

Overview of Malaria Disease detecting model

Objective:

Aim of this project is to distinguish the blood sample cell image of a person between parasitized and uninfected.

Tools and Techniques:

- Library used: NumPy, TensorFlow, Matplotlib, Glob , streamlit, vgg19
- Tools used: Google colab, ngrok

Method Summary:

- Downloading the data: - Data is originally downloaded from Kaggle.
- Importing the necessary libraries: - I have used deep learning techniques CNN to classify the image. Library used is TensorFlow and vgg19.
- Architecture of the model: - I build the framework of my model using vgg19. An inbuilt model under TensorFlow that consists of 19 layers, supporting image size of 224*224. I have not manually added the convolutional layer for the model instead I chose vgg19 because of high accuracy. I dropped the first and the last layer of the vgg19 model as, I used my input as the first layer and the number of classes as the neurons in the last layer, which is parasitized and uninfected in my case respectively. I decided not to train the 16 layers as it has already been trained under the default framework of vgg19. Finally, I used the Flatten library to flatten the outputs and the output is being produced using SoftMax function.
- Optimizer and Loss function: - I have used Adam optimizer to optimize the accuracy by reducing the loss. Adam optimizer uses momentum and dynamic learning rate to optimize the weights. Accuracy and convergence speed is higher relative to another optimizer. So, I used this technique. The loss function that I used is categorical cross entropy
- Data Augmentation: - To introduce variability and to increase the size of the dataset I have used this method. Operations used are flipping, shearing, zooming and scaling. For this, library I used is ImageDataGenerator. It also helps in dealing with the Overfitting issue.
- Splitting of Dataset: - Dataset is divided into train and test set. 80% of the data were used for training purpose and rest for the validation purpose.

- Fitting the model: - Model is finally being trained over 5 epochs. Accuracy of the model can be increased to much if I used 50 epochs. Since using 50 epochs were taking more time in hours. so, I decided to use 5 epochs by retaining the accuracy to almost 90%.
- Saved the model in hd5 format.

Deployment of the model using Streamlit:

Written the app.py file under the Google colab. I loaded my model using `tensorflow.keras.models.load_model`. I then uploaded the image `st.file_uploader` supporting png, jpg type . I have then imported PIL to do lots of computation on images. Then we define a function that accept the image and model and then predict the disease. Under this I converted the image into array and reshape its dimension. Finally, I used for if loop to print if any cell images is parasitized or not. Eventually it will create an ap.py file. So, this way I created my web app. Now I host this on the internet using ngrok using authentication token.

Note: - I tried deploying the file but somewhere it throws an error. My model runs perfectly, the only issue I face is in deploying