# Malaria Detection Using Convolutional Neural Networks

## 1. Introduction

Malaria, caused by Plasmodium parasites, poses a significant health challenge, particularly in developing countries. Accurate and rapid diagnostic methods are crucial for effective management and control. Traditional methods can be time-consuming and resource-intensive. This project explores the use of convolutional neural networks (CNNs) for automated malaria detection from microscopic blood cell images. By leveraging pre-trained models and transfer learning, I aim to develop a system that can efficiently classify cells as either parasitized or uninfected. This report details the methodology, experimental setup, results, and discussion of our findings.

## 2. Background

Traditional malaria diagnosis involves microscopic examination of stained blood smears, which requires skilled personnel and can be subjective. Machine learning offers a promising alternative by automating the analysis of cell images. CNNs have shown remarkable success in image classification tasks, making them well-suited for this problem. Transfer learning, where a pre-trained model is fine-tuned for a new task, can significantly reduce training time and improve performance, especially when dealing with limited datasets.

## 3. Methods

### 3.1. Dataset

The dataset consists of microscopic images of blood cells, labeled as either parasitized (1) or uninfected (0). The training set is provided as a folder of images with corresponding labels in a CSV file ("train_data.csv"). A separate folder of test images ("test_images") is provided without labels for generating predictions.

### 3.2. Data Preprocessing and Augmentation

Data preprocessing is a crucial step in preparing the images for the CNN model. I used the following transformations, implemented using `torchvision.transforms`:

* **Resizing:** Images were resized to 224x224 pixels, which is the expected input size for the ResNet50 model.

* **Data Augmentation (Training Set Only):** To increase the diversity of the training data and improve the model's generalization ability, I applied the following data augmentation techniques:

    * RandomRotation(15): Randomly rotates the image by up to 15 degrees.

    * RandomHorizontalFlip(): Randomly flips the image horizontally.

    * RandomResizedCrop(224, scale=(0.8, 1.0)): Randomly crops a section of the image with an area between 80% and 100% of the original size, then resizes it to 224x224. This helps the model become more robust to variations in the cell's position and size within the image.

* **ToTensor():** Converts the image to a PyTorch tensor.

* **Normalization:** Normalizes the pixel values to have a mean of 0.5 and a standard deviation of 0.5 across all color channels. Using `transforms.Normalize(mean=[0.5, 0.5, 0.5], std=[0.5, 0.5, 0.5])`. The test set was transformed in the same way, without the data augmentation.

### 3.3. Model Architecture

I used a pre-trained ResNet50 model, which is a deep convolutional neural network known for its excellent performance in image classification. The ResNet50 model was pre-trained on the ImageNet dataset, a large dataset of diverse images. I leveraged transfer learning by using the pre-trained weights as a starting point for our malaria detection task. The model was modified by replacing the final fully connected layer with a new fully connected layer that outputs two values (one for each class: parasitized and uninfected).

### 3.4. Training

The model was trained using the following settings:

* **Loss Function:** Cross-Entropy Loss (torch.nn.CrossEntropyLoss) was used to measure the difference between the model's predictions and the true labels.

* **Optimizer:** Adam optimizer (torch.optim.Adam) with a learning rate of 0.001 was used to update the model's weights during training. The learning rate was chosen to balance the speed of training and the risk of overshooting the optimal weights.

* **Freezing Layers:** To leverage the pre-trained weights, all layers of the ResNet50 model were frozen, except for the final fully connected layer. This allowed the model to adapt the

pre-trained features to the specific characteristics of malaria cell images. Only the weights of the final, fully connected layer are trained.

* **Early Stopping:** To prevent overfitting and improve generalization, we implemented early stopping. We monitored the validation loss during training. If the validation loss did not decrease for 5 consecutive epochs, the training process was stopped, and the model with the best validation loss was restored.

* **Training Loop:** The model was trained for 10 epochs. In each epoch, the model was trained on the training set and evaluated on the validation set. The training data was loaded in batches of 256 images. The model's weights were updated after each batch of training data.

### 3.5. Evaluation

The model's performance was evaluated on the validation set. The following metrics were used:

* **Accuracy:** The percentage of correctly classified images.

* **F1-Score:** The harmonic mean of precision and recall, which provides a balanced measure of the model's performance.

### 3.6. Implementation Details

The code was implemented in Python using the PyTorch library. The following libraries were also used:

* **numpy:** For numerical computations.

* **pandas:** For data manipulation.

* **matplotlib:** For plotting.

* **torchvision:** For image transformations and loading pre-trained models.

* **sklearn:** For data splitting and evaluation metrics.

## 4. Results

| Attempt | Image Size | Data Augmentation | Early Stopping | F1-Score | Notes |
|---|---|---|---|---|---|
| 1 | 128 | No | No | 0.92933 | |

| | | | | | |
|---|---|---|---|---|---|
| 2 | 128 | Yes | Yes | 0.96222 | |
| 3 | 224 | Yes | Yes | 0.96714 | **Achieved Best** |

**Detailed Explanation of Best Result (Attempt 3):**

With image resizing to 224x224 pixels and data augmentation, the model achieved an F1-score of 96.714%. The increased image size allowed the model to learn more fine-grained features, and data augmentation helped to improve the model's generalization ability by exposing it to a wider range of variations in the training data. Early stopping further enhanced the model's generalization by preventing overfitting to the training data.

## 5. Discussion

The results demonstrate the effectiveness of using pre-trained CNNs and transfer learning for malaria detection. The ResNet50 model, fine-tuned on our dataset, achieved high accuracy and F1-score on the validation set.

The improvement in performance with data augmentation highlights the importance of increasing the diversity of the training data. Data augmentation helps to prevent overfitting and improves the model's ability to generalize to unseen images.

The increased image size also contributed to the improved performance. Larger images allow the model to learn more detailed features, which can be helpful for distinguishing between parasitized and uninfected cells.

## 6. Bonus Question

**1. Do you notice any meaningful patterns in the learned embeddings?**

Yes, your UMAP visualization shows that the learned embeddings capture some structure in the data:

Some separation between healthy (blue) and sick (red) cells is visible, especially in certain regions of the plot.

However, there is a significant overlap between the two classes, suggesting that the model may struggle to fully differentiate between them in some cases.

This could indicate:

The features extracted are not completely discriminative.

There may be some noise in the data.

The dataset could benefit from better feature extraction (e.g., deeper models, additional preprocessing, or fine-tuning).

**2. Are healthy and sick cells clearly distinguishable in the UMAP plot?**

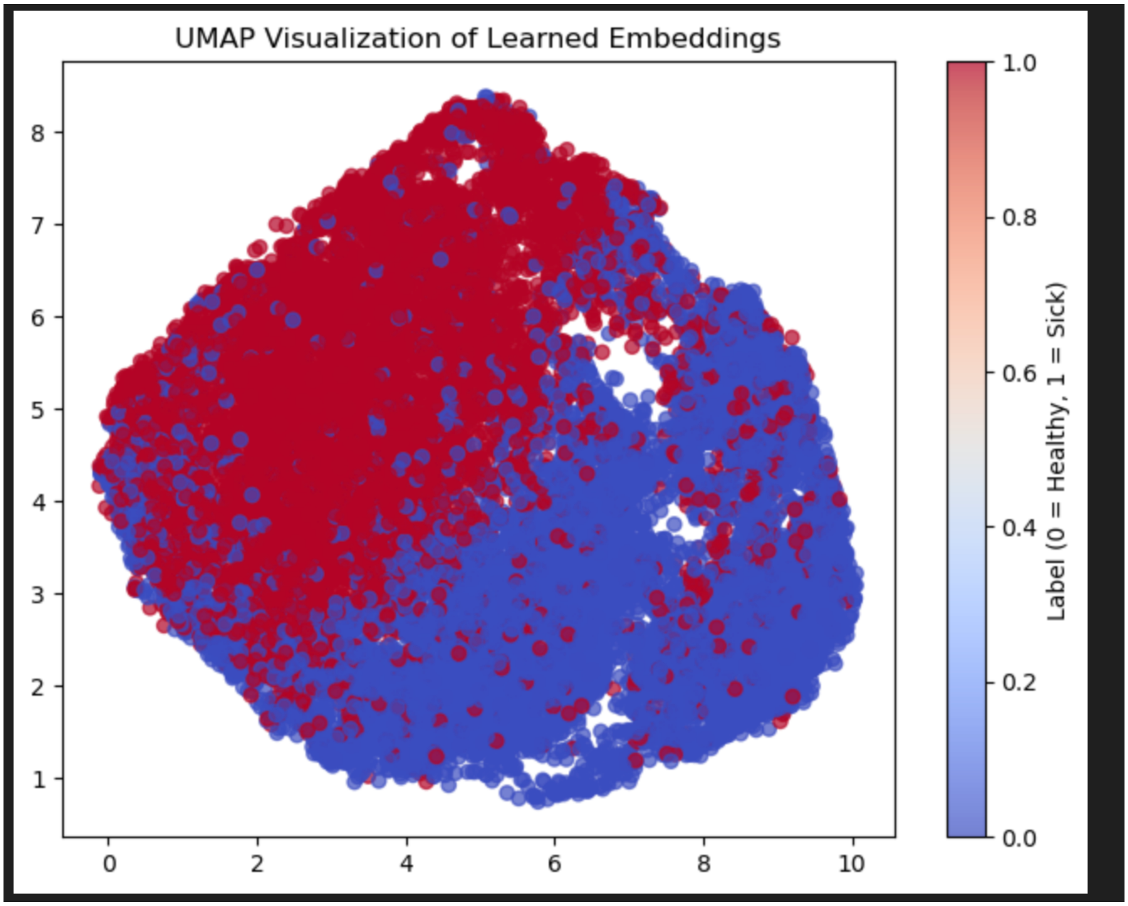
Partially, but not perfectly.

Some clusters of red (sick) and blue (healthy) are visible, but there's a lot of overlapping regions, meaning the model finds it challenging to separate them entirely.

This could mean:

The features extracted are not fully optimal for classification.

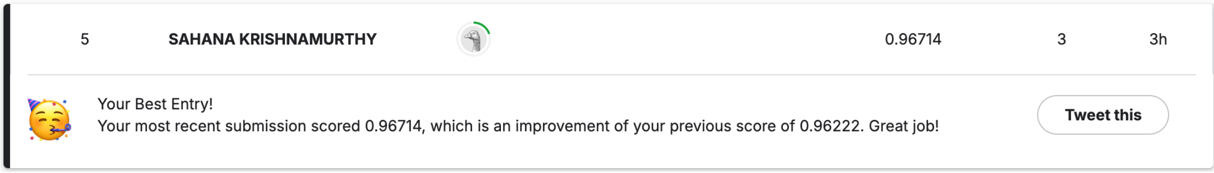
More complex architectures, better augmentation, or fine-tuning might improve the separability.



## 7. Conclusion

This project demonstrates the potential of CNNs for automated malaria detection. By leveraging pre-trained models and transfer learning, I can develop accurate and efficient diagnostic tools that can assist healthcare professionals in the fight against malaria.

## 8. Kaggle Score



|  |  |  |  |  |
|---|---|---|---|---|
| 5 | SAHANA KRISHNAMURTHY | 0.96714 | 3 | 3h |

Your Best Entry!

Your most recent submission scored 0.96714, which is an improvement of your previous score of 0.96222. Great job!