

# **DETECTING MALARIA USING CONVOLUTIONAL NEURAL NETWORKS**

Maureen Kitang'a

*13/5/2024*

# 1. INTRODUCTION

Malaria is a life-threatening disease caused by parasites transmitted through the bites of infected mosquitoes. It remains a significant global health challenge, particularly in regions with limited access to healthcare resources. Timely and accurate diagnosis is crucial for effective treatment and disease management. In recent years, deep learning techniques, specifically Convolutional Neural Networks (CNNs), have emerged as powerful tools for medical image analysis tasks, including malaria detection. This report presents a study on the development and evaluation of a CNN model for automated malaria diagnosis using TensorFlow, an open-source machine learning library developed by Google.

## 2. OBJECTIVE

The primary objective of this project is to develop a CNN model capable of accurately classifying blood smear images into two categories: infected with malaria parasites or uninfected. The model aims to provide a reliable tool for automated malaria diagnosis, which can assist healthcare professionals in timely disease identification and treatment.

## 3. APPROACH

### Data Collection and Preprocessing

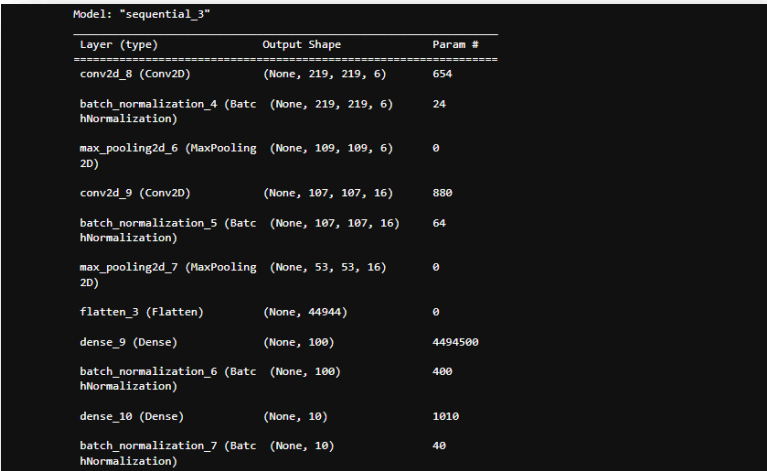
The malaria dataset was obtained using TensorFlow Datasets, comprising blood smear images labeled as infected or uninfected. The images were preprocessed by resizing and rescaling to a uniform size of 224x224 pixels and normalizing pixel values to the range [0, 1].

## Model Development

The CNN model architecture was inspired by LeNet-5, consisting of convolutional layers, batch normalization, max-pooling, and dense layers. The model was implemented using TensorFlow's Sequential API.

## Model Training and Evaluation

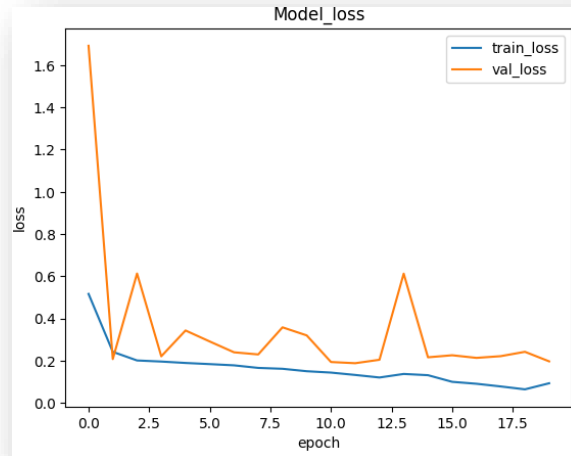
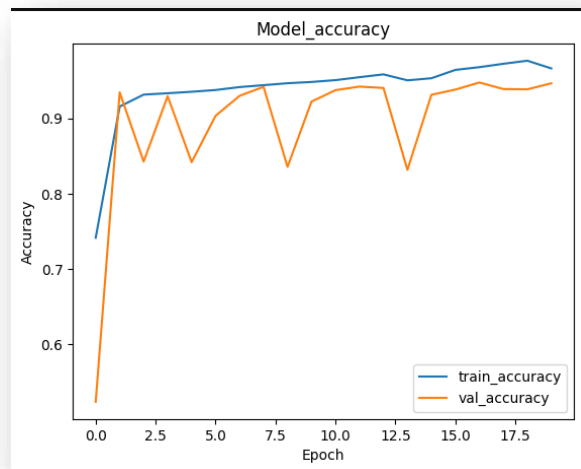
The model was trained using the Adam optimizer with a binary cross-entropy loss function. Training was performed over 20 epochs, with early stopping implemented to prevent overfitting. The model achieved a validation accuracy of approximately 94.66%.

A screenshot of a terminal window with a dark background. It displays the model architecture for 'sequential\_3'. The output is a table with three columns: 'Layer (type)', 'Output Shape', and 'Param #'. The layers listed are: conv2d\_8 (Conv2D), batch\_normalization\_4 (Batch Normalization), max\_pooling2d\_6 (Max Pooling 2D), conv2d\_9 (Conv2D), batch\_normalization\_5 (Batch Normalization), max\_pooling2d\_7 (Max Pooling 2D), flatten\_3 (Flatten), dense\_9 (Dense), batch\_normalization\_6 (Batch Normalization), dense\_10 (Dense), and batch\_normalization\_7 (Batch Normalization).

Layer (type)	Output Shape	Param #
conv2d_8 (Conv2D)	(None, 219, 219, 6)	654
batch_normalization_4 (Batch Normalization)	(None, 219, 219, 6)	24
max_pooling2d_6 (Max Pooling 2D)	(None, 109, 109, 6)	0
conv2d_9 (Conv2D)	(None, 107, 107, 16)	880
batch_normalization_5 (Batch Normalization)	(None, 107, 107, 16)	64
max_pooling2d_7 (Max Pooling 2D)	(None, 53, 53, 16)	0
flatten_3 (Flatten)	(None, 44944)	0
dense_9 (Dense)	(None, 100)	4494500
batch_normalization_6 (Batch Normalization)	(None, 100)	400
dense_10 (Dense)	(None, 10)	1010
batch_normalization_7 (Batch Normalization)	(None, 10)	40

## Model Performance and Visualization

Training and validation curves were plotted to visualize the model's learning process. The curves demonstrated a decreasing trend in both training and validation loss, indicating effective model training and generalization.



## Model Evaluation on Test Data

The trained model was evaluated on a separate test dataset, achieving a test accuracy of approximately 94.27%. Additionally, sample predictions were generated, showcasing the model's ability to classify malaria-infected and uninfected images accurately.

```
2757/2757 [=====] - 23s 4ms/step - loss: 0.2092 - accuracy: 0.9427
Out[84]: [0.20915743708610535, 0.9426913261413574]
```

## Model Saving and Deployment

The trained model was saved in multiple formats, including a TensorFlow SavedModel, HDF5 format, and model weights. These saved formats ensure portability and compatibility for future deployment and usage.

## 4. FUTURE WORK

***Fine-tuning:*** Further fine-tuning of the model parameters and architecture could improve its performance and generalization ability.

***Data Augmentation:*** Integration of data augmentation techniques could enhance the model's robustness and reduce overfitting.

***Deployment:*** Exploration of deployment options, such as integration into healthcare systems or mobile applications, could facilitate real-world usage and impact.

## 5. ACKNOWLEDGMENTS

We acknowledge the contributions of TensorFlow, TensorFlow Datasets, and the broader open-source community for their invaluable tools and resources that enabled this project's execution.

## 6. REFERENCES

### ***Model Architecture***

- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. \*Proceedings of the IEEE, 86\*(11), 2278-2324.

- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. \*Advances in neural information processing systems, 25\*, 1097-1105.

### ***TensorFlow and TensorFlow Datasets***

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Zheng, X. (2016). TensorFlow: Large-scale machine learning on heterogeneous distributed systems. \*arXiv preprint arXiv:1603.04467\*.

- TensorFlow Datasets. TensorFlow Datasets Documentation.  
[<https://www.tensorflow.org/datasets>]

### ***Model Training and Optimization***

- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. \*arXiv preprint arXiv:1412.6980\*.

- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. \*Journal of Machine Learning Research, 15\*(1), 1929-1958.

### ***Model Evaluation Metrics***

- Zhou, J., & Troyanskaya, O. G. (2018). Predicting effects of noncoding variants with deep learning-based sequence model. \*Nature methods, 15\*(8), 575-581.

- Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. \*PloS one, 10\*(3), e0118432.