THE COMPARISON RESULT ANALYSIS OF THE ACCURACY FROM THE PROPOSED MODEL

Model	Optimizer	Epoch	Accuracy
CNN (3 convolutional layers)[33]	Adam	14	0.95
Proposed Model	RMSprop	50	0.97

Table VIII presents that the proposed model generates a train accuracy of 96%, which is greater than 0.01 compared to the previous model using the ResNet50 model [8] and the CNN model with 3 convolutional layer architecture [33]. In sum, the comparison of performance based on the accuracy is depicted in Table IX, confirming that the proposed scenario has an accuracy of 0.02%, which is deemed superior to previous studies.

IV. CONCLUSION

Based on the test scenario, which has modeled the Inception-V3 architecture, the application of the RMSprop optimizer with a learning rate of 0.0001 is profound to generate superior performance than the test scenario, previously employing the Adam learning rate optimizer of 0.001 and SGD learning rate of 0.001. The scenario proposed in this study can improve the accuracy from the previous study reaching 97%, thereby providing better performance.

REFERENCES

- [1] S. v. Militante, "Malaria Disease Recognition through Adaptive Deep Learning Models of Convolutional Neural Network," *ICETAS 2019 - 2019 6th IEEE International Conference on Engineering*,

- Technologies and Applied Sciences*, Dec. 2019, doi: 10.1109/ICETAS48360.2019.9117446.
- [2] M. M. Qanbar and S. Tasdemir, "Detection of Malaria Diseases with Residual Attention Network," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 7, no. 4, pp. 238–244, Dec. 2019, doi: 10.18201/ijisae.2019457677.
 - [3] T. Kami, A. Zein, and M. I. Analisis, "Sainstech : Jurnal Penelitian dan Pengkajian Sains dan Teknologi Pendeteksian Penyakit Malaria Menggunakan Medical Images Analisis Dengan Deep Learning Python," vol. 29, no. 1, pp. 28–29, 2019.
 - [4] D. R. Loh, W. X. Yong, J. Yapeter, K. Subburaj, and R. Chandramohanadas, "A deep learning approach to the screening of malaria infection: Automated and rapid cell counting, object detection and instance segmentation using Mask R-CNN," *Computerized Medical Imaging and Graphics*, vol. 88, p. 101845, Mar. 2021, doi: 10.1016/J.COMPAMEDIMAG.2020.101845.
 - [5] A. Rahman *et al.*, "Improving Malaria Parasite Detection from Red Blood Cell using Deep Convolutional Neural Networks," Jul. 2019, doi: 10.48550/arxiv.1907.10418.
 - [6] A. Abubakar, M. Ajuji, and I. U. Yahya, "DeepFMD: Computational Analysis for Malaria Detection in Blood-Smear Images Using Deep-Learning Features," *Applied System Innovation 2021, Vol. 4, Page 82*, vol. 4, no. 4, p. 82, Oct. 2021, doi: 10.3390/ASI4040082.
 - [7] Y. Supranelfy and R. Oktarina, "Gambaran Perilaku Pencegahan Penyakit Malaria di Sumatera Selatan (Analisis Lanjut Riskekdas 2018)," *Balaba: Jurnal Litbang Pengendalian Penyakit Bersumber Binatang Banjarnegara*, pp. 19–28, 2021, doi: 10.22435/blb.v17i1.3556.
 - [8] A. Sai Bharadwaj Reddy and D. Sujitha Juliet, "Transfer learning with RESNET-50 for malaria cell-image classification," *Proceedings of the 2019 IEEE International Conference on Communication and Signal Processing, ICCSP 2019*, no. April 2019, pp. 945–949, 2019, doi: 10.1109/ICCSP.2019.8697909.
 - [9] M. Masud *et al.*, "Leveraging Deep Learning Techniques for Malaria Parasite Detection Using Mobile Application," *Wirel Commun Mob Comput*, vol. 2020, 2020, doi: 10.1155/2020/8895429.
 - [10] A. Q. M. S. Sayyed, D. Saha, A. R. Hossain, and C. Shahnaz, "Effectiveness of Convolutional and Capsule network in Malaria Parasite Detection," *2019 IEEE International Conference on Signal Processing, Information, Communication and Systems, SPICSCON 2019*, pp. 68–73, Nov. 2019, doi: 10.1109/SPICSCON48833.2019.9065074.
 - [11] M. H. D. Alnussairi and A. A. Ibrahim, "Malaria parasite detection using deep learning algorithms based on (CNNs) technique," *Computers and Electrical Engineering*, vol. 103, p. 108316, Oct. 2022, doi: 10.1016/J.COMPELECENG.2022.108316.
 - [12] K. K. Jena, S. Kumar Bhoi, C. Mallick, D. Mohapatra, and P. Swain, "Classification of Malaria Parasitized and Uninfected Images Using Machine Learning Approach," *Proceedings of the 5th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2021*, pp. 1274–1279, 2021, doi: 10.1109/I-SMAC52330.2021.9640905.
 - [13] W. R. W. M. Razin, T. S. Gunawan, M. Kartiwi, and N. Md. Yusoff, "Malaria Parasite Detection and Classification using CNN and YOLOv5 Architectures," pp. 277–281, Nov. 2022, doi: 10.1109/ICSIMA55652.2022.9928992.
 - [14] W. Deelder *et al.*, "Using deep learning to identify recent positive selection in malaria parasite sequence data," *Malar J*, vol. 20, no. 1, pp. 1–9, Dec. 2021, doi: 10.1186/S12936-021-03788-X/TABLES/3.
 - [15] M. O. Arowolo, M. Adebisi, A. Adebisi, and O. Okesola, "PCA Model for RNA-Seq Malaria Vector Data Classification Using KNN and Decision Tree Algorithm," *2020 International Conference in Mathematics, Computer Engineering and Computer Science, ICMCECS 2020*, Mar. 2020, doi: 10.1109/ICMCECS47690.2020.240881.
 - [16] K. M. F. Fuhad, J. F. Tuba, M. R. A. Sarker, S. Momen, N. Mohammed, and T. Rahman, "Deep Learning Based Automatic Malaria Parasite Detection from Blood Smear and Its Smartphone Based Application," *Diagnostics 2020, Vol. 10, Page 329*, vol. 10, no. 5, p. 329, May 2020, doi: 10.3390/DIAGNOSTICS10050329.
 - [17] G. Comert, N. Begashaw, and A. Turhan-Comert, "Malaria Outbreak Detection with Machine Learning Methods," *bioRxiv*, p. 2020.07.21.214213, Jul. 2020, doi: 10.1101/2020.07.21.214213.
 - [18] O. Nkiruka, R. Prasad, and O. Clement, "Prediction of malaria incidence using climate variability and machine learning," *Inform Med Unlocked*, vol. 22, p. 100508, Jan. 2021, doi: 10.1016/J.IMU.2020.100508.
 - [19] Y. W. Lee, J. W. Choi, and E. H. Shin, "Machine learning model for predicting malaria using clinical information," *Comput Biol Med*, vol. 129, p. 104151, Feb. 2021, doi: 10.1016/J.COMPBIOMED.2020.104151.
 - [20] A. Q. M. S. Sayyed, D. Saha, A. R. Hossain, and C. Shahnaz, "Effectiveness of Convolutional and Capsule network in Malaria Parasite Detection," *2019 IEEE International Conference on Signal Processing, Information, Communication and Systems, SPICSCON 2019*, no. November, pp. 68–73, 2019, doi: 10.1109/SPICSCON48833.2019.9065074.
 - [21] L. A. Andika, H. Pratiwi, and S. S. Handajani, "Lingga Aji Andika 1, Hasih Pratiwi 2, and Sri Sulistijowati Handajani 3 1," *Indonesian Journal of Statistics and Its Applications*, vol. 3, no. 3, pp. 331–340, 2019.
 - [22] K. Sriporn, C. F. Tsai, C. E. Tsai, and P. Wang, "Analyzing Malaria Disease Using Effective Deep Learning Approach," *Diagnostics 2020, Vol. 10, Page 744*, vol. 10, no. 10, p. 744, Sep. 2020, doi: 10.3390/DIAGNOSTICS10100744.
 - [23] M. Suriya, V. Chandran, and M. G. Sumithra, "Enhanced deep convolutional neural network for malarial parasite classification," *International Journal of Computers and Applications*, vol. 0, no. 0, pp. 1–10, 2019, doi: 10.1080/1206212X.2019.1672277.
 - [24] "Cell Image Malaria Dataset Available : <https://www.kaggle.com/iarunava/cell-images-for-detecting-malaria>."
 - [25] "NIH: Cell Image Malaria Dataset Available : <https://ceb.nlm.nih.gov/repositories/malaria-datasets/>".
 - [26] A. Rahman, H. Zunair, T. R. Reme, M. S. Rahman, and M. R. C. Mahdy, "A comparative analysis of deep learning architectures on high variation malaria parasite classification dataset," *Tissue Cell*, vol. 69, p. 101473, Apr. 2021, doi: 10.1016/J.TICE.2020.101473.
 - [27] M. Harahap, J. Jefferson, S. Barti, S. Samosir, and C. A. Turnip, "Implementation of Convolutional Neural Network in the classification of red blood cells have affected of malaria," *Sinkron*, vol. 5, no. 2, pp. 199–207, 2020, doi: 10.33395/sinkron.v5i2.10713.
 - [28] S. D. H. Permana, G. Saputra, B. Arifitama, Yaddarabullah, W. Caesarendra, and R. Rahim, "Classification of bird sounds as an early warning method of forest fires using Convolutional Neural Network (CNN) algorithm," *Journal of King Saud University - Computer and Information Sciences*, no. xxxx, 2021, doi: 10.1016/j.jksuci.2021.04.013.
 - [29] Z. Liang *et al.*, "CNN-based image analysis for malaria diagnosis," *Proceedings - 2016 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2016*, pp. 493–496, 2017, doi: 10.1109/BIBM.2016.7822567.
 - [30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 2818–2826, Dec. 2016, doi: 10.1109/CVPR.2016.308.
 - [31] I. K. A. Wirayasa, "Comparison of Convolutional Neural Networks Model Using Different Optimizers for Image Classification International Journal of Sciences: Comparison of Convolutional Neural Networks Model Using Different Optimizers for Image Classification," no. September, 2021.
 - [32] Y. Bhatia, A. Bajpayee, D. Raghuvanshi, and H. Mittal, "v2 and Recurrent Neural Network," *2019 Twelfth International Conference on Contemporary Computing (IC3)*, pp. 1–6, 2019.
 - [33] D. Shah, K. Kawale, M. Shah, S. Randive, and R. Mapari, "Malaria Parasite Detection Using Deep Learning: (Beneficial to humankind)," *Proceedings of the International Conference on Intelligent Computing and Control Systems, ICICCS 2020*, no. Iciccs, pp. 984–988, 2020, doi: 10.1109/ICICCS48265.2020.9121073.
 - [34] A. Sai Bharadwaj Reddy and D. Sujitha Juliet, "Transfer learning with RESNET-50 for malaria cell-image classification," *Proceedings of the 2019 IEEE International Conference on Communication and Signal Processing, ICCSP 2019*, pp. 945–949, Apr. 2019, doi: 10.1109/ICCSP.2019.8697909.