CNNs against Malaria

Abstract

Malaria, a life-threatening disease caused by protozoan parasites of the genus Plasmodium, remains a major public health challenge, particularly in developing countries. Traditional diagnostic methods, such as staining thin and thick blood smears and using rapid test kits like OptiMAL and ICT, require skilled labor, specialized equipment, and can be costly and time-consuming. These limitations hinder large-scale malaria screening and early diagnosis, making automated detection methods a crucial area of research.

With advancements in machine learning and computer vision, deep learning models—particularly convolutional neural networks (CNNs)—have shown promising results in automating malaria detection from blood smear images. These models can effectively extract features from cell images and classify them as parasitized or uninfected with high accuracy.

This project aims to implement a CNN-based machine learning model to classify blood smear images of single cells as either parasitized (infected with Plasmodium) or uninfected. The provided dataset consists of labeled microscopic images, where each image contains a single cell annotated as 0 (uninfected) or 1 (parasitized). The model can be built from scratch or fine-tuned using a pre-trained deep learning model.

Additionally, the project will explore the learned representations within the CNN model. By extracting and visualizing learned embeddings from convolutional filters or multi-layer perceptron (MLP) layers, we aim to identify meaningful patterns in feature extraction. Further analysis using dimensionality reduction techniques such as t-SNE or UMAP will help assess the separability of healthy and parasitized cells in the learned feature space.

Project Deliverables

1. **Model Development:** Implementation of a CNN-based classifier for malaria detection.
2. **Prediction Submission:** Use the trained model to generate predictions on the test set and submit results to Kaggle.
3. **Feature Visualization:** Analyze learned embeddings and visualize them using t-SNE/UMAP to determine distinguishability of infected and healthy cells.
4. **Final Report:** A detailed report summarizing the background, methodology, results, and discussion of findings.

By leveraging deep learning techniques, this project aims to contribute to the automation of malaria diagnosis, potentially improving screening efficiency and aiding early detection efforts in resource-limited settings.

CNNs against Malaria

Methodology

Data Preprocessing

The dataset used for this study consisted of microscopic images of blood samples labeled as either infected with malaria or uninfected. Data preprocessing was performed to ensure consistency, enhance model generalization, and mitigate biases.

1. **Data Loading & Labeling:** The dataset was read from a CSV file containing image filenames and corresponding labels. Image file paths were constructed dynamically for efficient loading.
2. **Train-Validation Split:** The dataset was split into training (80%) and validation (20%) subsets using stratified sampling to maintain label distribution across both sets.
3. **Image Processing:** The images were loaded in RGB format and resized to 128×128 pixels to standardize input dimensions for the neural network.
4. **Data Augmentation:** To improve model generalization, various transformations were applied:
    1. Random affine transformations (rotation up to 30°, shear up to 10°, scaling in the range of 0.8 to 1.2).
    2. Color jittering (brightness, contrast, saturation, and hue variations).
    3. Normalization with a mean of [0.5, 0.5, 0.5] and a standard deviation of [0.5, 0.5, 0.5] to scale pixel values.

Model Architecture

A convolutional neural network (CNN) was developed for binary classification of malaria-infected and uninfected cells. The architecture consisted of:

- An initial convolutional layer (7x7 kernel, stride 2, 64 filters) followed by batch normalization and ReLU activation.
- Four convolutional blocks with increasing depth (64, 128, 256, and 512 channels), each containing convolutional layers, batch normalization, and ReLU activation.
- A global average pooling layer for feature extraction.
- A fully connected layer with a softmax output for binary classification.

The model was optimized using the Adam optimizer with a learning rate of 0.001 and trained for 10 epochs. The loss function used was cross-entropy loss.

CNNs against Malaria

Training Process

The model was trained using the PyTorch framework on a GPU, with a batch size of 32. The training loop included:

- Forward propagation to obtain predictions.
- Backpropagation and optimization using the Adam optimizer.
- Validation at the end of each epoch.

Weights and biases (WandB) were used to log and visualize model performance metrics.

Testing and Evaluation

The trained model was evaluated on a separate test set. Predictions were made using the trained model, and results were saved as a CSV file for submission.

Additionally, t-SNE (t-Distributed Stochastic Neighbor Embedding) was used for visualization of learned feature embeddings.

Results

Training and Validation Performance

- The model achieved a training accuracy of **96.17%** and a validation accuracy of **96.44%** at the final epoch.
- Training loss decreased from **0.2892** in epoch 1 to **0.1087** in epoch 10.
- Validation loss fluctuated but showed a downward trend, reaching **0.1060** in the final epoch.
- The best validation accuracy observed was **96.44%**.

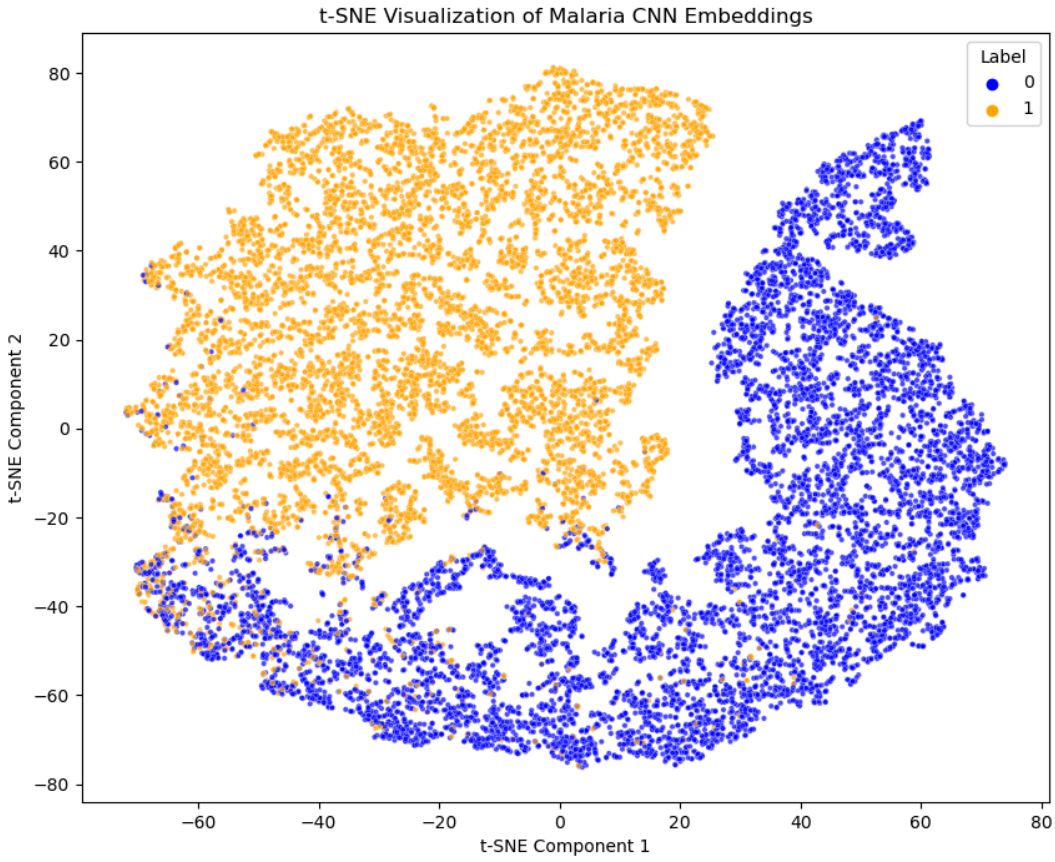
Test Performance

- The model achieved a Kaggle competition score of **0.95964**, indicating strong generalization to unseen test data.

t-SNE Visualization

- The t-SNE plot of the learned embeddings showed clear separation between the two classes, demonstrating the model's ability to extract discriminative features for malaria classification.

CNNs against Malaria



Conclusion

The CNN-based malaria classification model successfully learned to distinguish between infected and uninfected cell images, achieving high accuracy on both validation and test sets. The model's feature representations, as visualized using t-SNE, further validated its effectiveness. Future improvements could involve fine-tuning the architecture, experimenting with different augmentation techniques, and using additional ensemble methods for further performance gains.