Dissertation Submitted for the partial fulfillment of the **B.Sc. as a part of M.Sc. (Integrated) Five Years Program AIML/Data Science** degree to the Department of AIML & Data Science.

Project Dissertation

DIABETIC RETINOPATHY DETECTION

submitted to

By

Sneh Shah
Semester-VI

M.Sc. (Integrated) Five Years Program AIML/Data Science

Department of AIML & Data Science.
School of Emerging Science and Technology
Gujarat University

June 2022

DECLARATION

This is to certify that the research work reported in this dissertation entitled "**DIABETIC RETINOPATHY DETECTION**" for the partial fulfilment of B.Sc. as a part of M.Sc. (Integrated) in Artificial Intelligence and Machine Learning/Data Science degree is the result of investigation done by myself.

Place: Ahmedabad

Date:

Name of Student

Sneh Shah

ACKNOWLEGEMENT

On this occasion of submitting my project work, I would like to thank all the people who have made this possible. The role of Department of Data Science, Gujarat University in shaping my professional capability is very important.

I thank Dr. Ravi Gor, coordinator of the Department of AIML, Data Science & Actuarial Science, for all his support.

Rashmi Ma'am my guide has always guided me on the right path and has been a motivating source for innovative research work. This work would have not been possible without her guidance, support, and encouragement. With the help of her guidance, I magnificently overcame the difficulties during my work and learned at each step.

I am heartily and loving thankful to my parents for their sacrifice, support and endless love which encouraged me to efficiently overcome the difficulties in my pursuit of the project and who always helped me in my work.

Last but not the least, I would not be able to complete the work without the blessings of almighty and would like to thank God for giving me strength and patience to carry out my work with full of dedication.

~ Sneh Shah

Index

Chapter 1
Abstract & Key Words

Diabetic Retinopathy (DR) is human eye disease among people with diabetics which causes damage to retina of eye and may eventually lead to complete blindness. Detection of diabetic retinopathy in early stage is essential to avoid complete blindness. Effective treatments for DR are available though it requires early diagnosis and the continuous monitoring of diabetic patients. It is caused by the mutilations of the blood vessels of the light sensitive tissues at the back of the human retina. It is the most recurrent form of blindness in the working age group of people and is highly likely when diabetes is poorly controlled. Although method for detection of Diabetic retinopathy exist, they involve manual examination of the retinal image by an ophthalmologist. Also many physical tests like visual acuity test, pupil dilation, and optical coherence tomography can be used to detect diabetic retinopathy but are time consuming. The objective of this project is to give decision about the presence of diabetic retinopathy by applying different machine learning algorithms and deep learning algorithms also using algorithms on images to extract features from different retinal image. It will give us accuracy of which algorithm will be suitable and more accurate for prediction of the disease. Decision making for predicting the presence of diabetic retinopathy is performed using K-Nearest Neighbor, Random Forest , Support Vector Machine(SVM) , Decision Tree and deep learning algorithms like convolution neural network(CNN). The proposal approach of diabetic retinopathy detection aims to detect the complication using machine learning and deep learning.

Chapter 2
Introduction

Diabetes is a chronic and organ disease that occurs when the pancreas does not secrete enough insulin, or the body is unable to process it properly. Over time, diabetes affects the circular system, including that of the retina. Diabetes retinopathy (DR) is a medical condition where the retina is damaged because of fluid leaks from blood vessels into the retina. It is one of the most common diabetic eye diseases and a leading cause of blindness. Nearly 415 million diabetic patients are at risk of having blindness because of diabetics. It occurs when diabetes damages the tiny blood vessels inside the retina, the light sensitive tissue at the back of the eye. This tiny blood vessel will leak blood and fluid on the retina forms features such as micro-aneurysms, haemorrhages, hard exudates, cotton wool spots or venous loops. Diabetic retinopathy can be classified as non-proliferative diabetic retinopathy and proliferative diabetic retinopathy (PDR). Depending on the presence of features on the retina, the stages of DR can be identified. The disease can advance from mild, moderate to severe stage with various levels of features except less growth of new blood vessels. PDR is the advanced stage where the fluids sent by the retina for nourishment trigger the growth of new blood vessels. They grow along the retina and over the surface of the clear, vitreous gel that fills the inside of the eye. If they leak blood, severe vision loss and even blindness can result.

Currently, detecting DR is a time-consuming and manual process that requires a trained clinician to examine and evaluate digital colour fundus photographs of the retina. By the time human readers submit their reviews, often a day or two later, the delayed results lead to lost follow up, miscommunication, and delayed treatment.

1.1 Objective

The objective of this project is to give decision about the presence of diabetic retinopathy by applying different machine learning algorithms and using deep learning algorithms on images to extract features from different retinal image. Machine learning algorithms such as KNN, RF, SVM, DT etc. can be trained by providing training datasets to them and then these algorithms can predict the data by comparing the provided data with the training datasets. Our objective is to train our algorithm by providing training datasets to it and our goal is to detect diabetic retinopathy using different types of classification algorithms. It will give us accuracy of which algorithm will be suitable and more accurate for prediction of the disease.

1.2 Motivation

I wanted to do my undergraduate thesis on research that will assist a huge amount of people in their healthy lives. The number of people with diabetic retinopathy is growing higher day by day. It is estimated that the number will grow from 126.6 million to 191.0 million by 2030 and the number with vision-threatening diabetic retinopathy (VTDR) will increase from 37.3 million to 56.3 million, if any proper action is not taken . Despite growing evidence documenting the effectiveness of routine DR screening and early treatment, it is frequently leads to poor visual functioning and represents the leading cause of blindness. Most of the time it has been neglected in health care and in many low-income countries because of inadequate medical service. While researching about these factors I get motivated to work with this topic. As there are insufficient ways to detect about diabetic retinopathy, I will try to build a system which will give prediction about diabetic retinopathy. Thus I have decided to use Machine Learning Algorithms for the prediction of this disease.

Chapter 3
Basic Terminology

a) Euclidean distance : Distance between two points (or) the straight line distance. Let us assume that (x1,y1)(x1,y1) and (x2,y2)(x2,y2) are two points in a two-dimensional plane.

The Euclidean distance formula says:

$d = \sqrt{[(x22 – x11)2 + (y22 – y11)2]}$

where,

(x11, y11) are the coordinates of one point.

(x22, y22) are the coordinates of the other point.

d is the distance between (x11, y11) and (x22, y22).

b) Convolution: Convolution is a mathematical operation on two functions (f and g) that produces a third function ( ) that expresses how the shape of one is modified by the other. The term convolution refers to both the result function and to the process of computing it.

c) Epochs: Term used in machine learning and indicates the number of passes of the entire training dataset the machine learning algorithm has completed. Datasets are usually grouped into batches (especially when the amount of data is very large).

d) Maxpooling : Max Pooling is a pooling operation that calculates the maximum value for patches of a feature map and uses it to create a down sampled (pooled) feature map.

e) Flatten : Flattening is used to convert all the resultant 2-Dimensional arrays from pooled feature maps into a single long continuous linear vector. The flattened matrix is fed as input to the fully connected layer to classify the image

f) ReLu: It helps to prevent the exponential growth in the computation required to operate the neural network. If the CNN scales in size, the computational cost of adding extra ReLUs increases linearly.

Chapter 4
Literature Review

This chapter contains literature review related with supervised learning model, classification algorithms like Decision Tree, CNN, random forest, and SVM. This chapter also refers the related works and research. Besides it will give information about our research activity. Machine learning, a branch of artificial intelligence, concerns the construction and study of systems that can learn from data. Machine learning algorithms use computational methods to "learn" information directly from data without relying on a predetermined equation as a model. The algorithms adaptively improve their performance as the number of samples available for learning increases. The core of machine learning deals with representation and generalization. Representing the data instances and functions evaluated on these instances are part of all machine learning systems. Generalization is the ability of a machine learning system to perform accurately on new, unseen data instances after having experienced a learning data instance. The training examples come from some generally unknown probability distribution and the learner has to build a general model about this space that enables it to produce sufficiently accurate predictions in new cases.

Since there are so many algorithms for machine learning, it is not possible to use all of them for analysis. For this research paper, we will be using five of them Convolution neural networks (CNN), random forest (RF),Decision Tree(DT) and support vector machine (SVM), K-nearest Neighbor(KNN).

Neural networks are well-suited to identifying non-linear patterns, as in patterns where there isn't a direct, one-to-one relationship between the input and output. This is a learning training. Neural networks are characterized by containing adaptive weights along paths between neurons that can be tuned by a learning algorithm that learns from observed data to improve model. One must choose an appropriate cost function. The cost function is what is used to learn the optimal solution to the problem being solved. In a nutshell, it can adjust itself to the changing environment as it learns from initial training and subsequent runs provide more information about the world.

Random forest algorithm can use both for classification and the regression kind of problems. It is supervised classification algorithm which creates the forest with several trees. In general, the more trees in the forest the more robust the forest looks like. It could be also said that the higher the number of trees in the forest gives the high accuracy results. There are many advantages of

random forest algorithms. The classifier can handle the missing values. It can also model the random forest classifier for categorical values.

A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules

Support Vector machine constructs a hyper plane or set of hyper planes in a high or infinite-dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyper plane that has the largest distance to the nearest training data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier

K-nearest Neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure. KNN has been used in statistical estimation and pattern recognition. KNN makes prediction for a new instance (x) by searching through the entire training set for the k most similar instances and summarizing the output variable for those k instances. For regression this might be the mean output variable, in classification this might be the mode class determine which of the k instances in the training dataset are most similar to new input many distance measure is used like Euclidean distance, Manhattan distance, Minkowski distance.

4.1 Related Works & Research

Diabetic retinopathy is the leading cause of blindness in the working-age population of the developed world. Since 1982, the quantification of diabetic retinopathy and detection of features such as exudates and blood vessels on fundus images were studied. A lot of work has been done in this field. Before starting implementation of main task we go through similar paper to know about the whole system such as what are the things we need to consider in order to detect diabetic retinopathy. AkaraS. ,Matthew N. Dailey has proposed a “Machine learning approach to automatic exudate detection in retinal images from diabetic patients”[15]. In their paper they presented a series of experiments on feature selection and exudates classification using K- nearest Neighbor (KNN) and support vector machine (SVM) classifiers.

Rajendra Acharya U.,E. Y. K. Ng, Kwan-Hoong Ng and Jasjit S. Suri introduced algorithms for the automated detection of diabetic retinopathy using digital fundus images [16] where they improved an algorithm used for extraction of some features from digital fundus images. Moreover, Varun G. and Lily P. has used deep learning for detection of diabetic retinopathy [3].

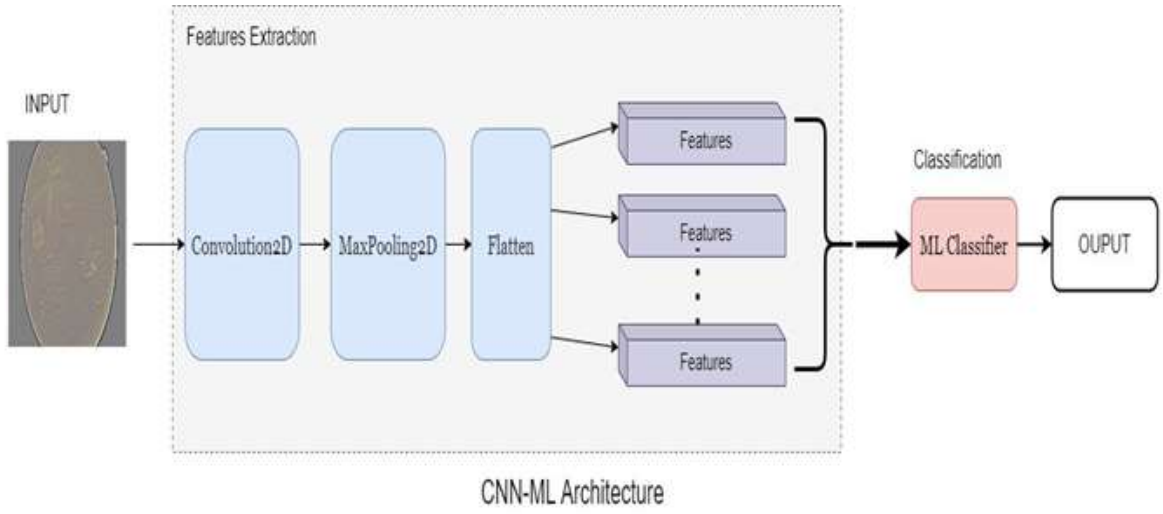
In "Diagnosis of Diabetic Retinopathy using Machine Learning" research paper S. Gupta and K. AM tried to detect retinal micro-aneurysms and exudates retinal funds from images [17]. After pre-processing, morphological operations are performed to find the feature and the features are get extracted such as GLCM and splat for classification. They achieved the sensitivity and specificity of 87% and 100% respectively with accuracy of 86%.

Tiago T.G. in his paper "Machine Learning on the Diabetic Retinopathy Debrecen Dataset" has used R language for predicting diabetic retinopathy [18]. He used a dataset in which the features were extracted from images of the eye of a diabetic patient. In his work he used eight different classification algorithms and also shown some comparisons. He achieved 78% accuracy from his work.

Those are some related paper of our topic from where we took knowledge and idea to develop new version. In our work we will use different machine learning classification algorithms to classify diabetic retinopathy.

Chapter 5
Methodology

5.1 Project Workflow



CNN-ML Architecture

5.2 Data Collection

I have seen different kind of datasets in Kaggle, GitHub and other websites which was used for different kind of projects based on diabetic retinopathy. As I wanted to work with detection of diabetic retinopathy, the dataset I have used will be appropriate for our work as it has been already pre-processed. I have collected the data from Kaggle. The original dataset is available at APTOS 2019 Blindness Detection in Kaggle competition I have directly collected the pre-processed data of the original dataset

5.3 Data Description

The dataset consists of 3362 retina scan images.

The images consist of gaussian filtered retina scan images to detect diabetic retinopathy.

These images are resized into 224x224 pixels so that they can be readily used with many pre-trained deep learning models.

The images are already saved into their respective folders according to the severity/stage of diabetic retinopathy using the train.csv file provided. Here DR is an abbreviation for diabetic retinopathy

You will find five directories with the respective images:

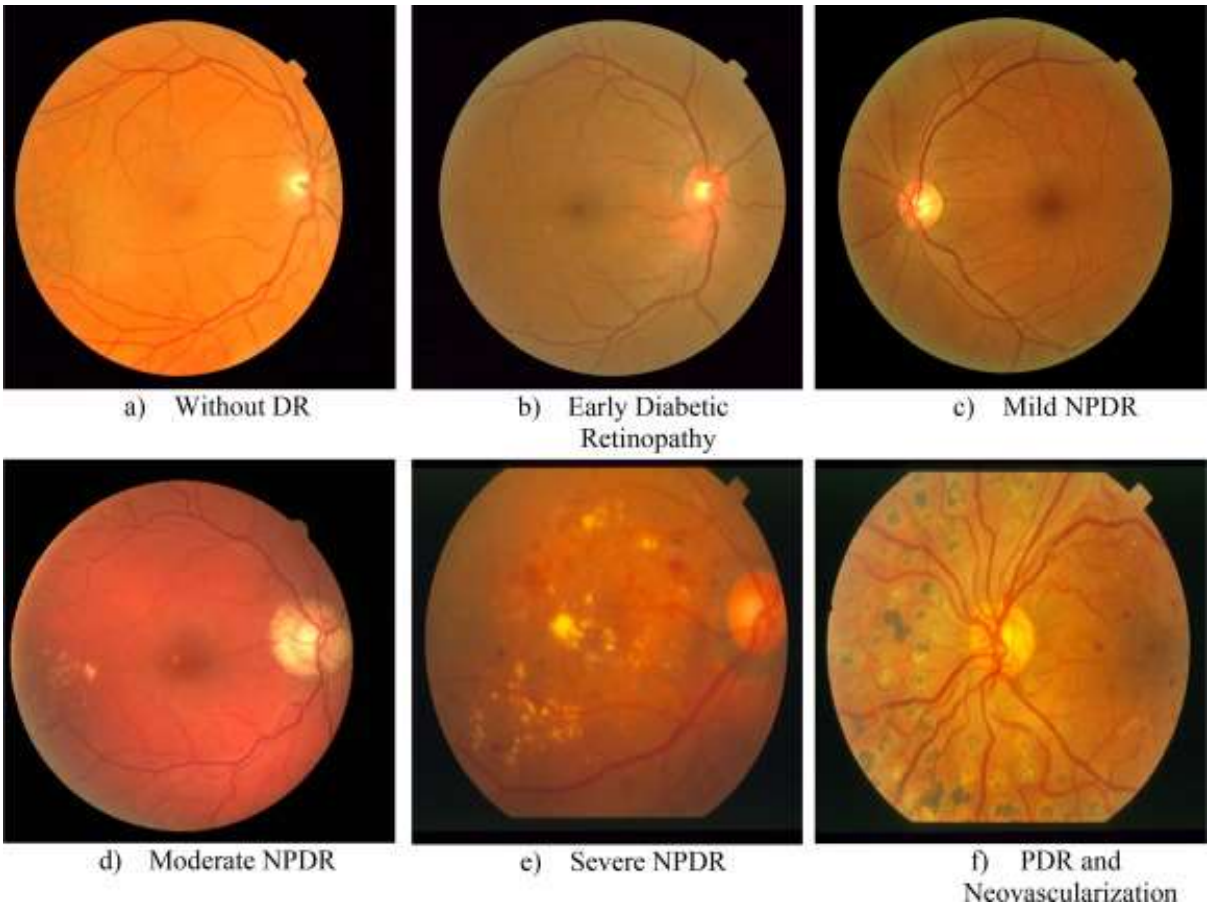
0 - No_DR

1 - Mild

2 - Moderate

3 - Severe

4 -Proliferate_DR



a) Without DR

b) Early Diabetic Retinopathy

c) Mild NPDR

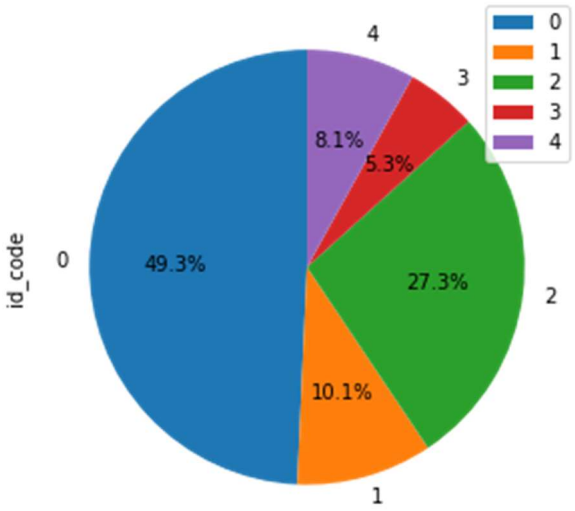
d) Moderate NPDR

e) Severe NPDR

f) PDR and Neovascularization

Chapter 6
Data Analysis

6.1 Data Visualization



As we can see the data is very unbalanced: -

- 50% of the data contains no diabetic retinopathy images.
- 10% contains mild diabetic retinopathy images.
- 27%contains moderate diabetic retinopathy images.
- 5%contains severe diabetic retinopathy images.
- 8% contains proliferative diabetic retinopathy images.

6.2 Split Dataset

Separating data into training and testing sets is an important part of evaluating data mining models. Typically, when separating a data set into two parts, most of the data is used for training, and a smaller portion of the data is used for testing. We have also split our dataset into two sets. One is for training and another for testing. The training set contains a known output and the model learns on this data in order to be generalized to other data later on. After the model has been processed by using the training set, we have tested the model by making predictions against the test set. Because the data in the testing set already contains known values for the attribute that

we want to predict, it is easy to determine whether the model's guesses are correct or not. In addition, we have used 80% of our data for training and 20% for testing.

6.3 Applying Algorithms

6.3.1 Feature Extraction

Since the data has been already pre-processed, the images will now extract features which will be used by the model to predict diabetic retinopathy. Convolution neural network (CNN) has been used to extract features of diabetic retinopathy images.

CNN is a neural network that extracts input image features, and another neural network classifies the image features. The input image is used by the feature extraction network. The extracted feature signals are utilized by the neural network for classification. The neural network classification then works based on the image features and produces the output.

The neural network for feature extraction includes convolution layer piles and sets of pooling layers. As its name implies, the convolution layer transforms the image using the process of the convolution.

The layer of pooling transforms the neighboring pixels into a single pixel. The pooling layer then decreases the image dimension. As CNN's primary concern is the image, the convolution and pooling layers procedures are intuitively in a two-dimensional plane.

6.3.2 Machine learning Algorithms

I went through algorithms that provides better result as we are working on classification of diabetic retinopathy, we used some machine learning classification algorithms. The Machine Learning system uses the training data to train models to see patterns and uses the test data to evaluate the predictive quality of the trained model. Machine learning system evaluates predictive performance by comparing predictions on the evaluation data set with true values using a variety of metrics.

Evaluation of CNN (convolution neural network) and four different machine learning algorithms using the features extracted from CNN

• Support Vector Machine (SVM):

Support Vector Machine constructs a hyper plane or set of hyper planes in a high or infinite-dimensional space, which can be used for classification, regression, or other tasks. SVM belong to the general category of kernel methods

• Decision tree:

A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g., whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes)

• K-Nearest Neighbor (KNN):

K-nearest Neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure. KNN has been used in statistical estimation and pattern recognition.

• Random Forest:

Random Forest algorithm can use both for classification and the regression kind of problems. It is supervised classification algorithm which creates the forest with a number of trees.

Library used:-

```
import keras
import tensorflow
import os
from keras.layers import Input
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from keras.layers import Flatten, Dense, Conv2D, MaxPooling2D, Dropout
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Input
from keras.layers import BatchNormalization
from tensorflow.keras import Sequential
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.preprocessing.image import load_img
from tensorflow.keras.utils import to_categorical
from sklearn.preprocessing import LabelBinarizer
from sklearn.model_selection import train_test_split
from imutils import paths
import matplotlib.pyplot as plt
from keras.models import Model
from sklearn.utils import shuffle
from cv2 import imread
import numpy as np
import pandas as pd
```

Chapter 7
Result & Discussion

CNN MODEL: -

```
h = cnn_model.fit(x_train,y_train,epochs=EPOCHS,validation_split=0.1,verbose=1, batch_size=32)
```

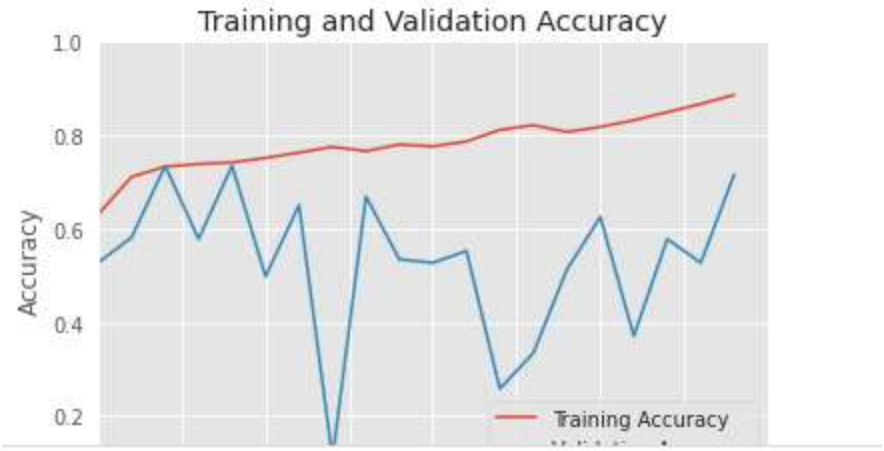
Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_5 (Conv2D)	(None, 148, 148, 16)	448
max_pooling2d_5 (MaxPooling 2D)	(None, 74, 74, 16)	0
conv2d_6 (Conv2D)	(None, 72, 72, 32)	4640
max_pooling2d_6 (MaxPooling 2D)	(None, 36, 36, 32)	0
conv2d_7 (Conv2D)	(None, 34, 34, 64)	18496
max_pooling2d_7 (MaxPooling 2D)	(None, 17, 17, 64)	0
conv2d_8 (Conv2D)	(None, 15, 15, 128)	73856
max_pooling2d_8 (MaxPooling 2D)	(None, 7, 7, 128)	0
conv2d_9 (Conv2D)	(None, 5, 5, 256)	295168
max_pooling2d_9 (MaxPooling 2D)	(None, 2, 2, 256)	0
batch_normalization_1 (Batc hNormalization)	(None, 2, 2, 256)	1024
flatten_1 (Flatten)	(None, 1024)	0
dropout_1 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 1024)	1049600
dense_3 (Dense)	(None, 5)	5125

Total params: 1,448,357
Trainable params: 1,447,845
Non-trainable params: 512

```
acc =  h.history["accuracy"]
val_acc = h.history['val_accuracy']

plt.plot(acc, label='Training Accuracy')
plt.plot(val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.ylabel('Accuracy')
plt.xlim([0,20])
plt.ylim([min(plt.ylim()),1])
plt.title('Training and Validation Accuracy')
plt.xlabel('Epoch')
plt.show()
```




```
pred = cnn_model.predict(x_test)
pred = np.argmax(pred,axis=1)
y_test_new = np.argmax(y_test,axis=1)
print(classification_report(y_test_new,pred))
```

```
              precision    recall  f1-score   support

           0       1.00      0.03      0.06       100
           1       0.49      0.96      0.65       245
           2       0.93      0.92      0.92       433
           3       0.17      0.01      0.02        79
           4       0.00      0.00      0.00        59

    accuracy                           0.70       916
   macro avg       0.52      0.38      0.33       916
weighted avg       0.69      0.70      0.62       916
```

Training Accuracy of SVM Algorithm

```
from sklearn.svm import SVC

svm = SVC(kernel='rbf')
svm.fit(feat_trainCNN,np.argmax(y_train,axis=1))

TrainSVMScoreCNN=svm.score(feat_trainCNN,np.argmax(y_train,axis=1))*100
print("SVM Training Accuracy Score:-",TrainSVMScoreCNN)

TestSVMScoreCNN=svm.score(feat_testCNN,np.argmax(y_test,axis=1))*100
print("\nSVM Testing Accuracy Score:-",TestSVMScoreCNN)

y_pred = svm.predict(feat_testCNN)

cm = confusion_matrix(np.argmax(y_test,axis=1), y_pred)
print('\nConfusion Metrics \n',cm)

print(classification_report(np.argmax(y_test,axis=1),y_pred))
```

```
SVM Training Accuracy Score:- 81.86453022578296

SVM Testing Accuracy Score:- 72.81659388646288

Confusion Metrics
 [[ 29  54  17   0   0]
 [ 18 210  17   0   0]
 [  5   2 426   0   0]
 [ 10  63   6   0   0]
 [  4  48   4   1   2]]
              precision    recall  f1-score   support

           0       0.44      0.29      0.35       100
           1       0.56      0.86      0.68       245
           2       0.91      0.98      0.94       433
           3       0.00      0.00      0.00        79
           4       1.00      0.03      0.07        59

    accuracy                           0.73       916
   macro avg       0.58      0.43      0.41       916
weighted avg       0.69      0.73      0.67       916
```

Training Accuracy of KNN Algorithm

```
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors=5,metric = 'euclidean')
knn.fit(feat_trainCNN,np.argmax(y_train,axis=1))

TrainKNNScoreCNN=knn.score(feat_trainCNN,np.argmax(y_train,axis=1))*100
print("KNN Training Accuracy Score:-",TrainKNNScoreCNN)

TestKNNScoreCNN=knn.score(feat_testCNN,np.argmax(y_test,axis=1))*100
print("\nKNN Testing Accuracy Score:-",TestKNNScoreCNN)

y_pred = knn.predict(feat_testCNN)

cm = confusion_matrix(np.argmax(y_test,axis=1), y_pred)
print('\nConfusion Metrics \n',cm)
print(classification_report(np.argmax(y_test,axis=1),y_pred))
```

```
KNN Training Accuracy Score:- 83.94027676620539

KNN Testing Accuracy Score:- 71.06986899563319

Confusion Metrics
 [[ 42  36  16   4   2]
 [ 39 180  14   8   4]
 [  4  10 418   0   1]
 [ 18  47   8   5   1]
 [ 10  35   6   2   6]]
              precision    recall  f1-score   support

           0       0.37      0.42      0.39       100
           1       0.58      0.73      0.65       245
           2       0.90      0.97      0.93       433
           3       0.26      0.06      0.10        79
           4       0.43      0.10      0.16        59

    accuracy                           0.71       916
   macro avg       0.51      0.46      0.45       916
weighted avg       0.67      0.71      0.68       916
```

Training Accuracy of Random Forest Algorithm

```
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 10, criterion = 'entropy', random_state = 0)
classifier.fit(feat_trainCNN,np.argmax(y_train,axis=1))
TrainRFScoreCNN=classifier.score(feat_trainCNN,np.argmax(y_train,axis=1))*100
print("random forest Training Accuracy Score:-",TrainRFScoreCNN)

TestRFScoreCNN=classifier.score(feat_testCNN,np.argmax(y_test,axis=1))*100
print("\nrandom forestTesting Accuracy Score:-",TestRFScoreCNN)

y_pred = classifier.predict(feat_testCNN)

cm = confusion_matrix(np.argmax(y_test,axis=1), y_pred)
print('\nConfusion Metrics \n',cm)
print(classification_report(np.argmax(y_test,axis=1),y_pred))
```

```
random forest Training Accuracy Score:- 98.2884195193008

random forestTesting Accuracy Score:- 70.96069868995633

Confusion Metrics
 [[ 34  41  20   4   1]
 [ 22 194  18   5   6]
 [  4  12 414   1   2]
 [ 16  47   9   4   3]
 [  7  37   6   5   4]]
              precision    recall  f1-score   support

           0       0.41      0.34      0.37       100
           1       0.59      0.79      0.67       245
           2       0.89      0.96      0.92       433
           3       0.21      0.05      0.08        79
           4       0.25      0.07      0.11        59

    accuracy                           0.71       916
   macro avg       0.47      0.44      0.43       916
weighted avg       0.65      0.71      0.67       916
```

Training Accuracy of Decision Tree Algorithm

```
from sklearn.tree import DecisionTreeClassifier

clf = DecisionTreeClassifier(criterion = 'gini',random_state=0)
clf = clf.fit(feat_trainCNN,np.argmax(y_train,axis=1))

TrainDecisionScoreCNN=clf.score(feat_trainCNN,np.argmax(y_train,axis=1))*100
print("Decision Tree Training Accuracy Score:-",TrainDecisionScoreCNN)


TestDecisionScoreCNN=clf.score(feat_testCNN,np.argmax(y_test,axis=1))*100
print("\nDecision Tree Testing Accuracy Score:-",TestDecisionScoreCNN)
y_pred = clf.predict(feat_testCNN)

cm = confusion_matrix(np.argmax(y_test,axis=1), y_pred)
print('\nConfusion Metrics \n',cm)
print(classification_report(np.argmax(y_test,axis=1),y_pred))
```

```
Decision Tree Training Accuracy Score:- 99.2352512745812

Decision Tree Testing Accuracy Score:- 64.8471615720524

Confusion Metrics
 [[ 40  34  12   9   5]
 [ 27 146  17  32  23]
 [  9  16 387  13   8]
 [ 12  38   6  15   8]
 [  8  29   6  10   6]]
              precision    recall  f1-score   support

           0       0.42      0.40      0.41       100
           1       0.56      0.60      0.57       245
           2       0.90      0.89      0.90       433
           3       0.19      0.19      0.19        79
           4       0.12      0.10      0.11        59

    accuracy                           0.65       916
   macro avg       0.44      0.44      0.44       916
weighted avg       0.65      0.65      0.65       916
```

Comparison between Algorithms

```
print("--Training Accuracy..")
print("CNN Accuracy:- {:.2f} %".format(trainCNNScore))
print("CNN-SVM Accuracy:- {:.2f} %".format(TrainSVMScoreCNN))
print("CNN-DT Accuracy:- {:.2f} %".format(TrainDecisionScoreCNN))
print("CNN-KNN Accuracy:- {:.2f} %".format(TrainKNNScoreCNN))
print("CNN-RF Accuracy:- {:.2f} %".format(TrainRFScoreCNN))

print("\n--Testing Accuracy..")
print("CNN Accuracy:- {:.2f} %".format(CNNScore))
print("CNN-SVM Accuracy:- {:.2f} %".format(TestSVMScoreCNN))
print("CNN-DT Accuracy:- {:.2f} %".format(TestDecisionScoreCNN))
print("CNN-KNN Accuracy:- {:.2f} %".format(TestKNNScoreCNN))
print("CNN-RF Accuracy:- {:.2f} %".format(TestRFScoreCNN))
```

```
--Training Accuracy..
CNN Accuracy:- 76.18 %
CNN-SVM Accuracy:- 81.86 %
CNN-DT Accuracy:- 99.24 %
CNN-KNN Accuracy:- 83.94 %
CNN-RF Accuracy:- 98.29 %

--Testing Accuracy..
CNN Accuracy:- 69.65 %
CNN-SVM Accuracy:- 72.82 %
CNN-DT Accuracy:- 64.85 %
CNN-KNN Accuracy:- 71.07 %
CNN-RF Accuracy:- 70.96 %
```

Chapter 8
Conclusion

We have tried to construct an ensemble to predict if a patient has diabetic retinopathy using features from retinal photos. After training and testing the SVM with CNN features is providing higher accuracy rate for predicting DR followed by KNN with CNN features. As the dataset of images was unbalanced, we can see that there was overfitting of models. To improve the model accuracy and reduce the overfitting we will have to balance the data where all types of diabetic retinopathy have equal or near to equal number of images.

I have also faced some problems while choosing algorithms. It was quite difficult for me to choose some specific machine learning algorithms that would give accurate classification of the disease. Possibility there be other parameter spaces that would yield better performing models.

I have also faced the computational problem which lead to less model training as the GPU and the ram of computer is every less. For any research, there is always room for improvement.

We have found some areas where this system can be improvised:

1. Work on more Categories: This can be improvised with a lot more categorized such as according to ages, genders, background studies, working facilities and so on. As an example, A matured man from the IT background has different eye condition that a matured women from Teaching background.

2. Work on more classes: As we working on only two classes whether it is good or bad. In future we are going to add more classes like low, medium, severe condition. In this way patients can know about their condition more accurately

3. Different Algorithms: CRF (Conditional Random Field), maximum entropy and other probabilistic graphical model can also be used to train our dataset in order to improve the algorithm.

4. More Analysis: To achieve more accuracy we could use more dataset. If we use huge amount of dataset, machine will train more and it would give us more accurate prediction and accuracy.

5. Hardware Implementation: A hardware product can be the best solution for patient. So, we are looking forward to build a hardware system where we can use our model to implement results on diabetic patients easily. We can then input the data of the patient and wait for the machine to create a new prescription integrated with Doctor's suggestion.

6. Software Implementation: We can build a website or an android app for this purpose. In this way patient will be able to upload their data into our server and our machine learning software will let them know about their disease through our website whether it is in a good or bad condition.

Chapter 9
Bibliography

[1] https://www.kaggle.com/datasets/sovitrath/diabetic-retinopathy-224x224-gaussian-filtered

[2] https://www.sciencedirect.com/topics/computer-science/feature-extraction-network

[3] https://ieeexplore.ieee.org/abstract/document/6680633

[4] http://ictactjournals.in/paper/IJSC_V3_I4_Paper_1_563-575.pdf

[5] Maisha Maliha ,Ahmed Tareque ,Sourav Saha Roy. “Diabetic Retinopathy Detection Using Machine Learning”. Published on 04,2018

[6] Jason B, Boinee P.“Machine Learning Algorithms”2(3), 138–147. Published on 15, 2016.

[7] Boser B. E, Guyon I. M.,Vapnik V. N. (1992). “A training algorithm for optimal margin classiers”.Proceedings of the 5th Annual Workshop on Computational Learning Theory COLT'92, 152 Pittsburgh, PA, USA. ACM Press, July 1992. On Page(s): 144-152

[8] Wikipedia.http://en.wikipedia.org/wiki/Support_vector_machine.

[9] Ben-Hur.A, Weston.J (2009) ."A User's Guide to Support Vector Machines". Data Mining Techniques for the Life Science.Humana Press. On Page(s): 223-239.

[10] Breiman. Pasting small votes for classification in large databases and on-line. Machine Learning, 36(1-2):85–103, 1999. (Cited on pages 169, 170, and 187.)

[11] Yau JW, Rogers SL, Kawasaki R, Lamoureux EL, Kowalski JW, Bek T, et al. Global prevalence and major risk factors of diabetic retinopathy. Diabetes Care. 2012;35(3):556–.

[12] Cheung N, Mitchell P, Wong TY. Diabetic retinopathyLancet. 2010;376(9735):124–36. doi: 10.1016/S0140-6736(09)62124-3