JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Malaria Parasite Detection Using Deep Learning

Sayyed Madiha Irfan, Dafedar Abir Jabir, Daredia Aalmeen, Dr Varsha Shah
Student, Student, Student, Principal , Rizvi College of Engineering
BE Electronics and Telecommunication Engineering,
Rizvi College of Engineering, Mumbai, India.

Abstract : Malaria is a deadly disease that affects hundreds of millions of people each year all around the world. It can be lethal if not treated promptly. Despite recent advances in malaria diagnostics, the most prevalent method for detecting malaria is microscopy. Unfortunately, the accuracy of microscopic diagnostics is dependent on the microscopist's skill, which limits the speed with which malaria can be diagnosed. With the advancement of Artificial Intelligence tools, particularly Deep Learning approaches, it is now possible to reduce costs while increasing overall accuracy. By adding deep neural networks, we may improve malaria diagnosis from patches segmented from digital microscopic images of red blood cells. In relation to clinical procedures that necessitate the time-consuming manual extraction of features, the suggested technique use a deep learning technique that extracts features from pixels and classifies the red blood cell directly from segmented patches. This study's dataset was obtained from the Kaggle online database.

Index Terms - **Deep learning, Malaria classification, Convolutional neural network , Red blood cells, Thin blood smears.**

I. INTRODUCTION

Malaria is a common disease that has claimed countless lives all across the world. These illnesses can affect humans as well as other animals. Female anopheles mosquitoes are the most common carriers of the disease. The parasite is injected into the afflicted person's blood by a mosquito bite, and it then travels to the liver to mature and reproduce. Many locations in Southeast Asia and Africa are afflicted by the disease's broad expansion. The state of the art for automating detection of malaria using a digitised microscopic image of red blood cells arises from the issues associated with manual diagnosis. The sickness will be detected precisely thanks to the automation of the detection process. To interpret medical images, profound learning algorithms are used in computer-aided systems. Understanding of traditional microscopy to computerised systems is required for automatic malaria diagnosis. Deep learning models have recently surpassed human performance in the recognition of complicated images. Deep learning is the most recent trend in artificial intelligence, and it has improved performance in a variety of medical fields. It is a multilayer neural network classifier that is trained via back propagation and has several additional layers for medical image categorization and detection. To train the software or machine, deep learning necessitates a large number of training sets. A convolutional neural network (CNN) is a deep learning system that takes an input image and classifies it using weights and biases applied to several layers of the network. The dataset is fetched from Kaggle.

II. LITERATURE SURVEY

Paper 1 : MALARIA DETECTION USING DEEP CONVOLUTIONAL NEURAL NETWORKS Swati Singh1 , Isha Gupta2 , Swapandeep Kaur3 , Gurjinder Kaur 41234 Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India

Description : In this published paper, a nine layer convolutional network was implemented , by performing various pre-processing techniques such as data augmentation, data splitting and rescaling to achieve better results. later on accuracy , precision, sensitivity and error rate was calculated and the result was given in a graph format.

Paper 2 : Malaria Parasite Detection Using Deep Learning Methods Kaustubh Chakradeo, Michael Delves, Sofya Titarenko. World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering Vol:15, No:2, 2021

Description : In this paper, few models of convolutional neural networks was compared and a customized model was developed. Accuracy rate was better than most of the models and later on precision is calculated and error rate was given in the paper.

Paper 3 : Rishika Kapoor, university of Cincinnati, 10th October

Description : In this paper various image pre-processing techniques were performed on the data sets and 3 models were implemented namely VGG19, VGG16 and transfer learning. Data augmentation gives a good accuracy rate which is done on datasets to get effective results.

III. Proposed System

3.1 Project Requirements

1)Dataset : The Dataset was taken from kaggle. The dataset contains 27,577 images, half images of malaria infected blood cells and half images of uninfected blood cells. These images are of thin blood smears.

2) Techniques :

Deep Learning : Deep learning is a subset of machine learning where artificial neural networks, algorithms inspired by the human brain , learn from large amounts of data.

Keras: Keras is an open-source software library that provides a python interface for artificial neural networks

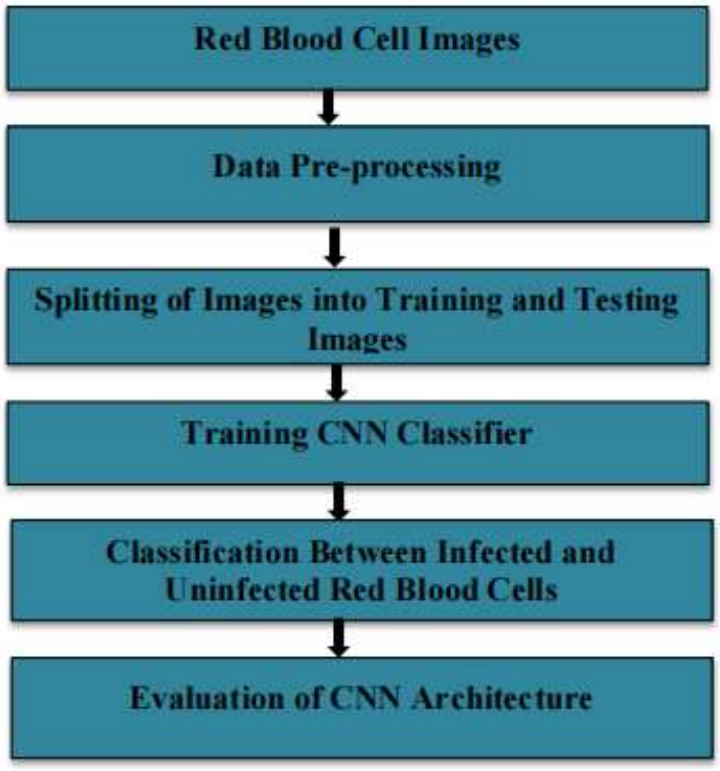
TensorFlow : TensorFlow is a free and open source library for Machine learning and Artificial Intelligence.

CNN Models: In Deep Learning, Convolutional Neural Network (CNN) is a type of an Artificial Neural Network. CNN or ConvNet is a class of deep, feed-forward artificial neural systems, most normally connected to examining visual representations. CNN is widely used for image recognition, image classifications, object detections, recognition faces, etc.

3) Platform : Google colab Notebook, PyCharm
Programming Language : Python
Libraries : 1) NumPy
2) TensorFlow
3) Keras
4) Matplotlib
5) Pandas

3.2 METHODOLOGY

BLOCK DIAGRAM



The proposed system for detection of malaria using slides of red blood cell images consists of various steps which are shown in figure . Initially the red blood cell images are pre-processed and then the splitting of samples into training and testing is done. The training of CNN classifiers is done using the training samples. Once the CNN classifier is trained, the trained CNN classifier gets the input of testing samples to detect malaria.

Data Pre-processing

In this section, experiment has been performed on various techniques and the results are as following:

A)Dividing Dataset

The malaria dataset consists of 27,557 images of which 13,778 are parasitized images and 13,779 are non-infected images. We merged files of both non-infected images and parasitized images and labeled them as healthy and malaria for the respective files. The dataset is divided into three sets for training, validation, and testing.

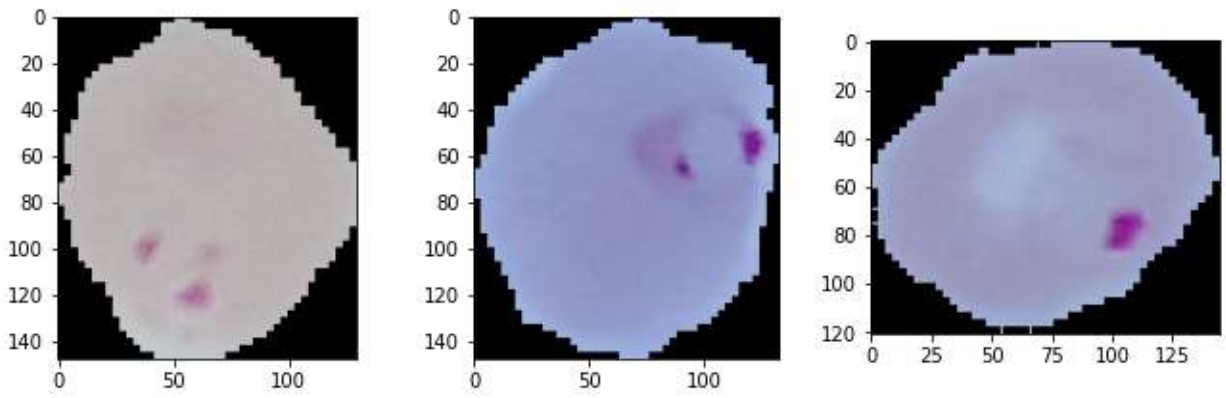
The ratio of train to validation to test is 63:7:30. We choose to take only 63% of dataset as training to avoid high computational intensity due to large dataset. The training dataset consists of 63% of the dataset with 8707 healthy images and 8653 malaria images, the validation dataset consists of 7% with 1001 healthy and 928 malaria images, and the testing dataset is 30% of the whole dataset having 4144 and 4124 healthy and malaria images, respectively. After analyzing the dataset, images in the dataset have different dimensions varying from [46, 58] to [385, 394] over the dataset. Scaling images up gives an advantage in performance but also requires a lot of computation time and memory space., we decided to resize all the images to [125,125]. This resulted in a better classification score with good processing speed.

We will be using Google colab notebook for training Neural Network. Firstly we have to import all the necessary Python Libraries in the Environment. The downloaded dataset from kaggle was already segmented in training and testing parts. We uploaded the whole file on Google drive, and while running the program the directory was mentioned in the code to read the Images

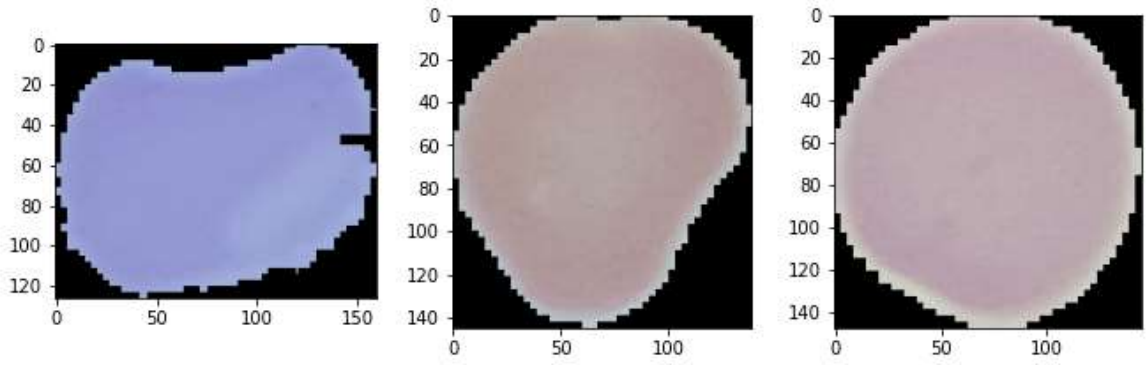
The below figure shows images of Infected blood cells and uninfected blood cells

The images were resized.

Samples of Parasitized Cells



Sample of Uninfected Cells



B) Normaliza**tion**

Normalization is an important preprocessing task which reduces the color and intensity variations present in stained images from different laboratories. Normalization has proven to significantly increase the accuracy of the unseen dataset., the images are collected from the pre-existing dataset of human blood cells which is prepared from in laboratory examination. The smear slides are prepared in the laboratory using various chemical stains which results in color variation due to the use of different chemicals and staining procedures. This staining results in the model to learn and deal with more complex models with a diverse set of images leading to maximizing error rate. A solution to standardize this is normalization.

Model Architecture

Many models exist in Deep Learning, such as VGG-16, VGG-19, ResNet-50, transfer learning, and so on. For this project, we choose to employ the Basic Convolutional Neural Network model in order to improve accuracy.

A CNN architecture consists of two key components: • A convolution tool that separates and identifies the distinct features of an image for analysis in a process known as Feature Extraction

• A fully connected layer that takes the output of the convolution process and predicts the image's class based on the features retrieved earlier.

• Convolutional layers, pooling layers, and fully-connected (FC) layers are the three types of layers that make up the CNN. A CNN architecture will be constructed when these layers are stacked. There are two more significant factors, the dropout layer and the activation function, which are defined below, in addition to these three layers.

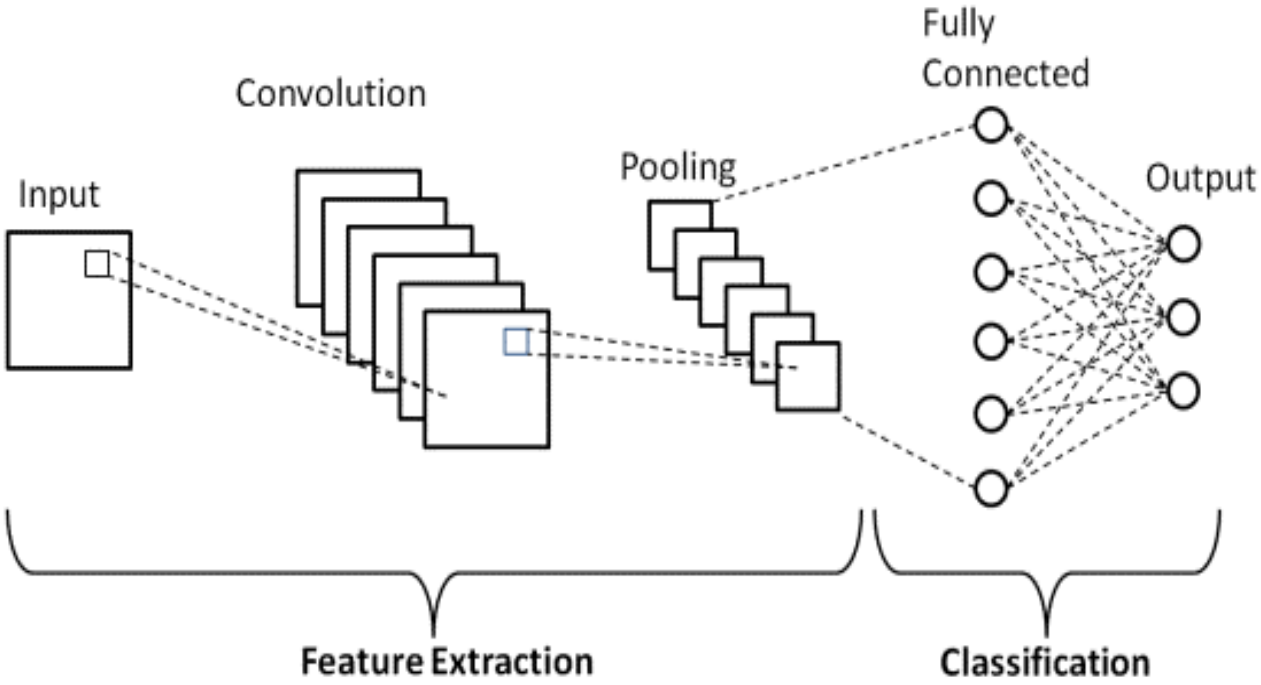


Figure: Blueprint of Deep Learning Model

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 48, 48, 32)        896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 24, 24, 32)        0
_________________________________________________________________
batch_normalization (BatchNo (None, 24, 24, 32)        128
_________________________________________________________________
dropout (Dropout)            (None, 24, 24, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 22, 22, 32)        9248
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 11, 11, 32)        0
_________________________________________________________________
batch_normalization_1 (Batch (None, 11, 11, 32)        128
_________________________________________________________________
dropout_1 (Dropout)          (None, 11, 11, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 9, 9, 32)          9248
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 4, 4, 32)          0
_________________________________________________________________
batch_normalization_2 (Batch (None, 4, 4, 32)          128
_________________________________________________________________
dropout_2 (Dropout)          (None, 4, 4, 32)          0
_________________________________________________________________
flatten (Flatten)            (None, 512)               0
_________________________________________________________________
dense (Dense)                (None, 512)               262656
_________________________________________________________________
batch_normalization_3 (Batch (None, 512)               2048
_________________________________________________________________
dropout_3 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 1026
=================================================================
Total params: 285,506
Trainable params: 284,290
Non-trainable params: 1,216
_________________________________________________________________
```

Figure : Model Summary

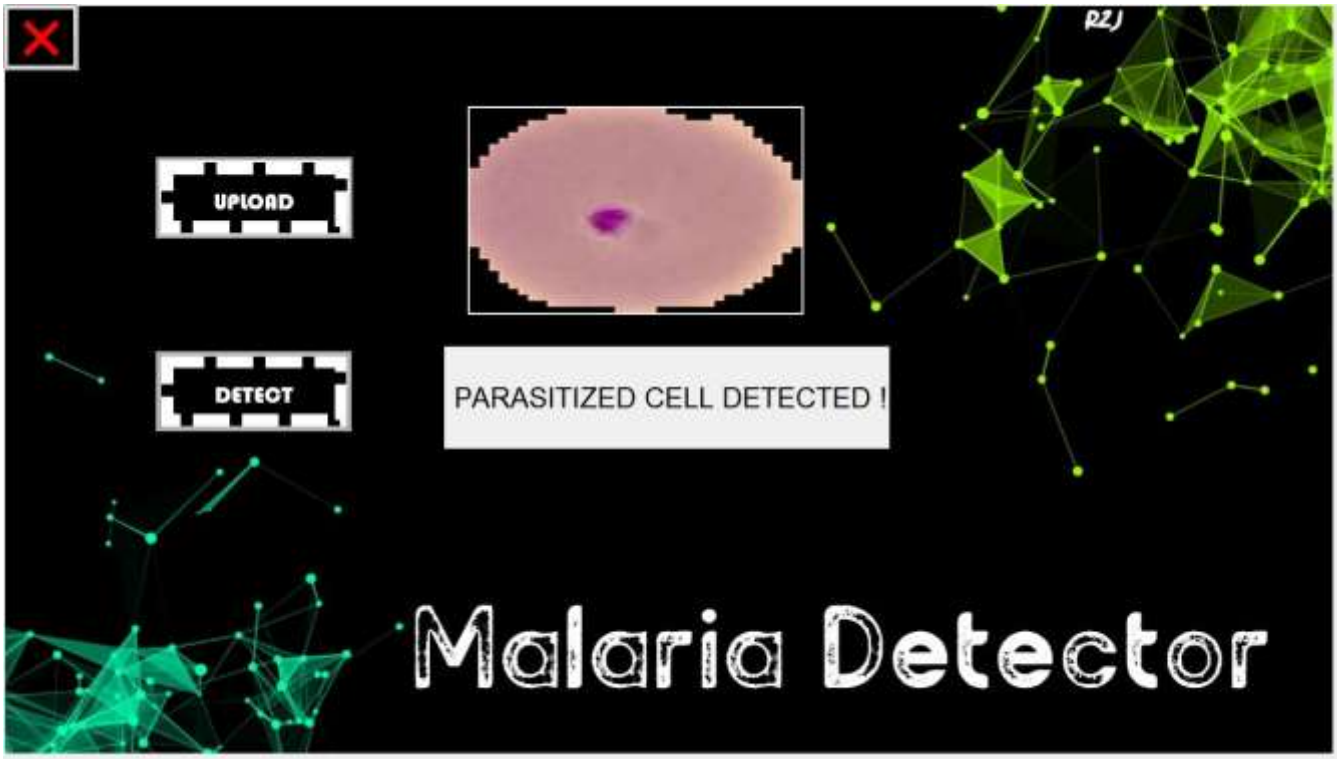
IV. Implementation Plan

Main aim of this project is to make an efficient disease detection application with GUI based (tkinter) frontend and a custom CNN model as backend which detects if a cell is parasitized or normal from its image in real time. GUI will be user friendly so that it will be easy for Health care staff to operate it and get effective results with less effort.
Below are the steps to make effective use of this project:

- **Upload Image:** The user can upload the medical test image through a workstation running on Windows OS. The image should be in jpeg, png or jpg format.
- **Read Image:** The image will be scanned before augmentation takes place.
- **Transform Image:** The scanned image is then transformed into a format that is needed by the saved custom model.
- **Evaluate image using saved model:** The saved custom model creates a feature map of the uploaded medical test image and predicts the output.
- **Determine and Analyze the Output:** The predicted output is then analyzed and converted to a user-friendly language.
- **Display the Output:** The analyzed result is then displayed to the user.

OUTPUT :

Parasitized



Uninfected

ACKNOWLEDGEMENT:

Firstly, we are grateful to Rizvi College of Engineering for giving us the opportunity to work on this project. We are fortunate to have worked under the supervision of our guide Dr.Varsha Shah.
Her valuable guidance and feedback has helped us in completing this project.
We are also thankful to the HOD of Electronics and Telecommunication Engineering,
Prof. Junaid Mandviwala for giving us access to all the resources that went into building this project.

CONCLUSION

Artificial intelligence is attracting the attention of healthcare organisations of all sizes, types, and specialties who want to improve patient care while lowering costs and enhancing efficiencies. The detection of malaria parasites using deep learning models is a fantastic feat of AI. Another good technique to implement this project is the way we did it. Before AI, microscopists' abilities were the only means to detect contaminated blood cells under a microscope, but now with AI, it can be done with less work and with more accurate results. We created a rudimentary CNN model in this project and tried to make it more user-friendly than the models discussed in earlier publications.

REFERENCES

- Rishika Kapoor, university of Cincinnati, 10th October
- MALARIA DETECTION USING DEEP CONVOLUTIONAL NEURAL NETWORKS Swati Singh1 , Isha Gupta2 , Swapandeep Kaur3 , Gurjinder Kaur 41234 Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India
- Malaria Parasite Detection Using Deep Learning Methods Kaustubh Chakradeo, Michael Delves, Sofya Titarenko. World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering Vol:15, No:2, 2021