# Malaria Detection Using Deep Learning

## Project Overview

This project aims to develop a Convolutional Neural Network (CNN) model capable of classifying malaria cell images as "Parasitized" or "Uninfected". The main goal is to assist in the early detection and treatment of malaria by leveraging automated image classification techniques. By analyzing cell images through a CNN, this project seeks to provide a tool for rapid and accurate malaria diagnosis.

## Features

- **Deep Learning Architecture**: Uses Keras Sequential API to build a CNN with convolutional, pooling, dropout, and batch normalization layers.
- **Pre-trained Models**: Implements VGG16, a pre-trained deep learning model fine-tuned for malaria cell image classification.
- **Data Augmentation**: Applies transformations like rotation, zooming, and flipping to increase the diversity of the training set and improve model generalization.
- **Model Evaluation**: Performance is evaluated using accuracy, precision, recall, F1-score, and confusion matrix.
- **Machine Learning Techniques**: Classical machine learning models like Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) are also tested for comparison.

**Installation Requirements**

To run this project, the following libraries are required:

pip install tensorflow keras matplotlib numpy scikit-learn

**Dataset Structure**

The dataset is structured as follows:

```
./Malaria Cells/
  /train/
    /Parasitized/
    /Uninfected/
  /test/
    /Parasitized/
    /Uninfected/
./Test_Images/
  /Uninfected.png
  /Parasitized.png
```

**Dataset Details**

The dataset consists of images of red blood cells that are labeled as either **Parasitized** (infected with malaria) or **Uninfected**. These images are used to train and evaluate the model.

**Progress Report**

**1. Initial Setup (Date: 19-12-2024)**

- **Completed Tasks**:
    - Imported necessary libraries (TensorFlow, Keras, Matplotlib).
    - Designed the CNN architecture with multiple convolutional layers, max-pooling, dropout, and fully connected output layers.
    - Defined model parameters such as learning rate, optimizer, and loss function.
- **Challenges**:
    - Balancing model complexity with computational cost for efficient performance.
- **Next Steps**:
    - Implement data preprocessing and augmentation.

**2. Data Preprocessing (Date: 20-12-2024)**

- **Completed Tasks**:
    - Applied extensive data augmentation techniques like rotation, zooming, flipping, etc., to increase the training data diversity.
    - Preprocessed test data by rescaling images.
    - Addressed class imbalance by assigning different class weights during training.
- **Challenges**:
    - Ensuring that augmented images retain meaningful features for effective classification.
- **Next Steps**:
    - Implement training strategies with callbacks for efficient training.

**3. Model Training (Date: 22-12-2024)**

- **Completed Tasks**:
    - Trained the CNN model for 20 epochs.
    - Implemented callbacks for early stopping and learning rate reduction to prevent overfitting and enhance training efficiency.
- **Challenges**:
    - Debugging callback functionality to ensure smooth training progress.
- **Next Steps**:
    - Evaluate the trained model's performance on the test dataset.

**4. Model Evaluation (Date: 23-12-2024)**

- **Completed Tasks**:
    - Evaluated model performance on the test dataset, achieving a test accuracy of 95.8%.
    - Plotted training and validation loss curves to assess model convergence and detect potential overfitting.
- **Challenges**:
    - Limited insight into performance metrics without precision, recall, and F1-score.
- **Next Steps**:
    - Perform comprehensive evaluation using confusion matrix and classification report.

**CNN Model Architecture**

The architecture of the CNN model consists of:

- **Convolutional Layers**: These layers apply filters to the input images to extract features such as edges, textures, and patterns.
- **Max Pooling**: Pooling layers are used to reduce the spatial dimensions of the feature maps, leading to more abstract representations and reduced computational cost.
- **Dropout**: Dropout layers are employed to prevent overfitting by randomly setting a fraction of input units to zero during training.
- **Batch Normalization**: This layer normalizes the activations of the neurons to speed up training and stabilize the learning process.
- **Fully Connected Layers**: After flattening the feature maps, the fully connected layers make final predictions using a sigmoid activation for binary classification.

**Results Summary**

**CNN Model Performance**

- **Training Accuracy**: 94.16%
- **Validation Accuracy**: 95.88%
- **Test Accuracy**: 95.93%

**VGG16 Performance**

- **Training Accuracy**: 87.99%
- **Validation Accuracy**: 91.30%

**Classical Machine Learning Models**

- **Support Vector Machine (SVM)**: Accuracy of 49.07%
- **K-Nearest Neighbors (KNN)**: Accuracy of 50.26%
- **Logistic Regression**: Accuracy of 50.68%

**Evaluation Metrics**

The model was evaluated using various metrics:

- **Accuracy**: Measures the overall correctness of the model.
- **Precision**: The percentage of correct positive predictions.
- **Recall**: The percentage of actual positive instances correctly identified by the model.
- **F1-Score**: A balance between precision and recall.
- **Confusion Matrix**: Provides a deeper understanding of the model's classification performance, showing the number of true positives, false positives, true negatives, and false negatives.

**Conclusion**

The malaria detection model using deep learning has shown promising results with high accuracy on both the training and test datasets. The CNN model, particularly with the use of pre-trained VGG16, achieved excellent performance in classifying malaria cell images. This model has the potential to assist in real-time malaria detection, which can be crucial for early diagnosis and treatment.

**Future Work**

- Fine-tuning the model further by experimenting with additional pre-trained models.
- Exploring deployment options for real-time malaria detection in healthcare settings.

---