

IMPLEMENT DEEP LEARNING TECHNIQUES FOR MALARIA DETECTION USING IBM CLOUD

Developed By: T. Soumya, D. Manikanta, B. Sony, G. Vineeth

Smart Bridge – Mini Project Report

1. INTRODUCTION

1.1 Overview:

Malaria is a significant burden on our healthcare system and it is the major cause of death in many developing countries. It is endemic in some parts of the world which means that the disease is regularly found in the region. Therefore, early testing is necessary to detect malaria and save lives. Thus, it gives us the motivation to make malaria diagnosis more effective and faster. A specialized technology proves essential to combat this problem.

For Malaria diagnosis, the RDT and microscopic diagnosis are the most used clinical methods. RDT is an effective and faster tool. Also, it does not require the presence of a trained medical professional. But RDT has few drawbacks like susceptibility to damage by heat and humidity and higher cost compared to a light microscope. The microscopic diagnosis systems do not suffer from these shortcomings, but it requires the presence of a trained microscopist. Machine learning has recently made headlines in the medical field. The machine learning approaches have proved to be successful in the diagnosis of a disease.

1.2 Purpose

The main aim of this malaria detection system is to address the challenges in the existing system by automating the process of malaria detection using Machine learning and image processing

2. LITERATURE SURVEY

2.1 Existing problem

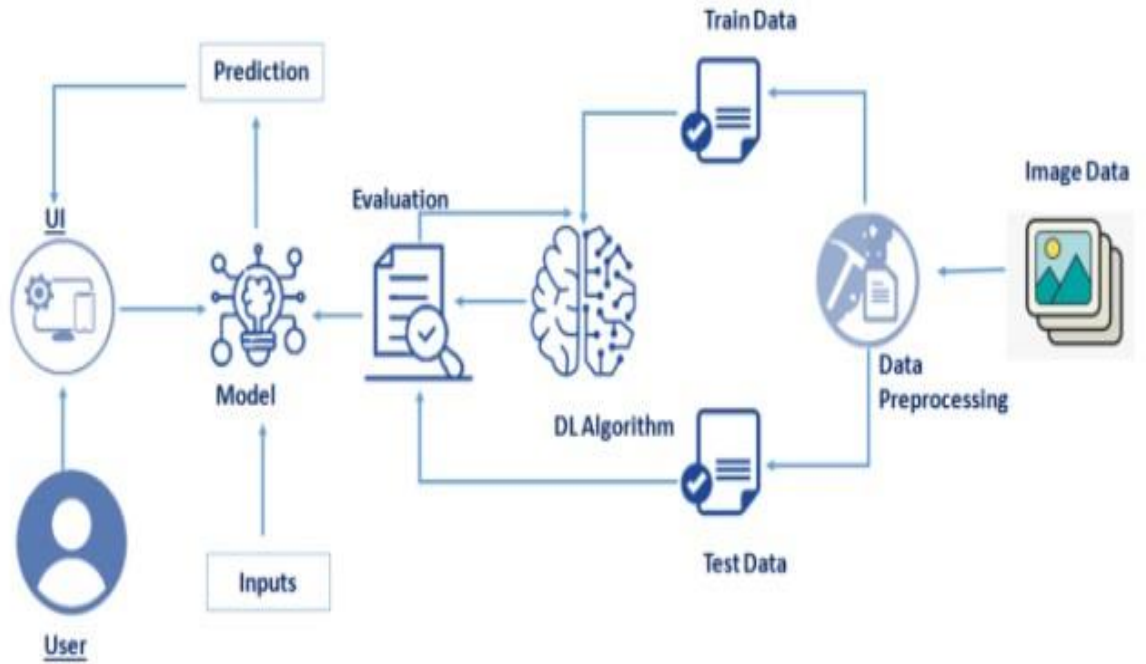
Malaria being a life threatening disease has caused deep research interests among the scientists all over the world. Earlier, malaria was mostly diagnosed in the laboratory setting requiring a great deal of human expertise. Automatic systems such as those relying on machine learning techniques were initially studied to overcome this problem. Techniques reported in this domain of study mostly considered the hand-crafted features in decision making. For example, relied on morphological factors for feature extraction and applied SVM and Principal Component Analysis (PCA) for the classification purpose. However, the accuracy achieved through these kinds of model is low compared to the more recently studied deep learning based techniques.

2.2 Proposed solution

For automatic detection of malaria pathogens from the microscopic images, Convolutional Neural Network (CNN) received much attention from the researchers in recent times. Dong et al. for example evaluated the performances of three popular Convolutional Neural Network , namely LeNet-5 , AlexNet and GoogLeNet . In addition, they trained an SVM classifier for comparison purposes and concluded that CNN is advantageous over SVM in terms of the capacity of learning image features automatically. To find the optimal layer of a pre-trained model to extract features from underlying malaria parasite data, Rajaraman et al. evaluated the performances of AlexNet, VGG-16 , ResNet-50 , Xception , DenseNet-121 along with their custom-built model . Liang et al. reported 97.37% accuracy in detection with the help of their 16-layered CNN model and claimed that their model is superior to transfer learning models . Hung et al. pre-trained a model on Imagenet but fine-tuned it on own data for detecting malaria parasite.

3.THEORITICAL ANALYSIS:

3.1 Block diagram



3.2 HARDWARE AND SOFTWARE REQUIREMENTS IN THE PROJECT:

For running a machine learning model on the system you need a system with minimum of 16 GB RAM in it and you require a good processor for high performance of the model. In the list of **software requirements** you must have:

- Jupyter Notebook for programming, which can be installed by Anaconda IDE.
- Python packages
- A better software for running the html and css files for application building phase e.g. spyder.

4. EXPERIMENTAL INVESTIGATIONS:

In the experiment of CNN-SVM and CNN-KNN, a CNN model was used to extract the features. Later these features were classified through SVM or KNN as per the objective of our experiment. The outline of this architecture is depicted in

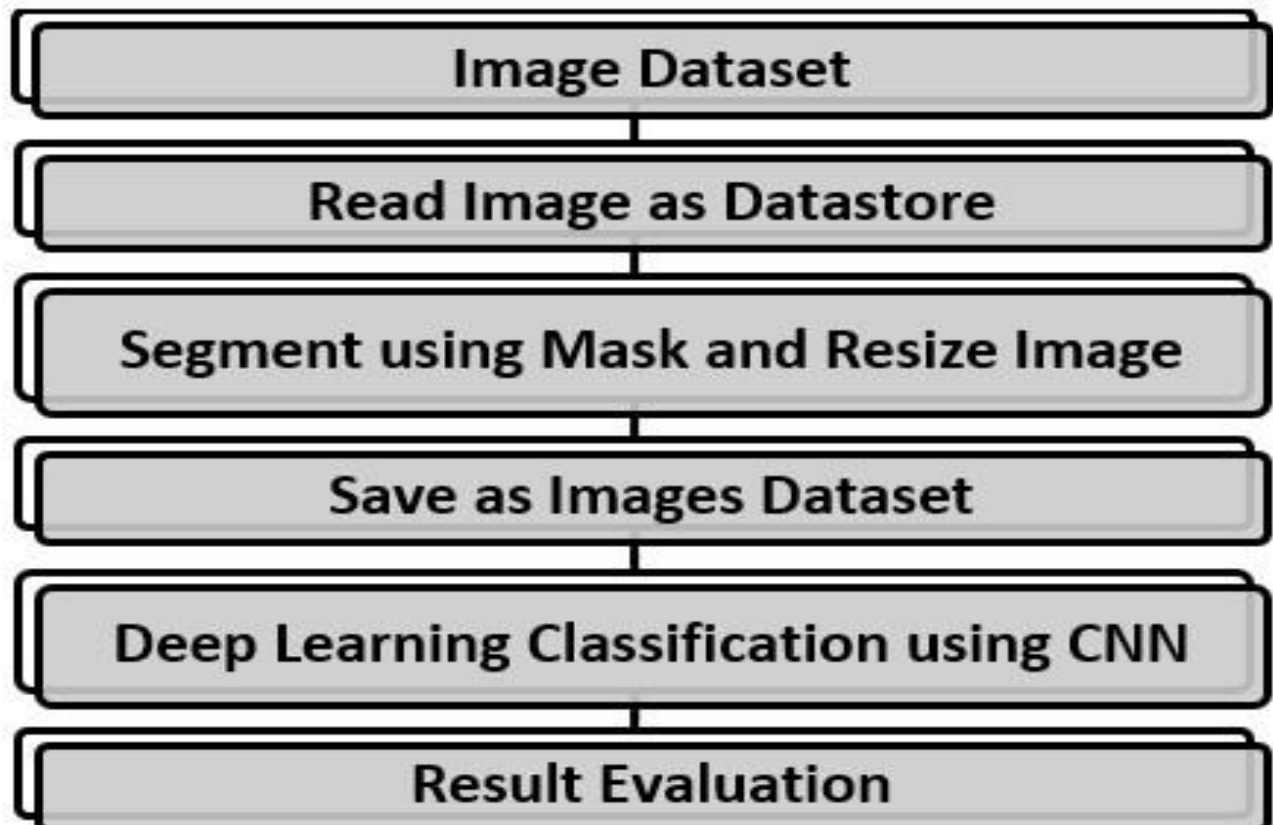
Support vector machine (SVM) and K-Nearest Neighbor (KNN) are the core machine

learning algorithms that were used as a classifier by implicitly mapping their inputs into high-dimensional feature spaces . Earlier work reported the effectiveness of SVM or KNN classifier on the Convolutional neural network's extracted features instead of using the softmax layer. Being motivated with this work, we trained SVM and KNN models after extracting features from a CNN model.

Autoencoder Training Finally, autoencoder based architecture was explored to increase the model's performance. It is found that the autoencoder based experiments outperforms other experiments. Autoencoder is composed of two main components which are Encoder and Decoder discussed in the proposed model subsection. The main aim of an Autoencoder is dimensionality reduction, but to detect malaria parasite, it is used as a classifier. This classifier can classify infected and parasitized images from a microscopic blood smear.

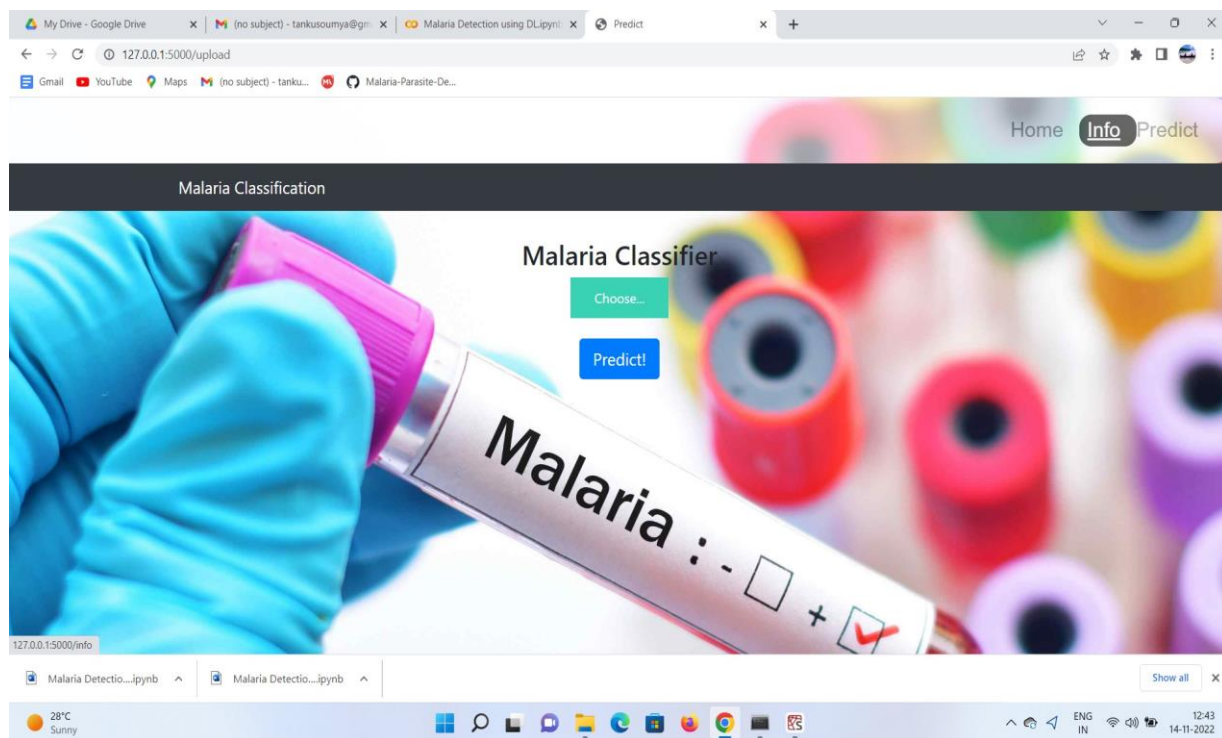
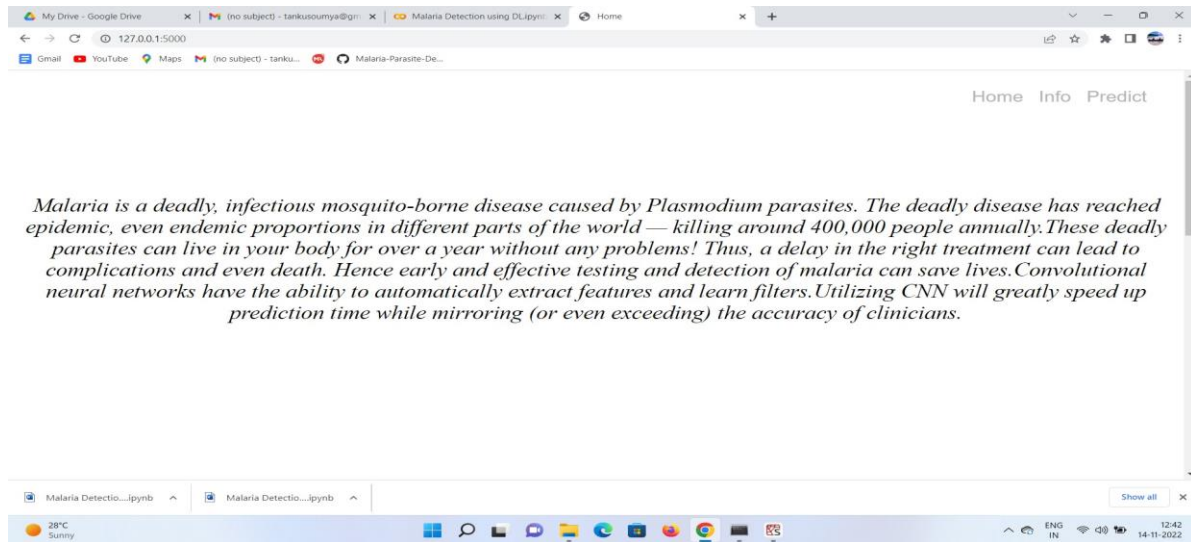
Stochastic Gradient Descent (SGD) was used as an optimizer with initial learning rate of 0.01 and momentum of 0.5 while training the student model. To update the weight, a decay of $1e-6$ was applied using Keras learning rate decay function. To calculate the loss and update the weights and biases of all trainable parameters Cross Entropy Loss function was applied. Combine validation accuracy

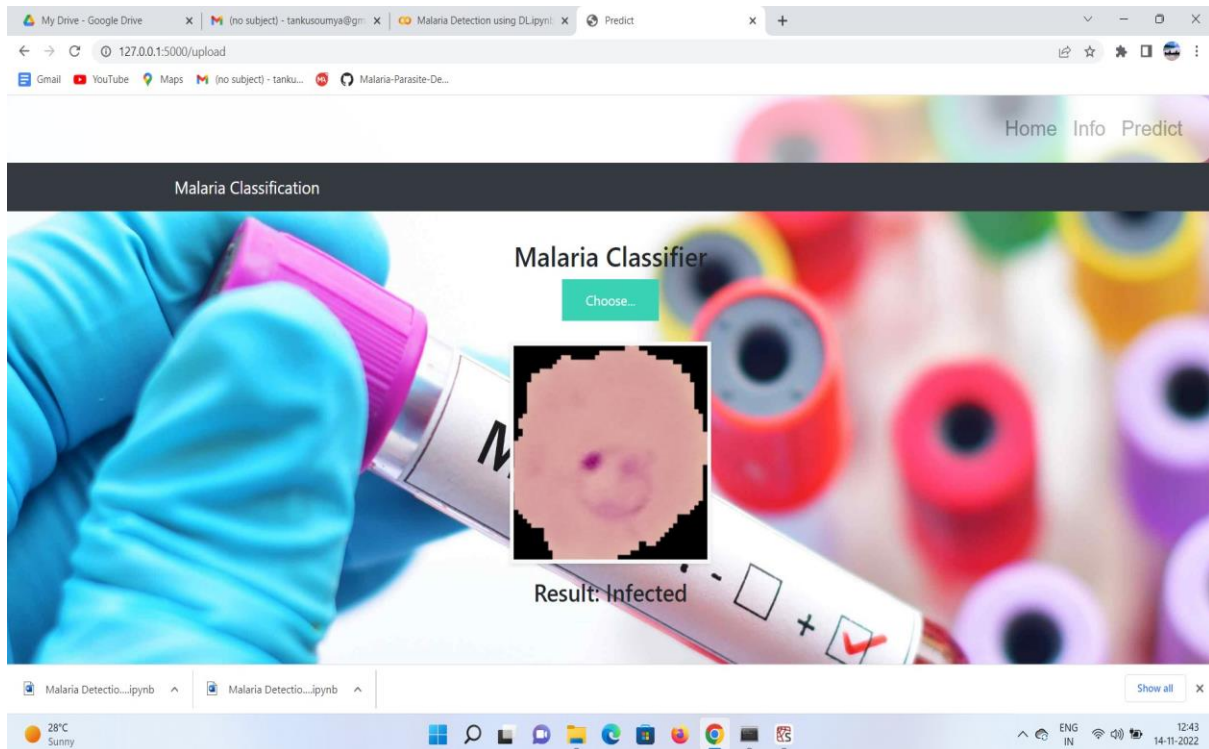
5.FLOW CHART:-



6.RESULTS:-

Final output of the project:





7. ADVANTAGES & DISADVANTAGES

ADVANTAGES

Blood slide microscopy makes it possible to count the number of parasites and is more useful than rapid diagnostic tests for monitoring the effectiveness of malaria treatment.

Magnification and higher resolution – as electrons rather than light waves are used, it can be used to analyze structures which cannot otherwise be seen.

DISADVANTAGES

But RDT has few drawbacks like susceptibility to damage by heat and humidity and higher cost compared to a light microscope. The microscopic diagnosis systems do not suffer from these shortcomings, but it requires the presence of a trained microscopist. Machine learning has recently made headlines in the medical field.

8.APPLICATIONS:-

Transfer learning can be effectively used for carrying out image recognition

tasks. For example, using transfer learning, a model that is pre-trained for identifying dogs can be used to identify cats. Neural networks are trained on large datasets of images to recognise objects.

9.CONCLUSION-

The correct diagnosis of malaria parasite species and correct treatment can reduce the number of malaria-infected individuals who carry the parasites between populations and may thus reduce the risk of re-introducing of malaria into other parts of Iran, outside this province, where an interruption of transmission had been previously achieved.

10.FUTURE SCOPE:

Practical validation of model efficiency was performed by deploying the model in 10 different mobile phones of varying computational capacity and a server-backed web application. Data gathered from these environments show that the model can be used to perform inference under 1 s per sample in both offline (mobile only) and online (web application) mode. These inference speeds coupled with the high classification accuracy lead us to believe that this work can play a part towards building a fully automated system for malaria parasite detection which may be useful in resource-constrained areas in the foreseeable future

11.BIBILOGRAPHY:-

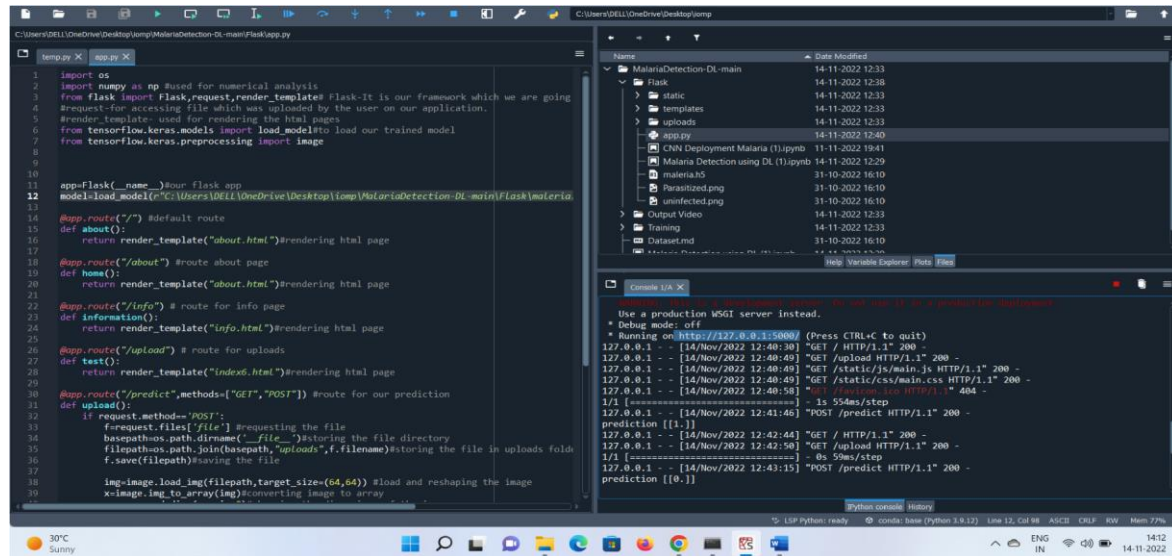
1. Schellenberg, J.A.; Smith, T.; Alonso, P.; Hayes, R. What is clinical malaria? Finding case definitions for field research in highly endemic areas. *Parasitol. Today* 1994, 10, 439–442. [[Google Scholar](#)] [[CrossRef](#)]
2. Tarimo, D.; Minjas, J.; Bygbjerg, I. Malaria diagnosis and treatment under the strategy of the integrated management of childhood illness (imci): Relevance of laboratory support from the rapid immunochromatographic tests of ict malaria pf/pv and optimal. *Ann. Trop. Med. Parasitol.* 2001, 95, 437–444. [[Google Scholar](#)] [[CrossRef](#)]
3. Díaz, G.; González, F.A.; Romero, E. A semi-automatic method for quantification and classification of erythrocytes infected with malaria parasites in microscopic images. *J. Biomed. Inform.* 2009, 42, 296–307. [[Google Scholar](#)] [[CrossRef](#)] [[PubMed](#)][[Green Version](#)]
4. Ohrt, C.; Sutamihardja, M.A.; Tang, D.; Kain, K.C. Impact of microscopy error on estimates of protective efficacy in malaria-prevention trials. *J. Infect. Dis.* 2002, 186, 540–546. [[Google Scholar](#)] [[CrossRef](#)] [[PubMed](#)]
5. Anggraini, D.; Nugroho, A.S.; Pratama, C.; Rozi, I.E.; Iskandar, A.A.; Hartono, R.N. Automated status identification of microscopic images obtained from malaria thin blood smears. In Proceedings of the 2011 International Conference on Electrical Engineering and Informatics, Bandung, Indonesia, 17–19 July 2011; pp. 1–6. [[Google Scholar](#)]

APPENDIX :-

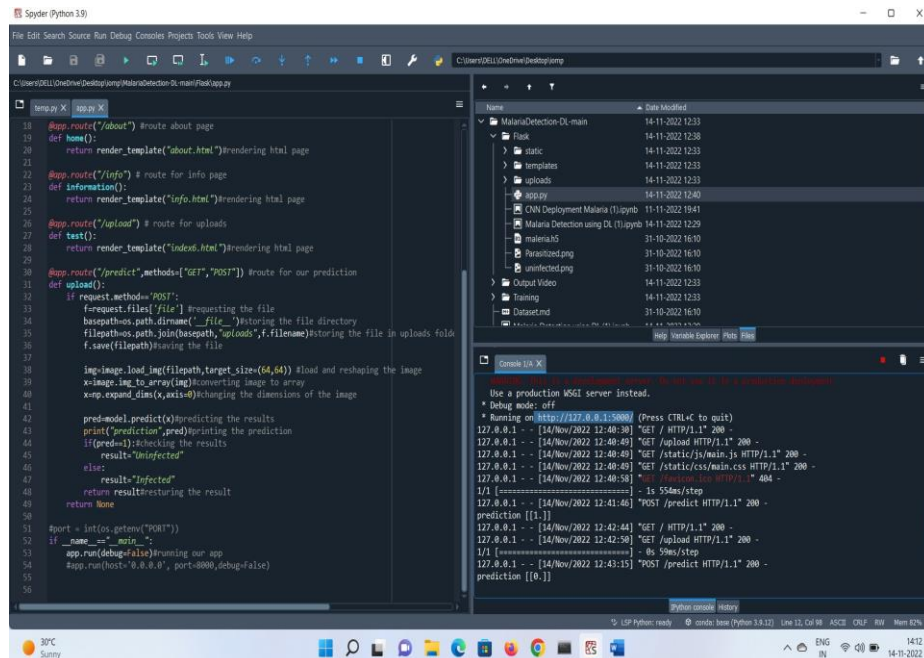
1. Google Collab

<https://drive.google.com/file/d/16rzVDqoB9OXszacIOUbaMjFQW55AKcRi/view?usp=sharing>

2. Flask Code



```
1 import os
2 import numpy as np #used for numerical analysis
3 from flask import Flask, request, render_template #Flask-It is our framework which we are going
4 #request for accessing file which was uploaded by the user on our application.
5 #render_template used for rendering the html pages
6 from tensorflow.keras.models import load_model #to load our trained model
7 from tensorflow.keras.preprocessing import image
8
9
10
11
12 app=Flask(__name__) #our Flask app
13 model=load_model(r"C:\Users\DELL\OneDrive\Desktop\jump\MalariaDetection-DL-main\Flask\app.py")
14
15 @app.route("/") #default route
16 def about():
17     return render_template("about.html")#rendering html page
18
19 @app.route("/about") #route about page
20 def home():
21     return render_template("about.html")#rendering html page
22
23 @app.route("/info") # route for info page
24 def information():
25     return render_template("info.html")#rendering html page
26
27 @app.route("/upload") # route for uploads
28 def test():
29     return render_template("index.html")#rendering html page
30
31 @app.route("/predict", methods=["GET", "POST"]) # route for our prediction
32 def upload():
33     if request.method=="POST":
34         f=request.files['file'] #requesting the file
35         filepath=os.path.dirname(f.filename)#storing the file directory
36         file_path=os.path.join(filepath, f.filename)#storing the file in uploads fold
37         f.save(file_path)#saving the file
38         img=image.load_img(file_path, target_size=(64,64)) #load and reshaping the image
39         x=image.img_to_array(img)#converting image to array
40         x=np.expand_dims(x,axis=0)#changing the dimensions of the image
41
42         pred=model.predict(x)#predicting the results
43         print("prediction:",pred)#printing the prediction
44         if(pred==1):#checking the results
45             result="Infected"
46         else:
47             result="Uninfected"
48         return result#returning the result
49     return None
50
51 if __name__ == "__main__":
52     app.run(debug=True)#running our app
53     app.run(host='0.0.0.0', port=5000, debug=False)
```



```
1 @app.route("/about") #route about page
2 def home():
3     return render_template("about.html")#rendering html page
4
5 @app.route("/info") # route for info page
6 def information():
7     return render_template("info.html")#rendering html page
8
9 @app.route("/upload") # route for uploads
10 def test():
11     return render_template("index.html")#rendering html page
12
13 @app.route("/predict", methods=["GET", "POST"]) # route for our prediction
14 def upload():
15     if request.method=="POST":
16         f=request.files['file'] #requesting the file
17         filepath=os.path.dirname(f.filename)#storing the file directory
18         file_path=os.path.join(filepath, f.filename)#storing the file in uploads fold
19         f.save(file_path)#saving the file
20         img=image.load_img(file_path, target_size=(64,64)) #load and reshaping the image
21         x=image.img_to_array(img)#converting image to array
22         x=np.expand_dims(x,axis=0)#changing the dimensions of the image
23
24         pred=model.predict(x)#predicting the results
25         print("prediction:",pred)#printing the prediction
26         if(pred==1):#checking the results
27             result="Infected"
28         else:
29             result="Uninfected"
30         return result#returning the result
31     return None
32
33 if __name__ == "__main__":
34     app.run(debug=True)#running our app
35     app.run(host='0.0.0.0', port=5000, debug=False)
```

