

Malaria Detection using Cell Images



Big Data Analytics and Applications(COMP_SCI 5542)

Team Name: Team Arrow

1. Kota Venkata Ramyasri
2. Trishanthi Thota
3. Rithik Reddy Boyapally
4. Sai Dattu Punati

Abstract

Malaria is an infectious disease caused by a blood parasite. According to WHO statistics there were 262 million malaria cases detected in 2000. Detecting this disease from the given cell image can help the health care system in the world. In this application, we are using a CNN architecture that gives the same classification accuracy as the VGG-16 and Resnet-50 models for thin blood smear RBC slide images for detection of malaria mean while decreasing the run time and computation power by an order of magnitude.

Key Words:

Deep Learning, Convolution Neural Network, Malaria Detection, Health Application, Malaria Cell Images Dataset

Objectives

- In this project, for the malaria images cell dataset we are building a deep neural network that can detect malaria.
- In the real-time world, this application can be able to save human lives by detecting malaria from a cell image.
- This model takes a cell image as the input and returns the class label(infected or uninfected) as the output.
- We have developed a website that can be accessible to everyone when deployed on the internet.

Scope

- This application can help the health care system to detect the malaria-infected cells from the images of patients.
- This application can help the common public to determine if they are infected with malaria or not using their cell images.
- In future, we are going to add audio and translation to the website which can add inclusivity to the website.

Problem Statement

Malaria being the sixth most prevalent disease in World, had infected around 200 million people in the world and nearly caused 400,000 deaths per every year. The malaria can be detected through the peripheral blood smear (PBS) test. The microscopic images of the cell are used for the detecting if they are infected or not. But, this process requires an expertised human that can detect malaria in the cell. To avoid this our application uses a convolutional neural network that can detect the infection in the cell. This can hugely impact the health system around the world in a positive manner.

Related work

- There are a wide variety of machine learning algorithms that can be used for the images classification problems.
- Among them VGG-16 is one of the famous models that is used in various research papers other than CNN. But VGG-16 is very slow to train the model. It requires GUPs with high computation power.
- VGG-16 is a pre-trained model trained on imagenet weights with 528MB, this could take up a lot of space and time while compiling which make it inefficient when compared to Convolutional Neural network.

Solution

We have designed and developed a website where we can upload a malaria cell image and click on the predict button, which will classify the image if it is infected or not. This can really help one which has to detect if the cell in the image is infected with malaria or not. Our main motive of the application is to bring a highly accessible application where users can directly upload their images in our website and save them by predicting Image Cells that contain Malaria or not.

The Model that we have used is Convolutional Neural Network ,which is generally used in image recognition and processing that is specifically designed to process pixel data.

We have saved the model in .h5 format and loaded that model using Tensorflow into the flask application.

Application:

Flask environment:

We have used a flask environment for creating web applications with a combination of python. Initially, we created a simple flask application and made changes to the project for our website. We have used Visual studio as our editor which can help to organize files in the folder structure. Initially, we created a virtual environment and activated the virtual environment.

After that open the command line interface and install the flask by using pip -> **pip install flask.**

We have two main files in this environment

- app.py
- index.html

app.py file: This is a file that initiates the website. We will start our website by giving **python app.py** command. This file mostly contains the routing information and we load the model in this file.

index.html file: This file mostly contains the information on how our website must look like and the structure of the website.

Dataset:

The dataset that we have used is **Malaria Cells Images Dataset**. The Dataset totally contains 27,558 images

The dataset contains 2 folders

- Infected
- Uninfected.

The Dataset is provided by the official NIH Website (National Institute of Health)

Preprocessing:

➤ **Reading and Resize the images**

For Malaria Detection we have used the Malaria Image Cells Dataset. We have uploaded the .Zip file to google drive and mounted it to google colab. We have used OpenCV, for reading the images from the dataset. We have resized all the images to (45,45) so that all the images are having the same size which can be fed to an Input layer

Libraries And Methods used :

cv2.imread() - For reading the images OpenCV library is used and method is imread

resize() - To resize all the images to the same size so that they can be fed to the input layer of the model.

➤ **Convert images to numpy array**

In this step, we are converting images to numpy array so that we can better perform numerical operations on the data rather than the images data

Libraries And Methods used :

np.array() - We have used the array method from the numpy library to convert the input to an array of images

➤ **Splitting images and Normalize the data**

Here we are splitting the dataset to train and test then further we are splitting the test dataset to evaluate the dataset for determining the accuracy of the model.

Libraries And Methods used :

astype(float32) - For converting to float values

train_test_split - For splitting the dataset to train and test.

➤ **One-hot encoding labels**

Here we are converting the classification labels related to train and test data to integer values. We have used one hot encoding to convert these labels in string format to the numeral data.

Libraries And Methods used :

to_categorical() - We have used this method from tensorflow.keras.utils library for encoding the labels.

Model:

In our Malaria Detection application, We have used a Convolutional neural network model that can be used for image prediction.

Libraries and methods used in the model as follows

compile() - To create learning for the model we have used a compile method. We have used binary cross entropy as the loss function and adam as our optimizer and metric as our accuracy. We have used binary cross entropy as our loss function because we have two labels at the end for prediction.

predict() - To determine the result of the test data we have used the predict method.

fit() - For training the model we have used fit method. We have used model_checkpoint as our call back function to save weights or model at some point in time.

save() - We have saved the model using save method in .h5 format.

Convolution Neural Network(CNN):

We created a sequential CNN model to classify the images. Since the sequential model can be accurate for 80% of the use cases we have trained this model on our dataset.

Additional Methods and Libraries used in CNN

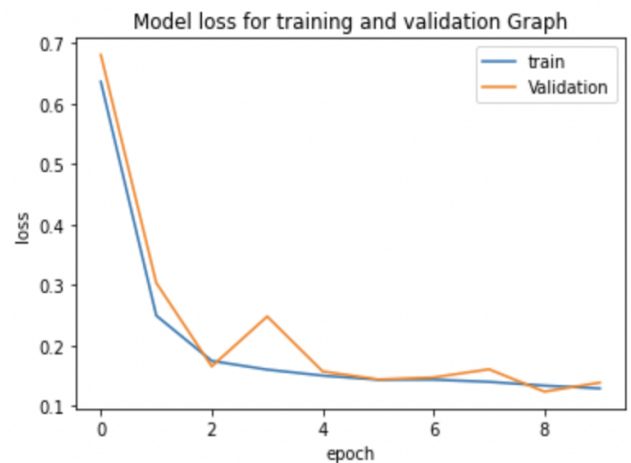
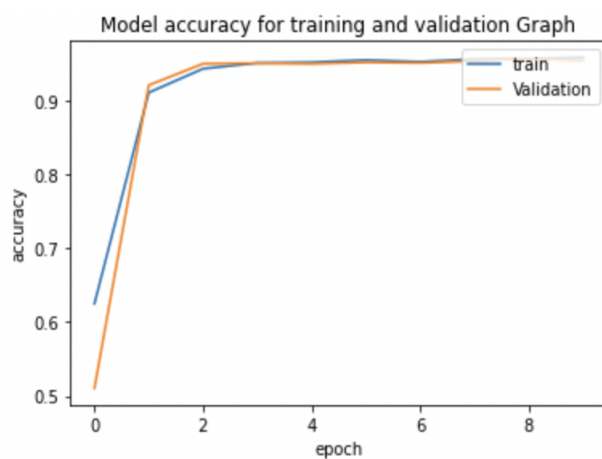
Sequential() - We have used Sequential to stack the layers to a model.

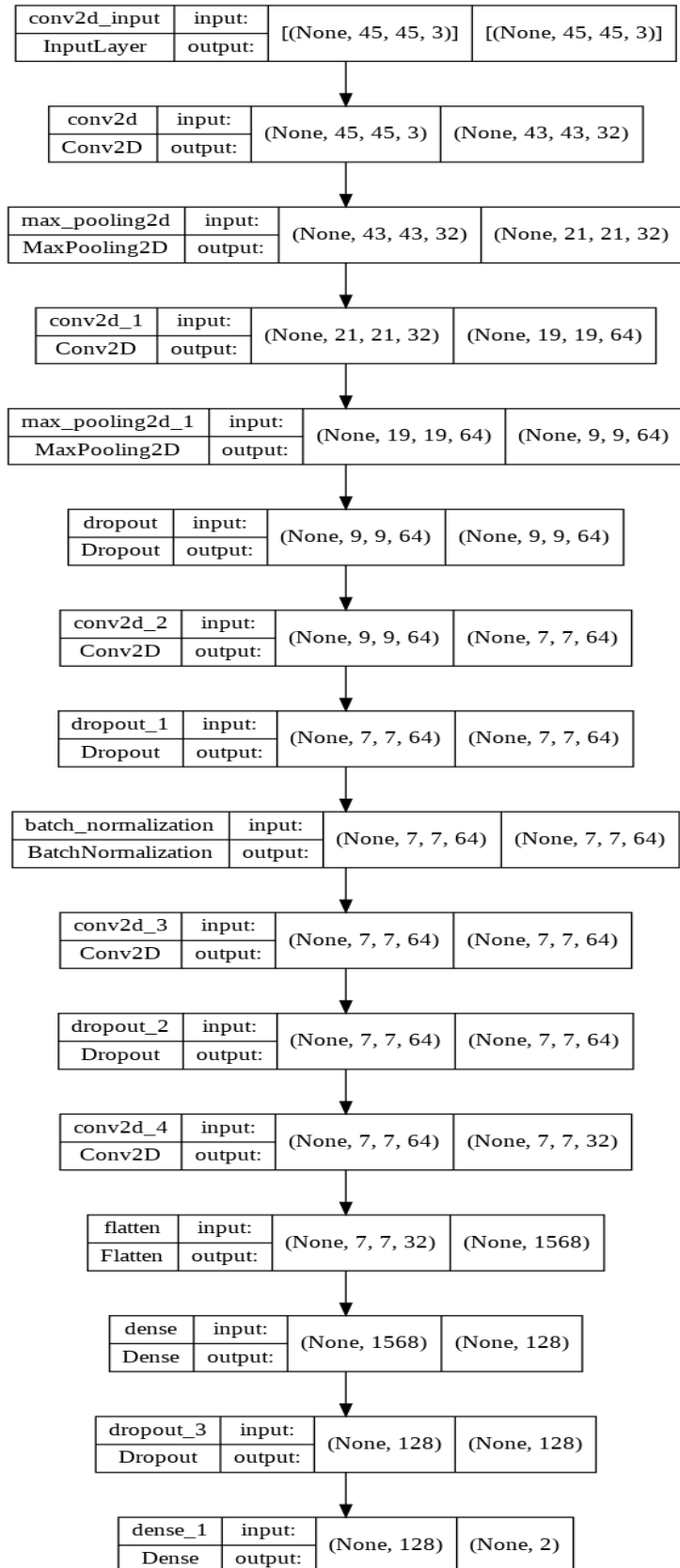
add() - We have used add method to add various layers to the model i.e, Convolutional Neural Network, Batch Normalization, Dropout layer etc.,

predict_classes() - Inorder to determine the class label of the test images we have used predic_classes

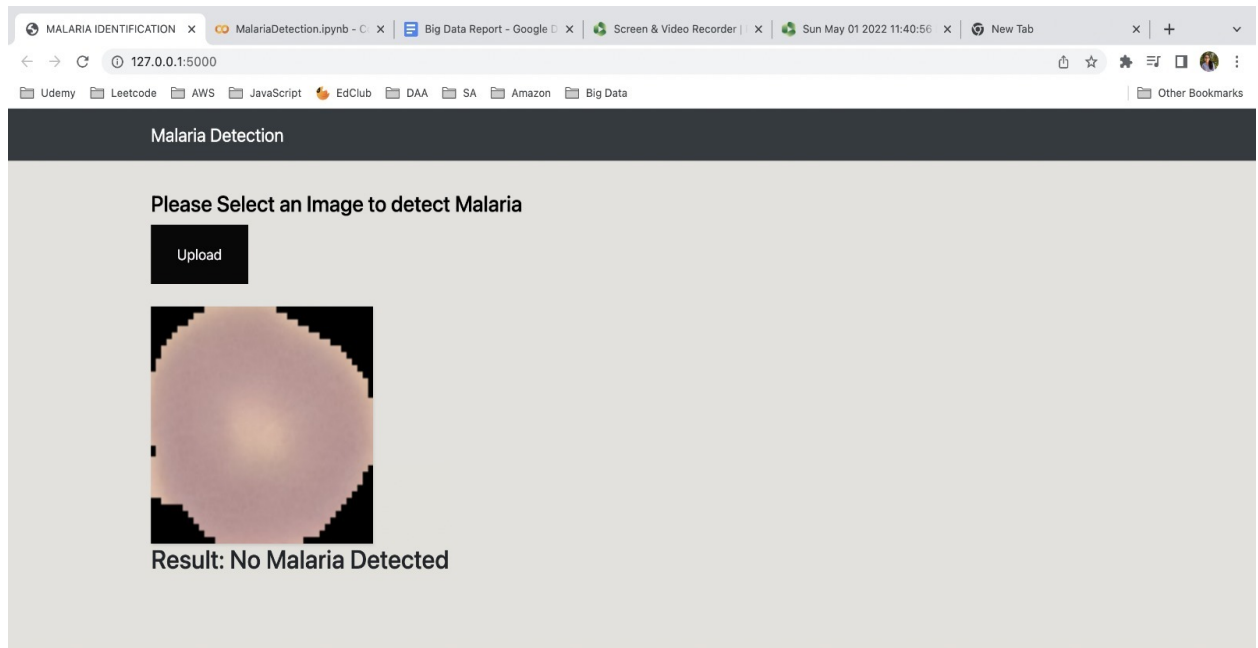
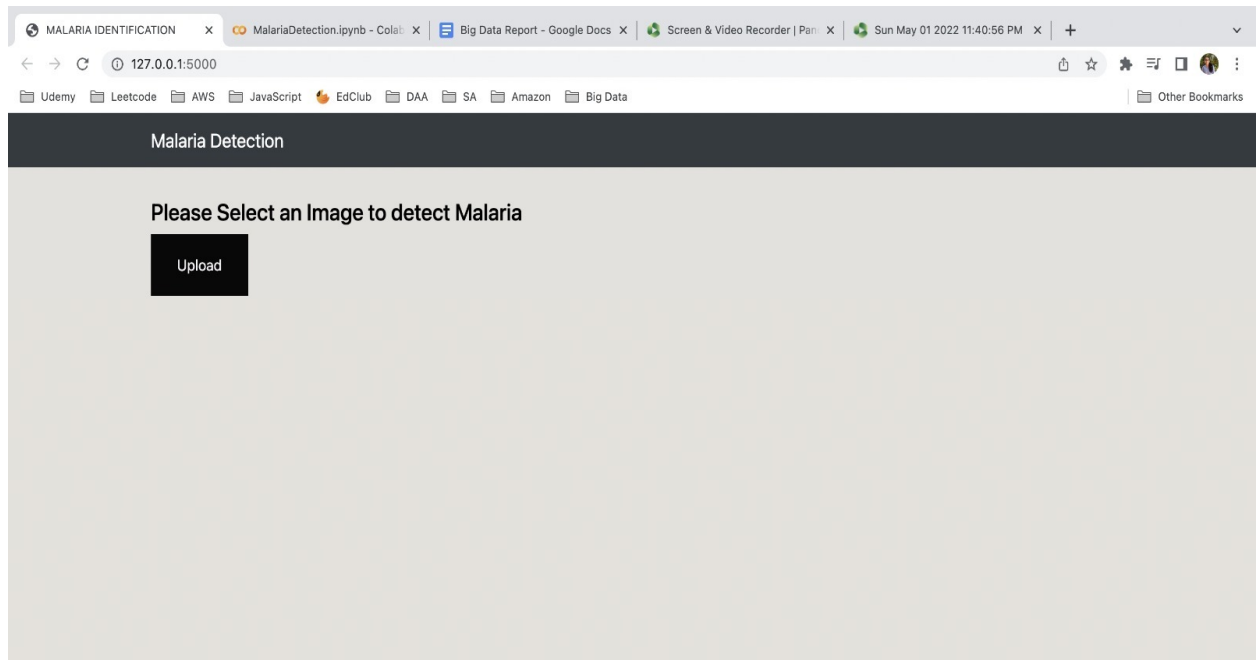
accuracy_score() - To determine the test accuracy of the model

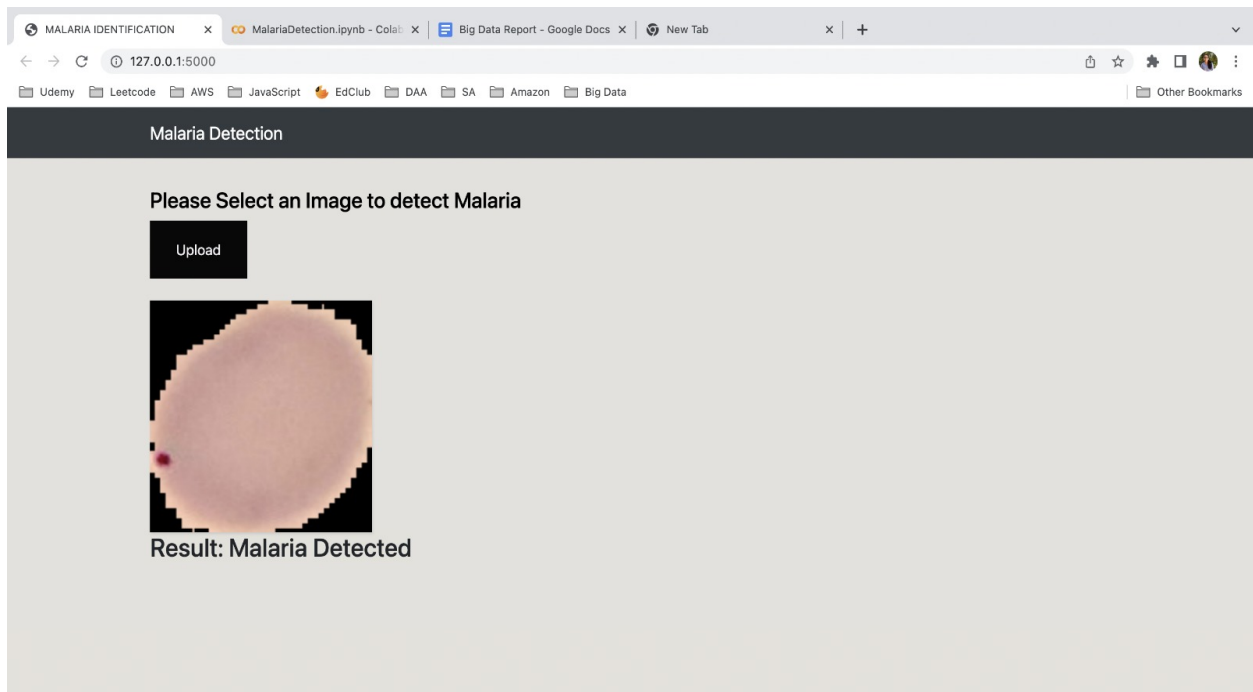
Test Accuracy: 0.9542





Results





Conclusion

The main contribution of this application is to detect if the red blood cell of a person is infected with malaria or not. The convolutional neural network model to determine the result. The test accuracy of the model is 0.9542. This shows that the accuracy is good but can be improved in future for the betterment.

References:

- [1]Abdiel Goni (2021) Malaria Cell Image Classification – An End-to-End Prediction
<https://www.analyticsvidhya.com/blog/2021/12/malaria-cell-image-classification-an-end-to-end-prediction/>
- [2] Angel Molina (2020) A Deep Learning Approach for Segmentation of Red Blood Cell Images and Malaria Detection <https://www.mdpi.com/1099-4300/22/6/657>
- [3]Amogh Manoj Joshi (2020) Deep Learning Based Approach For Malaria Detection in Blood Cell Images <https://ieeexplore.ieee.org/document/9293753>
- [4]Subrata Sarkar(2020) Malaria detection from RBC images using shallow Convolutional Neural Networks <https://arxiv.org/abs/2010.11521>
- [5]MahdiehPoostchi(2018) Image analysis and machine learning for detecting malaria
<https://www.sciencedirect.com/science/article/pii/S193152441730333X>