



PRESIDENCY UNIVERSITY

Private University Estd. in Karnataka State by Act No. 41 of 2013

BANGALORE



Name : Hemanth Akula

Class : 8DataScience -01

Roll No: 20201CSD0031

Introduction

Our project focuses on developing a website equipped with machine-learning capabilities to aid in malaria diagnosis. By utilizing convolutional neural networks (CNNs), a powerful type of machine learning model for image analysis, we aim to create a platform where we can upload microscope images of blood cells to determine if they are infected with malaria or not.

Data collection

We downloaded a dataset from Kaggle that includes images of malaria-infected cells (parasitized cells) and healthy cells (uninfected cells). This dataset helps us study and differentiate between these two types of cells. By analyzing these images, we aim to develop a method using computer algorithms that can automatically identify whether a cell is infected with malaria or not. This project is important because it can potentially assist in diagnosing malaria more quickly and accurately, which is crucial for effective treatment and control of the disease.

Data preprocessing

Data preprocessing for our malaria cell dataset involves several steps to make sure the data is clean and ready for analysis. First, we check the images to ensure they're clear and consistent. Then, we resize them if needed to make them uniform in size for our analysis. We also adjust the brightness or contrast to improve the image quality. After that, we label each image correctly as either parasitized (infected) or uninfected (healthy). Finally, we organize all the images into a format that our computer algorithms can easily understand and use for training. This preprocessing step is crucial because it helps our models learn effectively from the data and make accurate predictions about whether a cell is infected with malaria or not.

Splitting and Balancing Dataset:

Splitting the Data: We divided our dataset into two parts: one for training the model and another for testing its performance.

Training Data: About 89% of our dataset was used for training the model. This allowed the model to learn from a large variety of examples.

Testing Data: The remaining 11% of the dataset was kept aside for testing. This allowed us to see how well the model performs on new, unseen data.

Balancing the Data: Ensuring fair representation of both malaria-infected and healthy cells in the training data is crucial for the model to learn effectively.

we checked the distribution of infected and healthy cell images in our dataset.

If one type of cell was more common than the other, we adjusted the number of samples so that both types were equally represented in the training data.

This balanced approach helps the model learn to classify both types of cells accurately, avoiding bias towards the more prevalent class.

To train and test our model effectively, we took the following steps with our dataset:

Building and training the model

We created a computer model that learns to classify images of malaria-infected and healthy cells using a method called convolutional neural networks (CNNs). Here's how the model was constructed:

Layers: The model has several layers that process images step by step. Each layer extracts different features from the images.

Training Data: We used a technique called data augmentation. This involved creating new training images by slightly modifying the original ones. This helps the model become better at recognizing patterns in different versions of the same image.

Normalization: Before training, we scaled the pixel values of the images to a range between 0 and 1. This step ensures that the model learns more effectively.

Validation: During training, we set aside 20% of our augmented data to validate how well the model is learning. This helps us adjust the model to improve its performance.

We created a computer model that learns to classify images of malaria-infected and healthy cells using a method called convolutional neural networks (CNNs). Here's how the model was constructed:

Layers: The model has several layers that process images step by step. Each layer extracts different features from the images.

Training Data: We used a technique called data augmentation. This involved creating new training images by slightly modifying the original ones. This helps the model become better at recognizing patterns in different versions of the same image.

Normalization: Before training, we scaled the pixel values of the images to a range between 0 and 1. This step ensures that the model learns more effectively.

Validation: During training, we set aside 20% of our augmented data to validate how well the model is learning. This helps us adjust the model to improve its performance.

Evaluation of the Model:

Accuracy Plot:

The Training Accuracy Plot shows how well our model performed on the training data over each iteration.

Loss:

A loss value of 0.205 indicates that the model makes predictions with a high level of accuracy. It's performing well in minimizing errors between predicted and actual results.

Accuracy:

An accuracy of 0.924 means that the model correctly identifies the type of cell (malaria-infected or healthy) 92.4% of the time. This demonstrates the model's ability to effectively classify images based on its training.

If both curves are close and high, it suggests our model learned well from the training data and can generalize to new data effectively.

Classification Report:

The Classification Report summarizes how well our model classified the test data:

Precision: Indicates the accuracy of the model's positive predictions. For example, when the model predicts a cell is parasitized (infected), it is correct 97% of the time.

Recall: Measures how well the model correctly identifies parasitized or uninfected cells. For instance, the model identifies 92% of all parasitized cells and 97% of all uninfected cells.

F1-Score: Harmonic mean of precision and recall. It provides a balance between the two metrics.

Support: Number of occurrences of each class in the test set.

Deployment:

Streamlit Integration: Streamlit is a tool used to build interactive web applications for machine learning models.

Application Purpose: Our application aims to classify images to detect diseases, specifically malaria-infected cells, using a deep learning model.

How it Works: When we upload an image (in JPG or PNG format) through the application interface.

Prediction Process: The uploaded image is processed by our pre-trained TensorFlow model to predict if the cell is infected (Parasitized) or healthy (Uninfected).

Output: The application displays the predicted disease class and provides an estimate of how confident the model is in its prediction.

conclusion:

In this project, we developed a user-friendly website that uses advanced machine learning to diagnose malaria from microscope images of blood cells. By training a convolutional neural network (CNN) with a dataset of infected and healthy cells, our website can accurately identify malaria-infected cells