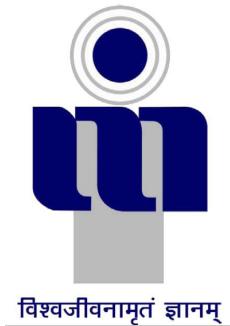


# **AI-POWERED MULTIPURPOSE WEB PLATFORM FOR MEDICAL IMAGE ANALYSIS**

*INTEGRATED POST GRADUATE  
Minor Project- 2020*

*by*

**Guramritpal Singh Saggū (2017IMT-036)**  
**Keshav Gupta (2017IMT-048)**  
**Devwrat Sahu (2017IMT-032)**  
**Shivam Maheshwary (2017IMT-078)**



**ABV INDIAN INSTITUTE OF INFORMATION  
TECHNOLOGY AND MANAGEMENT  
GWALIOR-474 015**

**2020**

## CANDIDATES DECLARATION

We hereby certify that the work, which is being presented in the report, entitled **AI-POWERED MULTIPURPOSE WEBPLATFORM FOR MEDICAL IMAGE ANALYSIS**, in partial fulfillment of the requirement for the award of the Degree of **Bachelor of Technology** and submitted to the institution is an authentic record of our own work carried out during the period *Jan 2020* to *May 2020* under the supervision of **Dr. Somesh Kumar**. We also cited the reference about the text(s)/figure(s)/table(s) from where they have been taken.

Date: \_\_\_\_\_ Signatures of the Candidates

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Date: \_\_\_\_\_ Signatures of the Research Supervisors

## ABSTRACT

This project's main aim is to provide assistance to medical practitioners and a reliable second opinion to people without medical background about diseases that can be detected from medical images or scans taken by medical professionals or people themselves. The diseases covered under this project are fatal if not detected in their early stages. Since people and professionals are not equipped with adequate resources so many cases concerning these may go undetected and possibly endanger lives or chronic ailments. The platform analyses these medical images and classifies them under various classes using the power of artificial intelligence. This app is designed to provide accurate, reliable and swift results. This app detects diseases such as Skin cancer from skin images, Tuberculosis from X-ray scans, COVID-19 from chest X-ray, Pneumonia from chest X-ray, Diabetes from retinopathy samples and Malaria from blood samples..

*Keywords:* Deep Convolution Neural Network, Transfer Learning, Machine Learning, Medical Images, Classification Report

## ACKNOWLEDGEMENTS

We are highly indebted to **Dr. Somesh Kumar**, and are obliged for giving us the autonomy of functioning and experimenting with ideas. We would like to take this opportunity to express our profound gratitude to them not only for their academic guidance but also for their personal interest in our project and constant support coupled with confidence boosting and motivating sessions which proved very fruitful and were instrumental in infusing self-assurance and trust within us. The nurturing and blossoming of the present work is mainly due to their valuable guidance, suggestions, astute judgment, constructive criticism and an eye for perfection. Our mentor always answered myriad of our doubts with smiling graciousness and prodigious patience, never letting us feel that we are novices by always lending an ear to our views, appreciating and improving them and by giving us a free hand in our project. It's only because of their overwhelming interest and helpful attitude, the present work has attained the stage it has.

Finally, we are grateful to our Institution and colleagues whose constant encouragement served to renew our spirit, refocus our attention and energy and helped us in carrying out this work.

(Guramritpal Singh Sagg)  
(Shivam Maheshwary)

(Keshav Gupta)  
(Devwrat Sahu)

# TABLE OF CONTENTS

<b>ABSTRACT</b>	<b>ii</b>
<b>LIST OF FIGURES</b>	<b>vi</b>
<b>1 INTRODUCTION AND LITERATURE SURVEY</b>	<b>1</b>
1.1 INTRODUCTION . . . . .	1
1.1.1 BACKGROUND . . . . .	1
1.1.2 MOTIVATION . . . . .	2
1.1.3 DEEP LEARNING FOR IMAGE CLASSIFICATION . . . . .	3
1.2 PYTHON . . . . .	4
1.3 OPTIMIZERS . . . . .	5
1.3.1 ADAM . . . . .	5
1.3.2 RMSPROP . . . . .	5
1.4 ACTIVATION FUNCTION . . . . .	6
1.4.1 RELU . . . . .	6
1.4.2 SOFTMAX . . . . .	7
1.4.3 SIGMOID . . . . .	7
1.5 LOSS FUNCTION . . . . .	7
1.5.1 BINARY CROSS-ENTROPY . . . . .	7
1.5.2 CATEGORICAL CROSS-ENTROPY . . . . .	7
1.6 TRANSFER LEARNING . . . . .	8
1.6.1 RESNET-50 . . . . .	8
1.6.2 INCEPTION-V3 . . . . .	9
1.6.3 VGG-16 . . . . .	10
<b>2 DESIGN DETAILS AND IMPLEMENTATION</b>	<b>11</b>
2.1 DATA FLOW OF THE PROJECT . . . . .	11
2.2 DEVELOPMENT OF THE CLASSIFIERS . . . . .	12
2.2.1 TUBERCULOSIS MODEL . . . . .	12
2.2.1.1 DATA COLLECTION AND PREPROCESSING . . . . .	12
2.2.1.2 MODEL DESCRIPTION . . . . .	14
2.2.1.3 LOSS AND ACCURACY CURVES . . . . .	14

2.2.2	SKIN CANCER RECOGNITION:MALIGNANT OR BENIGN	14
2.2.2.1	DATA COLLECTION AND PREPROCESSING . . . . .	14
2.2.2.2	MODEL DESCRIPTION . . . . .	15
2.2.2.3	LOSS AND ACCURACY CURVES . . . . .	16
2.2.2.4	PREDICTIONS . . . . .	16
2.2.3	MALARIA DETECTION MODEL . . . . .	16
2.2.3.1	DATA COLLECTION PREPROCESSING . . . . .	16
2.2.3.2	MODEL DESCRIPTION . . . . .	17
2.2.3.3	PREDICTIONS . . . . .	18
2.2.4	COVID-19 DETECTION:POSITIVE OR NORMAL . . . . .	18
2.2.4.1	DATA COLLECTION AND PREPROCESSING . . . . .	18
2.2.4.2	MODEL DESCRIPTION . . . . .	19
2.2.4.3	LOSS AND ACCURACY CURVES . . . . .	19
2.2.5	PNEUMONIA DETECTION MODEL . . . . .	20
2.2.5.1	DATA COLLECTION AND PREPROCESSING . . . . .	20
2.2.5.2	MODEL DESCRIPTION . . . . .	20
2.2.5.3	PREDICTIONS . . . . .	21
2.2.6	DIABETIC RETINOPATHY DETECTION . . . . .	21
2.2.6.1	DATA COLLECTION AND PREPROCESSING . . . . .	21
2.2.6.2	MODEL DESCRIPTION . . . . .	22
2.2.6.3	ATTENTION MAP . . . . .	22
2.2.6.4	PREDICTIONS . . . . .	23
2.3	WORKING OF THE WEB APPLICATION . . . . .	23
<b>3</b>	<b>RESULTS AND DISCUSSION</b>	<b>26</b>
3.1	ACCURACY . . . . .	26
3.2	EVALUATION METRICS . . . . .	26
3.2.1	PRECISION . . . . .	27
3.2.2	RECALL . . . . .	27
3.2.3	SUPPORT . . . . .	27
3.2.4	F1 SCORE . . . . .	27
3.2.5	CLASSIFICATION REPORT . . . . .	28
3.2.6	CONFUSION MATRIX . . . . .	29
3.2.7	AUC SCORE AND ROC SCORE . . . . .	30
<b>4</b>	<b>CONCLUSION AND FUTURE SCOPE</b>	<b>31</b>
4.1	CONCLUSION . . . . .	31
4.2	FUTURE SCOPE . . . . .	31
<b>REFERENCES</b>		<b>31</b>

# LIST OF FIGURES

1.1	A BASIC CONVOLUTIONAL NEURAL NETWORK[15] . . . . .	4
1.2	RELU FUNCTION [14] . . . . .	6
1.3	RESNET BLOCK ALSO KNOW AS SKIP-CONNECTION . . . . .	9
1.4	INCEPTION-V3 ARCHITECTURE [13] . . . . .	9
1.5	VGG-16 ARCHITECTURE [12] . . . . .	10
2.1	DATA FLOW DIAGRAM . . . . .	12
2.2	ORIGINAL IMAGES[9] . . . . .	13
2.3	AUGMENTED IMAGES . . . . .	13
2.4	MODEL LOSS . . . . .	14
2.5	MODEL ACCURACY . . . . .	14
2.6	MODEL LOSS . . . . .	16
2.7	MODEL ACCURACY . . . . .	16
2.8	PREDICTIONS . . . . .	16
2.9	PREDICTIONS . . . . .	18
2.10	MODEL LOSS AND ACCURACY . . . . .	19
2.11	PREDICTIONS . . . . .	21
2.12	ATTENTION MAP. . . . .	22
2.13	PREDICTIONS . . . . .	23
2.14	HOME PAGE . . . . .	23
2.15	SERVICES . . . . .	24
2.16	PREDICTING FOR TUBERCULOSIS . . . . .	24
2.17	RESULTS . . . . .	25
3.1	ACCURACY TABLE FOR ALL CLASSIFIERS . . . . .	26
3.2	TUBERCULOSIS DETECTION MODEL REPORT . . . . .	28
3.3	SKIN-CANCER DETECTION MODEL REPORT . . . . .	28
3.4	MALARIA DETECTION MODEL REPORT . . . . .	28
3.5	COVID-19 DETECTION MODEL REPORT . . . . .	28
3.6	PNEUMONIA DETECTION MODEL REPORT . . . . .	29
3.7	DIABETIC RETINOPATHY DETECTION MODEL REPORT . . . . .	29
3.8	TUBERCULOSIS DETECTION CONFUSION MATRIX . . . . .	29

3.9 SKIN CANCER DETECTION CONFUSION MATRIX . . . . .	30
3.10 ROC CURVE FOR DIABETIC RETINOPATHY MODEL . . . . .	30

## **ABBREVIATIONS**

CNN	Convolutional Neural Network
GUI	Graphical User Interface
ReLU	Rectified Linear Unit
FC	Fully Connected
C	Channels
H	Height
W	Width
1D	One Dimensional
2D	Two Dimensional
VGG	Visual Geometry Group
RESNET	Residual Network
Covid	Coronavirus disease
DR	Diabetic Retinopathy
ML	Machine Learning
DL	Deep Learning
CT	Computed tomography
AUC	Area under the curve
ROC	Receiver operating characteristic

# CHAPTER 1

## INTRODUCTION AND LITERATURE SURVEY

This chapter includes the details of the background, our problem statement, the motivation and the objectives of our thesis. In this section we briefly describe our project that aims to automate the process of diagnosis and analysis of various diseases in order to detect their existence from respective medical images.

### 1.1 INTRODUCTION

The process of detection of **Tuberculosis, Skin Cancer, Malaria, Covid-19, Pneumonia and Diabetic Retinopathy** is a very complicated process. It starts with collecting Chest X-Ray sample, Skin Images, Blood sample and Retina images from potentially infected person. These samples are observed under microscope or by an expert to detect the presence of respective disease. Our aim is to remove the need of human expertise and thus make the process faster and more reliable as the bias in decision making is removed which gives more accurate results.

#### 1.1.1 BACKGROUND

- The diagnosis of these diseases being done accurately heavily depends on human expertise and can be impacted by the variability in observation and the biases of the expert.
- A wide variety of existing softwares that automate the process of diagnosis and analysis of images use Machine Learning techniques with the features extracted by hand for making decisions. However, this process stated above required human expertise in order to precisely retrieve the features in order to apply machine learning techniques on them.

- But in recent times some advancement have been made in the field of image processing to automatically extract complex features and build classification models based upon them.
- Currently, machine learning practitioners across the globe have started to apply Deep Learning in the field of health-care and diagnosis to obtain better results in a several associated problems.

### 1.1.2 MOTIVATION

The motivation for our project is based on the nature and fatality of the disease.

- Tuberculosis is a disease that affects many people in developing countries. While treatment is possible, it requires an accurate diagnosis first. In these countries projects there are in many cases available X-ray machines, but often the radiological expertise is missing for accurately assessing the images. An algorithm that could perform this task quickly and cheaply could drastically improve the ability to diagnose and ultimately treat the disease.
- Diabetic retinopathy is the leading cause of blindness in the working-age population of the developed world. It is estimated to affect over 93 million people.
- Statistics of World Health Organization (WHO) state that there are over 200 million malaria cases and there are approximately 400,000 deaths due to malaria every year. Due to this we need a fast, easy and effective way to detect malaria and diagnose the disease.
- The COVID-19 pandemic continues to have a devastating effect on the health and well-being of the global population. A critical step in the fight against COVID-19 is effective screening of infected patients, with one of the key screening approaches being radiology examination using chest radiography.
- The recent growth in the field of deep learning has helped in creating better classification models. These models work well with large training data and we don't need to do feature extraction by ourselves.
- Every 2 minute a child dies of malaria, 15% of all deaths of children under five years old is caused by pneumonia
- Skin cancer is the most common human malignancy, is primarily diagnosed visually, beginning with an initial clinical screening and followed potentially by dermoscopic analysis, a biopsy and histopathological examination. Automated classification of skin lesions using images is a challenging task owing to the fine-grained variability in the appearance of skin lesions.

### 1.1.3 DEEP LEARNING FOR IMAGE CLASSIFICATION

We have used deep learning for classifying image into classes. The deep learning binary classifier is trained on the images of these slides in order to detect various diseases. A new image can be run on the trained classifier and result is obtained.

A convolutional neural network [1] is type of deep neural network that is used for image processing tasks in deep learning. It takes in input in the form of image and assigns weights and biases to parameters and help tell the difference between images. These networks require very low amount of pre-processing. There are multiple types of layers used in CNNs, some of which are listed below-

- **Input Layer** - This layer consists the dimensions similar to the dimensions of input image provided. All the images have been re-sized to the same dimension to create consistency in the data
- **Hidden Layers** - These layers takes input from the previous layer and produces more complex output features. These are produced by applying a convolution filter on the previous layer.
- **Output Layer** - The layer uses the output of the previous hidden layer as input and produces 2 outputs in the form of probabilities corresponding to each class. The class with the higher probability is considered to be the correct answer
- **Activation Function** - The final output layer of our model makes use of **Sigmoid Activation Function** in order to make binary classification. The intermediate outputs of our hidden layers are feed to **ReLU Activation Function**.

The figure 1.1 shows the basic operations carried out in the CNN for image processing. The input is an image and the output is a probability vector which shows the probability corresponding to each class. The class with the maximum probability is the required answer. The operations used in the network are listed below-

- **Convolution** - This operation uses a filter/kernel of a dimension smaller than input image dimension. The filter is placed on the top left corner of the image and the values in the pixel matrix and the filter which overlap are multiplied and added to give the first value of the next feature map. The filter is then slided forward and operation is repeated. When the filter reaches the rightmost position, it is moved one position below and the operation is repeated from left to right. Considering the input of size of image be  $M \times N$  and filter size be  $5 \times 5$ , the output matrix is of size  $(M-5+1) \times (N-5+1)$ . Many such filters are used and the resultant matrices are stacked together to give output in second image with  $n_1$  channels( $n_1$  filters are used). It can be seen that input image is  $28 \times 28$  size and  $n_1$  filters of size  $5 \times 5$  are used which give output of size  $24 \times 24 \times n_1$ .

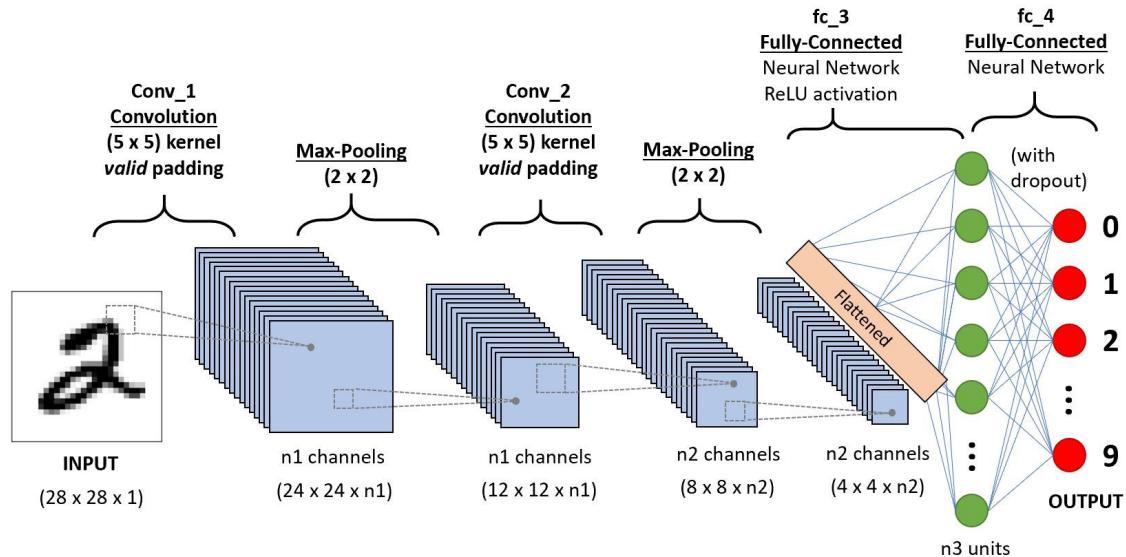


Figure 1.1: A BASIC CONVOLUTIONAL NEURAL NETWORK[15]

- **Max Pooling** - This operating uses a filter( $2 \times 2$  in the figure) which is滑动 over the feature map with a stride of 2 and we take maximum of each  $2 \times 2$  matrix. Therefore the dimensions are reduced by half.
- **Fully Connected** - The fully connected layer is a typical dense neural network layer which is obtained from previous layer by using a parameter matrix. Considering previous layer of size ' $M$ ' and output layer of size ' $N$ ', the parameter matrix has dimensions ' $M \times N$ '.

## 1.2 PYTHON

We have used the PYTHON to implement our project. PYTHON is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, it was released in the year 1991 and designed by Guido van Rossum. Python considers whitespace significant and it emphasizes code readability by design. We have used deep learning and machine learning libraries for the development of the model used in classification. These libraries include Keras, Scikit Learn, Matplotlib, pandas and numpy for deep learning part to develop the classifier. We use Django which is python based web framework used to develop the web application to make the process of process of malaria detection easily accessible to the general public.

### Key Features

- GUIs for building deep learning model and viewing and analyzing results.
- Integrated libraries for creating fuzzy inference systems.

- High community support for supporting and helping others with their projects.
  - Built in framework for web development.
- Version - PYTHON (3.6.4)

## 1.3 OPTIMIZERS

### 1.3.1 ADAM

Categorical cross entropy also known as log loss is used to measure the performance of the classifier and its output is always a value between 0 and 1. Closer the value is to zero, lesser is the loss and hence better is the performance and vice versa [6].

$$\hat{m}_t = m_t / (1 - \beta_1^t)$$

$$\hat{v}_t = v_t / (1 - \beta_2^t)$$

$$w_t = w_{t-1} - \eta \times [\hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)]$$

Here,  $m$  and  $v$  are estimators of first and second moments. The reason for using Adam optimizer is that it handles noisy gradients very well and it is easy to implement.

### 1.3.2 RMSPROP

RmsProp is an optimizer that utilizes the magnitude of recent gradients to normalize the gradients. We always keep a moving average over the root mean squared (hence Rms) gradients, by which we divide the current gradient. Let  $f'(\theta_t)$  be the derivative of the loss with respect to the parameters at time step  $t$ . In its basic form, given a step rate  $\alpha$  and a decay term  $\gamma$

we perform the following updates:

$$\begin{aligned} r_t &= (1 - \gamma) f'(\theta_t)^2 + \gamma r_{t-1}, \\ v_{t+1} &= \frac{\alpha}{\sqrt{r_t}} f'(\theta_t), \\ \theta_{t+1} &= \theta_t - v_{t+1}. \end{aligned}$$

In some cases, adding a momentum term  $\beta$  is beneficial. Here, Nesterov momentum is used:

Additionally, this implementation has adaptable step rates. As soon as the components of the step and the momentum point into the same direction (thus have the same

$$\begin{aligned}\theta_{t+\frac{1}{2}} &= \theta_t - \beta v_t, \\ r_t &= (1 - \gamma) f'(\theta_{t+\frac{1}{2}})^2 + \gamma r_{t-1}, \\ v_{t+1} &= \beta v_t + \frac{\alpha}{\sqrt{r_t}} f'(\theta_{t+\frac{1}{2}}), \\ \theta_{t+1} &= \theta_t - v_{t+1}\end{aligned}$$

sign) the step rate for that parameter is multiplied with  $1 + \text{stepadapt}$ . Otherwise, it is multiplied with  $1 - \text{stepadapt}$ . In any way, the minimum and maximum step rates `stepratemin` and `stepratemax` are respected and exceeding values truncated to it.

RmsProp has several advantages; for one, it is a very robust optimizer which has pseudo curvature information. Additionally, it can deal with stochastic objectives very nicely, making it applicable to mini batch learning.

## 1.4 ACTIVATION FUNCTION

### 1.4.1 RELU

ReLU stands for rectified linear unit, and is a type of activation function. Mathematically, it is defined as  $y = \max(0, x)$ . ReLU is the most commonly used activation function in neural networks, especially in CNNs. Visually, it looks like the following:

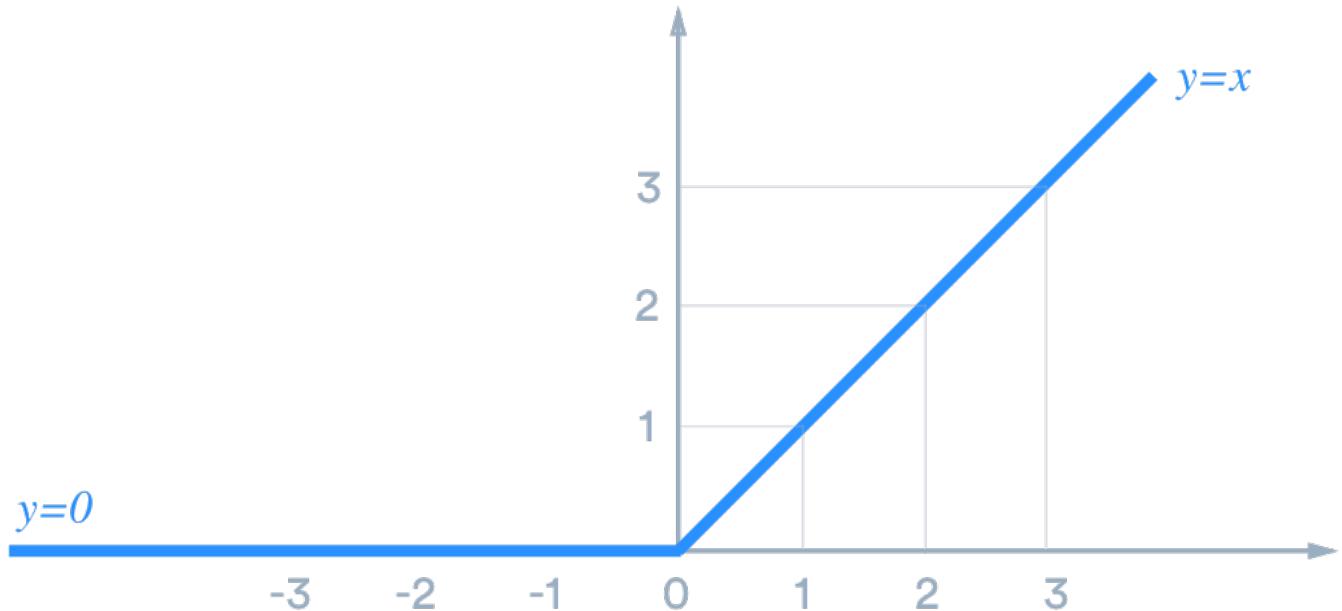


Figure 1.2: RELU FUNCTION [14]

### 1.4.2 SOFTMAX

In mathematics, the softmax function, also known as softargmax or normalized exponential function, is a function that takes as input a vector  $z$  of  $K$  real numbers, and normalizes it into a probability distribution consisting of  $K$  probabilities proportional to the exponentials of the input numbers.

### 1.4.3 SIGMOID

The Sigmoid Function curve looks like a S-shape. The main reason why we use sigmoid function is because it exists between (0 to 1). Therefore, it is especially used for models where we have to predict the probability as an output. Since probability of anything exists only between the range of 0 and 1, sigmoid is the right choice.

## 1.5 LOSS FUNCTION

### 1.5.1 BINARY CROSS-ENTROPY

The Cross-Entropy Loss is actually the only loss we are discussing here. The other losses names written in the title are other names or variations of it. The CE Loss is defined as:

$$CE = - \sum_i^C t_i \log(s_i)$$

Where  $t_i$  and  $s_i$  are the groundtruth and the CNN score for each class  $i$  in  $C$ . As usually an activation function (Sigmoid / Softmax) is applied to the scores before the CE Loss computation, we write  $f(s_i)$  to refer to the activations.

In a binary classification problem, where  $C'=2$ , the Cross Entropy Loss can be defined also as

$$CE = - \sum_{i=1}^{C'=2} t_i \log(s_i) = -t_1 \log(s_1) - (1 - t_1) \log(1 - s_1)$$

### 1.5.2 CATEGORICAL CROSS-ENTROPY

Categorical cross entropy also known as log loss is used to measure the performance of the classifier and its output is always a value between 0 and 1. Closer the value is to zero, lesser is the loss and hence better is the performance and vice versa [4].

$$L(y, \hat{y}) = -1/N \sum_i^N [y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)]$$

## 1.6 TRANSFER LEARNING

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. It is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks given the vast compute and time resources required to develop neural network models on these problems and from the huge jumps in skill that they provide on related problems.

A pre-trained model has been previously trained on a dataset and contains the weights and biases that represent the features of whichever dataset it was trained on. Learned features are often transferable to different data. For example, a model trained on a large dataset of bird images will contain learned features like edges or horizontal lines that you would be transferable your dataset.

Pre-trained models are beneficial to us for many reasons. By using a pre-trained model you are saving time. Someone else has already spent the time and compute resources to learn a lot of features and your model will likely benefit from it.

### 1.6.1 RESNET-50

ResNet-50 [2]is a deep residual network. The “50” refers to the number of layers it has. It’s a subclass of convolutional neural networks, with ResNet most popularly used for image classification.

The main innovation of ResNet is the skip connection. As you know, without adjustments, deep networks often suffer from vanishing gradients, ie: as the model back-propagates, the gradient gets smaller and smaller. Tiny gradients can make learning intractable.

The skip connection in the diagram below is labeled “identity.” It allows the network to learn the identity function, which allows it pass the the input through the block without passing through the other weight layers.

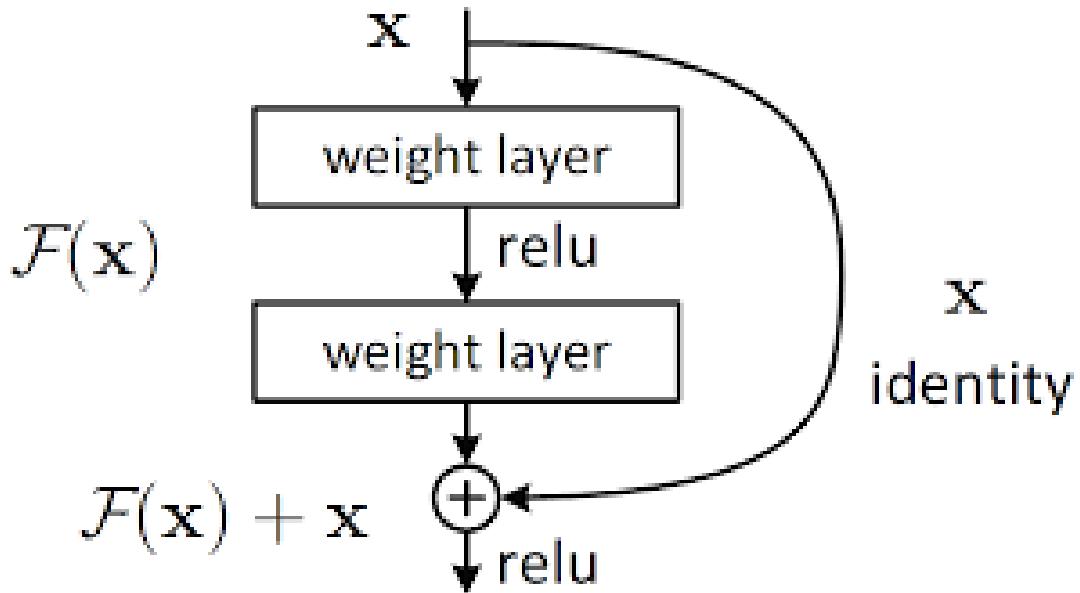


Figure 1.3: RESNET BLOCK ALSO KNOW AS SKIP-CONNECTION

### 1.6.2 INCEPTION-V3

Inception V3 [3] by Google is the 3rd version in a series of Deep Learning Convolutional Architectures. Inceptionv3 is a convolutional neural network for assisting in image analysis and object detection, and got its start as a module for Googlenet. It is the third edition of Google's Inception Convolutional Neural Network, originally introduced during the ImageNet Recognition Challenge.

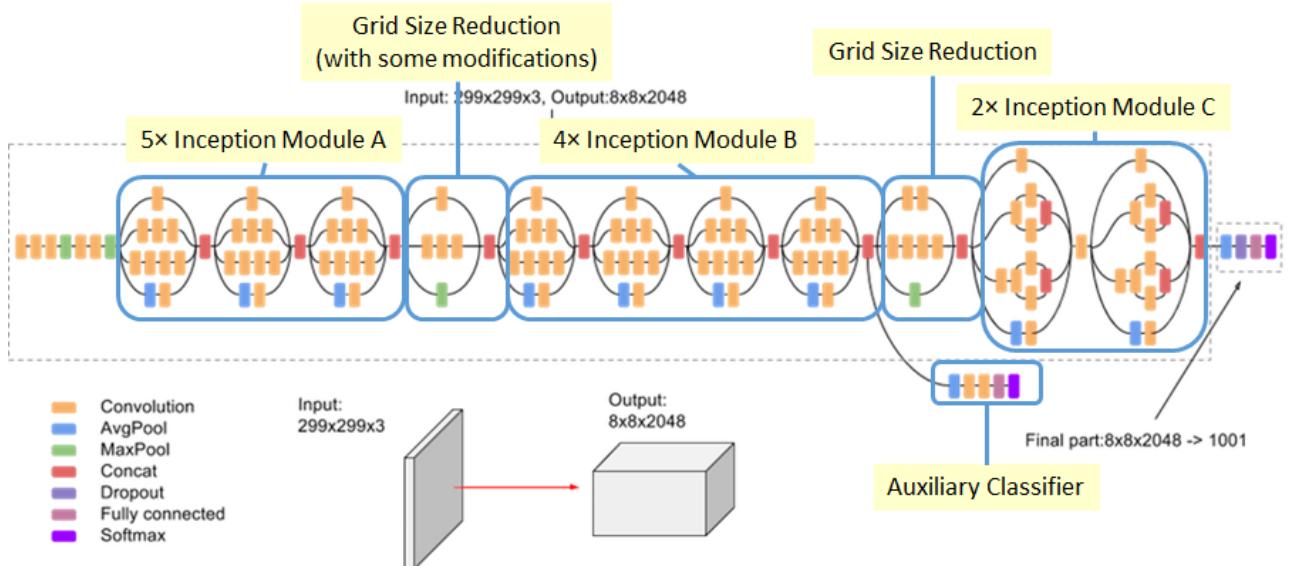


Figure 1.4: INCEPTION-V3 ARCHITECTURE [13]

### 1.6.3 VGG-16

VGG16 [5] is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper “Very Deep Convolutional Networks for Large-Scale Image Recognition”. The model achieves 92.7 percent top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. It was one of the famous model submitted to ILSVRC-2014. It makes the improvement over AlexNet by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple 3x3 kernel-sized filters one after another.

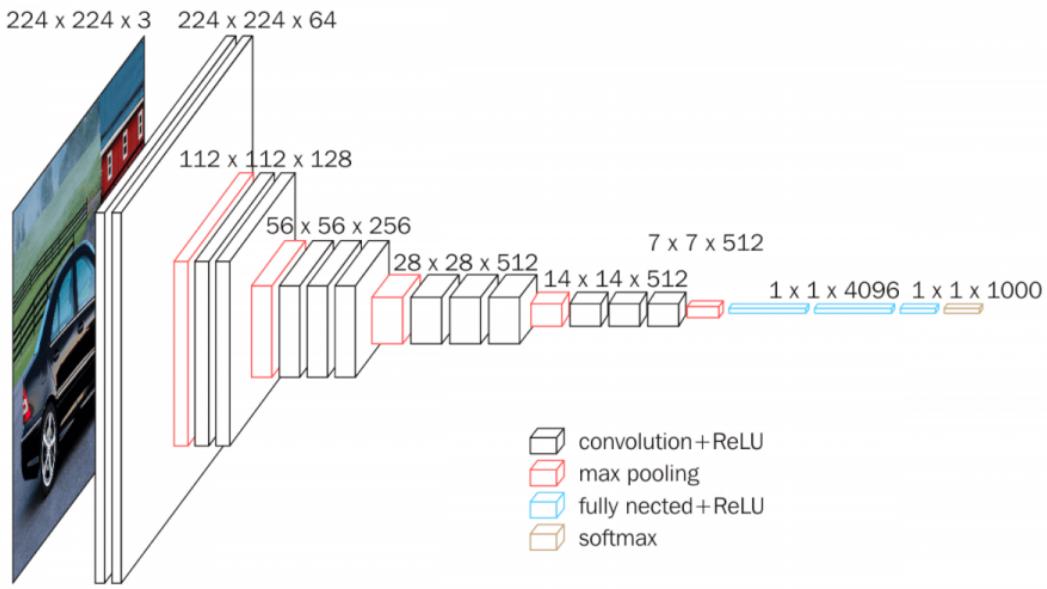


Figure 1.5: VGG-16 ARCHITECTURE [12]

# **CHAPTER 2**

## **DESIGN DETAILS AND IMPLEMENTATION**

The image classifier models will be running at the backend in website. We will discuss both working of website and image classifiers in this section.

### **2.1 DATA FLOW OF THE PROJECT**

The figure 2.1 below depicts the working our web application. Our system can be broadly divided into 3 sections- Frontend, Backend and CNN image classifiers. The application contains a user friendly landing page at the front end where the user is required to select the respective disease they want to detect. Then the user is required to upload the medical image of respective disease sample. The image submitted by the user is received at the backend and the presence of various diseases is detected using a pre-trained CNN based classifier. The model process the image and produces the result which are sent to the front-end application. Results are displayed on the web page with their respective class and probabilities.

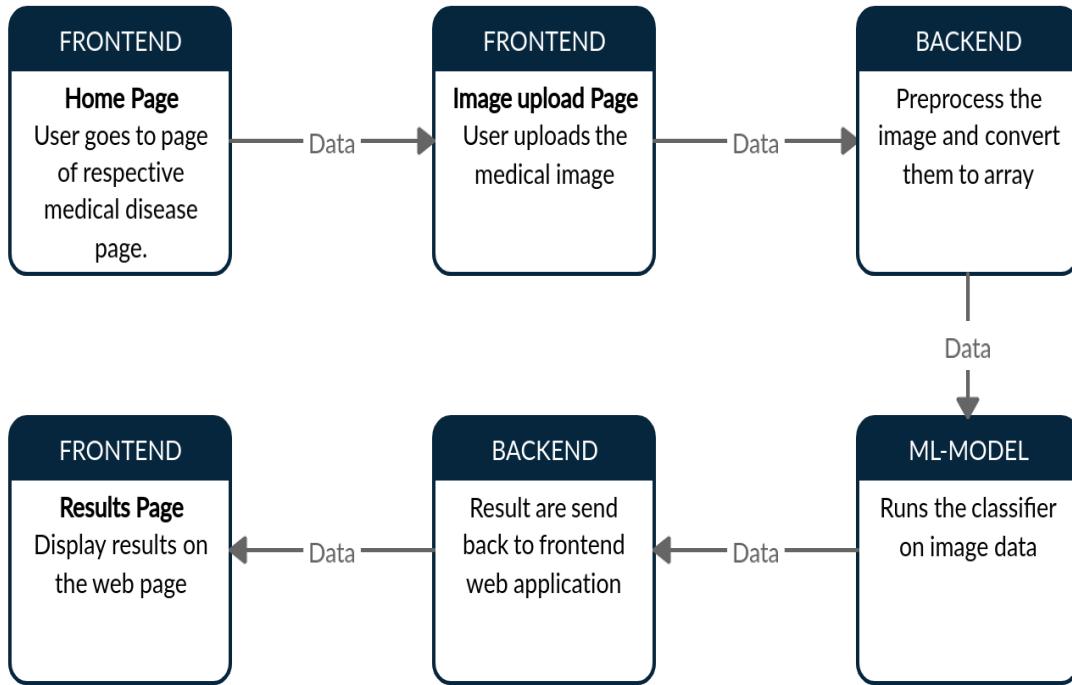


Figure 2.1: DATA FLOW DIAGRAM

## 2.2 DEVELOPMENT OF THE CLASSIFIERS

### 2.2.1 TUBERCULOSIS MODEL

#### 2.2.1.1 DATA COLLECTION AND PREPROCESSING

The model takes a look at a chest x-ray and predicts whether a person has TB or not. The model will be trained on a dataset[9][10][11] of 800 images from two sources:

Shenzhen, China

Montgomery, USA

Both the datasets contain almost an equal amount of cases of tuberculosis and normal images. The Shenzhen dataset contains images of 336 tuberculosis cases and 326 normal cases, and the Montgomery dataset has 80 examples of normal cases and 56 examples of tuberculosis.

After assigning the labels we combine the two dataframes and shuffle them. We need a training set to train the model on and a testing set to test its accuracy. So in this case we create a folder for containing training set and a folder for containing test set. Each folder further has 2 folders which would contain images of tuberculosis and normal images. Because we need to use as many images as possible for training, the test set will contain only 120 images. This is 15 percent of the data. First we read images using open-cv's imread functionality, then the read images are resized to size of 96x96x3. Then the read images are saved into either training or test set.

Since the dataset is quite small we will use data augmentation in order to increase its perceived size and model's generality. Now we copy images from the training set into a temporary folder. We use the ImageDataGenerator of keras to generate augmented images from images stored in temporary folder and then the generated augmented images are stored with original images in the training folder.

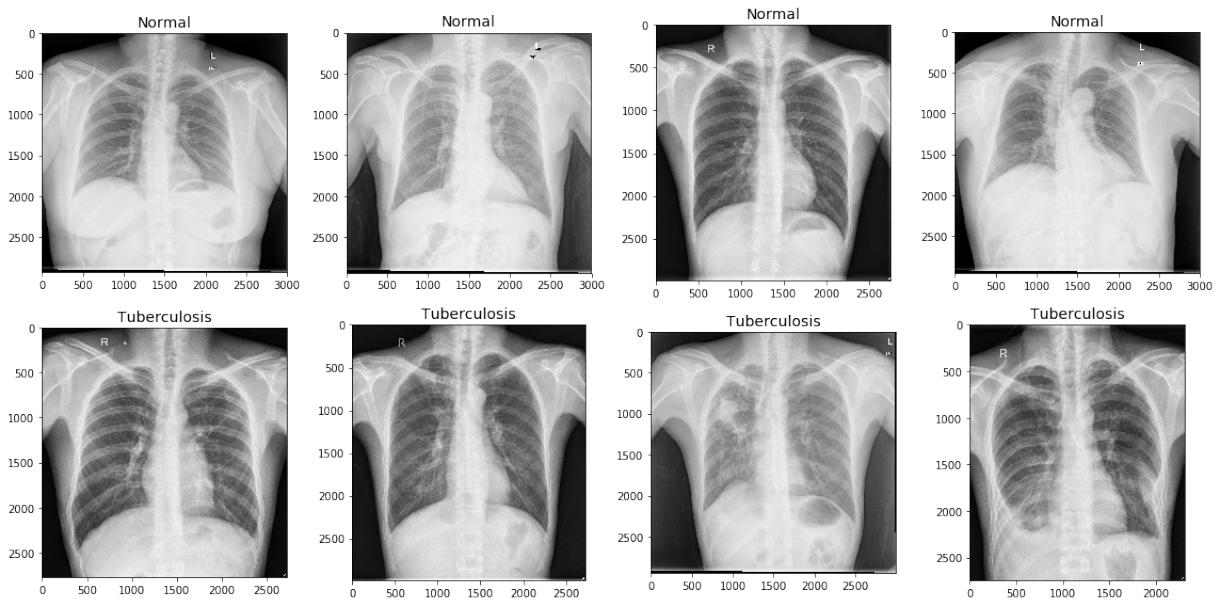


Figure 2.2: ORIGINAL IMAGES[9]

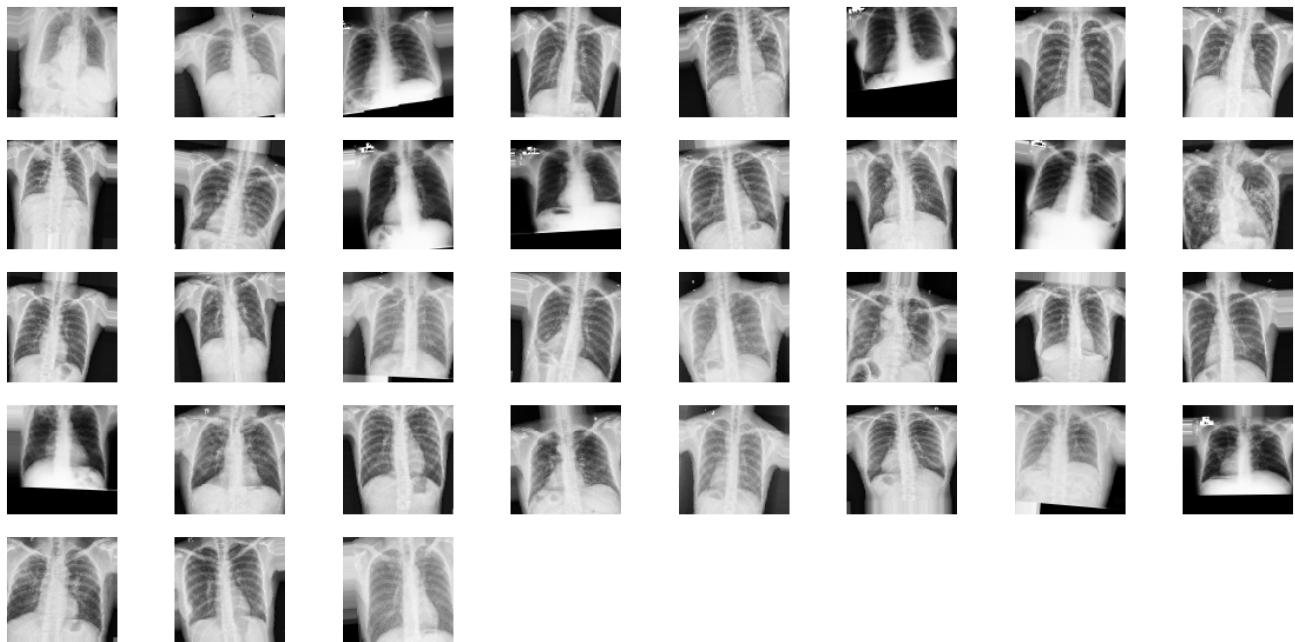


Figure 2.3: AUGMENTED IMAGES

### 2.2.1.2 MODEL DESCRIPTION

The neural network has Convolutional layers, Max Pooling layers, Dropout layers and fully connected layers. The Architecture used consists of 3 blocks, each having three Convolution layers followed by a max pooling layer and a Dropout layer. The number of filters for all convolution layers belonging to particular block is same. The first block's convolution layers have 32 filters, second have 64 and the third have 128 filters. The last block is followed by a Flatten layer and then we have 2 fully connected layers with 256 units and 2 units respectively. Dropout Layer is applied between the first fully connected layer and the second fully connected layer. The last layer is the output layer which predicts if the given image of chest X-ray belongs to tuberculosis class or the normal class. In both the Convolutional layers and fully connected layers the activation function being used is ReLU activation function except in case of last fully connected layer which makes use of softmax activation function.

To train the neural network we used the Adam Optimiser and binary crossentropy as loss function.

### 2.2.1.3 LOSS AND ACCURACY CURVES

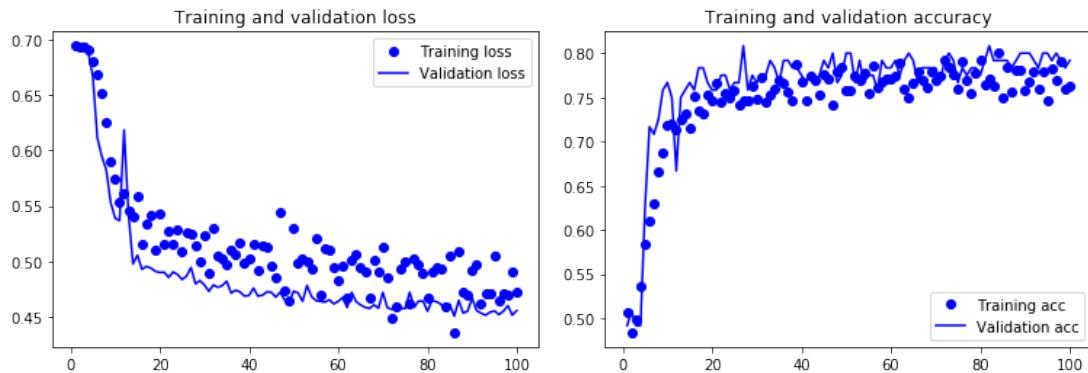


Figure 2.4: MODEL LOSS

Figure 2.5: MODEL ACCURACY

## 2.2.2 SKIN CANCER RECOGNITION: MALIGNANT OR BENIGN

### 2.2.2.1 DATA COLLECTION AND PREPROCESSING

The dataset[16] is taken from the ISIC (International Skin Image Collaboration) Archive. It consists of 1800 pictures of benign moles and 1497 pictures of malignant classified moles. The pictures have all been resized to low resolution (224x224x3) RGB. The task of this kernel is to create a model, which can classify a mole visually into benign and malignant.

The images are loaded and then turned into numpy arrays using the images RGB value. The pictures have been resized to a square, 224x224, and then the images are

given labels as there have been no labels created for the images. Once the labels are created and the images are turned into numpy arrays, the images are then added to a training set and shuffled. We created categorical labels for the training and testing data. Then normalized all the values of the images by diving the RGB values by 255. By normalizing the images, we are changing the range of pixel's intensity values.

### 2.2.2.2 MODEL DESCRIPTION

ResNet50 is short for Residual Networks, and it is essentially a classic neural network used as a backbone for many computer vision tasks, like the one we are doing. The reason we use a ResNet50 is that it allows us to train deep neural networks successfully without having the problem of vanishing gradients.

We added one layer at a time from the starting input as a Keras Sequential API was used. The first layer is a convolutional layer, it is a set of learnable filters. For the first two convolutional layers, the number of filters was set to 64. Each filter has a role to transform a part of the image using the kernel size. The kernel filter matrix is applied to the whole image, and the filters can be visualized as a transformation of the image. The second layer in the CNN is the pooling layer, and what this layer does is it looks at the two neighboring pixels and picks the maximal value. These are used for two reasons, one is to reduce the computational cost and the second is to reduce overfitting. By combining the two types of layers we used, the convolutional layer and the pooling layer, CNN is able to combine the local features and learn more global features of the image. Local features describe image patches around interest points, and global features describe an image as a single vector. We used dropout which is a regularization method, where a proportion of nodes are randomly ignored for each training sample. The proportion of nodes is randomly dropped forcing the network to learn features in a distributed way. This method is used to improve generalization and prevent overfitting. Relu which is also used to add non-linearity to the network. The flattened layer of the network is the layer that converts the final feature and maps it into a single 1D vector. This layer combines all the local features with the previous convolutional layers. In the end, I used the features in one fully connected layer which works as classifier.

We cross validated the model and the goal of this is to test the model's ability to predict new data and to flag problems such as overfitting or selection bias. This will also give us an insight as to how the model will generalize to an independent dataset

### 2.2.2.3 LOSS AND ACCURACY CURVES

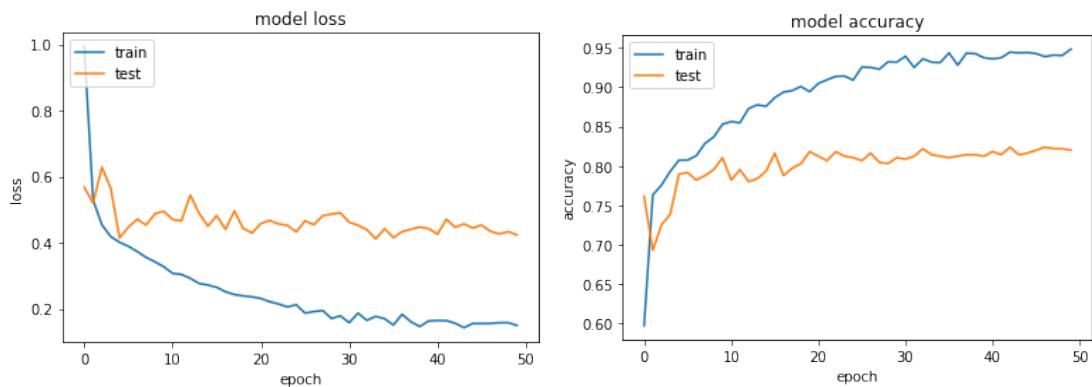


Figure 2.6: MODEL LOSS

Figure 2.7: MODEL ACCURACY

### 2.2.2.4 PREDICTIONS

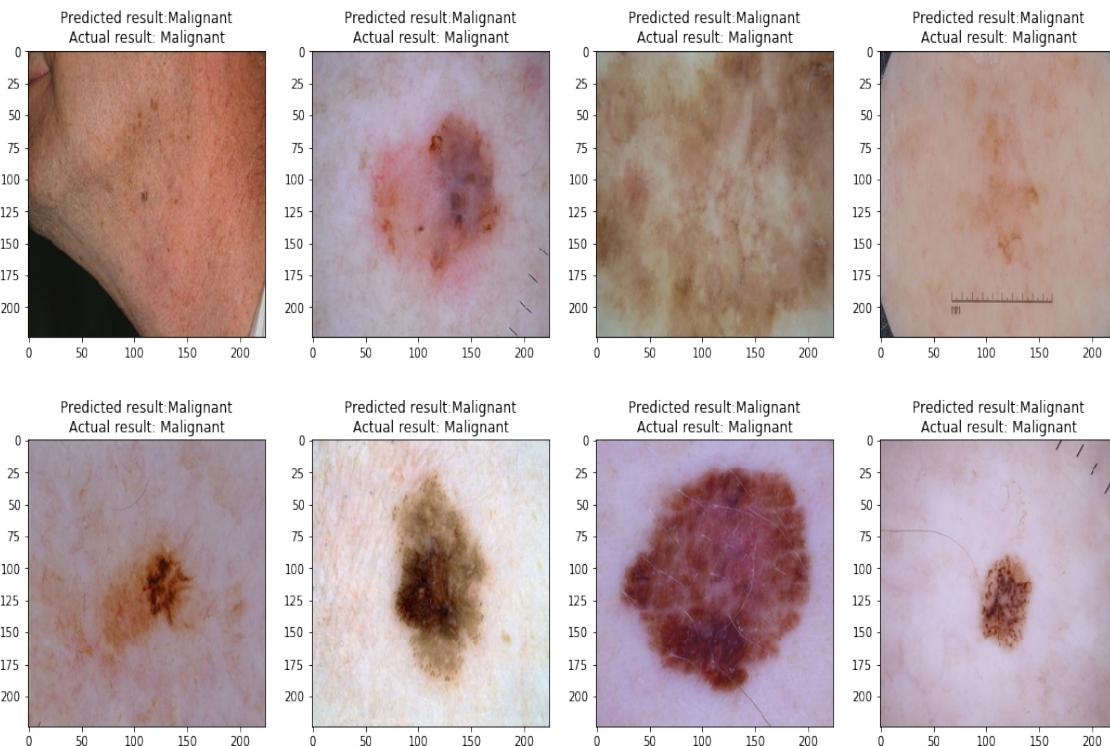


Figure 2.8: PREDICTIONS

## 2.2.3 MALARIA DETECTION MODEL

### 2.2.3.1 DATA COLLECTION PREPROCESSING

The dataset [8] is taken from the NIH (National Institute of Health) Website. It hosts a repository of segmented cells from the thin blood smear slide images from the Malaria

Screener research activity. The dataset contains a total of 27,558 cell images with equal instances of parasitized and uninfected cells. All the images are then resized to the resolution of 125x125x3 RGB. The task of this kernel is to create a model, which can look at the blood smeared slide image and predict with high accuracy if the image contains blood slide which is parasitic or not.

We need a training set to train the model on and a testing to tests its accuracy and they are created using open-cv which reads the images from storage and converts them into a three dimensional array where the first dimension corresponds to the height of the array and second dimension corresponds to the widths of the array and third dimension corresponds to the channels of the array. Open-cv is then used to resize the read images into the desired size which in this case is 125x125x3. After resizing, a kernel of size 3x3 is run over the images, following which they are converted into YUV color space and histogram equalization is performed in the luminous channel and then the images are converted back in to RGB color space. The images are then loaded into the set along with their labels (whether they are parasitic or normal). Once the set consists of all the images along with their labels, it is split into 2 parts training and test set and this is done using sklearn's train test split module.

### 2.2.3.2 MODEL DESCRIPTION

Since we felt that number of images in the dataset was enough to train a relatively simple neural network from scratch. We decided to create custom neural network architecture instead of fine-tuning the pre-trained models or making changes to the pre-trained models.

The neural network has Convolutional layers, Max and Average Pooling layers, Dropout layers and fully connected layers. The used neural network consists of 5 Convolutional layers with 16, 32, 64, 128, 256 filters respectively. Second and Fourth have Max Pool layers and Dropout layers following them. After last layer we have Average Pool layer which is followed by Flatten layer and then we have fully connected layers. In the used architecture we make use of 4 fully connected layers with 120, 60, 10, 1 units respectively. The last layer is the output layer which predicts if the given image of blood smear is parasitic or not. In both the Convolutional layers and fully connected layers the activation function being used is ReLU activation function except in case of last fully connected layer which makes use of sigmoid activation function.

To train the neural network we used the Adam Optimizer and binary crossentropy as a loss function.

### 2.2.3.3 PREDICTIONS

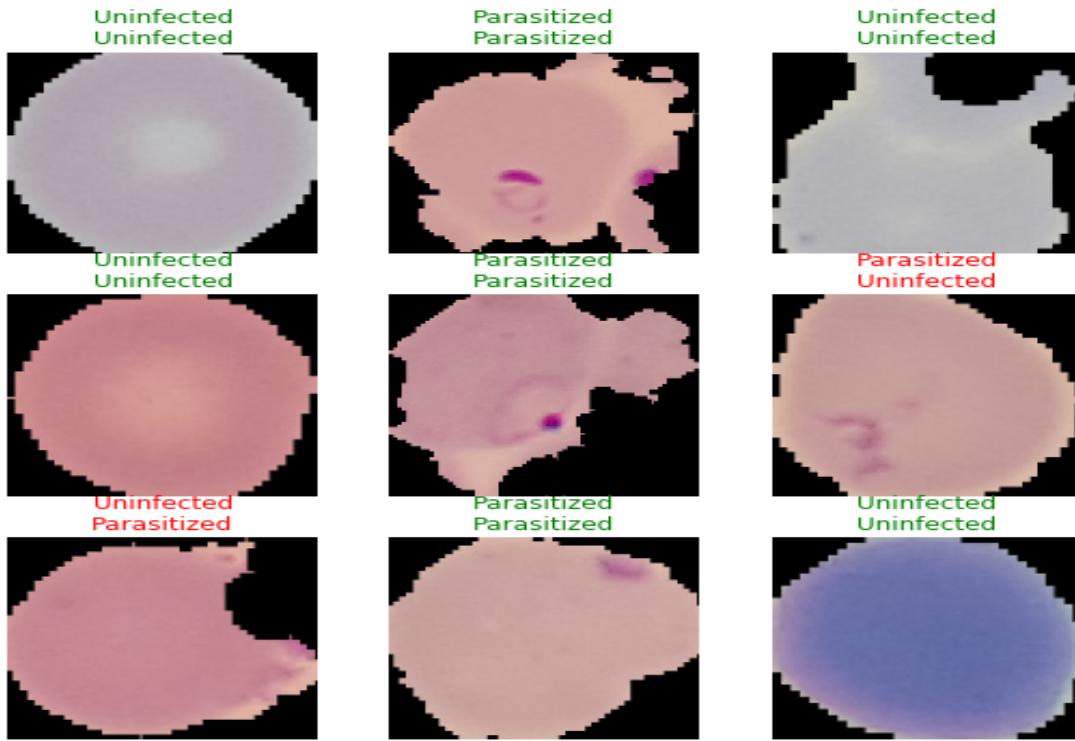


Figure 2.9: PREDICTIONS

### 2.2.4 COVID-19 DETECTION:POSITIVE OR NORMAL

#### 2.2.4.1 DATA COLLECTION AND PREPROCESSING

The data[18] we need for this type of problem is chest X-Ray for both COVID affected and fit patients. There is no direct link to the dataset but we can make-shift to get the data and start the operation. Dr.Joseph Paul Cohen recently open-sourced a database containing chest X-ray images of patients suffering from the COVID-19 disease. The dataset used is an open-source dataset which consists of COVID-19 images from publicly available research, as well as lung images with different pneumonia-causing diseases such as SARS, Streptococcus, and Pneumocystis. We have also used the Kaggle's Chest X-ray competitions dataset to extract X-rays of healthy patients and have sampled 100 images to have a balance with the COVID-19 available images. So, the dataset consists of COVID-19 X-ray scan images and also the angle when the scan is taken. It turns out that the most frequently used view is the Posteroanterior view and we have considered the COVID-19 PA view X-ray scans for my analysis. To stratify our data we will take an equal number of images and will blend them and later will divide into test and train data.

### 2.2.4.2 MODEL DESCRIPTION

We will create a deep learning model that is going to learn the difference between normal X-Ray and COVID-19 affected X-Ray and later can predict. We have used transfer learning with the VGG-16 model and have fine-tuned the last few layers. We tend to have 3 hidden layers, you can experiment with more or fewer layers that is up to you. We are going to follow the traditional approach of increasing the neurons as we go deep inside the layer; as it helps to learn more features from the image which returns us better certainty. We have (224,224,3) input neurons that are we are resizing our data to 224\*224 with 3 channels as it is considered to be the ideal size and therefore our model can grasp even minutiae and necessary features from the image. At last, we going to flatten our features and will use sigmoid as activation function as we are having binary classification problem, and thus our output will only contain one cell, adam as optimizer works well with sigmoid hence compiling the model with them in addition to cross binary entropy. So it's better to mold the data for better reception of features, therefore, we are performing shearing, zooming, and horizontal rotation on our training data. Once the images are sculpted we can convert the given images in the input shape that is 224\*224 with a batch size of 32 and can train our training set.

### 2.2.4.3 LOSS AND ACCURACY CURVES

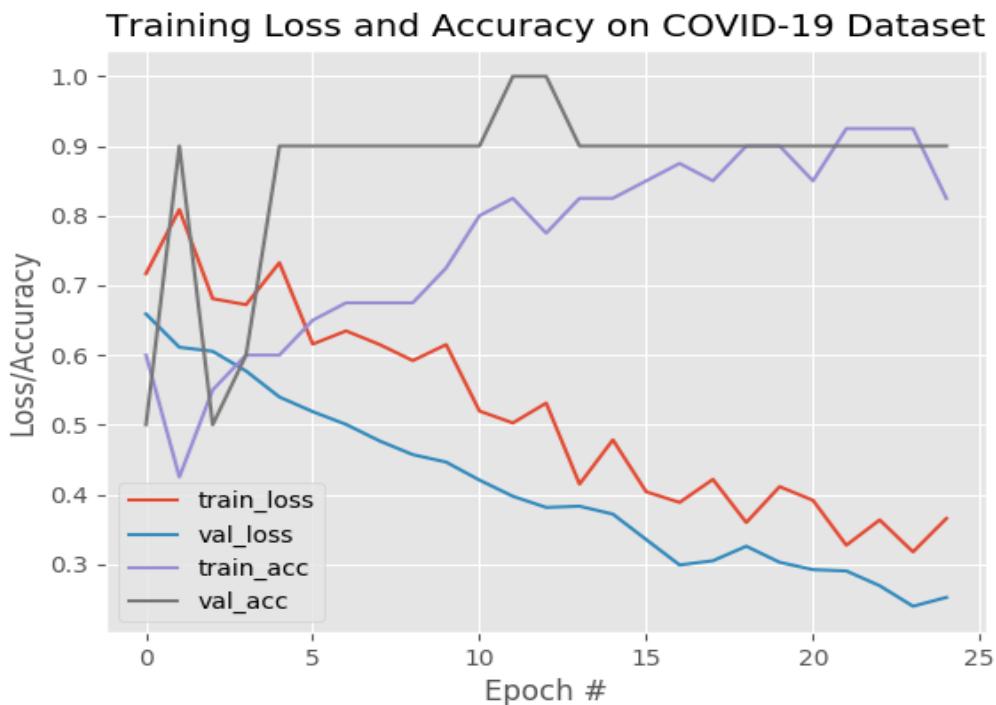


Figure 2.10: MODEL LOSS AND ACCURACY

## 2.2.5 PNEUMONIA DETECTION MODEL

### 2.2.5.1 DATA COLLECTION AND PREPROCESSING

The data[19] needed for this problem is the chest X-Ray images of patients who dont have tuberculosis and those who have tuberculosis. The dataset used for training this model is taken from mendeley.com. The dataset consists of 5,863 chest x-ray images, with 1583 belonging to normal and rest belonging to tuberculosis. All the images are resized to the resolution of 224x224x3 RGB. The task of the kernel is to create a model, which can look at the chest x-ray images and predict with high accuracy if the x-ray is of tuberculosis patient or not.

The dataset provided already has training and testing folder with each further having folder for tuberclosis and normal images. We use open-cv to read images from the folders, then the read images are resized to 224x224x3 and then saved to training set or testing set along with their labels. Next we create a custom generator which would take the training set or testing set and generate batches for neural network model. The generated images by the generator also contains augmented images. Augmentation is performed to increase the size of preceived dataset and increase the generality of the model.

### 2.2.5.2 MODEL DESCRIPTION

Since the dataset wasn't big enough and the complexity of the task was fairly high we decided to use transfer learning instead of designing our own custom model from scratch. We used VGG-16 which is trained on ImageNet Dataset. As the benefits of transfer learning have been discussed earlier we will skip that part here. The neural network used here has base of VGG 16 i.e only convolutional layers are taken from VGG 16. The base is followed by a Flatten layer which is then followed by 3 fully connected layers with 1024, 512, 2 units respectively and we have a dropout layer between the 3 fully connected layer as well. The base of the model is not trainable i.e the gradients wont travel beyond the fully connected layers. The last fully connected layer is the output layer of the model which tells whether the chest x-ray belongs to person with pneumonia or not. The activation function used in fully connected layers is ReLU except in last where softmax is used as an activation function. To train the neural network we use Adam Optimizer and binary cross entropy as loss function

### 2.2.5.3 PREDICTIONS

---

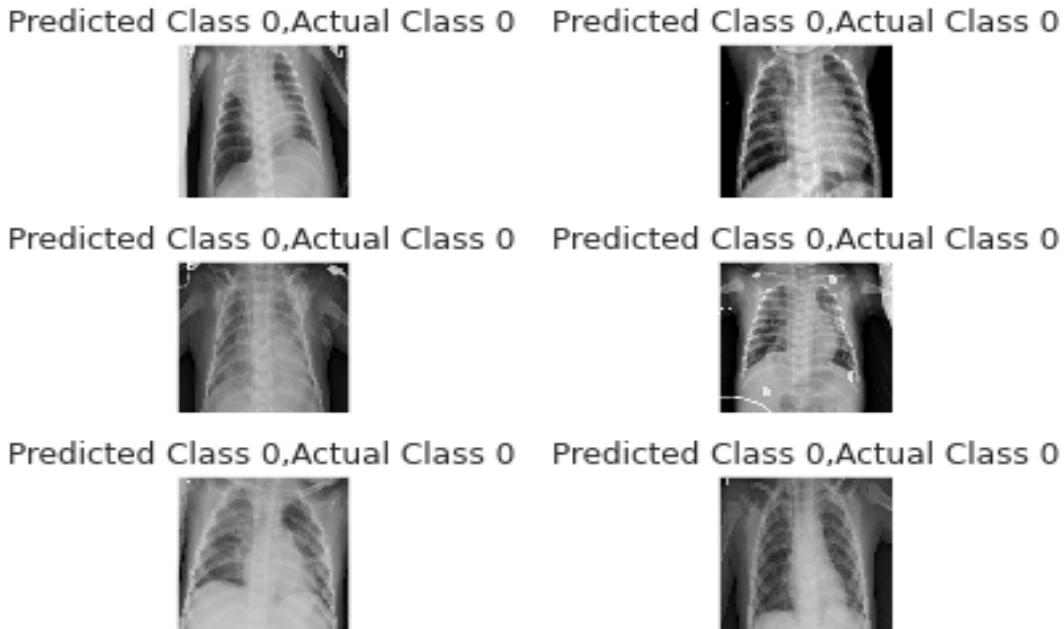


Figure 2.11: PREDICTIONS

## 2.2.6 DIABETIC RETINOPATHY DETECTION

### 2.2.6.1 DATA COLLECTION AND PREPROCESSING

Our main dataset[20] is based on the [Kaggle Diatebic Retinopathy Detection competition] (<https://www.kaggle.com/c/diabetic-retinopathy-detection>) which was carried out in 2016. The main dataset contains 35,000 eye images with 5 stages of DR disease. We also look at the Messidor dataset which contains 1,200 images with 4 stages of DR progression. Although the Messidor dataset is smaller, there are fewer labeling errors. The images in the dataset come from different models and types of cameras, which can affect the visual appearance of left vs. right. Some images are shown as one would see the retina anatomically (macula on the left, optic nerve on the right for the right eye). Others are shown as one would see through a microscope condensing lens.

There are total 35,126 images in our dataset, we split the dataset in train and validation using train test split module of sklearn such that training set distribution is balanced. We resize the images, then use image augmentation, rotation, flipping, rescaling, translation and cropping techniques to preprocess the image and convert them into an array so that they can be feed into neural network.

### 2.2.6.2 MODEL DESCRIPTION

We have opted for transfer learning with Inception v3 model pretrained on image-net data set. Transferred learning is used since the final categories are different between ImageNet (categorization for common objects) and DR (categorization for eye disease). We usually use the no-top-layer pretrained model and then add several layers of normalization and fully connected layer on the top. We will then retrain the final classification layer. Attention Map [24] is a mechanism to expand the capabilities of neural networks. They enable focusing on specific parts of the input, and it has been shown they can improve the performance results of neural processing. We have shown some fundus images and the attention maps exerted on the region to help processing. The basic idea is that a Global Average Pooling is too simplistic since some of the regions are more relevant than others. So we build an attention mechanism to turn pixels in the GAP on or off before the pooling and then rescale (Lambda layer) the results based on the number of pixels. The model could be seen as a sort of 'global weighted average' pooling. There is probably something published about it and it is very similar to the kind of attention models used in NLP. It is largely based on the insight that the winning solution annotated and trained a UNET model to segmenting the hand and transforming it. At last, we going to flatten our features and will use sigmoid as activation function as we are having binary classification problem, and thus our output will only contain one cell, adam as optimizer works well with sigmoid hence compiling the model with them in addition to cross binary entropy.

### 2.2.6.3 ATTENTION MAP

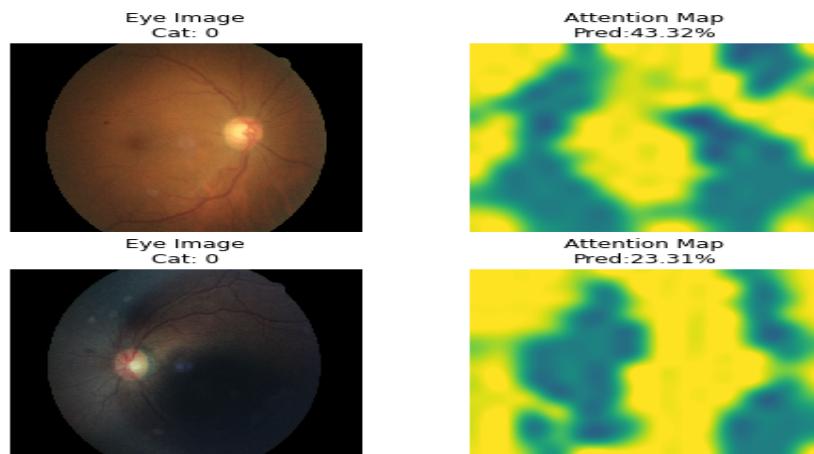


Figure 2.12: ATTENTION MAP.

### 2.2.6.4 PREDICTIONS

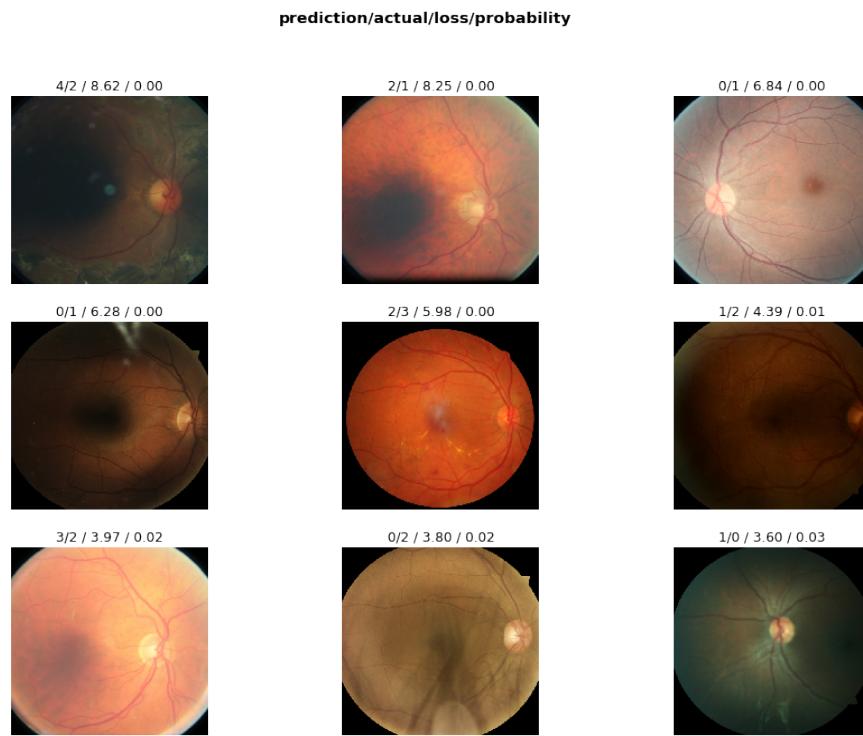


Figure 2.13: PREDICTIONS

## 2.3 WORKING OF THE WEB APPLICATION

- The user visits the web application. At the home page of the application, the user see about the application and services provided.

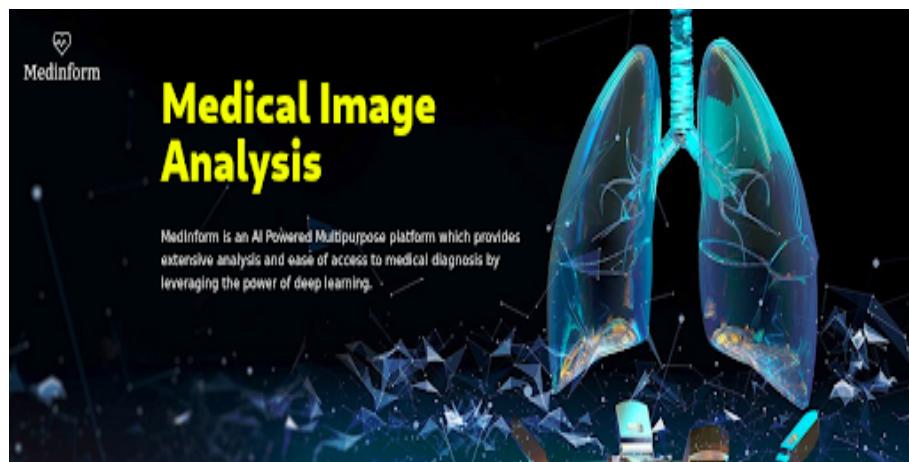


Figure 2.14: HOME PAGE

- The user clicks on the respective disease button to go the Image Upload Page of that disease.

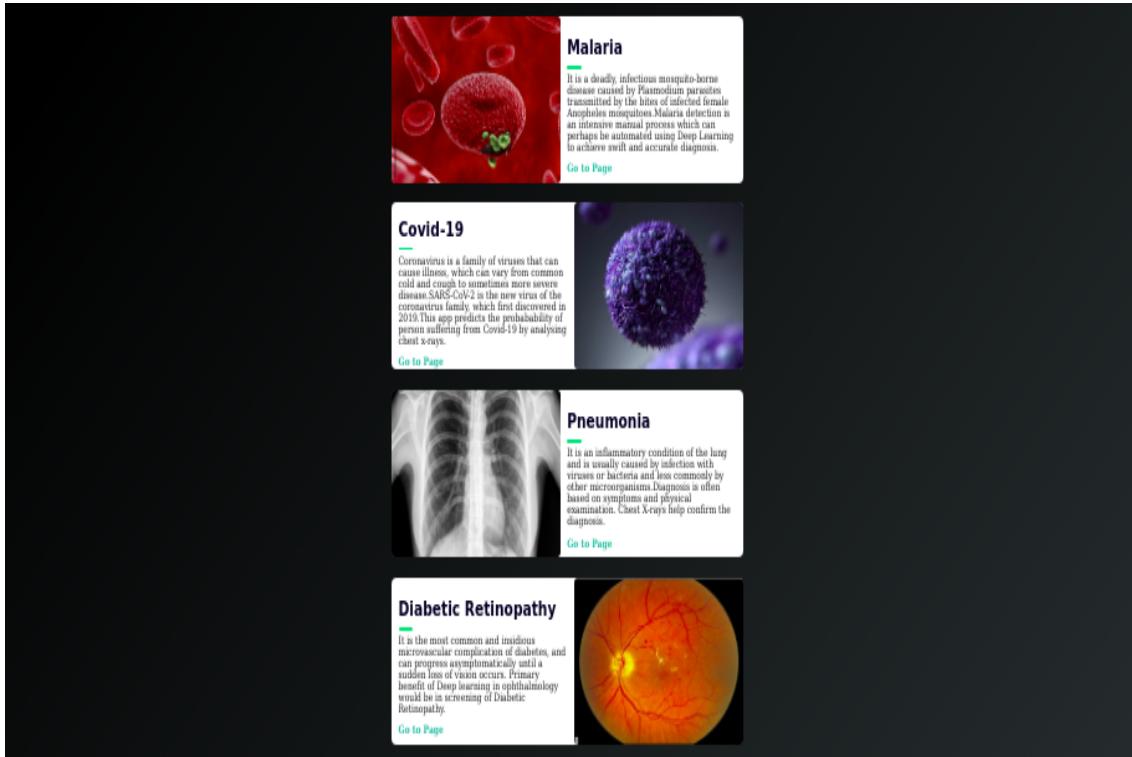


Figure 2.15: SERVICES

- The user uploads the medical image and the application send the image to back-end to preprocess the image and pre-trained classifier then runs on the image.

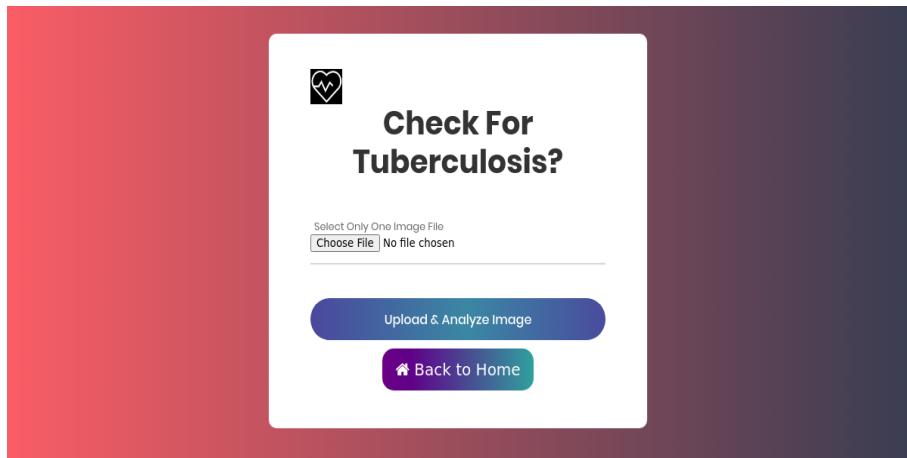


Figure 2.16: PREDICTING FOR TUBERCULOSIS

- The result of the prediction are display on the Results page of the application.

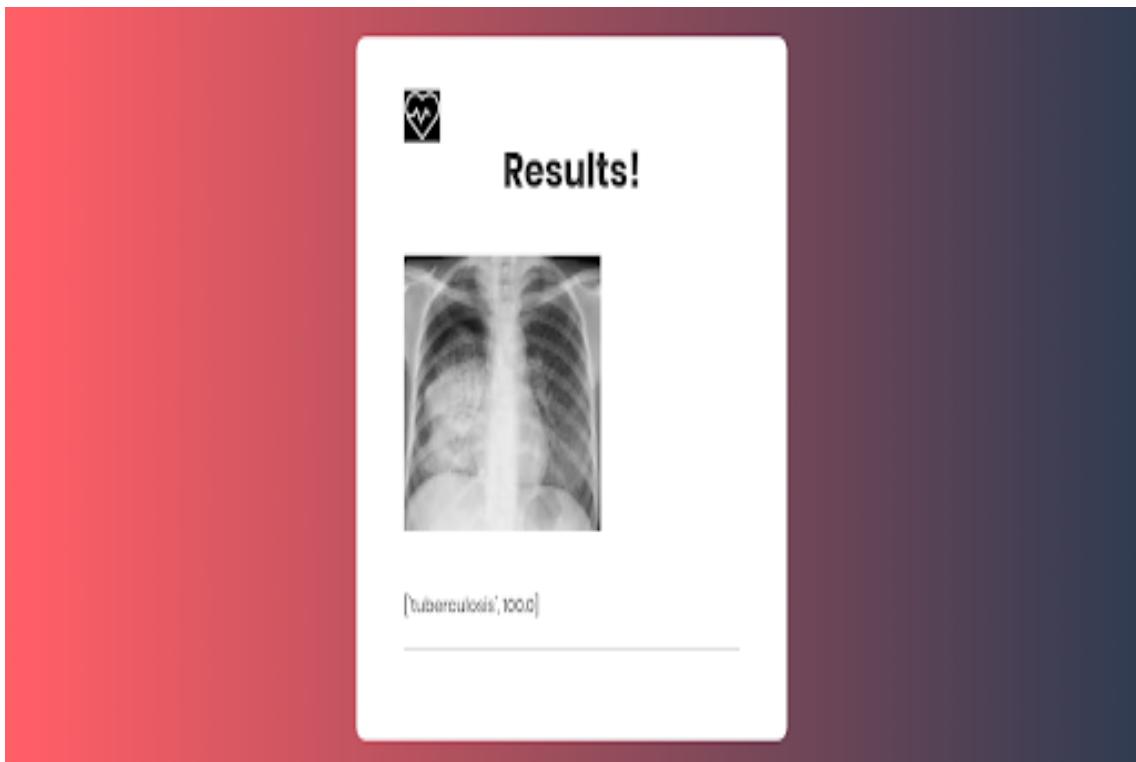


Figure 2.17: RESULTS

# CHAPTER 3

## RESULTS AND DISCUSSION

### 3.1 ACCURACY

We can see that our models achieved really good accuracy score on the test dataset.

ML-Model(Classifier)	Accuracy(%)
Tuberculosis Model	84
Skin Cancer Recognition	91
Malaria Detection	96
Covid-19 Detection	91
Pneumonia Detection	86
Diabetic Retinopathy	88

Figure 3.1: ACCURACY TABLE FOR ALL CLASSIFIERS

We now define the various metrics i.e. precision, recall, support, F1 score that gives detailed information about the performance of our classifiers.

### 3.2 EVALUATION METRICS

We now define the various metrics i.e. precision, recall, support, F1 score that gives detailed information about the performance of our classifier [22].

### 3.2.1 PRECISION

The precision is the ability of the classifier not to label a sample that is negative as positive [22]. For each class, it is defined as the ratio of true positives to the sum of true and false positives.

$$\text{Precision} = \frac{\text{truepositives}}{\text{truepositives} + \text{falsepositives}}$$

### 3.2.2 RECALL

The recall is intuitively the ability of the classifier to find all the positive samples [22]. For each class, it is defined as the ratio of true positives to the sum of true positives and false negatives.

$$\text{Recall} = \frac{\text{truepositives}}{\text{truepositives} + \text{falsenegatives}}$$

### 3.2.3 SUPPORT

Support is the number of actual occurrences of the class in the specified dataset. Imbalanced support in the training data may indicate structural weaknesses in the reported scores of the classifier and could indicate the need for stratified sampling or rebalancing [22].

### 3.2.4 F1 SCORE

The F1 score is the harmonic mean of the precision and recall, where an F1 score reaches its best value at 1 (perfect precision and recall). The F1 score is also known as the Sørensen–Dice coefficient or Dice similarity coefficient (DSC).

### 3.2.5 CLASSIFICATION REPORT

Tuberculosis Model	Precision	Recall	Support	F1-Score
Normal	0.77	0.89	61	0.82
Tuberculosis	0.86	0.73	59	0.79

Figure 3.2: TUBERCULOSIS DETECTION MODEL REPORT

Skin Cancer Model	Precision	Recall	Support	F1-Score
Malignant Cancer	0.80	0.81	360	0.84
Benign Cancer	0.84	0.83	360	0.84

Figure 3.3: SKIN-CANCER DETECTION MODEL REPORT

Malaria Model	Precision	Recall	Support	F1-Score
Normal	0.95	0.98	2805	0.96
Parasitic	0.97	0.98	2706	0.96

Figure 3.4: MALARIA DETECTION MODEL REPORT

Covid-19 Model	Precision	Recall	Support	F1-Score
Normal	1.00	0.80	300	0.89
Covid	0.93	1.00	150	0.94

Figure 3.5: COVID-19 DETECTION MODEL REPORT

Pneumonia Model	Precision	Recall	Support	F1-Score
Normal	0.60	0.91	234	0.70
Pneumonia	0.97	0.80	390	0.88

Figure 3.6: PNEUMONIA DETECTION MODEL REPORT

Diabetic Retinopathy Model	Precision	Recall	Support	F1-Score
0 - No Diabetic Retinopathy	0.68	0.87	129	0.76
1 - Mild	0.08	0.06	18	0.07
2 - Moderate	0.21	0.09	33	0.13
3 - Severe	0.00	0.00	7	0.00
4 - Proliferative DR	0.00	0.00	5	0.00

Figure 3.7: DIABETIC RETINOPATHY DETECTION MODEL REPORT

### 3.2.6 CONFUSION MATRIX

A confusion matrix is a table that is often used to describe the performance of a classification model (or “classifier”) on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm.

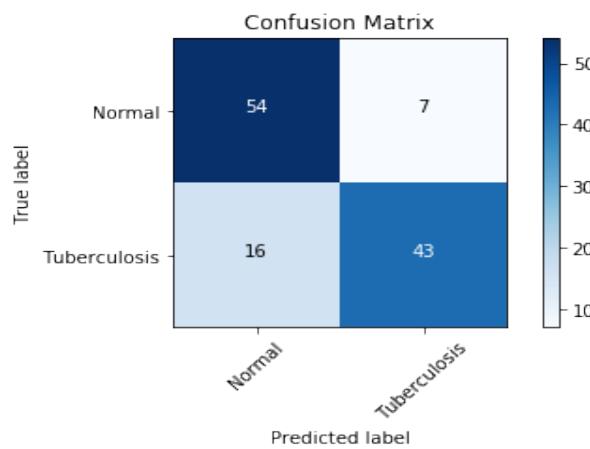


Figure 3.8: TUBERCULOSIS DETECTION CONFUSION MATRIX

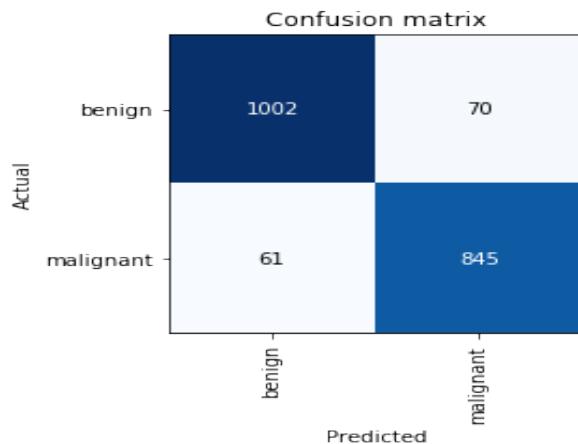


Figure 3.9: SKIN CANCER DETECTION CONFUSION MATRIX

### 3.2.7 AUC SCORE AND ROC SCORE

AUC or area under the curve is a performance measurement that denotes the degree of separability i.e. how good our model is in differentiating between the classes [7]. The higher the value of the area under the curve, the better is the performance of our model. The AUC score for our intermediate model is 0.97 which is pretty good. ROC or the receiver operating characteristics represents the probability curve. It is plotted by keeping the TPR on y-axis and FPR on the x-axis. The TPR rate is also called as sensitivity and the FPR rate is also called as the fall-out.

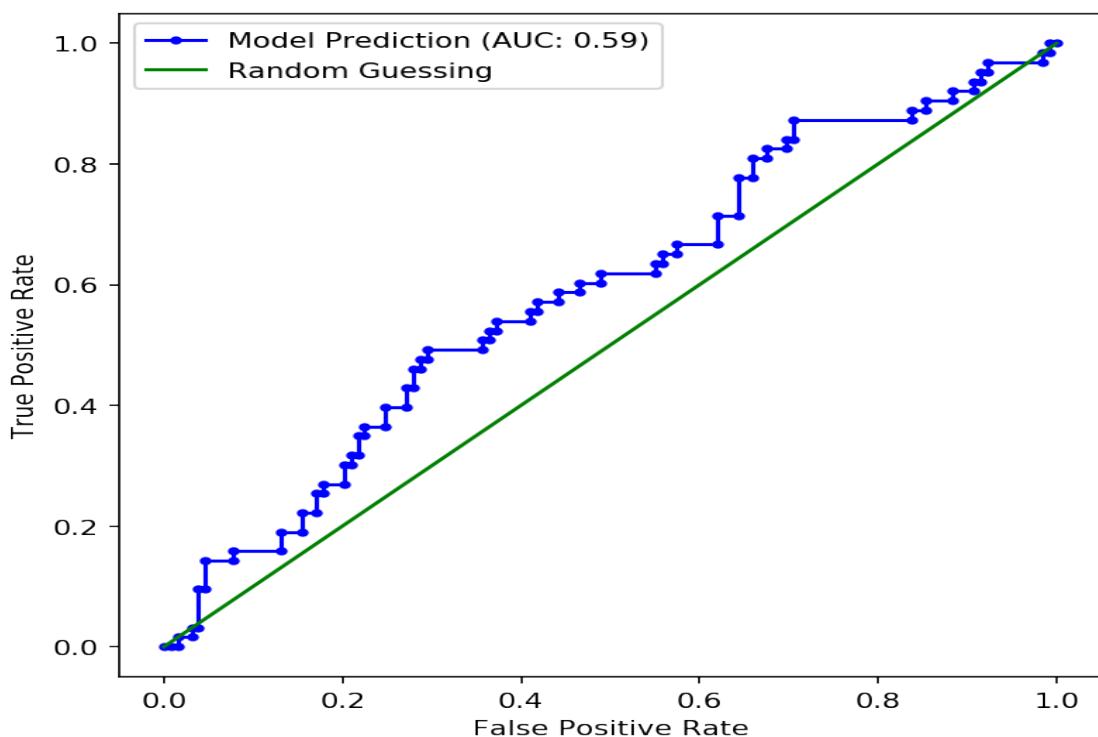


Figure 3.10: ROC CURVE FOR DIABETIC RETINOPATHY MODEL

# **CHAPTER 4**

## **CONCLUSION AND FUTURE SCOPE**

### **4.1 CONCLUSION**

We have analyzed the performance of CNN-based deep learning models in doing image analysis and thus trying to classify the images into their respective categories as accurately as possible. By observing the results, it is clear that the models were able to perform satisfactorily on various evaluation metrics. On analysis of website and deployment of the models on cloud again satisfactory results were found, while getting prediction majority of time was consumed by uploading of an image which is dependent on one's internet uploading speed. Prediction time were pretty much independent of the size and depth of the model deployed and since a single central model serves request for a single disease we can be sure about reproducibility of result.

### **4.2 FUTURE SCOPE**

We've built web-platform for detecting six diseases and it is possible to extend the platform for other diseases like scabies or tasks like brain tumor segmentation from CT scan and other medical images. We can add functionality to process multiple files at once for corporate organizations. In the current implementation deep learning models are static but it is possible to implement advanced deep learning techniques like federated learning [23] so that the deep learning models are continuously learning and improving by developing a pipeline.

# REFERENCES

- [1] Convolutional neural network. [Online]. Available: [https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neuralnetworks-the-eli5-way-3bd2b1164a53](https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53)
- [2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun,"Deep Residual Learning for Image Recognition",<https://arxiv.org/abs/1512.03385>
- [3] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna,"Rethinking the Inception Architecture for Computer Vision" , <https://arxiv.org/abs/1512.00567>
- [4] S.Mannor,D.Peleg, and R.Rubinstein, "The crossentropy method for classification," in Proceedings of the 22nd international conference on Machine learning. ACM, 2005, pp. 561–568
- [5] Karen Simonyan, Andrew Zisserman,"Very Deep Convolutional Networks for Large-Scale Image Recognition",<https://arxiv.org/pdf/1409.1556.pdf>
- [6] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proceeding of the International Conference on Learning Representations. San Diego, 2015, pp. 1–5..
- [7] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in Proceedings of the 23rd international conference on Machine learning. ACM, 2006, pp. 233–240.
- [8] Malaria datasets. [Online]. Available: <https://ceb.nlm.nih.gov/repositories/malaria-datasets/>
- [9] Tuberculosis Dataset. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4256233/>  
The datasets come from Shenzhen and Montgomery respectively.
- [10] Jaeger S, Karargyris A, Candemir S, Folio L, Siegelman J, Callaghan F, Xue Z, Palaniappan K, Singh RK, Antani S, Thoma G, Wang YX, Lu PX, McDonald CJ. Automatic tuberculosis screening using chest radiographs. IEEE Trans Med Imaging. 2014 Feb;33(2):233-45. doi: 10.1109/TMI.2013.2284099. PMID: 24108713

- [11] Candemir S, Jaeger S, Palaniappan K, Musco JP, Singh RK, Xue Z, Karargyris A, Antani S, Thoma G, McDonald CJ. Lung segmentation in chest radiographs using anatomical atlases with nonrigid registration. *IEEE Trans Med Imaging*. 2014 Feb;33(2):577-90. doi: 10.1109/TMI.2013.2290491. PMID: 24239990
- [12] VGG-16 Architecture [Online]  
<https://neurohive.io/wp-content/uploads/2018/11/vgg16-1-e1542731207177.png>
- [13] Inception V3 Architecture [Online]  
<https://missinglink.ai/guides/tensorflow/tensorflow-image-classification/>
- [14] ReLU — Most popular Activation Function for Deep Neural Networks  
[https://miro.medium.com/max/1140/1\\*D0F3UiQ2l5Q0Ak-tjEdJg.png](https://miro.medium.com/max/1140/1*D0F3UiQ2l5Q0Ak-tjEdJg.png)
- [15] Convolutional neural network. [Online]. Available:  
<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
- [16] Skin-Cancer Dataset. [Online]. Available:<https://www.kaggle.com/fanconic/skin-cancer-malignant-vs-benign>
- [17] CovidAID: COVID-19 Detection Using Chest X-Ray Arpan Mangal, Surya Kalia, Harish Rajgopal, Krithika Rangarajan, Vinay Namboodiri, Subhashis Banerjee, Chetan Arora arXiv:2004.09803 [eess.IV]
- [18] Covid-19 Dataset. [Online]. <https://github.com/ieee8023/covid-chestxray-dataset>
- [19] Pneumonia Dataset. [Online]. <https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia>
- [20] Diabetic Retinopathy Dataset. [Online]. <https://www.kaggle.com/c/diabetic-retinopathy-detection/data>
- [21] Prerained Keras and Tensorflow Models <https://keras.io/api/applications/>
- [22] Hossin, Mohammad AU - M.N, Sulaiman PY - 2015/03/31 SP - 01 EP - 11 T1 - A Review on Evaluation Metrics for Data Classification Evaluations VL - 5 DO - 10.5121/ijdkp.2015.5201 JO - International Journal of Data Mining Knowledge Management Process ER
- [23] Advances and Open Problems in Federated Learning Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet arXiv:1912.04977 [cs.LG]
- [24] Embedding Human Knowledge into Deep Neural Network via Attention Map Masahiro Mitsuhashara, Hiroshi Fukui, Yusuke Sakashita, Takanori Ogata, Tsubasa Hirakawa, Takayoshi Yamashita, Hironobu Fujiyoshi arXiv:1905.03540 [cs.CV]