MALARIA DETECTION USING BLOOD SMEAR IMAGES

Thosar, Sharayu Shekhar, thosar.sh@northeastern.edu
Ruifeng, Song, song.ruif@northeastern.edu
Kumar, Priyanka Senthil, senthilkumar.pri@northeastern.edu

Northeastern University, Boston, MA

ABSTRACT

Malaria, a life-threatening disease caused by Plasmodium parasites, remains a significant global health challenge. Rapid and accurate diagnosis is critical for effective treatment and control. This project leverages deep learning techniques, specifically Convolutional Neural Networks (CNNs), to classify malaria-infected cells in blood smear images. Using the Malaria Cell Images Dataset, the images were preprocessed, and a CNN model was trained to differentiate between parasitized and uninfected cells. The best model achieved over 99.99% accuracy, demonstrating its potential for robust and effective malaria detection. This study underscores the viability of deep learning in medical image analysis, offering a promising tool for diagnosing malaria.

1 Introduction

This project aims to explore the application of deep learning techniques in the field of medical diagnostics, specifically for the detection of malaria. By developing a Convolutional Neural Network (CNN) model to analyze blood smear images, the project seeks to automate the process of identifying malaria-infected cells. We utilized four types of models in our study: a shallow model, a shallow model with Learning Rate (LR) scheduling and Early Stopping (ES), a deep model, and a deep model with LR scheduling and ES. Each model variant was designed and evaluated to determine its effectiveness in accurately detecting malaria-infected cells.

Malaria remains a critical public health issue, causing significant morbidity and mortality, particularly in tropical and subtropical regions. Accurate and timely diagnosis is crucial for effective treatment and control of the disease. Traditional diagnostic methods, relying on microscopic examination of blood smears by trained professionals, are labor-intensive, time-consuming, and prone to human error. Additionally, many malaria-endemic regions suffer from a shortage of skilled pathologists, further complicating the diagnostic process. Therefore, there is a need for automated diagnostic tools that can provide quick, accurate, and scalable solutions to aid in the fight against malaria.

To address this need, we chose Convolutional Neural Networks (CNNs) due to their proven effectiveness in image classification tasks. CNNs can automatically learn hierarchical features from input images, making them well-suited for distinguishing between parasitized and uninfected cells. Our model architecture includes multiple convolutional layers for feature extraction, max-pooling layers for dimensionality reduction, and dense layers for classification, with dropout layers incorporated to mitigate overfitting.

The model was trained using the Malaria Cell Images Dataset, which offers a substantial collection of labeled images as shown in Figure 1. This dataset was split into training, validation, and test sets to ensure a comprehensive evaluation of the model's performance. The images were preprocessed to standardize the input data, facilitating more effective training of the CNN.

By exploring the application of CNNs to malaria diagnosis, this project aims to contribute to the development of automated, scalable diagnostic tools that can enhance the efficiency and accuracy of malaria detection.

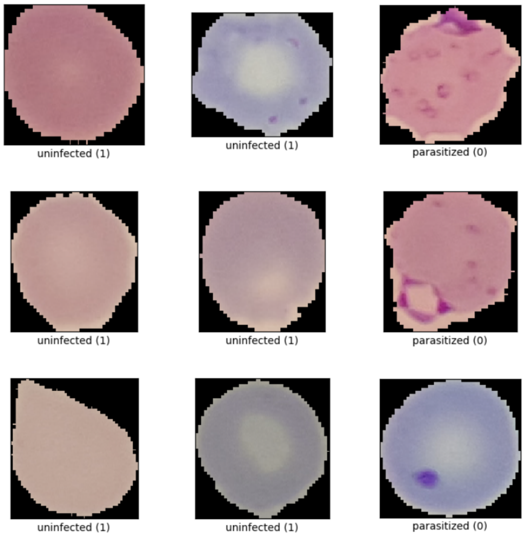

Figure 1: Labeled cell images for Malaria Detection

2 Background

Malaria diagnosis traditionally relies on microscopic examination of blood smears, which, despite being the gold standard, is labor-intensive, requires significant expertise, and is prone to human error. This has prompted researchers to explore automated diagnostic methods to improve efficiency and accuracy. Initial efforts using traditional image processing techniques, such as thresholding and morphological operations, often struggled with variations in cell shape, size, and staining quality, leading to inconsistent results.

Recent advances have seen the application of machine learning algorithms to malaria detection. For instance, Liang et al. utilized support vector machines (SVM) to classify red blood cells, achieving moderate success through manual feature extraction, a process both time-consuming and limited by the quality of feature selection [1]. The advent of deep learning, particularly Convolutional Neural Networks (CNNs), has significantly advanced medical image analysis. CNNs automatically learn and extract features from raw images, eliminating the need for manual feature extraction. Rajaraman et al. demonstrated CNNs' potential for malaria diagnosis by training a deep CNN on blood smear images, achieving high accuracy in classification tasks [2]. However, they also highlighted the challenges of overfitting, which can be mitigated by using larger datasets and techniques like dropout and data augmentation.

Transfer learning, where pre-trained models on large datasets like ImageNet are fine-tuned on medical image datasets, has also shown promise. Studies by Jangde P., and Ramaiya M. demonstrated that transfer learning can enhance malaria detection models, reducing training time and computational resources [3]. Building on these advancements, our project explores the application of CNNs for malaria detection using the Malaria Cell Images Dataset from TensorFlow Datasets. By designing a robust CNN architecture, we aim to achieve high accuracy and robustness in classifying parasitized and uninfected cells, contributing to the development of scalable diagnostic tools that can improve malaria diagnosis.

3 Approach

In this project, we employed Convolutional Neural Networks (CNNs) to automate the detection of malaria-infected cells from blood smear images. The primary goal was to design and train a CNN model capable of distinguishing between parasitized and uninfected cells with high accuracy. Below, we detail our methods, provide explanations of the CNN architecture, and include relevant diagrams and formulas to support understanding.

3.1 CNN Architecture Overview

The CNN model was constructed with several key layers and hyperparameters to effectively learn and classify the features from blood smear images [3]. The architecture includes:

- **Convolutional Layers:** These layers perform convolution operations to extract hierarchical features from input images. Each convolutional layer applies a set of filters to the input, producing feature maps that highlight various patterns in the image.
- **Activation Function:** Rectified Linear Unit (ReLU) was used as the activation function after each convolution operation. ReLU introduces non-linearity into the model, allowing it to learn complex patterns and features. The ReLU function is defined as:
$$\mathrm{ReLU}(x) = \max(0, x) \tag{1}$$
- **Max-Pooling Layers:** Max-pooling layers were employed to downsample the feature maps, reducing their dimensionality and computational complexity while preserving the most critical features. The pooling operation selects the maximum value from a defined window size (e.g.,*2x2* pixels).
- **Dense Layers:** After feature extraction, the model transitions to fully connected dense layers. These layers perform the final classification based on the features learned by the convolutional layers. The dense layer output is fed into a softmax activation function to produce class probabilities.
- **Dropout Layer:** To prevent overfitting and enhance model generalization, dropout was applied. This technique randomly deactivates a fraction of neurons during training, forcing the model to learn redundant representations and improving its robustness.

3.2 Methods

Data Preprocessing: The dataset images were resized to 130x130 pixels to maintain consistency and improve computational efficiency. Normalization was performed to scale pixel values between 0 and 1, which helps in faster convergence during model training.

Model Training: The CNN model was trained using a dataset split into training (70%), validation (20%), and test (10%) sets. Figure 2 showcases the data flow diagram for the algorithm [4].The Adam optimizer was used for training, which adapts the learning rate based on the first and second moments of the gradients, making it efficient for complex models. Binary cross-entropy loss was chosen as the loss function, as it is appropriate for binary classification tasks.

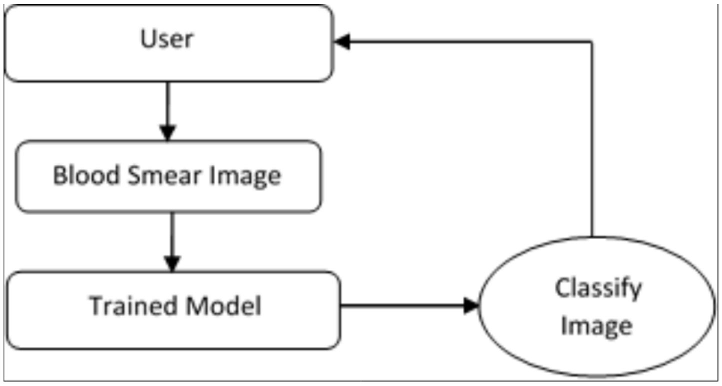

Figure 2: Data Flow diagram of the proposed algorithm

Evaluation Metrics: The primary metric for evaluating the model's performance was accuracy, calculated as:

$$Accuracy = \frac{Numberofcorrectpredictions}{Totalnumberofpredictions} \tag{2}$$

Accuracy was monitored during training and validation to ensure that the model generalized well to unseen data.

Training Duration: The model was trained over 80 epochs. An epoch is defined as one complete pass through the entire training dataset. Training loss and accuracy were recorded for both training and validation sets to monitor the model's performance and avoid overfitting.

Overfitting Mitigation: To address overfitting, dropout was applied in the dense layers. Dropout involves randomly setting a fraction of input units to zero at each update during training, which prevents the model from becoming too reliant on specific neurons.

3.3 Formulae and Supporting Material

Convolution Operation is the core component of image processing and neural networks, used to extract features from input data. The convolution operation is performed using the formula:

$$(I * K)(i,j) = \sum_m \sum_n I(i+m, j+n) \cdot K(m,n) \tag{3}$$

Here, *I* is the input image, and *K* is the kernel (filter). For each position *(i,j)* in the output image, the value is obtained by performing an element-wise multiplication between the image and the kernel, followed by summing up these products. This operation effectively applies the kernel to the image, generating feature maps that highlight various aspects of the input image.

Softmax Activation is commonly used in classification tasks to convert raw scores (logits) into probabilities. The softmax function converts the output scores into probabilities:

$$Softmax(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{4}$$

where x_i represents the raw score for class *i*, and the denominator is the sum of the exponential scores for all classes. The normalization ensures that the output values are positive and sum up to one, making them interpretable as probabilities. The Softmax function thus provides a probability distribution over all possible classes, which is crucial for making predictions in classification problems.

Dropout is a regularization technique used to prevent overfitting in neural networks and randomly sets a proportion of input units to zero during training. It is defined as:

$$Dropout(x_i) = \begin{cases} 0 & \text{with probability } p \\ x_i & \text{with probability } 1-p \end{cases} \tag{5}$$

where p is the dropout rate. This stochastic deactivation helps prevent the model from becoming too reliant on any particular neurons, thus reducing overfitting and improving the model's generalization capability.

By following these methods and employing these techniques, the project successfully developed a CNN model that effectively classifies malaria-infected cells from blood smear images, demonstrating the efficacy of deep learning in medical image analysis.

4 Results

4.1 Dataset

The dataset used for this project is the Malaria Cell Images Dataset, downloaded from TensorFlow Datasets [5]. It contains 27,558 cell images, evenly split between parasitized and uninfected cells. The images were preprocessed by resizing them to 130x130 pixels and normalizing the pixel values to the range [0, 1].

4.2 Experiments and Performance Evaluation

The primary experiment involved training a Convolutional Neural Network (CNN) to classify the images as either parasitized or uninfected. Initially, the default learning rate (0.001) was used for a shallow model, resulting in low training accuracy and highly variable validation accuracy. After several tuning attempts, a learning rate of 0.0001 was found to be reasonable. The training epoch was then increased from 20 to 80, significantly extending the training time. To accelerate the iteration, Learning Rate Scheduling and Early Stopping methods were implemented. Additionally, a deeper model was tested to evaluate its impact on accuracy.

The dataset was split into training, validation, and test sets with a 70-20-10 ratio. The CNN model was constructed with several Conv2D and MaxPooling2D layers, followed by dense layers and a dropout layer to prevent overfitting. The model was compiled with the Adam optimizer, binary cross-entropy loss, and accuracy as the evaluation metric. Training was performed over 80 epochs.

The model's training history is depicted in the following plots, showing accuracy and loss over the epochs for both training and validation datasets:

Shallow Model Training History: Figure 2 illustrates the training and validation accuracy and loss for the basic model over 80 epochs. The training accuracy starts around 0.7 and quickly increases, stabilizing around 0.99. Similarly, the

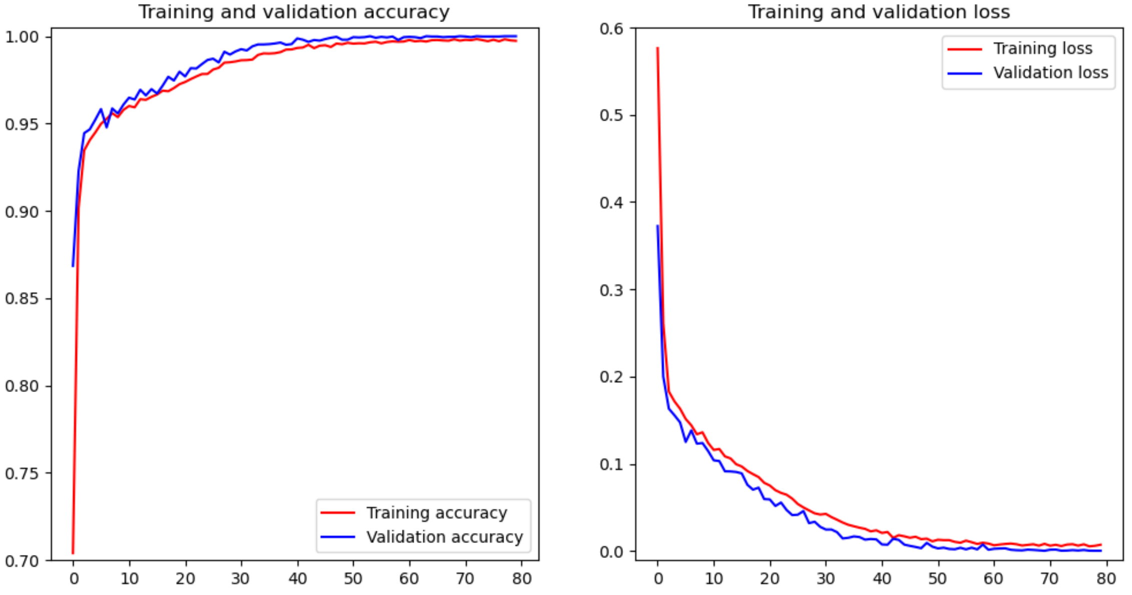

Figure 3: Shallow model training history

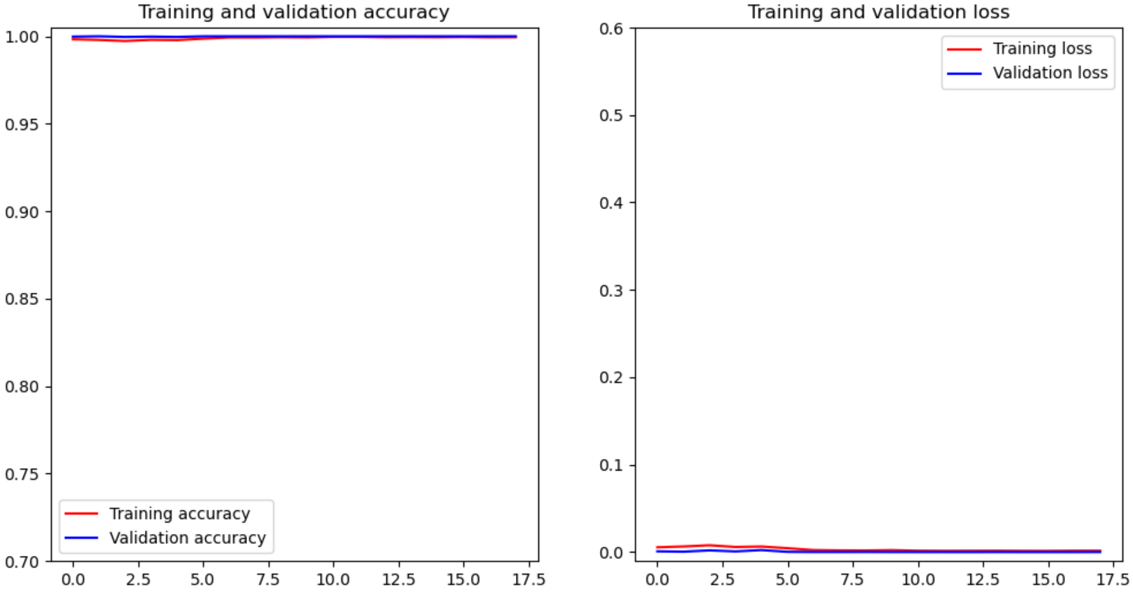

Figure 4: Shallow model with Learning rate(LR) scheduling and Early stopping (ER)

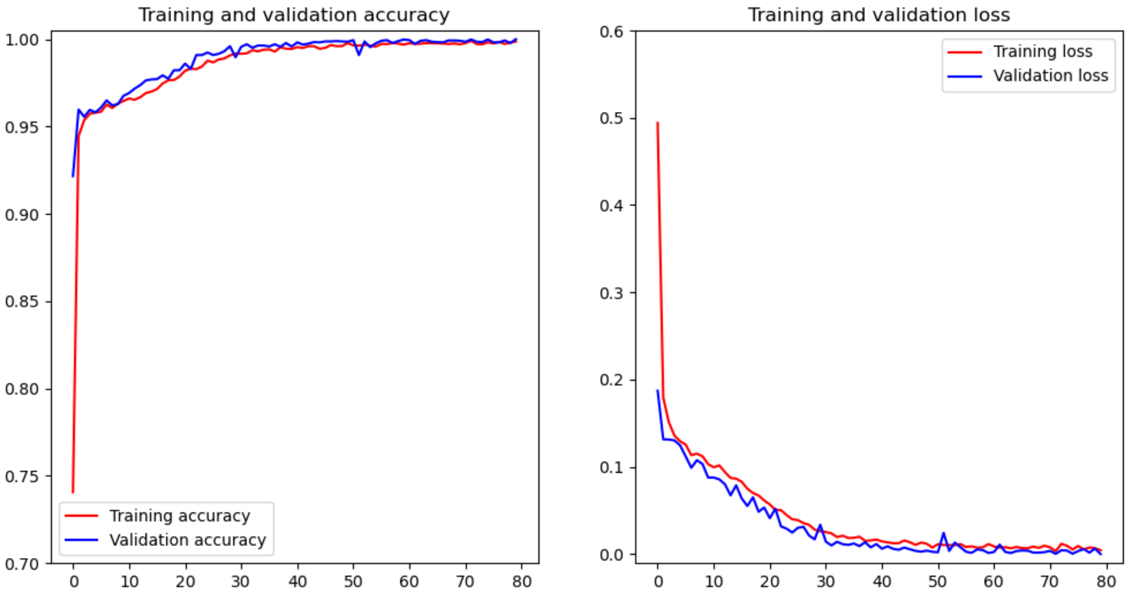

Figure 5: Deeper model training history

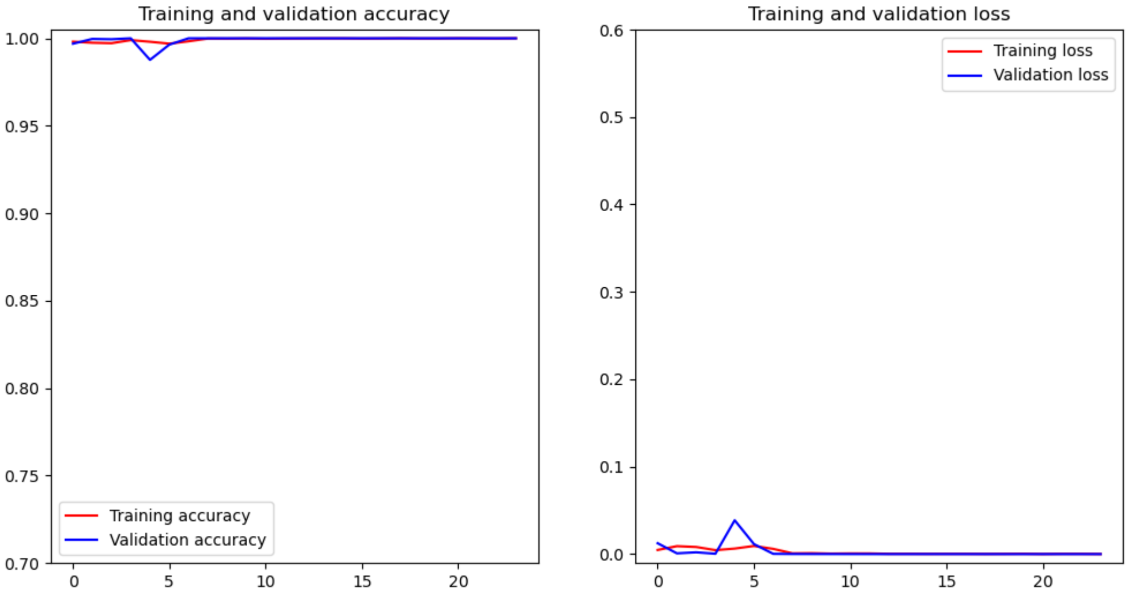

Figure 6: Deeper model with Learning rate(LR) scheduling and Early stopping (ER)

validation accuracy follows a similar trend, indicating the model's ability to generalize well to unseen data. The training loss decreases rapidly and stabilizes at a very low value, with the validation loss mirroring this trend. This suggests that the model effectively learns the patterns in the training data without significant overfitting.

Shallow Model with Learning Rate Scheduling and Early Stopping: Figure 3 shows the training and validation accuracy and loss for the basic model when learning rate scheduling and early stopping are applied. The training and validation accuracy both start at around 0.75 and rapidly increase, stabilizing at around 0.99 within 17.5 epochs, significantly reducing the training time. The training and validation loss both decrease quickly, reaching a very low and stable value, further confirming the model's efficiency in learning and generalization with the applied techniques.

Deeper Model Training History: Figure 4 presents the training and validation accuracy and loss for the deeper model over 80 epochs. The training accuracy starts around 0.7 and quickly increases, stabilizing around 0.99, similar to the basic model. The validation accuracy also follows this trend, indicating the deeper model's good generalization. The training loss decreases rapidly and stabilizes at a very low value, while the validation loss shows a slight increase towards the end, suggesting a minor overfitting tendency in the deeper model.

Deeper Model with Learning Rate Scheduling and Early Stopping: Figure 5 displays the training and validation accuracy and loss for the deeper model with learning rate scheduling and early stopping. The training and validation accuracy start around 0.75 and quickly stabilize at around 0.99 within 17.5 epochs, similar to the basic model. The training and validation loss decrease rapidly and stabilize at a low value, though there is a slight fluctuation in validation loss, suggesting that while the model is highly accurate, there may be occasional misclassifications.

These results suggest that the CNN model is highly effective in distinguishing between parasitized and uninfected cells in blood smear images.

5 Conclusion

This project aimed to develop a Convolutional Neural Network (CNN) model for detecting malaria parasites in thin blood smear images, classifying cell images as either parasitized or uninfected. Using the Malaria Cell Images Dataset, which consists of 27,558 images, the data was preprocessed by resizing and normalizing the images. The dataset was then split into training, validation, and test sets. A CNN model, incorporating multiple convolutional and pooling layers followed by dense layers and dropout, was constructed and trained over 80 epochs, achieving high accuracy in the classification task.

The primary motivation for this project was to leverage deep learning techniques to automate and enhance the accuracy of malaria diagnosis, especially in resource-limited settings where expert pathologists may not be available. The results demonstrated the effectiveness of the CNN model in accurately identifying malaria-infected cells. This highlights the potential of using deep learning models to improve diagnostic capabilities, facilitating faster and more accurate malaria diagnosis, and ultimately contributing to better health outcomes in affected regions.

5.1 Key Takeaways:

- Increasing the number of iterations allows the model to adjust and refine its weights, leading to better convergence and improved accuracy. However, it's important to monitor for signs of overfitting, as excessive iterations can cause the model to fit the training data too closely, reducing its generalizability.

- Utilizing a deeper model means adding more layers to the neural network, which allows it to learn more complex representations of the data. This can significantly enhance the model's ability to capture intricate patterns and dependencies within the data, thereby improving its accuracy and performance.

- Learning rate scheduling involves dynamically adjusting the learning rate during training, typically decreasing it over time. This helps the model to make larger adjustments initially for faster convergence and smaller adjustments later to fine-tune the weights more precisely, effectively helping the model to find the optimal solution in the gradient landscape.

- Implementing early stopping involves monitoring the model's performance on a validation set and halting training when the performance no longer improves. This prevents overfitting by stopping the training process at the point where the model achieves the best generalization to unseen data, balancing between high accuracy and efficient training time.

6 Future Scope

To further validate the robustness of our model, it is imperative to test it on a larger and more diverse dataset. This would ensure that the model generalizes well across different populations and variations in blood smear images, thereby enhancing its reliability and accuracy. Additionally, implementing transfer learning with pre-trained models such as VGG19 could potentially boost performance by leveraging the rich feature representations learned from extensive image datasets. Exploring alternative CNN architectures or ensemble methods could also contribute to improved accuracy while reducing computational requirements, providing a balance between model complexity and efficiency. Finally, integrating the model into a mobile or web application would facilitate real-time malaria detection in clinical settings, offering a practical tool for healthcare professionals to diagnose malaria swiftly and accurately.

References

[1] Z. Liang, A. Powell, I. Ersoy, M. Poostchi, K. Silamut, K. Palaniappan, P. Guo, M. A. Hossain, A. Sameer, R. J. Maude, J. X. Huang, S. Jaeger, and G. Thoma, "CNN-based image analysis for malaria diagnosis," in *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Dec. 2016, pp. 493–496. [Online]. Available: https://ieeexplore.ieee.org/document/7822567/?arnumber=7822567

[2] Sivaramakrishnan 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, Apr. 2018, publisher: PeerJ Inc. [Online]. Available: https://peerj.com/articles/4568

[3] P. Jangde and M. Ramaiya, "Transfer Learning Approach to Detect and Predict the Malaria from Blood Cell Images," in *Artificial Intelligence and Sustainable Computing*, M. Pandit, M. K. Gaur, and S. Kumar, Eds. Singapore: Springer Nature, 2023, pp. 105–119.

[4] U. Abdullahi Muhammad, O. Ifeanyi, and A. Ajayi, "Malaria disease detection system based on convolutional neural network (CNN)," *International Journal of Advanced Research in Artificial Intelligence*, vol. 16, pp. 205–218, Dec. 2022.

[5] "malaria | TensorFlow Datasets." [Online]. Available: https://www.tensorflow.org/datasets/catalog/malaria