काशी हिन्दू विश्वविद्यालय

BANARAS HINDU UNIVERSITY

ESTABLISHED BY PARLIAMENT BY NOTIFICATION NO. 225 OF 1916

PROJECT

DEEP LEARNING FOR EARLY DETECTION OF DIABETIC RETINOPATHY

Dissertation Report submitted to Banaras Hindu University for the Partial Fulfilment of the Requirement for the Award of the Degree of

MASTER'S OF SCIENCE

IN

MATHEMATICS AND COMPUTING

SESSION:2020-2021

UNDER THE SUPERVISION OF

DR. MANJARI GUPTA

DST- CENTRE FOR INTERDISCIPLINARY MATHEMATICAL SCIENCS ,BHU

SUBMITTED BY

ANUSHKA PANDEY

EXAM ROLL NO:-19419MAC013

ENROLLMENT NO:-412105

&

VISHAL KUMAR

EXAM ROLL NO:-19419MAC043

ENROLLMENT NO:-413330

DEPARTMENT:- DST- CENTRE FOR INTERDISCIPLINARY MATHEMATICAL SCIENCE

BANARAS HINDU UNIVERSITY VARANASI ,

UTTAR PRADESH, 221005

CERTIFICATE

THIS IS TO CERTIFY THAT THE DATA GIVEN IN THIS DISSERTATION "DEEP LEARNING FOR EARLY DETECTION OF DIABETIC RETINOPATHY" WAS PRIMARILY COMPILED , ANALYSED & PRESENTED BY VISHAL KUMAR AND ANUSHKA PANDEY ,M.SC.(MATHEMATICS AND COMPUTING) 4TH SEM STUDENTS OF DST- CENTRE FOR INTERDISCIPLINARY MATHEMATICAL SCIENCE, INSTITUTE OF SCIENCE BHU,VARANASI.

THIS PROJECT HAS BEEN COMPLETED UNDER MY SUPERVISION & GUIDANCE IN SESSION 2021.

DATE:-30/07/2021

DR. MANJARI GUPTA

DST-CIMS

BANARAS HINDU UNIVERSITY

VARANASI,PIN:- 221005

ACKNOWLEDGEMENT

FIRST OF ALL, I BOW TO THE ALMIGHTY FOR GIVING ME STRENGTH AND INSPIRATION. I PAY TRIBUTE TO PT. MADAN MOHAN MALVIYA JI, THE FOUNDER OF OUR ALMA MATER, FOR HIS LIFE TIME SACRIFICE AND EFFORTS IN ESTABLISHING A TEMPLE OF LEARNING. I AM IMMENSELY GRATEFUL TO MY SUPERVISOR ***DR. MANJARI GUPTA****, WHO ENTICED ME TO START THIS WORK WITH HIM. I REALLY WISH TO EXPRESS MY DEEP SENSE OF GRATITUDE & INDEBTEDNESS TO HIS VALUABLE SUGGESTIONS, ACADEMIC FREEDOM, CONSTANT ENCOURAGEMENTS, INSPIRING GUIDANCE THROUGHOUT THIS PERIOD WHICH HELPED ME GATHER CONSIDERABLE KNOWLEDGE AND PERSPECTIVE.*

ANUSHKA PANDEY

&

VISHAL KUMAR

M.SC. (MATHEMATICS AND COMPUTING)

DST- CENTRE FOR INTERDISCIPLINARY MATHEMATICAL SCIENCES BANARAS HINDU UNIVERSITY

CONTENTS

Chapter Name	Page No.
1. Intriduction	**1**
1.1 Diabetes ……	1
1.2 Diabetes Retinopathy ……..2	
2. Deep Learning	**3**
2.1 Architectures……..3	
3. Methodology	**6**
3.1 Data Collection ……..6	
3.2 Implementation on Python ……..,7	
4. Result & Disscussion	**8-14**
5. Conclusion	**15**
6. Bibliography	**16**

Chapter 1

INTRODUCTION:-

Diabetic is a disease that increases the amount of Glucose in the blood caused by a lack of insulin. It effects 425 million adults worldwide.Diabetes effects the retina, heart,nerves,and Kidneys.Therefore, this project presents a method to detect a diabetic disease on a person by using the fundus images of their eyes. Diabetes can have many effects on the eye's such as fluctuation in vision clarity, flurry and/or distortions in vision, eye pain, cataracts, slower in healing after the eyes injuries and blind ness. Moreover, most of the scariest things that effect to the eye of diabetes patient is the blindness. Basically, diabetic retinopathy was something that harm to the retina in the eyes. Basically, retina was laying at the back of the eyes and its sensitive to the light. Therefore, if it is damaged, its can causing someone to be a blindness. fering from this disease. there were two types of diabetes which is diabetes type 1 and type 2.

1.1 Diabetes

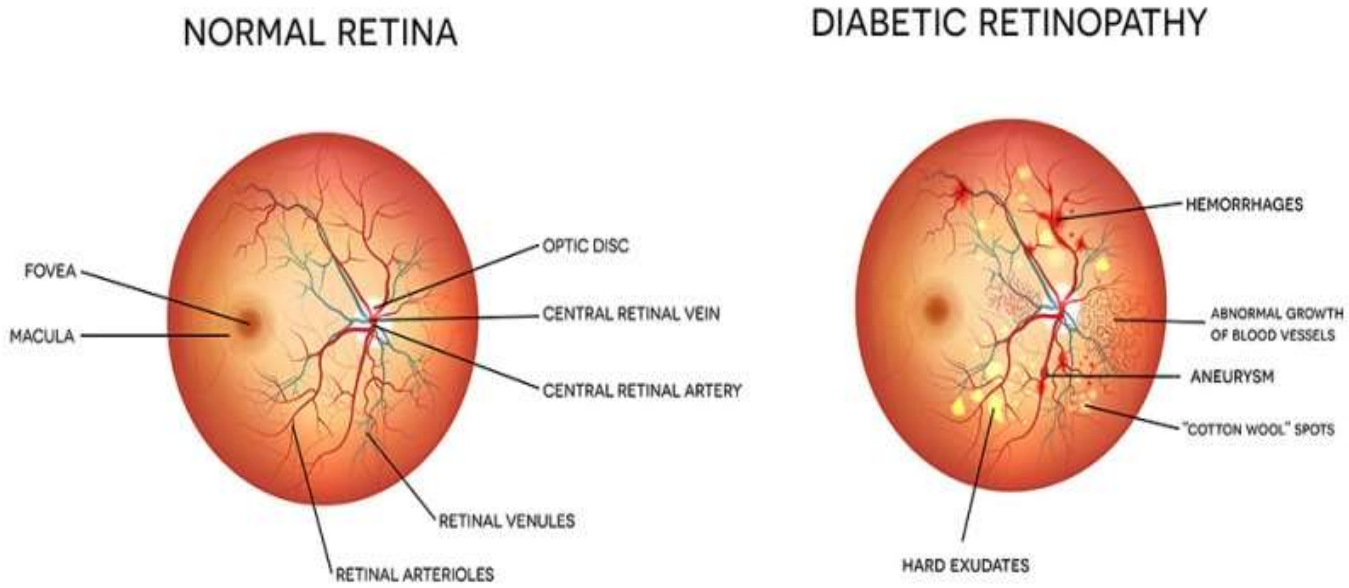
Type 1 diabetes is called as insulin-dependent diabetes. It is often beginning in childhood and since it starts from childhood it used to be called juvenile-onset diabetes. Type 1 diabetes is an auto immune condition and it's caused by the body to attacking its own pancreas with antibodies. In addition, a person that has type 1 diabetes, their pancreas will be damaged and it will not make any insulin. Basically, this type of diabetes may be caused by a genetic predisposition which is comes from their parents. It could also be the result of faulty beta cells in the pancreas that normally produce insulin. Therefore, the treatment for type 1 diabetes involves taking insulin, which needs to be injected through the skin into the fatty tissue below.

Then, for type 2 diabetes it is called as adult-onset diabetes, which is an obese and overweight kids. teenagers now adays were possesses with this type of diabetes. Scientifically, type 2 diabetes was called as non-insulin-dependent diabetes and it is the higher case rather than type 1 diabetes. In addition, it can make some organ didn't fully functionally, especially on the smallest blood vessels in the body which working to sustaining the nerve of kidneys and eyes. After that, it also can affect the pancre as where it is usually produces an insulin and it causes the amount of insulin produced is not enough for the body's needs, or the body's cells are resist to receive the insulin. Typically, obese per sons which have height 20% lower than their ideal body are easier to have this disease because they have an insulin resistance in their body

However, type 2 diabetes can be preventing by having some exercise, taking nutrition and managing their weight. Unfortunately, type 2 diabetes tends to progress and the diabetes medications are often needed.

1.2 Diabetic Retinopathy

Diabetic Retinopathy(DR) is a serious eye diesease associated with long-standing diabetes That results in progressive damage to the retina,eventually leading to blindness. We have devoloped an AI-based DR diagnostic Tool to help doctors detect and grade the level of DR disease based on the fundus images. Diabetic Retinopathy is a complication of diabeties that causes the blood vessels of the retina to swell and leak fluids and blood DR can lead to a loss of vision if it is an advanced stage.worldwide DR causes 2.6% of blindness .the possibility of DR presence increses for diabetes patients who suffer from the dieses for a long period.Retina regular screening is essential for diabetes patientes patients to diagnose amd to treat DR at an early stage to avoid the risck of blindness.



Chapter 2

Deep Learning:-

Deep learning is a branch of Machine Learning Trchniques that involves hierarchical layers of no-linear processing stages for unsupervised features learning as well as for classifiying patterns. DL is one computer-aided medical diadnosis method .DL applications to medical image analysis include the classification,segmentation,detection,retrival,and registration of the images. Recently,DL has been widely used in DR detection and Classification. It can successfully learn the features of input data even when many heterogeneous sources integrated. There are many DL-based methods such as restricted Boltzmann Machines,convolutional neural network(CNNs), auto encoder,and sparse coding.the preformance of these methods increses when the number of training data increses due to the increses in the learned features unlike machine learning methods. Also,DL methods did not require hand-crafted feature extraction. CNNs are more widley used than the other methods in medical image analysis,and it is highly effective.There are three main layers in the CNN architechture,which are convolution layers,pooling layers,and fully connected layers . The number of layers,size,and rhe number of filters of the CNN vary according to the author's vision.each layer in CNN architecture playes a specific role in the convolutin layers,different filters convole an image to extract the feature.

In human brain approximately 100 billion neurons all together this is a picture of an individual neuron and each neuron is connected through thousand of their neighbours.

The question here is how do we recreate these neurons in a computer. So, we create an artificial structure called an artificial neural net where we have nodes or neurons. We have some neurons for input value and some for output value and in between, there may be lots of neurons interconnected in the hidden layer.

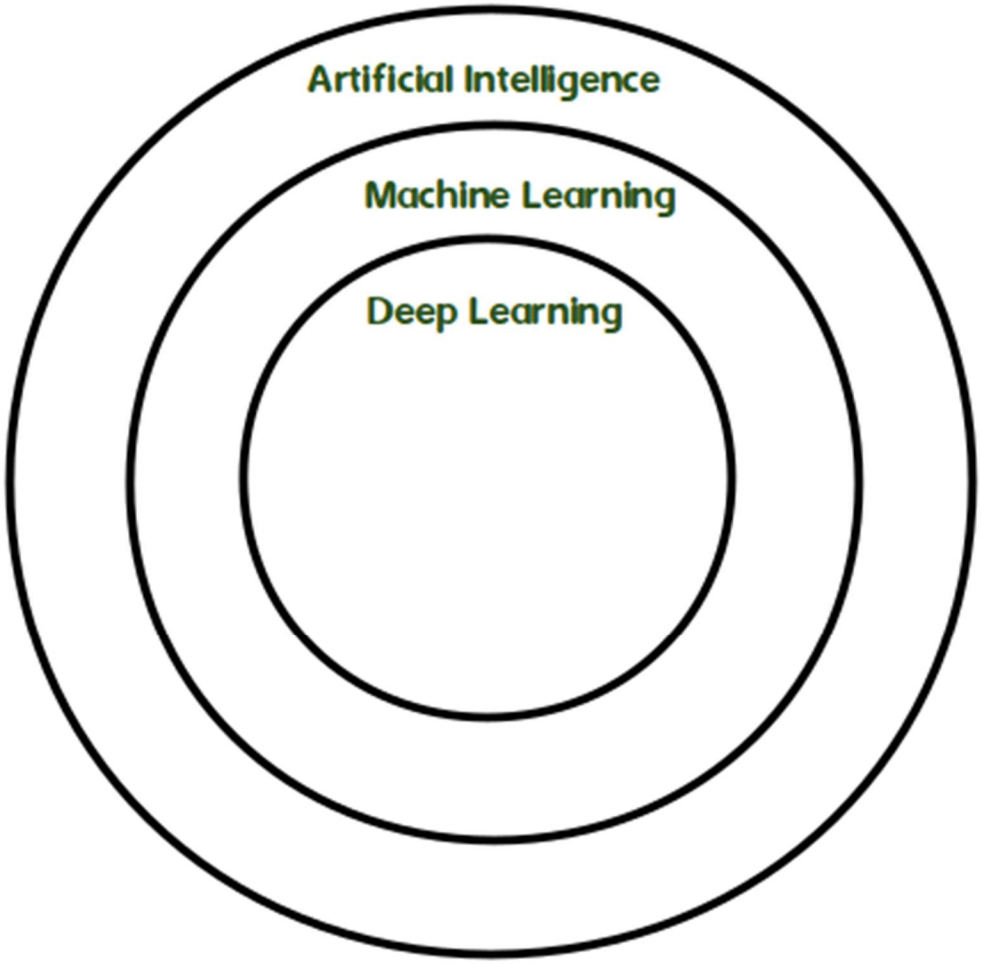
2.1 Architectures:-

- **Deep Neural Network –** It is a neural network with a certain level of complexity (having multiple hidden layers in between input and output layers). They are capable of modeling and processing non-linear relationships.
- **Deep Belief Network(DBN)** – It is a class of Deep Neural Network. It is multi-layer belief networks.
  Steps for performing DBN :
  a) Learn a layer of features from visible units using Contrastive Divergence algorithm.

b) Treat activations of previously trained features as visible units and then learn features of features.

c) Finally, the whole DBN is trained when the learning for the final hidden layer is achieved.

- **Recurrent** (perform same task for every element of a sequence) **Neural Network** – Allows for parallel and sequential computation. Similar to the human brain (large feedback network of connected neurons). They are able to remember important things about the input they received and hence enables them to be more precise.



Difference between Machine Learning and Deep Learning :

Machine Learning	Deep Learning
Works on small amount of Dataset for accuracy	Works on Large amount of Dataset.
Dependent on Low-end Machine.	Heavily dependent on High-end Machine
Takes less time to train.	Takes longer time to train.
Testing time may increase.	Less time to test the data.

Tools used :

Anaconda, Jupyter, Pycharm, etc.

Languages used :

R, Python, Matlab, CPP, Java, Julia, Lisp, Java Script, etc.

Limitations :

1. Learning through observations only.
2. The issue of biases.

Advantages :

1. *Best in-class performance on problems.*
2. *Reduces need for feature engineering.*
3. *Eliminates unnecessary costs.*
4. *Identifies defects easily that are difficult to detect.*

Disadvantages :

1. *Large amount of data required.*
2. *Computationally expensive to train.*
3. *No strong theoretical foundation.*

Applications :

1. **Automatic Text Generation** – Corpus of text is learned and from this model new text is generated, word-by-word or character-by-character.
   Then this model is capable of learning how to spell, punctuate, form sentences, or it may even capture the style.

2. **Healthcare** – Helps in diagnosing various diseases and treating it.
3. Automatic Machine Translation – Certain words, sentences or phrases in one language is transformed into another language (Deep Learning is achieving top results in the areas of text, images).
4. **Image Recognition** – Recognizes and identifies peoples and objects in images as well as to understand content and context. This area is already being used in Gaming, Retail, Tourism, etc.
5. **Predicting Earthquakes** – Teaches a computer to perform viscoelastic computations which are used in predicting earthquakes.

Chpter 3

Methodology:-

3.1Data Collection:-

The MESSIDOR-2 dataset is a collection of Diabetic Retinopathy(DR) examinations,each consisting of two macula-centered eye fundus images(one per eye). Part of the dataset(Messidor –Original)was kindly Provided by the Messidor programpartners.the remainder(Messidor – Extension) consists of never-before-published examinations form Brest University Hospital.

In the original Messidor dataset,some fundus images came in pairs,some others were single. Messidor-Original consists of all image pairs from the original Messidor dataset,that is 529 examinations(1058 images,saved in PNG format).

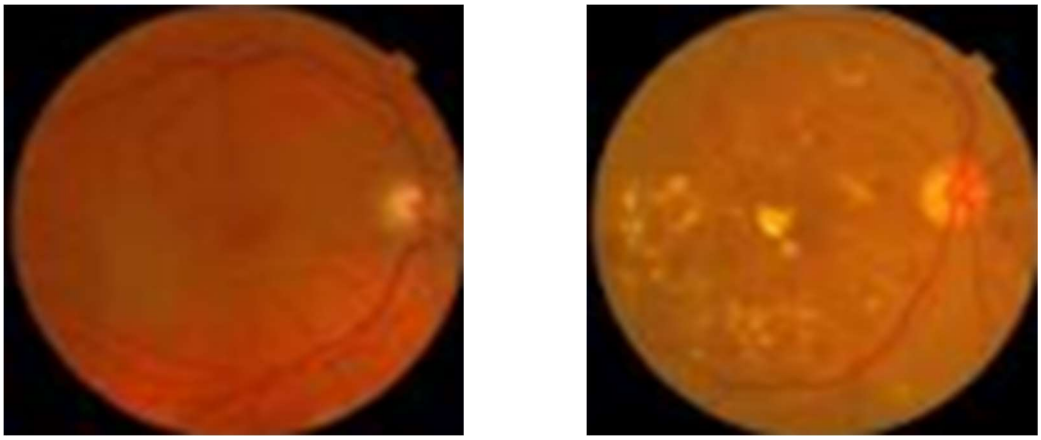
In order to populate Messidor-Extension,diabetic patients were recruited in the Ophthalmology department of Brest University Hospital(France) between October16,2009 and September 6,2010.

Eye fundi were imaged,whitout pharmacological dilation,using a Topcon TRC NW6 non-mydriaatic fundus camera with a 45 degree field of view.Only macula-centered images were included in the dataset.Messidor-Extension contains 345 examinations(690 images, in JPG format).

Overall,Messidor-2 dataset contains 874 examinations(1748 images). The dataset

comes with a spreadsheet containing image pairing.

As can be seen in figure (i),it shown some exudates appears as white spots on the fundus image.Nevertheless,figure(ii) shown the fundus image is completely without spots.



(i) (ii)

Fig. 1: (i) Normal eyes, (ii) Exudates eyes

3.2 Implimentation on Python:-

The Data Pre-Processing (Image Processing) is performed using Python's Scipy library's Miscellaneous Routines. The CPU is used as the interface with Python's Keras Library (TensorFlow in the backend).

Split the data set in train and test with ratio 8:2 with random. normalise the trani and test data set.

Use label encoder.

Use flatten data set (convert data setin vector form).

CNN architecture (mulltipe dense and dropout layers with softmax function.

Finally use Adam optimezer learning rate0.0001. finally use adam optimezer.

CODE:-

```
import tensorflow as tf
tf.test.gpu_device_name()
```

/device:GPU:0

```
!nvidia-smi
```

```
Sun Apr 18 18:19:49 2021
+-----------------------------------------------------------------------------
----+
| NVIDIA-SMI 460.67       Driver Version: 460.32.03    CUDA Version: 11.2
|
|-------------------------------+----------------------+----------------------
----+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr.
ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute
M. |
|                               |                      |               MIG
M. |
|===============================+======================+======================
====|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |
0 |
| N/A   56C    P0    28W /  70W |    222MiB / 15109MiB |      0%
Default |
|                               |                      |
N/A |
+-------------------------------+----------------------+----------------------
----+

+-----------------------------------------------------------------------------
----+
| Processes:
|
|  GPU   GI   CI        PID   Type   Process name                  GPU
Memory |
|        ID   ID                                                   Usage
|
|=============================================================================
====|
+-----------------------------------------------------------------------------
----+
```

#here we mount a google drive

```
from google.colab import drive
```

```
drive.mount('/content/drive', force_remount=True)
Mounted at /content/drive

!ls "/content/drive/My Drive"
'Colab Notebooks'   Dataset

pip install Kaggle
#here we import all Libreies

from sklearn.preprocessing import LabelEncoder
from keras.utils import np_utils
from keras.layers import Conv2D,MaxPooling2D
import os
import numpy as np
import tensorflow as tf
import keras
from keras import models,layers
from keras.models import Sequential, Model
from keras.layers import Dense,Dropout,Flatten,Input
import matplotlib.pyplot as plt
from keras.preprocessing import image
from sklearn.model_selection import train_test_split


path="/content/drive/MyDrive/Dataset/IMAGES/"


data=os.listdir(path)
data


array=[]
for index,y in enumerate(data):
 img=image.load_img(path+y,target_size=(224,224))
 img_data=image.img_to_array(img)
 array.append(img_data)


img_data
array([[[0., 1., 0.],
        [0., 0., 0.],
        [0., 1., 0.],
        ...,
        [6., 4., 7.],
        [8., 2., 2.],
        [9., 4., 1.]],

       [[0., 0., 0.],
        [0., 0., 0.],
```

```
        [3., 0., 0.],
        ...,
        [3., 3., 3.],
        [1., 2., 4.],
        [4., 2., 3.]],

       [[0., 1., 0.],
        [0., 0., 0.],
        [0., 0., 0.],
        ...,
        [3., 1., 2.],
        [4., 3., 1.],
        [4., 3., 8.]],

       ...,

       [[0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.],
        ...,
        [5., 4., 9.],
        [2., 1., 6.],
        [2., 0., 1.]],

       [[0., 0., 0.],
        [0., 0., 2.],
        [0., 1., 0.],
        ...,
        [5., 1., 2.],
        [8., 0., 2.],
        [3., 2., 0.]],

       [[0., 1., 0.],
        [0., 0., 0.],
        [0., 0., 0.],
        ...,
        [1., 1., 3.],
        [5., 2., 0.],
        [1., 3., 2.]]], dtype=float32)
```

```
import csv
grade={}
with open('/content/drive/MyDrive/Dataset/messidor_data.csv',newline=None)
as csvfile:
  spamreader=csv.reader(csvfile,delimiter=',',quotechar='1')
  for row in spamreader:
    if (row[3]=='0'):
      grade[row[0]]='0'
    else:
      grade[row[0]]=row[1]
label=[]
for i in data:
  label.append(grade[i])
```

#link texttrain and test the images

```
train_array,test_array,y_train,y_test=train_test_split(array,label,test_si
ze=0.2,random_state=13)


label_train=np.array(y_train)
label_test=np.array(y_test)
test_array=np.array(test_array)
train_array=np.array(train_array)


np.max(train_array)
255.0
```

here try to leran train_test_split

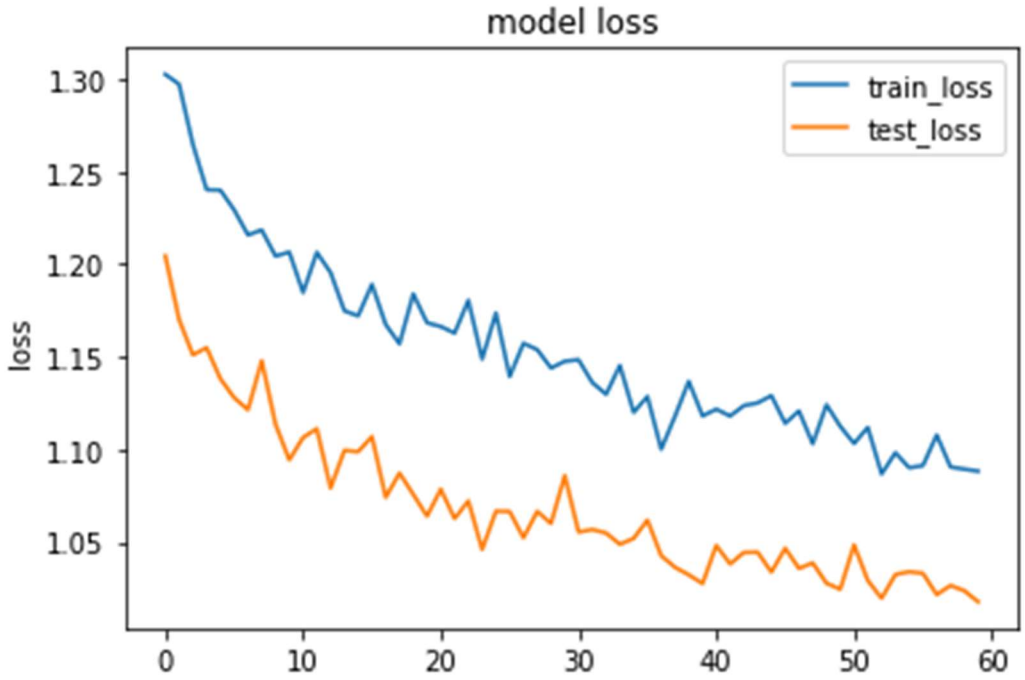
```
train_array=train_array/np.max(train_array)
test_array=test_array/np.max(test_array)


lb=LabelEncoder()
y_train=np_utils.to_categorical(lb.fit_transform(label_train))
y_test=np_utils.to_categorical(lb.fit_transform(label_test))
train_x,valid_x,train_label,valid_label=train_test_split(train_array,y_tra
in,test_size=0.1,random_state=13)


Input=Input(shape=(224,224,3), name='image_input')


Input
<KerasTensor: shape=(None, 224, 224, 3) dtype=float32 (created by layer
'image_input')>



flat=Flatten()(Input)
x=Dense(1024,activation='relu')(flat)
x=Dropout(0.2)(x)
x=Dense(512,activation='relu')(x)
x=Dropout(0.2)(x)
x=Dense(128,activation='relu')(x)
x=Dropout(0.2)(x)
x=Dense(64,activation='relu')(x)
x=Dropout(0.2)(x)
x=Dense(32,activation='relu')(x)
prediction=Dense(5,activation="softmax")(x)
```

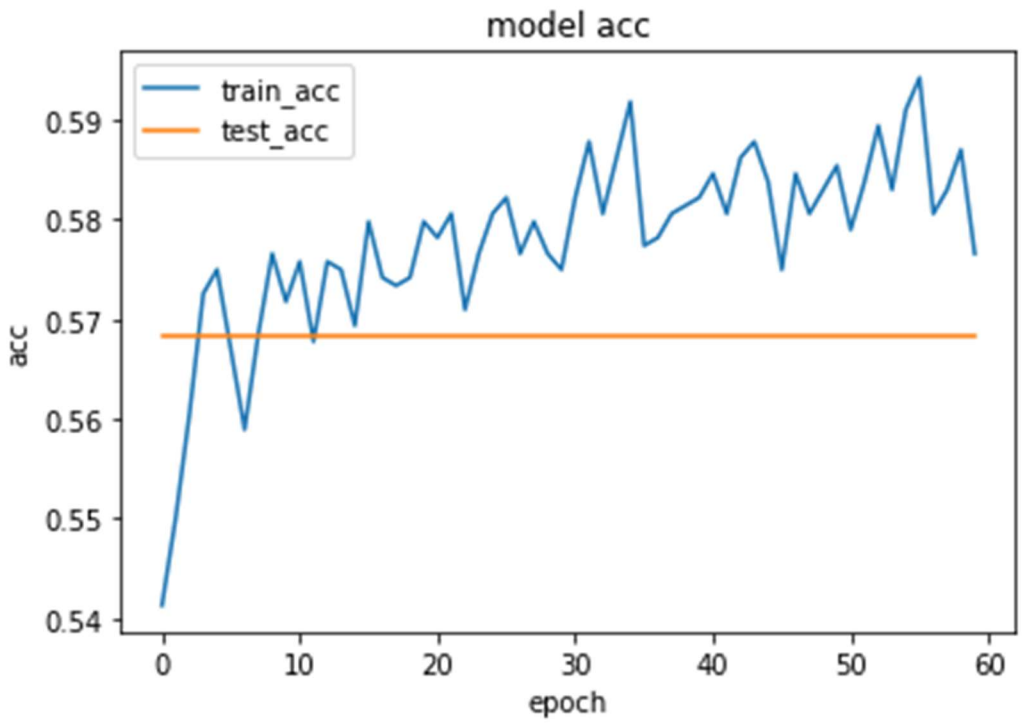
```
prediction
<KerasTensor: shape=(None, 5) dtype=float32 (created by layer 'dense_5')>

model=Model(inputs=Input,outputs=prediction)
model.summary()
```

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
image_input (InputLayer)     [(None, 224, 224, 3)]     0
_________________________________________________________________
flatten (Flatten)            (None, 150528)            0
_________________________________________________________________
dense (Dense)                (None, 1024)              154141696
_________________________________________________________________
dropout (Dropout)            (None, 1024)              0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               524800
_________________________________________________________________
dropout_1 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 128)               65664
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_3 (Dense)              (None, 64)                8256
_________________________________________________________________
dropout_3 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_4 (Dense)              (None, 32)                2080
_________________________________________________________________
dense_5 (Dense)              (None, 5)                 165
=================================================================
Total params: 154,742,661
Trainable params: 154,742,661
Non-trainable params: 0
```

```
#adam=keras.optimizers.SGD(learning_rate=0.00001)
adam=keras.optimizers.Adam(learning_rate=0.00001)
model.compile(loss='categorical_crossentropy',optimizer=adam,metrics=['acc
uracy'])
cnnhistory=model.fit(train_x,train_label,batch_size=32,epochs=60,verbose=2
,validation_data=(valid_x,valid_label))


plt.plot(cnnhistory.history['loss'])
```

```
plt.plot(cnnhistory.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.legend(['train_loss','test_loss'],loc='upper right')
plt.show()
plt.title('model acc')
plt.ylabel('acc')
plt.xlabel('epoch')
plt.plot(cnnhistory.history['accuracy'])
plt.plot(cnnhistory.history['val_accuracy'])
plt.legend(['train_acc','test_acc'],loc='upper left')
plt.show()
```






```
model_name='dr_cnn.h5'
save_dir=os.path.join(os.getcwd(),'save_models')
if not os.path.isdir(save_dir):
  os.makedirs(save_dir)
model_path=os.path.join(save_dir,model_name)
model.save(model_path)
print('Saved trained model at %s'%model_path)

Saved trained model at /content/save_models/dr_cnn.h5

from keras.models import load_model
model=load_model('save_models/dr_cnn.h5')
test_eval=model.evaluate(test_array,y_test,verbose=1)
print('Test loss:',test_eval[0])
print('Test accuracy:',test_eval[1])

11/11 [==============================] - 0s 14ms/step - loss: 1.0253 -
accuracy: 0.5937
Test loss: 1.025274395942688
Test accuracy: 0.5936599373817444
```

Goole Colab link:

https://colab.research.google.com/drive/1WLbDjd3YFYhBtQxhfMl5ZseVU1Ukzl2V?usp=sharing

Chapter 5

Conclusion:-

This project successfully detects the diabetes by using deep learning on a fundus images and it can be used as one of method to detect the diabetes on the future.

The DR stages are based on the type of lesions that appear on the retina. The common fundus DR datasets that are publicly available have been described, and deep-learning techniques have been briefly explained. Most researchers have used the CNN for the classification and the detection of the DR images due to its efficiency. This review has also discussed the useful techniques that can be utilized to detect and to classify DR using DL.

Chapter 6

Bibliograpghy:-

- Saikat Dutt,Subramaniam Chandramouli,Amit Kumar Das(Machine Leraning Book)
- 2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)
- Diabetic retinopathy detection through deep learning techniques: A review paper by ScienceDirect
- Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems 2nd Edition
- Wikipedia
- www.geeksforgeeks.org
- https://youtu.be/pMGLFlgqxuY
- https://youtu.be/LOLeueZnsSo
- https://youtu.be/wMow3mgIpS0
- International Journal of Grid and Distributed Computing Vol. 11, No. 1 (2018), pp.89-106 http://dx.doi.org/10.14257/ijgdc.2018.11.1.09 ISSN: 2005-4262 IJGDC Classification of Diabetic Retinopathy Images by Using Deep Learning Models

THANK YOU!