Project Report: Malaria Infected Cell Classification

1. Introduction

Malaria remains a significant global health challenge, particularly in regions with limited access to healthcare resources. One promising approach to addressing this challenge is the use of deep learning models for the automatic classification of infected cells from microscopic images. The aim of the project is to develop a **deep learning model capable of accurately identifying and classifying infected cells from images of blood smears**. This would provide a reliable and efficient tool for the early detection and diagnosis of malaria, which can significantly improve patient outcomes and aid in disease surveillance efforts.

The basic approach involves training a **convolutional neural network (CNN)** on a dataset of labeled images of infected and uninfected cells. The trained model will then be able to classify new images into one of these two classes. The objective of the project is to leveraging the power of deep learning to achieve high accuracy and robust performance, even when dealing with noisy and heterogeneous datasets.

2. Problem Definition and Algorithm

2.1 Task Definition

The problem we are addressing is the automated classification of malaria-infected cells from microscopic images of blood smears. Given an input image of a blood smear, the task is to determine whether the cells in the image are infected with the malaria parasite or not.

This problem is of utmost importance in the field of medical diagnostics, as early and accurate detection of malaria can significantly improve patient outcomes and aid in disease management and control efforts.

2.2 Algorithm Definition

Our approach involves training a **convolutional neural network (CNN)** for image classification. The CNN architecture consists of multiple convolutional and pooling layers followed by fully connected layers. We use rectified linear unit (ReLU) activation functions and dropout regularization to prevent overfitting.

Pseudocode for our CNN-based classification algorithm:

1. Load the dataset of labeled images (infected and uninfected cells).
2. Preprocess the images (resize, normalize pixel values, etc.).
3. Split the dataset into training and testing sets.
4. Define the CNN architecture:
    - Convolutional layers with ReLU activation
    - Max pooling layers
    - Fully connected layers with dropout regularization
    - Output layer with softmax activation
5. Compile the model with appropriate loss function and optimizer.

6. Train the model on the training data.
7. Evaluate the model on the testing data.
8. Save the trained model for future use

3. Dataset Description:

The dataset contains two classes: Parasitized and Uninfected. Each class consists of images of cells captured under a microscope. The images are stored in two separate folders, with each folder containing images belonging to one of the classes. The dataset was collected from Kaggle.

4. Data Preprocessing:

Before building the model, we preprocessed the data in the following steps:

- **Read images from the folders**: We used Python libraries such as **OpenCV** and **PIL** to read images from the Parasitized and Uninfected folders.
- **Resize images**: All images were resized to a common size of 128x128 pixels to ensure uniformity.
- **Normalize pixel values**: The pixel values of the images were normalized to the range [0, 1] by dividing by 255.0.

5. Model Architecture:

We chose a Convolutional Neural Network (CNN) architecture for this image classification task due to its effectiveness in learning spatial hierarchies of features. The model architecture consists of:

- Convolutional layers: **Four convolutional layers with ReLU activation** followed by max-pooling layers to extract features from the images.
- Flatten layer: Flattens the output from the convolutional layers to prepare it for input to the fully connected layers.
- Fully connected layers: Two dense layers with ReLU activation followed by a final dense layer with **softmax activation** for classification.

6. Training and Evaluation:

We split the dataset into training and testing sets using an 80-20 split. The model was trained using the training set and evaluated on the testing set. During training, we used binary cross-entropy loss and Adam optimizer. The model was trained for 10 epochs with a batch size of 32.

7. Results:

After training the model, we evaluated its performance on the testing set. The model achieved an accuracy of **95.45%** on the testing set, indicating its effectiveness in classifying cell images as infected or uninfected. A web application was created using **Streamlit** and deployed it using **Render.com**.

8. Hotspot Analysis:

To gain insights into the regions of interest (hotspots) that influence the model's predictions, we implemented a method to overlay hotspots on the images in the app.py file. This involved generating heatmaps from the model's predictions and overlaying them on the original images. The hotspots highlighted areas of the images that contributed the most to the model's decision.

9. Conclusion:

In conclusion, we successfully built a deep learning model to classify cell images as infected or uninfected with high accuracy. The model's performance was evaluated on a separate testing set, and insights into the regions of interest influencing the predictions were provided through hotspot analysis and finally a web application was created and deployed showcasing the working of the model. This approach demonstrates the potential of deep learning models in medical image analysis tasks and opens avenues for further research and development in this field.

10. Future Work:

In future work, we could explore additional techniques for improving model performance, such as data augmentation, transfer learning, or fine-tuning the model architecture. Additionally, conducting further analysis on the hotspots identified by the model could provide valuable insights into the characteristics of infected cells and contribute to the development of more effective diagnostic tools.