Malaria Disease Detection

Project Report

Submitted for the partial fulfillment

of B.Tech. Degree

in

COMPUTER SCIENCE AND ENGINEERING

by (Group 17)

Preksha Rai (1805231039)

Poonam Agrawal (1805213036)

Anjali Shivhare(1805213010)

Under the supervision of

Dr Parul Yadav

Dr Jasvant Kumar

Department of Computer Science and Engineering

Institute of Engineering and Technology

Dr. A.P.J. Abdul Kalam Technical University, Lucknow, Uttar Pradesh.

Contents

Declaration

We therefore declare that this post is our work and that, in our belief as well information, does not contain anything previously published or written by another person or property which is a major mistake accepted in the awarding of any degree or university diploma or another institution of higher learning, unless the confession is published in a book text. We did not send the project to any other institution for any other qualification requirement.

(1) Name: Poonam Agrawal

Roll No.: 1805213036

Branch: CSE

Signature:

(2) Name: Preksha Rai

Roll No.: 1805231039

Branch: CSE

Signature:

(3) Name: Anjali Shivhare

Roll No.: 1805213010

Branch: CSE

Signature:

Certificate

This is to confirm that the project report entitled “Malaria Detection” presented by Preksha Rai, Poonam Agrawal, and Anjali Shivhare at the completion of the Bachelor of Technology in Computer Science and Engineering, a record of the work done by them under me supervision and guidance at the Department of Computer Science and Engineering at the CenterEngineering and Technology, Lucknow.
It is also confirmed that this project has not been submitted to any other institution for any awards other degrees to my knowledge.

Dr Parul Yadav

Department of Computer Science and Engineering

Institute of Engineering and Technology, Lucknow

Dr Jasvant Kumar

Department of Computer Science and Engineering

Institute of Engineering and Technology, Lucknow

Acknowledgement

We owe a special debt of gratitude to Dr. Parul Yadav and Dr. Jasvant Kumar, Department Computer Science, Institute of Engineering and Technology, Lucknow with their regular support as well guidance during our work. Their sincerity, awareness and determination have been an ongoing source of encouragement to us. It is only in their efforts to recognize our own attempts to see the light of day. We also take this opportunity to acknowledge the contribution of Prof D.S. Yadav, Head of Department of Computer Science and Engineering, Institute of Engineering and Technology, Lucknow with his full help and assistance during job development. We do not like to miss out an opportunity to acknowledge the contribution of all the intelligence of the department and staff on their behalf kind assistance and cooperation during the development of our project.

Name: Poonam Agrawal
Roll no.: 1805213036
Signature:

Name: Preksha Rai
Roll no.: 1805231039
Signature:

Name: Anjali Shivhare
Roll no.: 1805213010
Signature:

Abstract

Through this project, we are performing Malaria Detection using Deep Learning techniques. We have taken the blood sample images of parasitized and uninfected images. The splitting of the dataset is done into testing and training categories. Then, the image is fed to the MobileNet model, which is then passed through various convolution and pooling layers. The image is classified as Parasitized if the output is ‘0’ and Uninfected if the output is ‘1’.

We have also implemented the VGG16 model and compared the accuracy of the VGG16 model and MobileNet model. The MobileNet model provided better performance.

The technology used here is Deep Learning and Convolution Neural Network. Deep Learning functions just like the human brain does. No explicit programming is done. It makes use of various non-linear processing units which perform the extraction of features and also do the transformation. The next layer receives the output from the previous layer and so on.

Deep Learning works by extracting the features by themselves and doesn’t require any knowledge from the programmer and thus solves the problems of dimensionality. It is used when we have multiple inputs and outputs. Deep learning follows the idea of mimicking the brain.

An artificial neural network (ANN) is a calculational model that consists of many processing units that receive inputs and deliver outputs based on their already defined activation functions.

A CNN is a feed-forward neural network, often leveraged to examine visible images with a grid-like matrix by processing data. It is also known as ConvNet. A convolutional neural network is leveraged for the classification and detection of features in an image.

For this project, we have used deep learning and CNN to detect malaria. We have differentiated the parasitized and uninfected samples of blood smear.

In future, we will be implementing our project using other ANN models such as Alexnet, VGG19, Resnet etc to see how the model will react to the data sets and also how better would be the accuracy of the output.

List of Figures

List of Tables

Chapter 1

Introduction

1.2) About Malaria:

Malaria is a serious illness root cause are insects that infect humans through the bite of female Anopheles mosquitoes. It can be cured if the right steps are taken. According to the World Health Organization, there are 228 million active cases in Africa and 405000 deaths by 2020 [1]. Malaria is a major problem in Africa as it is clearly evident in the data. Approximately 94% of these deaths occurred in the African region. It is known to us that about 90 percent of cases of toxic malaria are in children under 5 years of age. An approximate 3.2 billion people in 95 countries are at high danger, according to a 2016 World Malaria report [5]. Millions of blood tests are performed each year to detect the effects of malaria, including manual parasites and contaminated red blood cells using a skilled microscopist. Correct parasite statistics are important not only in diagnosing malaria. They are also very prominent in assessing drug resistance, drug efficiency, and illness acuteness . However, microscopic diagnoses are poorly maintained and rely heavily on microscopist ability and knowledge. It is very common for microscopes to operate alone in low-cost settings, without a solid system in place that can guarantee the conservation of their capabilities and thus curable quality. This results in wrong diagnostic conclusion in this area. In untrue scenarios, this results to non prominent use of antimicrobials, second vision, loss of working days, and in few scenarios the spread of terrible malaria. In fictitious scenarios, the diagnosis leads to non prominent use of anti-malarial drugs and pain from possible reactions , such as nausea, abdominal pain, diarrhoea, and in some scenarios serious issues. Therefore, these facts have encouraged us to take up this building project. Diagnosis of Malaria through ML will not only benefit Health Care but also assist in our studies as Machine Learning is a new benefit to the industry.

1.2 Objectives

- Study previously done work by researchers in the line of our project.
- Proposing a model based on our study.
- Implementing our model.
- Comparative study of pre-existing models and our model.

1.3 Organization of Report

In this chapter we have described Malaria disease and its severity . In the following chapter, headed Chapter 1 "Literature Review," we present an overview of Deep Learning ,CNN and the more current methods used in this project. In the next chapter,Chapter 2 we have explained the proposed models in detail .In Chapter 3 we have elaborated “Methodology” of our proposed model 150 CNN. In Chapter 4 we have presented the “Experimental work” ,compared the accuracies of our models and in the last chapter ,Chapter 5 we have discussed “Conclusions” and future scope of our project.

Chapter 2

Literature review

Malaria is an infection related to the blood, Plasmodium parasite is the root cause of this illness. This infection is passed on from one person to another person because of a mosquito named Female Anopheles. Malaria is one of the world's death-dealing illnesses and it comes to the top list of child-killing diseases on earth. If the treatment for this illness is delayed, this infection can cause other blood-related illnesses e.g. Anaemia (lack of blood), hypoglycemia, or brainy malaria, in which paths for capillaries to the brain for blood are blocked. Brainy malaria can cause coma, disabilities for life, and death. The detection of malaria at an early stage can reduce the death rate. Microscopic diagnosis is accepted by many and it is the most popular malaria diagnosis method. In this diagnosis, thick and thin blood films are stained on a glass slide to visualize malaria parasites under a microscope. Expert technicians and laboratorians under a microscope examine the blood slides that are prepared . This method is widely accepted because of its simplicity and its potential to detect the existence of parasites. However, the process needs labor, a lot of time for examination, and requires expertise and well-trained healthcare workers. The shortage and unavailability of trained technicians and health facility resources in remote rural areas make the microscopic diagnosis inefficient. Most people in rural areas use some sort of self-treatment based on symptoms which results in a false diagnosis and improper treatment.

Misdiagnosis of malaria can be due to two major reasons, immense workload and limited human resources with the primary reason being the shortage of skilled technicians. In the countries with high malaria cases, it is evident that there is a high requirement for skilled technicians, as per the survey conducted in Ghana. Other countries where malaria is not that common do not have skilled technicians; the technicians are not well trained as they lack to witness the malaria cases very often. Automating the detection process will help in easy and effective diagnosis all around the world. Data can be collected globally and the model can be trained on the global data set to give more accurate results.

This chapter is is divided into two phases :

- Previously Implemented Models
- Technologies used for the construction of models

2.1 Previous Models

Some of the previous works done related to Malaria Detection are described below .Researchers have come up with many models for the detection of this life threatening disease.

2.1.1 Autoencoder

One of the proposed models was to detect malaria parasites in a blood smear using autoencoder-based architecture [3]. Autoencoder is a common type of artificial neural network that compress input data into a low-resolution hidden area and ultimately output rebuilt using a decoder.

2.1.2 CAD

Also, Computer-aided diagnostic (CAD)[2] tools using machine learning (ML) algorithms put in an application to microscopic blood smear images have the capacity to decrease clinical burden by supporting priority and disease interpretation. These tools process medical pictures for typical appearances and emphasize pathological features to supplement clinical decision-making. For these reasons, Computer-aided diagnostic tools have gained approval in picture-based medical diagnosis and risk estimation. The CAD system has as many layers of FLAN and SSAE as it is a neural network. learning models, in which the parameters of the output of each layer are connected to the input and trained in the background after the layer. This scheme network can be used not only to reduce the size but also to reduce the complexity of the calculation. This model focuses on two phases of study to find the best malaria parasites in small blood smear images.

2.1.3 ResNet-50

In the research[6], two deep Convolutional Neural Networks, VGG-19, and ResNet-50, are experimented on to analyze and compare the best performing model on the malaria dataset. Their results showed that the ResNet-50 outperforms the VGG-19 by achieving an accuracy of 97 percent.

These were the previous research works related to Malaria Detection.We have implemented the existing VGG16 model and achieved an accuracy of 90.64%.

2.1.4 VGG16

Objective

ImageNet Database contains 224 * 224 standard images and contains RGB channels. Therefore, there is a tensor (224, 224, 3) as our input. The model performs input image processing and renders output in the form of a 1000 value vector

Architecture

The size of the input image used in the network is : (224,224,3). There are 64 channels in the first two layers having 3*3 filter size. They also have the same compression. Then a layer of stride pool (2,2) which is large, two layers having convolution layers with size of the filter 256 and another size is (3,3). Which is then followed by a binding layer which is large in size of step (2,2) same as the previous layer. This is followed by 2 layers having convolution of filter size (3,3) and 256 filters. Next, there are two sets of three layers consisting of convolution and a pooling layer of large in size. Each has got filters of size (3,3) having the same pads. The particular image is transferred to a two-layer convolution stack. For these large integration and layers of convolution, the filters that we have used are of size 3*3 instead of 11*11 on AlexNet and 7*7 on ZF-Net. In few layers, there is also the usage of 1*1 pixel to manipulate the number of input channels. One pixel termination is done. This is done after each layer of rotation so that the local element of the image is prevented.

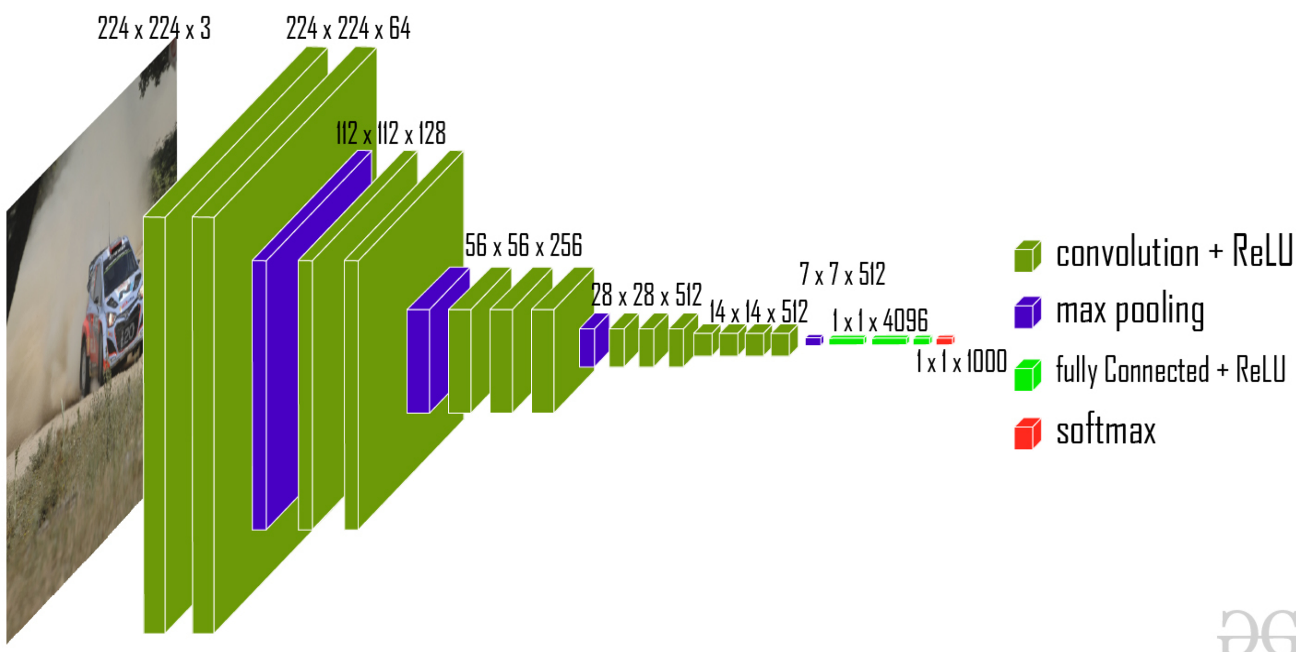


Figure 2.1 VGG16 CNN Architecture

After so many max-pooling layers and convolution layers, it is found that (7,7,512) we have got a feature map. Then the output is decrypted so that a component vector is made, i.e, (1,25088). This is then followed by a three fully connected layer, the 1st layer gets the input from the vector of the last element. Then the vector of size (1,4096), is subtracted. The 2nd layer extracts a vector size (1,4096), but 1000 channels are removed from 1000 class ILSVRC challenge. After the release of the 3rd layer which is fully integrated is then transferred to a softmax layer so that the separation vector is made normal. The release of the fifth vector segment is categorised for testing. ReLu is used as its unlock function by all the hidden layers. This is because ReLu works well mathematically as it leads to faster learning thus decreasing the chances of having a collapse problem of gradient.

Challenges of VGG16

It's one of the challenges is that training is done very slow . So, a lot of bandwidth and disk space is used which has made it inefficient.

We have implemented 150 CNN model which is based on MobileNet and achieved an accuracy of 93.06%.

2.1.5 MobileNet

MobileNet uses a wide variety of deep convolutions. It does a significant reduction in many variable compared to a network with standard conversion with the equal depth in the nets. Deep neural network becomes less heavy. Light neural network, MobileNet has very few variables and segment precision . You want to decrease the number of variables used in the network and upgrade the precision of Partition, dense blocks presented on DenseNets are then imported into MobileNet. Because of Dense-MobileNet models, convulsive layers of the equal size that include input maps on MobileNet models are considered dense blocks and within dense blocks, dense links are made. The new network structure can make better use of the previously identified mapping feature Layers of convolution in dense blocks, so a large number of feature maps will be produced in a few convolution cores and those who use features regularly. By keeping the growth rate low, the network also decreases variables and calculation costs. Two Dense-MobileNet models, Dense2-MobileNet and Dense1-MobileNet, are built. Test results show Dense2-MobileNet can get higher identification precision than MobileNet, while having lesser number of variables and calculation costs.

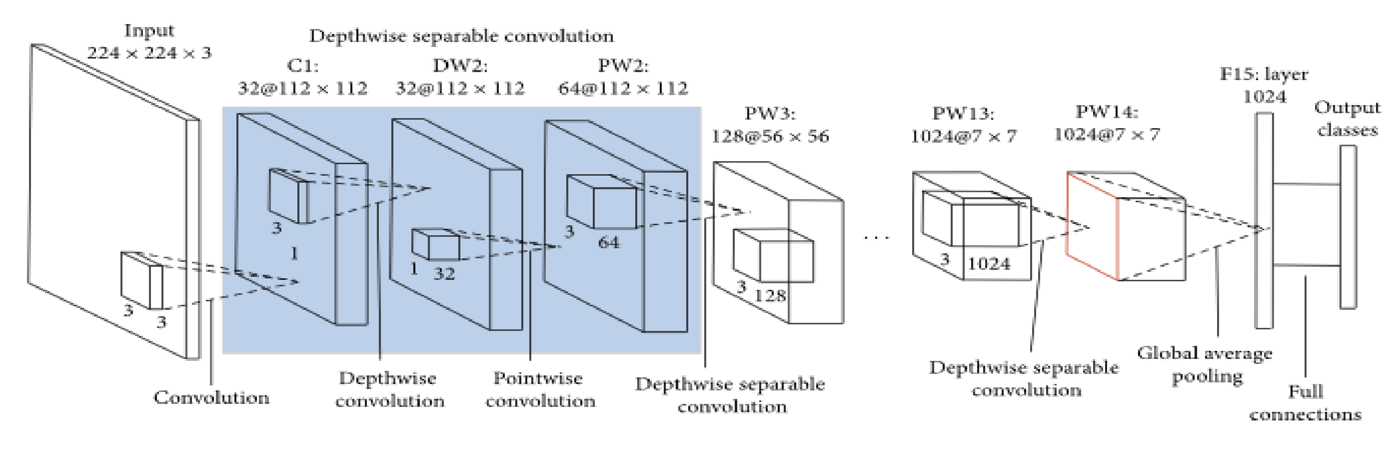


Figure 2.2 MobileNet Architecture

The technologies which are used to implement the models related to Malaria Detection are as follows:

2.2.1 Deep Learning

Machine Learning is the subset of Artificial Intelligence, and Deep Learning is the branch of it. Deep Learning functions just like the human brain does. No explicit programming is done. It makes use of various non-linear processing units which perform the extraction of features and also do the transformation. The next layer receives the output from the previous layer and so on.

Deep Learning works by extracting the features by themselves and doesn't require any knowledge from the programmer and thus solves the problems of dimensionality. It is used when we have multiple inputs and outputs. The idea of deep learning is to mimic the brain. The neural network helps in the implementation of Deep Learning and the idea behind is how biological neurons work just like brain cells.

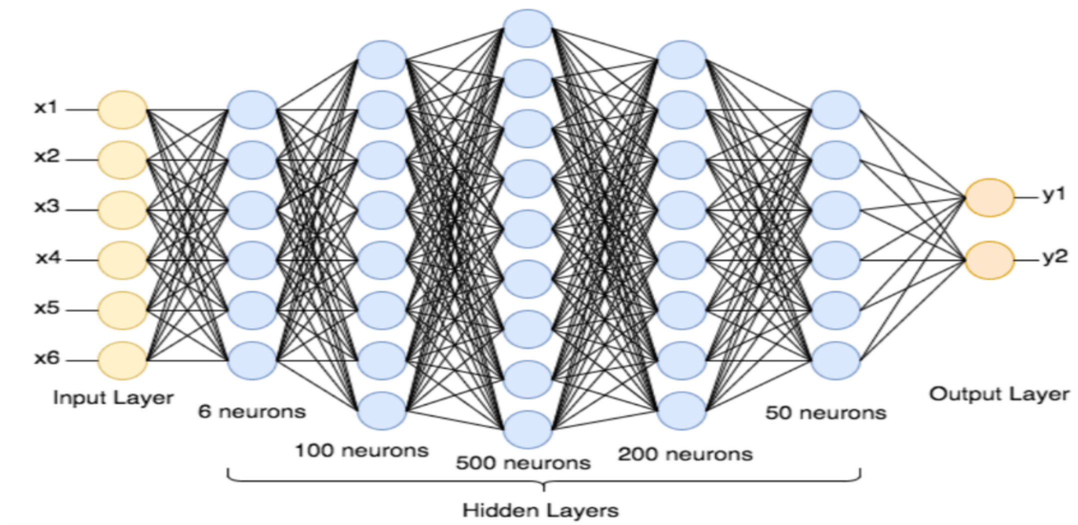


Figure 2.3 Deep learning Architecture

In the above example, the first layer gets the raw data items of the input layer. Then, the patterns of the local contrast are determined by the input layer which is then differentiated on the basis of colors, luminosity, etc. Then the face feature is determined by the first hidden layer which means the fixation of the eyes, lips, nose, etc. Then, facial features will be adjusted to the appropriate face template. Therefore, within the second hidden layer, it will ensure that the appropriate surface here is visible inside the top image when it is sent to the output layer. Similarly, additional hidden layers are added to solve more burdensome issues, e.g, if we want to see if a particular surface has a large or light color. Therefore, when the hidden layers increase, we will solve additional machine problems.

Deep Neural Networks is a neural network that introduces complexity to a precise level, which proposes a large number of hidden layers transmitted between input and output layers. They are an extremely capable model for making indirect organizations.

Deep Belief Networks can be a category of Deep Neural Networks made up of multi-layer network beliefs.

Steps to Creating Deep Belief Networks: The layer layer is identified in the detectable units leveraging the Contrastive Divergence algorithm.

Then, the earlier instructed properties are served as visual units, and the function of the learning features is performed. Finally, the full Deep Belief Networks is trained when the final hidden layer reading is completed.

Recurrent Neural Networks: It permits both sequential computations as well as parallel computation, and it works just like the human brain. Since they are capable enough to remember all of prominent things in relation to the input that they have got, they are more accurate.

How Deep Learning Works: Many neural network architectures are used by many deep learning models, that is the main reason why deep learning models are known as deep neural networks. The term "depth" usually means the number of hidden layers present in the neural network. Typically, neural networks have only two or three hidden layers, while deep neural networks can have as many as up to fifty.Large labeled data sets are used for the training of in-depth learning models and neural network architectures that acquire skill to detect properties directly with the data. There is no need to remove the properties in person now

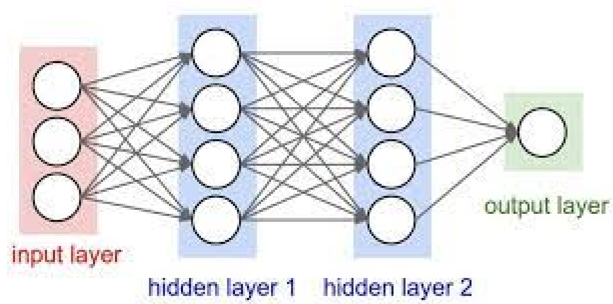


Figure 2.4 Simplified View Of Deep Learning Model

Convolution neural networks (CNN or ConvNet) are the important variety of deep neural networks. A Convolution Neural Network convolves learned options with input files, and 2nd convolution layers are used, and this design is used for the processing of 2nd knowledge like pictures.

The importance of manual feature extraction is eliminated. So, because of this, we have a tendency to set up the options to classify pictures. The options are extracted directly from the pictures using CNN. The antecedent training of the connected options is not done. They are only learned and whereas the training of the network is done using a group of pictures. The deep learning models become extremely correct using this automatic feature extraction for tasks like object classification.

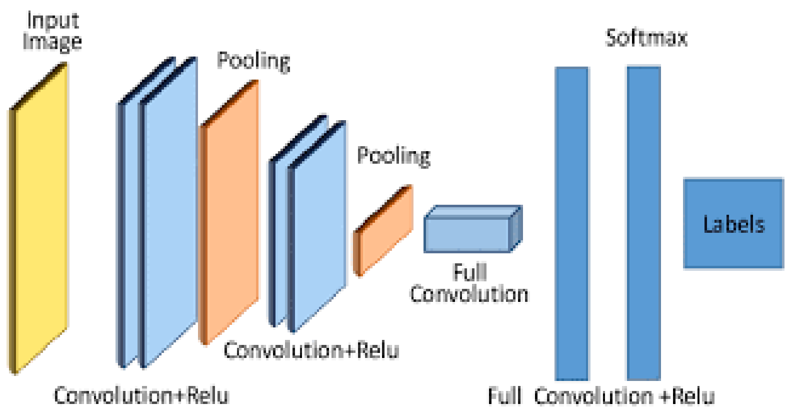


Figure 2.5 Architecture of CNN.

CNN detects many image elements using a few hidden layers. All hidden layers increase the complexity of the image elements learned. For example, the main hidden layer learns to find the edges, and the last layer learns more advanced conditions that are taken into account, especially the shape of the object we usually see.

How we can train and Create Deep Learning Models : The three commonest ways that folks are using deep learning to perform object classification are:

Training from Scratch : To train a deep network from scratch, a set of data with a very large label is collected and the network architecture is intended to learn features and models. This can be great for very new programs or applications that can have a huge variety of output categories. This can be a very unusual method because, with large amounts of data and level of learning, these networks usually take weeks to train.

Transfer Learning

Various in-depth learning applications leverage a transfer learning method, a method that incorporates a well-developed pre-trained model. We often start with an existing neural network such as AlexNet or GoogLeNet, and new information is introduced by containing previously unknown categories. After the tweaks have been created on the network, we can now do some new work, such as categorizing dogs or cats into a thousand completely different things. This will have the advantage of needing very little data (processing only 1000 images, there are millions), so counting time is reduced to minutes or hours.

2.2.2 Artificial Neural Network:

A computational unit that counts based on another connected unit known as the artificial neural network. In the scenario of a one synthetic neuron, it will be linked to the input illustration of the corresponding degree object to which you wish to retrieve details. Synthetic neurons will learn information from inputs and perform a specific calculation and calculate their value. Based on the number, the neuron will determine whether a specific specification is available for input or not.

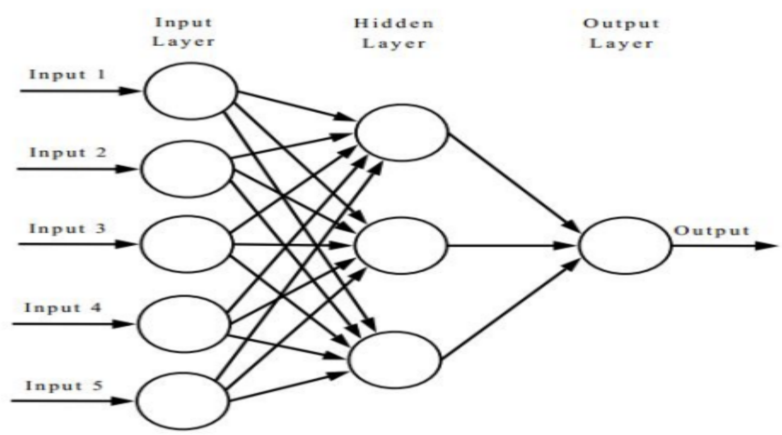


Figure 2.7 Neural Network

● **Input layer:** It is a set of input neurons, where each of the neurons represents the feature in our dataset. The initial input is taken at this layer and will be forwarded to the next layer.

● **Hidden layer**: A set of neurons where each neuron has a weight associated with it. It takes the input from the previous layer and after performing the dot product of inputs and weights, the activation function is applied, generates the result and passes the info to the next layer.

● **Output layer**: it is the same as the hidden layer except it gives the final result. For the given input, the networks calculate the results based on the inputs bypassing the inputs in the first layer of the network and passing the values to the subsequent layers. This process is known as 'forward propagation' and depend on the values generated by the network, weights are adjusted to minimize the error calculation

2.2.3 Convolutional Neural Network

CNN is an advanced neural network, often used to analyze visual images with a grid-like Matrix for data processing. Also known as ConvNet. The convolutional neural network is used to distinguish and detect objects in the image.

Below is a neural network used to identify two types of flowers: Orchid and Rose.

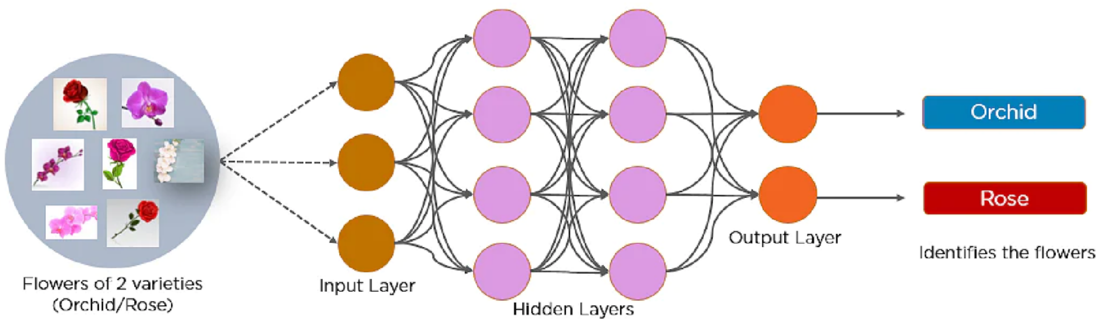


Figure 2.8 CNN Model For Identifying Flowers

In CNN, each picture is considered using an array of pixel values.

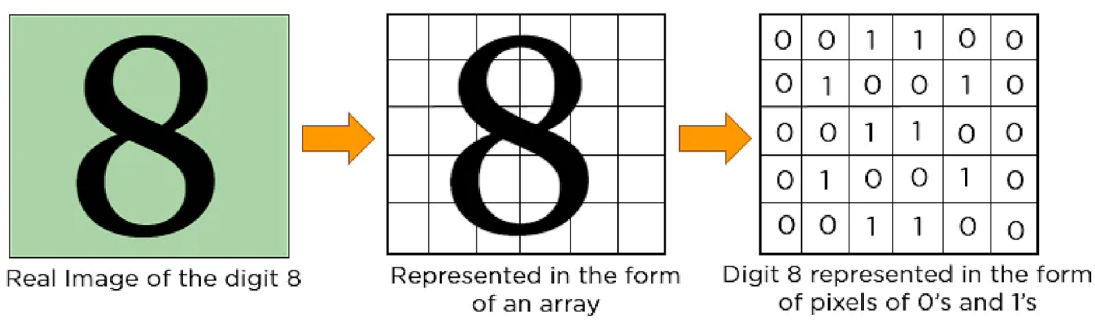


Figure 2.9 Image Conversion to pixel

Layers in a Convolutional Neural Network : A convolution neural network has many hidden layers that are used in extracting information from the image. The four essential layers in CNN are:

- Convolution layer
- ReLU layer
- Pooling layer
- Fully connected layer

Convolution Layer

This is the first step in the process of removing important elements from a picture . The convolution layer has many filters that are used to execute the conversion function. The whole image is represented as a matrix of pixel values.
Suppose a 5x5 image with a value of 0 or 1 pixel. There is a used filter matrix with a size of 3x3. We move the filter matrix over the convolution picture and calculate the product of the dots to get the convolved matrix element as output.

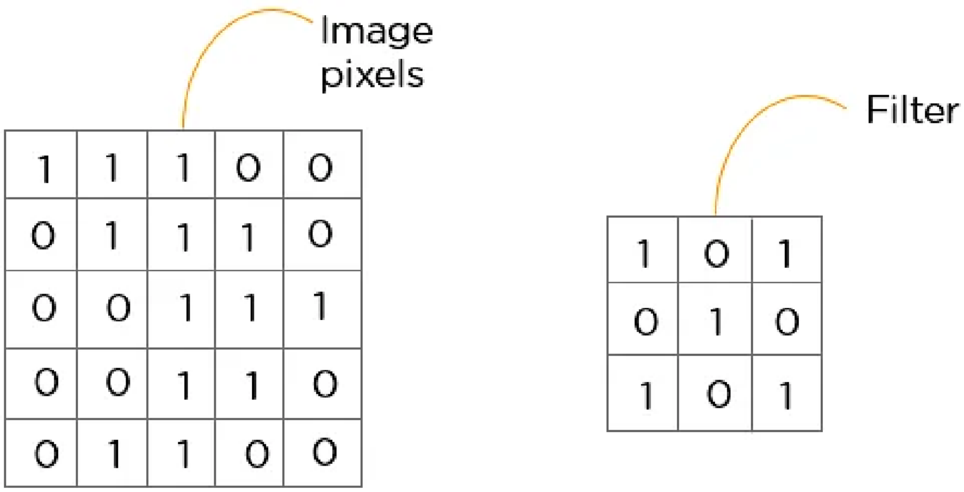


Figure 2.10 Sample Convolution Kernel and Sample Filter Kernel

ReLu Layer

It represents a modified line unit. Once the feature maps are calculated, the next step is to move them to a fixed line unit layer.The intelligent functions of the element are performed on the ReLu layer. Converts the value of all negative pixels to 0. It brings non-linearity to the network, and the output is a fixed feature map.

Pooling Layer
Low sample performance is included. Feature map size is reduced to this layer. The modified feature map is now sent to the composite layer to produce an integrated feature map.

Fully Connected Layer

Fully Connected Layer neural network for easy feed transfer which is the last few part of the network. The input taken by this layer is the output from the Final Layer of Converts or Converts.

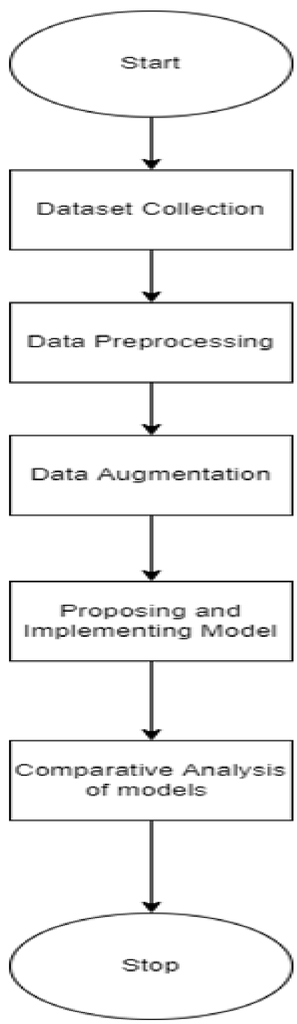
Chapter 3

Methodology

The following steps are involved in the Methodology of our model -

- Dataset Collection
- Data Preprocessing
- Data Augmentation
- Proposing and Implementing Model

Flowchart of the Steps followed in Methodology:



3.1 Dataset Description:

The dataset consisted of 27,560 cell images with same number of parasitized and uninfected cells instances. Positive instances accommodate Plasmodium and the negative instances contained no Plasmodium but other kinds of objects including staining artifacts/ impurities.We used randomized splitting to split entire data in the ratio of 80:20 as training and testing data respectively.

	Training	Testing
Parasitized	11024	2756
Uninfected	11024	2756
Total	22048	5512

Table 3.1 : Dataset Table

3.2 Data Preprocessing:

Data Preprocessing contains the steps we need to follow to interpret or encode data so that it may be smoothly parsed by the machine. Accuracy and precision in prediction that the algorithm should be able to easily interpret the data's features is the main purpose of the model.

3.3 Data Augmentation:

Data Augmentation is a technology that is applied to the dataset to enhance the type of data present for training models. It has been used in the various medical dataset to improve classification performance . Majorly the dataset with small size uses data augmentation to enhance the number of samples to improve model performance by supplying variable data values for the model to train on. The dataset consists of equal instances in different classes keeping it a balanced dataset. Data augmentation is performed on the training dataset which eliminates the chances of having similarity with the test dataset. Data augmentation applied includes rotation, flip, horizontal and vertical shifts, lightening the images, darkening, zoom in, and zoom out.

3.4 Proposed Model:

In this chapter we have proposed our model for the detection of Malaria disease. Our model is 150CNN for which the base model is MobileNet .We have used the concept of transfer learning which is also elaborated below in details.

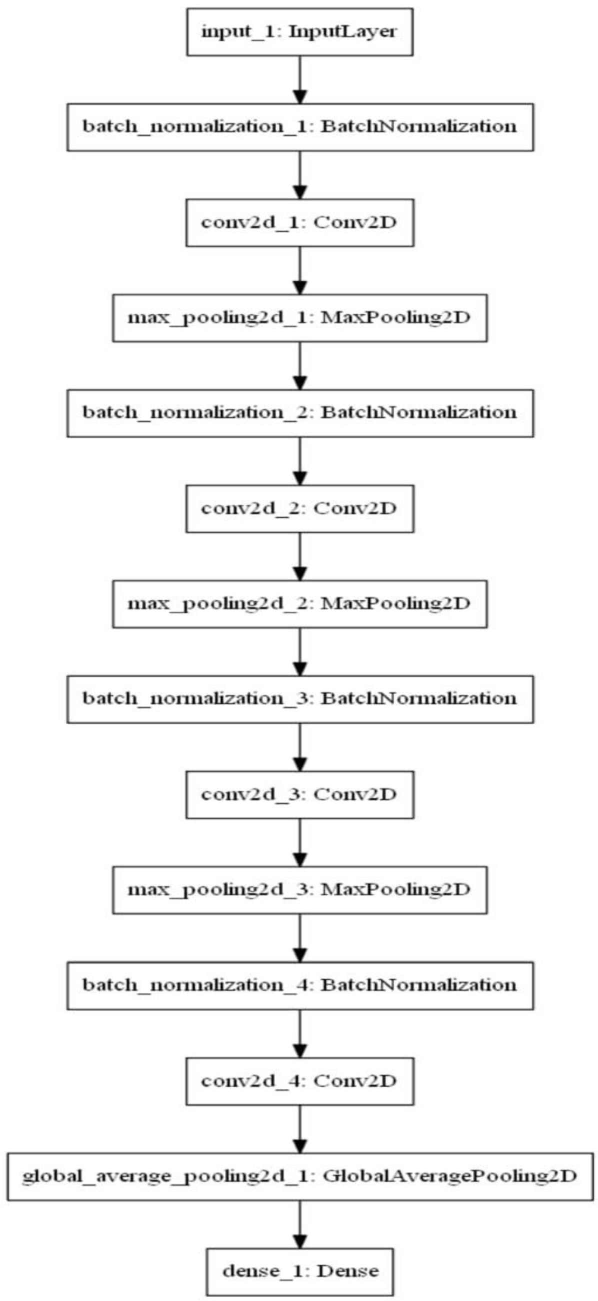
3.4.1 Transfer Learning:

Through the transfer learning, it enables the user to transfer the knowledge of already trained models so that it can be useful in their own set of problems. There is no need to create the model from scratch in the project, we are using the models that are trained on larger datasets such as ImageNet having 100,000 datapoints and explored the efficiency of transfer learning which is proven to be very proficient in many types of image classification of research. We have used transfer learning on the VGG-19 model and the same for MobileNet and evaluating their performances which are discussed in further sections.

3.4.2 150CNN:

A wide range of convolutions deep in nature is used by 150CNN. It makes a remarkable reduction in the number of parameters when compared to a network with standard modifications having similar depths present in the network. This makes deep neural networks less difficult.

As a deep network and simple neural network, 150CNN has very few parameters and partial accuracy. If there is a need to decrease the number of used variables in the network and upgrade the accuracy of the partition, the dense blocks suggested on DenseNets are then imported into MobileNet because MobileNet are dynamic models, having layers of the similar size that insert input maps into MobileNet models are considered as dense blocks and within dense blocks, dense connections are made. The new network architecture can make better use of the pre-identified map element layers of convolution in dense blocks, so a large number of feature maps will be produced in a few convolution cores and those who use features regularly. By keeping the growth rate low, the network also reduces parameters and calculation costs. Two versions of Dense-MobileNet have been developed, Dense2-MobileNet and Dense1-MobileNet. Test results show that Dense2-MobileNet can achieve higher recognition accuracy than MobileNet while having fewer restrictions and calculation costs.



Flowchart of 150CNN

3.6 Implementation

We have used the MobileNet(150 CNN) and VGG16 artificial neural network model for the classification of images. It is found out that 150CNN provides better accuracy than VGG16 in our project. We have preferred it over other models because MobileNet uses convolutions that are depthwise separable. It proficiently reduces the number of parameters when the comparison is done with the network having convolutions that are regular having the same depth in the nets. This results in lightweight deep neural networks. It gives better accuracy also. We have divided the data set into train and test sets. Both consist of parasitized and uninfected images. The total size of the dataset is 27,560. For training, Parasitized: 11024 and uninfected:11024. For testing, Parasitized: 2756 and uninfected:2756.

Initially, the image of input size : (150,150,3) is provided. The image is resized as a consequence of the model's input. Based on the input image's size, various data augmentation techniques are applied to the image which makes the images more model friendly. After the data augmentation, the image is sent to the model for training and the results are calculated. The technology we have used is TRANSFER LEARNING because the weights are trained only in the last layer keeping other layers frozen. There are two output classes, i.e, parasitized and uninfected. The input image is passed through various convolution layers, pooling layers and Batch Normalisation to finally classify it as to whether it is parasitized or uninfected. ReLu is used to introduce non-linearity to the final image and also to make the model classify better and improve the computational time as other models used sigmoid and tanh functions. This proved much better than that. MobileNets work very efficiently and these are small deep learning architectures especially made for mobile devices. Due to the small size in nature, there is a tradeoff of accuracy when compared with the larger fully convolutional architectures, but that is very minute.

The model is firstly trained in multiple epochs where the weights are updated using Backpropagation and the loss function is minimized eventually. At last, the test images are provided to get them classified as parasitized or uninfected. The result is displayed at '0' if Parasitized or '1' if uninfected.

Chapter 4

Experimental Work

Firstly, we have trained and tested the existing model, that is, VGG16. Then, we have implemented the MobileNet model and changed the last 40 layers of it to make it a new model as 150CNN. Afterwards, their performance is compared.

4.1 Required Libraries and Modules:

In our model we have used many inbuilt libraries and modules which includes tensorflow, keras ,numpy ,pandas,matplotlib,sklearn etc.Tensorflow is a library of python that is used when there is a requirement of fast calculation. Google has created and released this library.Tensorflow library is leveraged for the construction of models of deep learning directly or by leveraging other pre-existing libraries.Google has also developed a high level API that is keras.

```
from keras import models
from keras import layers
from keras import optimizers
import numpy as np
import pandas as pd
import math
import matplotlib.pyplot as plt
import cv2
from sklearn.model_selection import train_test_split, StratifiedKFold
import keras
import tensorflow as tf
from keras.models import Sequential, Model
from keras.layers import Input, Flatten, Dense, Dropout, Convolution2D, Conv2D, MaxPooling2D, Lambda, GlobalMaxPooling2D, GlobalAveragePool
from keras.preprocessing.image import ImageDataGenerator
from keras.callbacks import EarlyStopping, ModelCheckpoint, ReduceLROnPlateau
from keras.utils import np_utils
```

Existing model :

4.2 VGG16

The existing model which we have worked upon is VGG16. The size of the input image provided is (125,125,3). The activation functions used are ReLu and Sigmoid.

```
def getVggAngleModel22():
    input_2 = Input(shape=[1], name="angle")
    base_model = VGG16(weights='imagenet', include_top=False,
                 input_shape=(125,125,3), classes=2)
    x = base_model.get_layer('block5_pool').output


    x = GlobalMaxPooling2D()(x)
    merge_one = Dense(16, activation='relu', name='fc5')(x)

    predictions = Dense(2, activation='sigmoid')(merge_one)

    model = Model([base_model.input], predictions)
    return model
```

The loss function used is binary crossentropy. The model is trained with 15 epochs.

```
model.compile(loss='binary_crossentropy',
              optimizer=tf.keras.optimizers.SGD(learning_rate=0.01),
              metrics=['acc'])
# Train the model
history = model.fit(
      train_set,
      steps_per_epoch=train_steps,
      epochs=15,
      validation_data=validation_generator,
      validation_steps=vaid_steps,
      verbose=1,callbacks = [checkpoint, early])
```

4.3 150CNN

We have implemented 150CNN model. The input image is passed through Batch Normalisation, Convolution and Max Pooling layers. The activation function used is ReLu.

```
def get_model():

    x = Input((150, 150, 3))
    model = BatchNormalization(axis = 3)(x)
    model = Convolution2D(filters = 32, kernel_size = (3,3), strides = (1,1), padding = 'same', activation='relu')(model)
    model = MaxPooling2D()(model)

    model = BatchNormalization(axis = 3)(model)
    model = Convolution2D(filters = 64, kernel_size = (3,3), strides = (1,1), padding = 'same', activation='relu')(model)
    model = MaxPooling2D()(model)

    model = BatchNormalization(axis = 3)(model)
    model = Convolution2D(filters = 128, kernel_size = (3,3), strides = (1,1), padding = 'same', activation='relu')(model)
    model = MaxPooling2D()(model)

    model = BatchNormalization(axis = 3)(model)
    model = Convolution2D(filters = 64, kernel_size = (3,3), strides = (1,1), padding = 'same', activation='relu')(model)
    model = GlobalAveragePooling2D()(model)

    model = Dense(2, activation = 'sigmoid')(model)

    model = Model(x,model)
```

The last 40 layers of the model are trained.

```
from keras.applications.mobilenet import MobileNet
mob_net= MobileNet(weights='imagenet', include_top=False, input_shape=(image_size, image_size,3))
for layer in mob_net.layers[:-40]:
    layer.trainable = False

 # Check the trainable status of the individual layers
for layer in mob_net.layers:
    print(layer, layer.trainable)
```

4.4 Results:

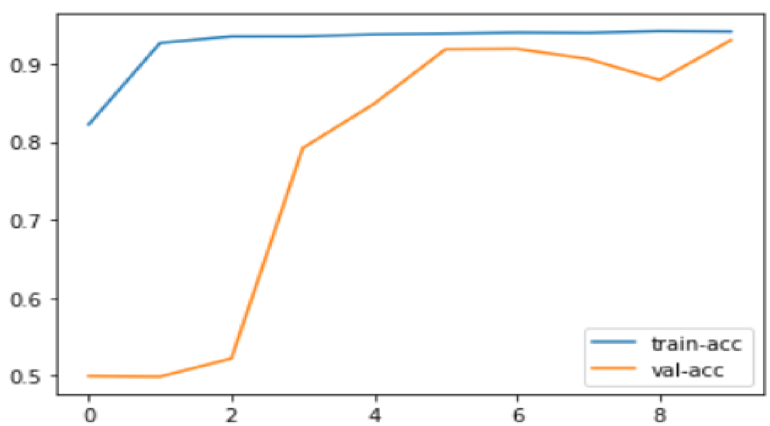
Table 4.1 : Comparison of Model's accuracy

Architecture	Training Accuracy(%)	Training loss	Validation Accuracy(%)	Validation Loss
VGG16	93.99	0.1751	90.64	0.2976
150CNN	94.17	0.1701	93.06	0.2387

So, 150CNN model turned out to be better than VGG16 in terms of accuracy and performance.

Accuracy of 150CNN : 93.06%

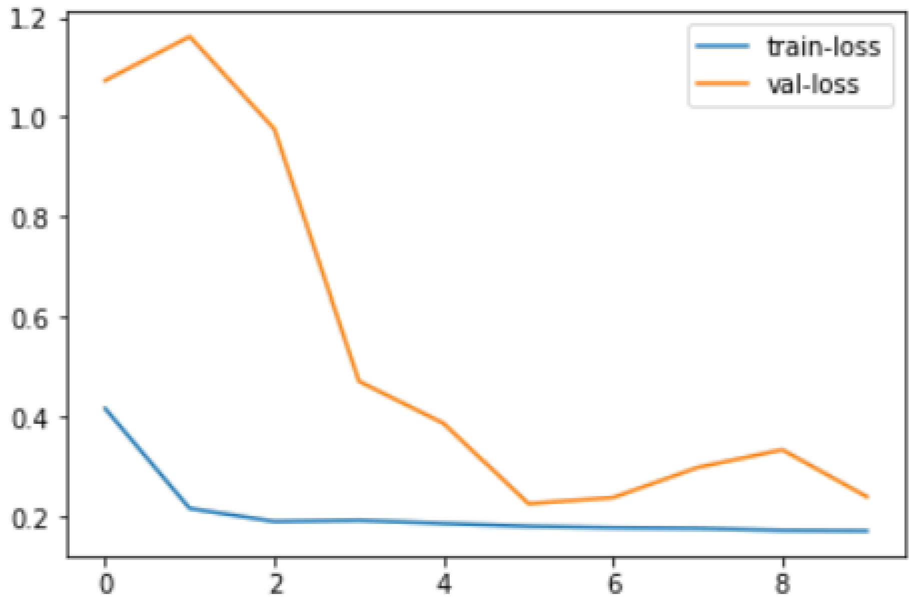
4.4.1 Accuracy Plot of 150CNN Model



x-axis	y-axis
Epoch	Accuracy

Figure 4.1 : It is the plot of accuracy vs epoch comparing training accuracy and validation accuracy of 150CNN.

4.4.2 Loss Plot of 150CNN Model



x-axis	y-axis
Epoch	Loss

Figure 4.2 : It is the plot of loss vs epoch comparing training loss and validation loss of 150CNN.

Chapter 5

Conclusions

5.1) Conclusion

In this project design and implementation of the deep neural networks, and learning are presented. We have used an approach and an algorithm to detect Malaria using Deep Learning. We have implemented an Artificial Neural Network and Convolution Neural Network used for the classification of the infected and uninfected images of blood samples. We have gone through various research papers published in this domain to get an idea as to what extent the work is done. We have tried our best to get an accurate result. The model that we have used gives 94.17% accuracy that outperforms existing model VGG16.

5.2) Future Work

Here, we have used MobileNet as the ANN model for our project. In future, we will be implementing our project using other ANN models such as Alexnet, Resnet etc to see how the model will react to the data sets and also how better would be the accuracy of the result. The solution we use is especially useful to find a single stage of parasitic malaria, usually a ring. However, the life cycle of malaria parasites is complex. The whole life cycle contains many of the morphological changes present in human blood. As malaria parasites lead to the formation of multi-stage forms with a variety of subdivisions present in the intraerythrocytic cycle, including trophozoites, rings, gametocytes, and schizonts. To date, an accurate multi-stage malaria detection program has not been available due to morphological changes in all parasites with multiple stages and diversity of images taken in various laboratories, specialists, clinics, and regions and we will try to use it. .

References

[1] Ministry of Health and Family Welfare, WHO World Malaria Report2020: "India continues to make Impressive Gains in reduction of Malaria Burden", 02 DEC 2020.

[2] Priyadarshini Adyasha Pattanaik, Mohit Mittal, Mohammad Zubair Khan, "Unsupervised Deep Learning CAD Scheme for the Detection of Malaria in Blood Smear Microscopic Images" IEEE, pp 94936 – 94946, 20 May 2020.

[3] K. M. Faizullah Fuhad, Jannat Ferdousey Tuba, Md. Rabiul Ali Sarker, Sifat Momen, Nabeel Mohammed, and Tanzilur Rahman "Deep Learning Based Automatic Malaria Parasite Detection from Blood Smear and Its Smartphone Based Application", 2020 May 20 .

[4] Mahdieh Poostchi, Kamolrat Silamut, Stefan Jaeger, George Thoma, "Image analysis and machine learning for detecting malaria" , 12 January 2018.

[5] Ministry of Health and Family Welfare, WHO World Malaria Report 2016: "Global Malaria Programme" 19 NOV 2016.

[6] Sumit Saha. A Comprehensive Guide to Convolutional Neural Networks. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53, 2018

[7] Andrej Karpathy. CS231n Convolutional Neural Networks for Visual Recognition. http://cs231n.github.io/convolutional-networks/, 2018.

[8] Dev Kumar Das, Madhumala Ghosh, Mallika Pal, Asok K. Maiti, and Chandan Chakraborty. Machine learning approach for automated screening of malaria parasite using light microscopic images. Micron, 45:97–106, feb 2013.

[9] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. Deep learning, may 2015.

[10] Wan Azani Mustafa, Ragunathan Santiagoo, Irfan Jamaluddin, Nurul S. Othman, Wan Khairunizam, and M. N.K.H. Rohani. Comparison of Detection Method on Malaria Cell Images. In 2018 International Conference on Computational Approach in Smart Systems Design and Applications, ICASSDA 2018. Institute of Electrical and Electronics Engineers Inc., sep 2018.