Detection of Diabetic Retinopathy from Retinal images

Sunayana Hubli
University of Massachusetts, Lowell
sunayana_hubli@student.uml.edu

Abstract **- Diabetic retinopathy is a diabetes complication that affects eyes. It's caused by damage to the blood vessels of the light-sensitive tissue at the back of the eye (retina). Detection of this disease has been predominantly done by the physician during retinal examination or scans. Detection relies on skilled readers and is both labor and time intensive. An automated system can assist in a more accurate and quicker detection of the disease. In this project a comparative study of different machine learning algorithms for the detection of diabetic retinopathy from retinal images has been performed. Some of the algorithms performed are KNN, SVM, Neural Networks, etc. The performance of each of these techniques are measured and analyzed.**

Index Terms – Image Classification, Neural Network, Diabetic Retinopathy, CNN

I. Introduction

The goal of this project is to do a comparative study of different machine learning algorithms for the detection of diabetic retinopathy from retinal images. Detection of this disease is a time-consuming and typical a process that requires a trained expert to go over the retinal images manually. This process usually leads to delay in follow-up and treatment
Clinicians can identify DR by the presence of lesions associated with the vascular abnormalities caused by the disease. While this approach is effective, its resource demands are high. The expertise and equipment required are often lacking in areas where the rate of diabetes in local populations is high and DR detection is most needed[1]. As the number of individuals with diabetes continues to grow, the infrastructure needed to prevent blindness due to DR will become even more insufficient.

In this paper we are using retinal images to detect the presence of diabetic retinopathy. We will apply multiple techniques like Naïve Bayes, Support Vector Machine, KNN, Neural Network, Resnet. Their performance is assessed and analyzed.

II. Background

The current research related to automated detection of diabetic retinopathy is done using deep learning, convolution neural network and propriety artificial intelligence algorithms from companies like Google and Eyenuk EyeArt[2][3]. There have also been various surveys, research and competitions regarding the same.

III. Data and Approach

A. Dataset

The images are sourced from Kaggle. The dataset contains retinal color images that have been resized to 224 x 224 pixels. It contains a total of 35126 images. It has the following categories: Mild, Moderate, Proliferate, Severe and No DR. Distribution of the images are shown below:

Figure 1: Data Distribution

As observe, there is data imbalance, with the data has more samples of retnal images without Diabetic Retinopathy than the ones with.

B. Image Processing:

Following are some of the data pre-processing done during the implementation of certain techniques. A combination of these were utilized through trial and error or based on the rationale for the specific techniques used for detection of the disease.

- Data size modifications
  - To offset the data imbalance, the samples from No Dr category were reduced
  - Under sampled categories were updated by data augmentation
- Converted to Grayscale
  - Images were converted to grayscale
- Increasing contrast
  - Rescalin and contrast modifications were done
- Resizing the images
  - All images are of size 224 x 224 pixels.

C. Approach

The approach used in this paper is described at a high level in Figure 2

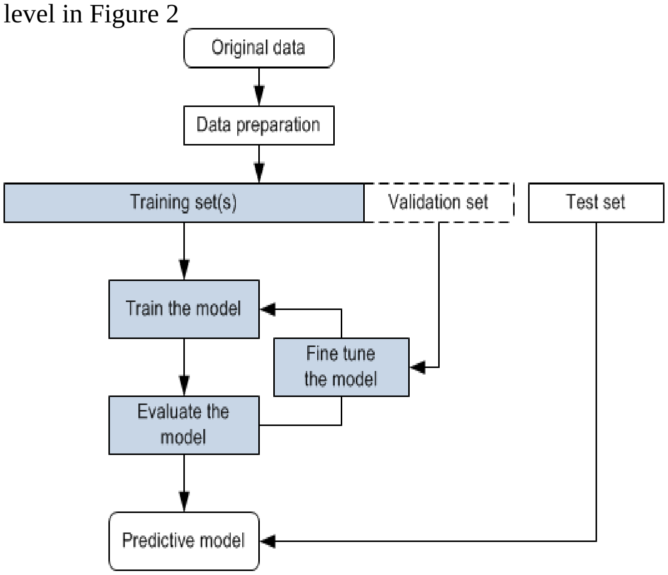


Figure 2: Flowchart for the general approach or each algorithm

IV. METHODOLOGY

A. Naïve Bayes

Naive Bayes Algorithm calculates the conditional probability of all classes and, because it is naive, **it will assume that every feature is independent** of one another. Baye's Theorem. it is a powerful tool in the field of probability, Bayes Theorem is also widely used in the field of machine learning. Including its use in a probability framework for fitting a model to a training dataset, referred to as maximum a posteriori or MAP for short, and in developing models for classification predictive modelling.

Baye's Theorem assume there is no correlation between any feature and so, their contribution to predicting the class is not impacted by other features.

This changes the model from a dependent conditional probability model to an independent conditional probability model and dramatically simplifies the calculation.
This means that we calculate P(data|class) for each input variable separately and multiple the results together, for example:

$$P(\text{class} \mid X1, X2, \ldots, Xn) = P(X1|\text{class}) * P(X2|\text{class}) * \ldots * P(Xn|\text{class}) * P(\text{class}) / P(\text{data})$$

We can also drop the probability of observing the data as it is a constant for all calculations, for example:

$$P(\text{class} \mid X1, X2, \ldots, Xn) = P(X1|\text{class}) * P(X2|\text{class}) * \ldots * P(Xn|\text{class}) * P(\text{class})$$

The word "*naive*" is French and typically has a diaeresis (umlaut) over the "i", which is commonly left out for simplicity, and "Bayes" is capitalized as it is named for Reverend Thomas Bayes.[4]

Naïve Bayes is the simplest classifier, which used the language of graphical models. This method assumes that each category has its own distribution over the codebook, and the distribution of each category is observably different from those of others.

Results:

- When we partition the dataset with a ratio of 80:20, which means 80% of the data goes to training and the remaining 20% is for testing, 62% for Naïve Bayes.

	precision	recall	f1-score	support
0	0.63	0.93	0.75	734
1	0.27	0.04	0.07	426
accuracy			0.61	1160
macro avg	0.45	0.49	0.41	1160
weighted avg	0.50	0.61	0.50	1160

Figure 3: Naïve Bayes Performance Metric

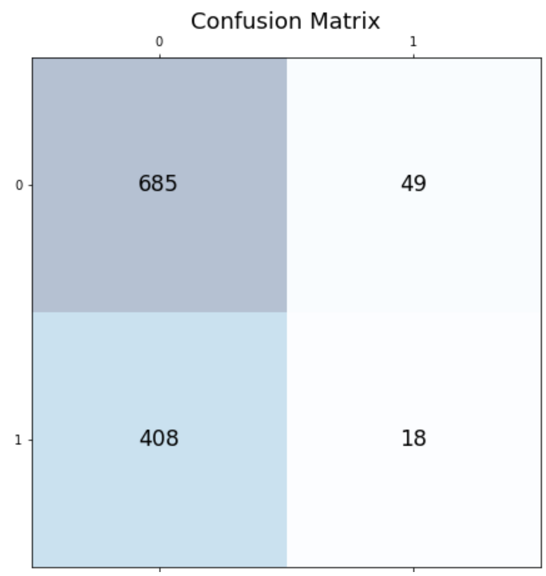


Figure 4: Naïve Bayes Confusion Matrix

B. KNN

The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another.[5]

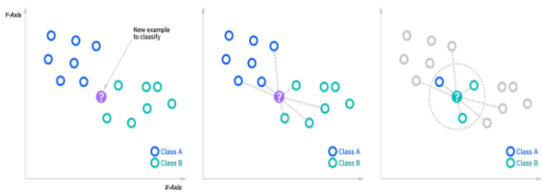


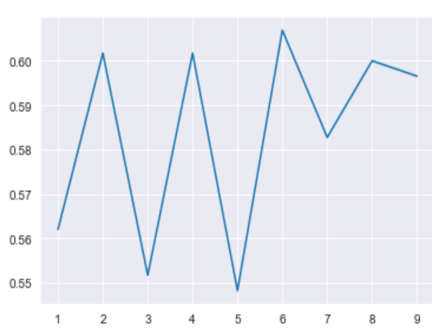
Figure 5: K-NN Classification

For classification problems, a class label is assigned on the basis of a majority vote—i.e. the label that is most frequently represented around a given data point is used. While this is technically considered "plurality voting", the term, "majority vote" is more commonly used in literature

KNN is easy to implement and adapts easily but is prone to overfitting. It also not always a great fit for image classification as owing to the large number of features, the KNN faces the issue of curse of dimensionality.

Results:
kNN classifier identifies the class of a data point using the majority voting principle. If k is set to 6, the classes of 6 nearest points are examined. Prediction is done according to the predominant class. Similarly, kNN regression takes the mean value of 6 nearest locations.[6]

```
Using SKLEARN
K: 1, Score:0.5620689655172414
K: 2, Score:0.6017241379310345
K: 3, Score:0.5517241379310345
K: 4, Score:0.6017241379310345
K: 5, Score:0.5482758620689655
K: 6, Score:0.6068965517241379
K: 7, Score:0.5827586206896552
K: 8, Score:0.6
K: 9, Score:0.596551724137931
```




```
max acc at k=6 acc of 0.6068965517241379
```

Figure 5: Selection of value K

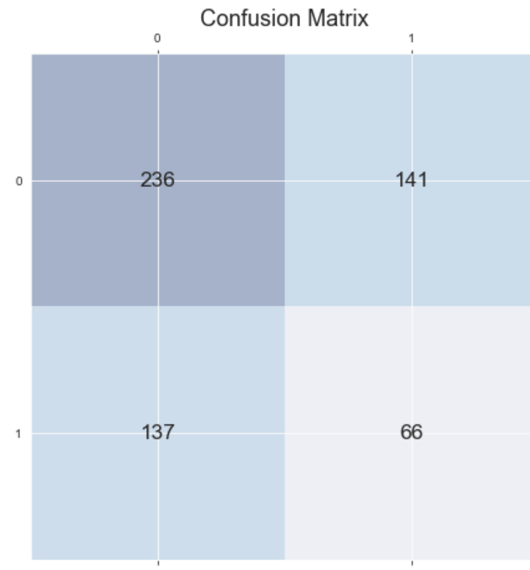


Figure 6: KNN Confusion Matrix

C. SVM

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points. To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence. Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM. SVM is a very good algorithm for doing classification. It's a supervised learning algorithm that is mainly used to classify data into different classes. SVM trains on a set of label data. The main advantage of SVM is that it can be used for both classification and regression problems. SVM draws a decision boundary which is a hyperplane between any two classes to separate them or classify them. SVM also used in Object Detection and image classification.

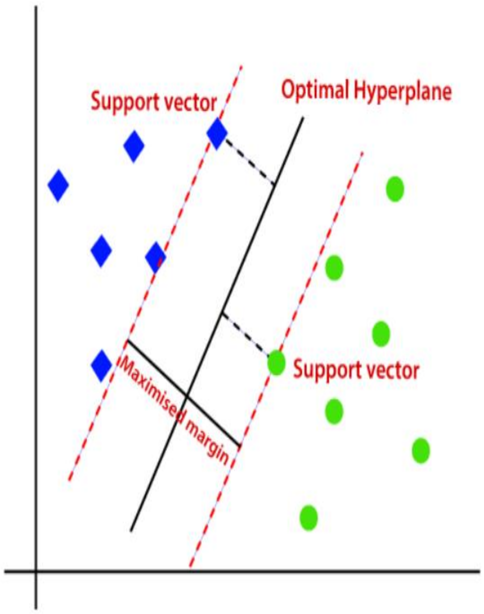


Figure 7: SVM Hyperplane

D. Convoluted Neural Network

An artificial neural network learning algorithm, or neural network, or just neural net, is a computational learning system that uses a network of functions to understand and translate a data input of one form into a desired output, usually in another form. he neural net learning algorithm instead learns from processing many labeled examples (i.e. data with with "answers") that are supplied during training and using this answer key to learn what characteristics of the input are needed to construct the correct output. Once a sufficient number of examples have been processed, the neural network can begin to process new, unseen inputs and successfully return accurate results.[8]

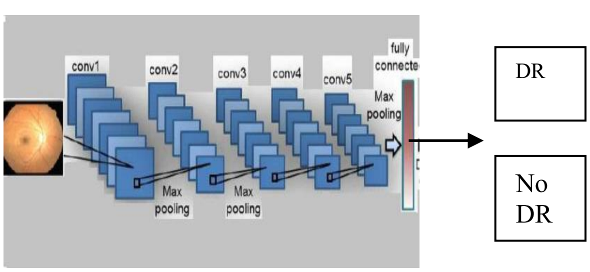


Figure 8: CNN Architecture

In this implementation Max Pooling is used with every convolution. A total of 5 Convultion layers are implemented. Relu function is used as activation in the hidden layers and SoftMax for the final.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 222, 222, 34)	340
batch_normalization (BatchNormalization)	(None, 222, 222, 34)	136
max_pooling2d (MaxPooling2D)	(None, 111, 111, 34)	0
dense (Dense)	(None, 111, 111, 84)	2940
max_pooling2d_1 (MaxPooling2D)	(None, 27, 27, 84)	0
dense_1 (Dense)	(None, 27, 27, 160)	13600
max_pooling2d_2 (MaxPooling2D)	(None, 13, 13, 160)	0
flatten (Flatten)	(None, 27040)	0
dense_2 (Dense)	(None, 100)	2704100
dense_3 (Dense)	(None, 2)	202

Total params: 2,721,318
Trainable params: 2,721,250
Non-trainable params: 68

Table 1: CNN Model Summary

Results:

Accuracy score is : 0.72
Precision score is : 0.54
Recall score is : 0.65

E. Residual Neural Network

Residual Network (ResNet) is one of the famous deep learning models that was introduced by Shaoqing Ren, Kaiming He, Jian Sun, and Xiangyu Zhang in their paper. The paper was named "Deep Residual Learning for Image Recognitionin 2015. The ResNet model is one of the popular and most successful deep learning models so far.

Using ResNet with Keras: Keras is an open-source deep-learning library capable of running on top of TensorFlow. Keras Applications provides the following ResNet versions. ResNet50 ResNet architecture uses the CNN blocks multiple times. Wecreate a ResNet class that takes the input of a number of blocks, layers, image channels, and the number of classes.

```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50 (Functional)       (None, 7, 7, 2048)        23581440

 flatten_3 (Flatten)         (None, 100352)            0

 dense_6 (Dense)             (None, 16)                1605648

 batch_normalization_3 (Batc  (None, 16)               64
 hNormalization)

 activation_3 (Activation)   (None, 16)                0

 dropout_3 (Dropout)         (None, 16)                0

 dense_7 (Dense)             (None, 2)                 34

=================================================================
Total params: 25,187,186
Trainable params: 1,605,714
Non-trainable params: 23,581,472
_________________________________________________________________
```

Table 2: ResNet Model Summary

Results:

Accuracy score is : 0.74
Precision score is : 0.5476
Recall score is : 0.74
F1 Score is : 0.6294252873563217

V. ◦RESULTS

The measurement of an accuracy for the network architecture is estimated by correctly classified DR suffered images from the pool of images in the different dataset. A whole objective is to minimizing the cost function of the deep convolutional neural network results significantly reflected in the testing datasets. In terms of diabetic retinopathy performance measurements Accuracy(Acc) is the crucial parameters for deciding the algorithms. Four parameters which take part in measuring those performances are :

True Positive(TP) - Correctly detected DR images
True Negative(TN) - Correctly detected Non-DR images
False Positive(FP) - Number of Non-DR images are detected wrongly as DR images
False Negative(FN) - Number of DR images are detected wrongly as Non-DR images

This is displayed in Confusion Matrix.

Hardware and Software requirements : For augmentation, Image editor tool is used for contrast adjustment, color balance adjustment, rotate or cropping. At pre-processing stage, monochrome conversion is done with the NumPy package. Convolutional Neural Network(CNN), multi-layer deep architecture are implemented using TensorFlow Keras libraries..
Dataset : Kaggle dataset[11] : A high-resolution retina images taken under a variety of imaging conditions. A clinician rated the presence of diabetic retinopathy and scale it as 0-4. It contain 35126 images. TEST dataset[12] : This database contains color eye fundus images used for validation.

Algorithm	Test Accuracy %	Accuracy with different dataset
Naïve Bayes	60.603	57.34
KNN	62.84	55.5
SVM	68.66	63.33
CNN	72.55	61.45
ResNet	74.3	67.4

Table 3: Accuracy Comparision

VI. CONCLUSION

ResNet and CNN display the best accuracy and performance amongst all the techniques used. SVM performs comparably and can act as a relevant classifier in detection of Diabetic Retinopathy. The traditional methods of KNN and Naïve Bayes perform well only with limited dataset and do not fare well when the data set is large or when predicting with new dataset i.e.., from a different source.

VII. FUTURE WORK

Implementation of AlexNet, Inception and other Deep learning techniques. Deep learning has seen increased performance when compared to traditional methods for image classification.

Move to Categorical prediction of disease. Currently we are only detecting the presence of the disease, it would be beneficial to detect the severity of the disease as well.

REFERENCES

[1] "Diabetic retinopathy" in "https://www.mayoclinic.org/diseases-conditions/diabetic-retinopathy/symptoms-causes/syc-20371611
[2] Dai, L., Wu, L., Li, H. et al. A deep learning system for detecting diabetic retinopathy across the disease spectrum. Nat Commun 12, 3242 (2021). https://doi.org/10.1038/s41467-021-23458-5
[3] https://www.sfchronicle.com/business/article/Google-sets-sights-on-frontier-of-artificial-12894136.php
[4] "A Gentle Introduction to Bayes Theorem for Machine Learning" by Jason Brownlee on https://machinelearningmastery.com/bayes-theorem-for-machine-learning/
[5] "Machine Learning Basics with the K-Nearest Neighbors Algorithm" by Onel Harrison on https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

[6] “K-Nearest Neighbors (KNN) in Python” By Isha Bansal on https://www.digitalocean.com/community/tutorials/k-nearest-neighbors-knn-in-python
[7] “Build an Image Classifier With SVM!” by Bahauddin Taha on https://www.analyticsvidhya.com/blog/2021/06/build-an-image-classifier-with-svm/
[8] “Neural Network” on https://deepai.org/machine-learning-glossary-and-terms/neural-network
[9] “Metrics and scoring: quantifying the quality of predictions” https://scikit-learn.org/stable/modules/model_evaluation.html
[10] “Build ResNet from Scratch With Python!” by Syed Abdul Gaffar Shakhadri https://www.analyticsvidhya.com/blog/2021/06/build-resnet-from-scratch-with-python/
[11] https://www.kaggle.com/datasets/sovitrath/diabetic-retinopathy-2015-data-colored-resized
[12] https://www.kaggle.com/datasets/gunavenkatdoddi/eye-diseases-classification

APPENDIX

Tasks	**Planned Timeline**	**Actual**
Data preprocessing	Oct 20th	Oct 20th
Naïve Bayes method	Oct 24th	Oct 24th
KNN	Nov 10th	Oct 10th
SVM	Nov 10th	Nov 10th
Neural network	Nov 20th	Nov 28th
Tuning Models	Nov 25th	Dec 8th
Comparative study and conclusion	Nov 30th	Dec 11th
Report Completion	Dec 2nd	Dec 10th
Project Presentation	Dec 2nd	Dec 12th