CLASSIFYING RETINA OCT IMAGES FOR THE PRESENCE OF EYE DISEASE WITH SUPPORT VECTOR MACHINE AND NEURAL NETWORK

[1]Mojeed Ajegbile, Bolanle Hafiz Matti[2], Adesoji Bello[3], Andrew Oguns[4]

[1]Department of Mathematics and Statistics, Austin Peay State University, Tennessee, USA

[1]Department of Mathematics and Statistics, Austin Peay State University, Tennessee, USA

[2]Department of Electrical and Computer Engineering, Michigan Technological University, USA

[3]Department of Electrical and Computer Engineering, Iowa State University, USA

ABSTRACT

Image classification is a fundamental challenge encountered in computer vision and data analysis. Images are unstructured data and sometimes hard to analyze because of the arduous feature extraction steps involved. Some algorithms often used in image classification tasks include minimum K-Nearest neighbor, Clustering, Fuzzy C -Means.

However, advancements in deep learning allow deeper feature extraction and better representations of image data and has shown it is more efficient and has wider applications in computer vision like classification and image detection. Support vector machine algorithm too is strong and useful in image classification. Even though it’s normally regarded as a binary classification technique, it can be tweaked and extended for multi-label classification tasks. Deep learning so far has provided successful results for machine learning problems. In this paper, we compare the accuracy of image classification performed using convolutional neural network which is became standard with the accuracy of image classification done with support vector machine which uses principal component analysis for feature extraction. The image data used are Retina optical coherence tomography (OCT) with four classes, three classes have diseased retina while one class is normal. The aim is to observe which classification algorithm performs better in identifying an image with a disease. At the end of this project, it was discovered that deep convoluted neural network performed better than other machine learning approaches used in classifying the OCT images.

INTRODUCTION

Retinal degenerative diseases are leading causes of blindness in the USA and worldwide. Age related retina blindness is expected to rise in the USA to 41% of total blindness cases. With early diagnosis, these diseases can be prevented and treated or else it can culminate into permanent eye damage and blindness. The picture of the retina can be taken using a technique called Optical coherence Tomography (OCT). This is an imaging test that uses.

measured quantum of light waves capable of penetrating the distinctive layers of the retina. The thickness of these layers can be measured using this technique. This helps in diagnosis, to determine the presence of abnormalities in the layers of the retina.

Optical coherence tomography (OCT) is a recently developed technology for performing high-quality cross-sectional imaging of the eye. OCT is like ultrasound imaging, but instead uses light rather than sound.

Deep learning is a specific subfield of machine learning, a novel technique of learning which is designed for better representations of data by enforcing learning from successive layers of data. According to [9], The *deep* indicates the concept of consecutive layers of representations. The *depth* of the model is the number of layers that contributes to the model. These layered representations are learned using *neural networks*, structured in l layers stacked on top of each other.
The term *neural network* originates from

neurobiology, but deep-learning models are less intricate compared to the models of the brain.

According to [8], the forerunner to deep learning is the multi-layer perceptron. It is simply a logistic regression classifier. The input undergoes a nonlinear transformation. This transformation projects the input data into a linearly separable space. This intermediate layer is referred to as a hidden layer.

The set of parameters to learn can be computed using a back propagation algorithm.

L*earning* is achieved by finding a set of values for the weights at all layers in the network, in such a way that the network will appropriately map an input vector to its corresponding targets [8]. This is the job of a *loss function*. The loss function takes the predicted output of the network and the true target and compares them by computing a score which represents the distance between them. This measures how good the network has done in predicting the outcome.

SVM objective is to attempt classification problems by finding *decision boundaries* between two sets of points
belonging to two different categories [5]. A decision boundary is an imaginary line or a barrier separating training data into two or more spaces. Classifying new data points is attained by checking which side of the boundary the training sets fall on. Kernel methods map data to a higher dimensional representation if data is not linearly separable.

The main objective of this project is to utilize different algorithms to train retina OCT images to be able to predict the presence of diseases on the retina layers. This is an image classification problem. At the end of the project, we want to create a machine learning framework that can classify diseased retina or normal retina with accuracy comparable to that of a trained ophthalmologist. The performance of the different algorithms used in the classification will be compared and the best performing one is identified.

LITERATURE REVIEW.

The data is from medical (X-ray) images, which can be converted into pixel matrix and color channels, then fed into our algorithms. We anticipate the implementation of many algorithms and compare their accuracy but for the sake of brevity, Support vector machines and deep learning with neural network are focused on.

SVM is used for supervised learning tasks. They originate from statistical learning [5], they are precise and efficient with small training samples. They are mostly used in binary classification but can be extended to multi-class classification [4]. SVM uses kernel methods to project the training data in the input space to a feature space of higher dimension if the data is not linearly separable. Also, the objective of the classifier is to maximize the margin between the data points and the decision boundaries [4].

According to [3], Neural Networks are the most used algorithms in creating deep learning models. Convolutional neural networks have shown high performance on image classification. For this project, a stochastic gradient descent algorithm will be employed to solve optimal parameters, while experimenting with different learning rates. According to [1], stochastic gradient descent is capable of large-scale machine learning because it has better computational complexity which is a limiting factor for other optimization techniques.

FEATURE EXTRACTION

Preprocessing

The method involved in the preprocessing step involved the following: the high-resolution images are converted to a lower resolution of 224×224 gray scaled image arrays These inputs are standardized using max-min normalization.
As a second preprocessing step, we reshaped the numPy arrays as needed to fit into the requirement of the training algorithms.

For deep learning algorithms, features were extracted with convoluted neural architecture.

Convolutional Layer: The function of convolutional layer is basically featuring extraction. It consists of many convoluted layers followed by one or more fully connected layers. According to [2], this arrangement is to support two-dimensional structure of input such as images. To maintain the image dimension from decreasing, we add zeros. The process of adding zeros is known as padding.

Pooling layer: [2] also explained that pooling layer functions to reduce the dimension of the feature map for faster processing time. It reduces the spatial size (height, width) which results in reduces parameters to be learned [2].

Flattening Layer: convolutional layer returns a multidimensional Tensor. This must be reshaped to a single dimension tensor. Hence a flattening layer is needed. This is done using by reshaping. Numpy can be used for this reshaping operation.

Fully connected layer: This layer that does the classification of the image based on the features already extracted by the previous convolutions [2]. All the neurons relate to every neuron in the previous layer. A probability function like SoftMax is required to map the output of neural network into a probability for each class [2].

For support vector classification, principal component analysis and whitening were used for feature extraction. The idea behind it is to reduce the dimension by finding a direction (vector) onto which to project the data to minimize the projection error. To implement a PCA algorithm, we normally

(1) perform mean normalization and feature scaling

(2) calculate a covariance matrix (sigma) with the following formula

(3) use singular value decomposition (svd) on sigma

(4) multiply the transpose of the first k columns of the resulting U matrix with the feature vector x that shall be reduced and return the resulting z feature vector

CLASSIFICATION ALGOITHMNS USED

K-MEANS CLUSTERING

The *k*-means algorithm finds clusters in an unsupervised dataset.

k-means is one of the simplest unsupervised machine learning algorithms.

Support Vector Machine

According to [5], Support vector machine technique finds a hyper plane in limitless dimensional space. The nearest data point to this hyperplane is called functional margin [5].

The non-linear classifier in SVM is based on kernel function and it was proposed by [5]. It is similar in concept but replaces the dot product by a non-linear kernel function [6].

Multiclass SVM

Multiclass SVM aims to assign labels to instances by using SVM, where the labels are more than two.

Several methods have been propounded for this method, there can be combination of many binary classifiers. Some methods considered all classes at once [6]. The last approach is computationally expensive to solve multiclass.

METHODS

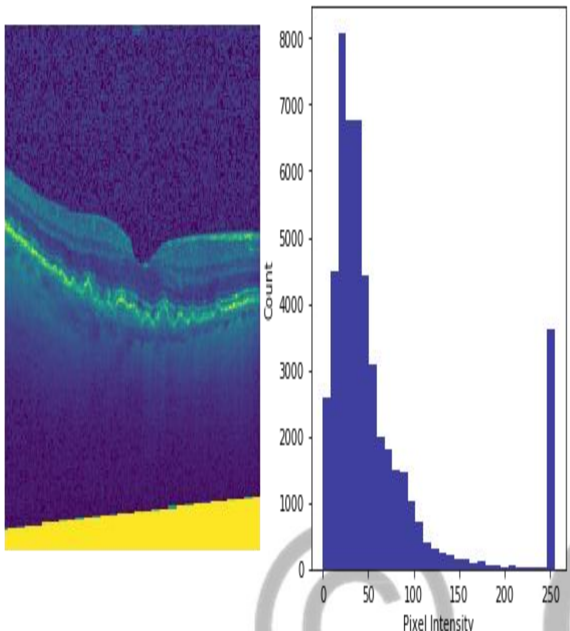
There are 4000 x-ray images (JPEG) and four categories (NORMAL, CNV, DME, DRUSEN). Those are the class labels that an image can be assigned to. Each category is grouped in a sub-directory of its own and the name of the sub-directory is the label of that group of images. The data is now organized into three folders namely train, test and val. Up to 60 percent of the images are used as training example while approximately 10% are used as test and validation set respectively.

Python programming language is the preferred tool for this project. It is a high level, interpreted, object-oriented scripting language that is easy to use and maintain. It also has support for both functional and procedural programming styles. It is a very popular choice among machine learning experts, and it can be used to quickly represent complex mathematical ideas [2]. Other frameworks and libraries that will help to simplify the task are NumPy which is a scientific computing to simplify array operations, Scikit-Learn which contains machine learning predefined operations.

Keras is a high-level library on Tensorflow and it is used extensively in this project.

This is a supervised learning task where input images are assigned to labels from set of categories of retina image labels. The width and height pixels of the images as well as the color channels serve as the input to our algorithms. The algorithms predicted categories/labels from the input data. Classification is a supervised learning technique for which the labels of the classes of the data are given in the training data. Training data are example data from which our classification models learn from. To determine the accuracy of the classifier, the classification error rate is measured. The error rate is the number of observations that are misclassified over the sample size.

The picture below shows a plot of an example image data and its pixel intensity.



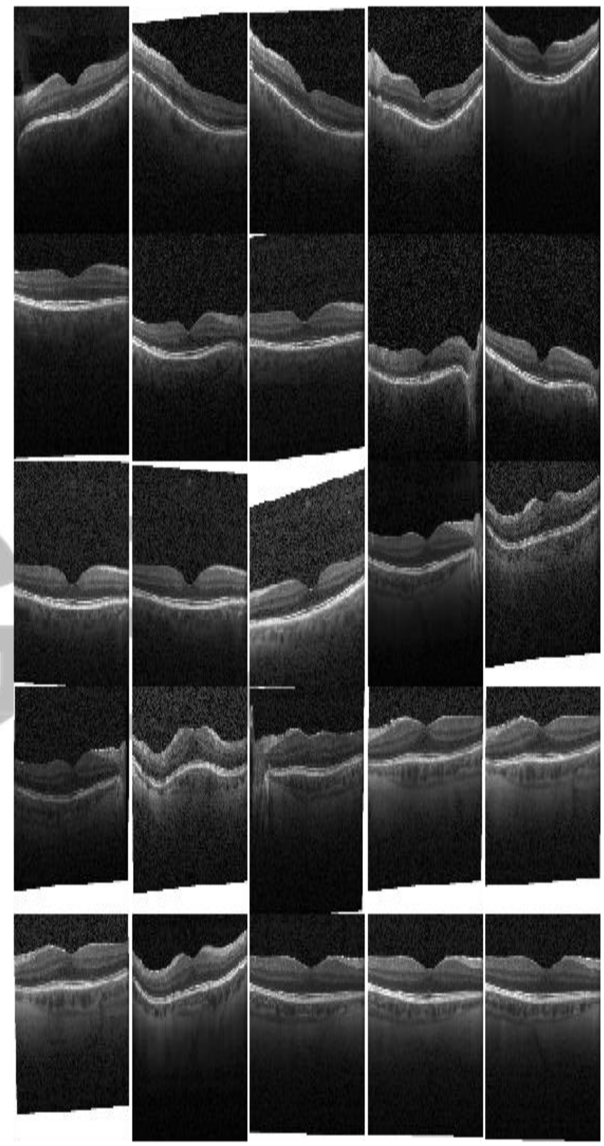
Different algorithms can be employed for a classification problem, for this project however, we are considering linear classifiers. Principal component analysis and support vector machines were considered. K-Means clustering was also used. The most complex algorithms that will be adopted on this project are deep learning algorithms. Deep learning transforms its inputs into more abstract and composite representation. In our project, the raw input into the deep learning framework is a matrix of pixels. For training our deep learning model, backpropagation algorithm will be used to optimize the loss function.

The criteria for judging the credibility of these algorithms will be their prediction accuracy. The objective functions of the models are expected to be optimized so that the error rate is reduced. Prediction metrics is the yardstick to be used to determine which model does the best job in classifying the retinal OCT images.

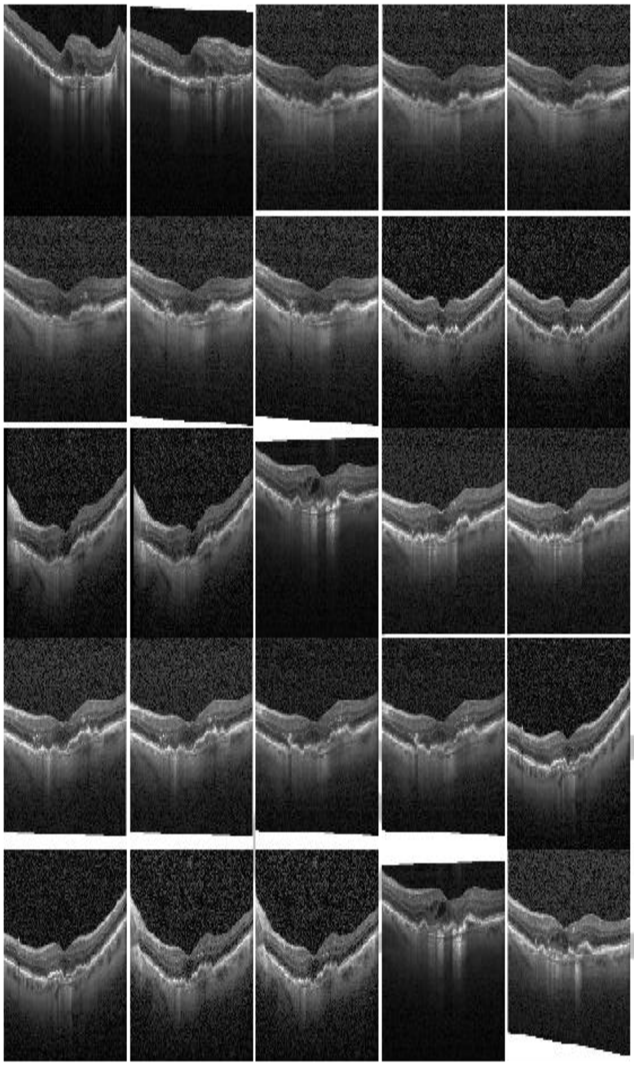
DATA EXPLORATION

Four categories of images were classified, three of the classes have one form of defect while one category is normal. Below is a plot showing example data from each category.

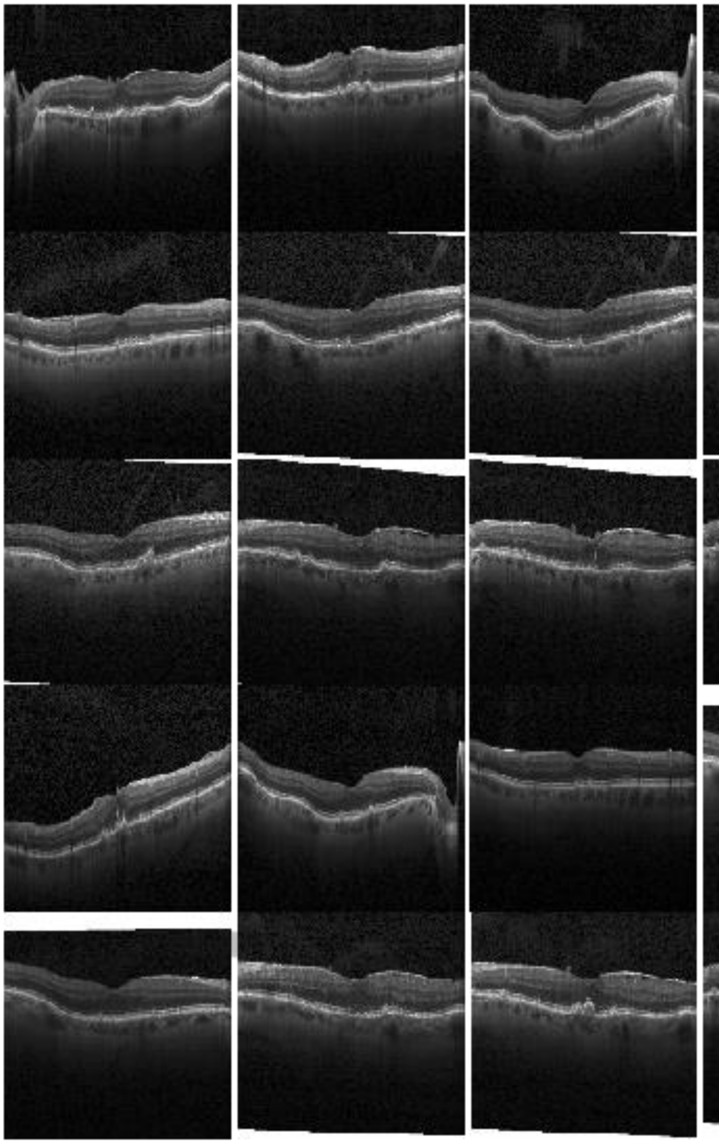
NORMAL



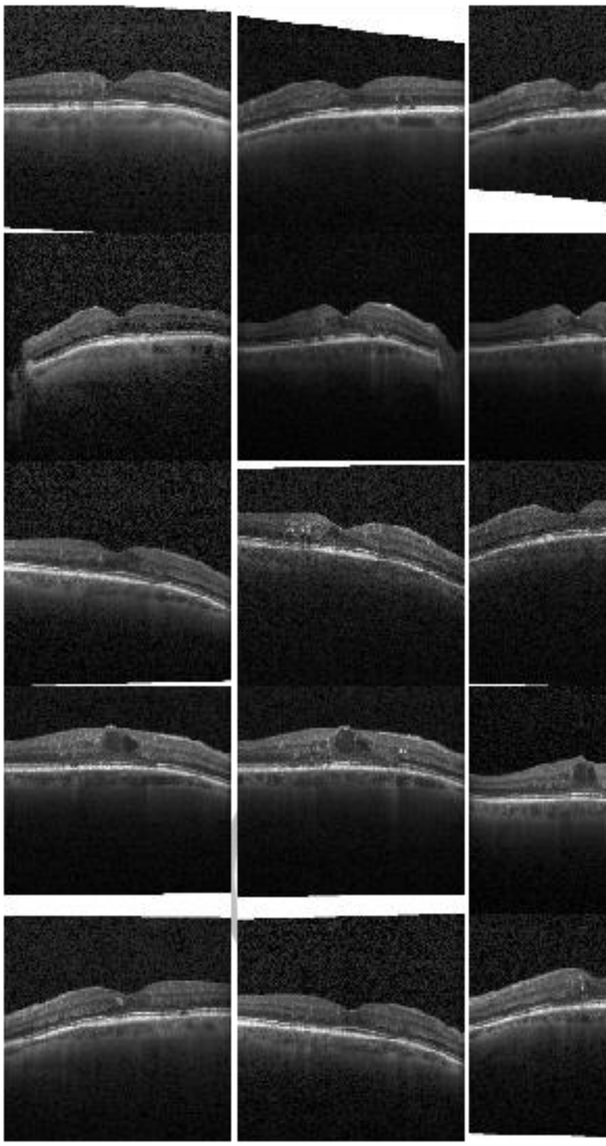
CNV



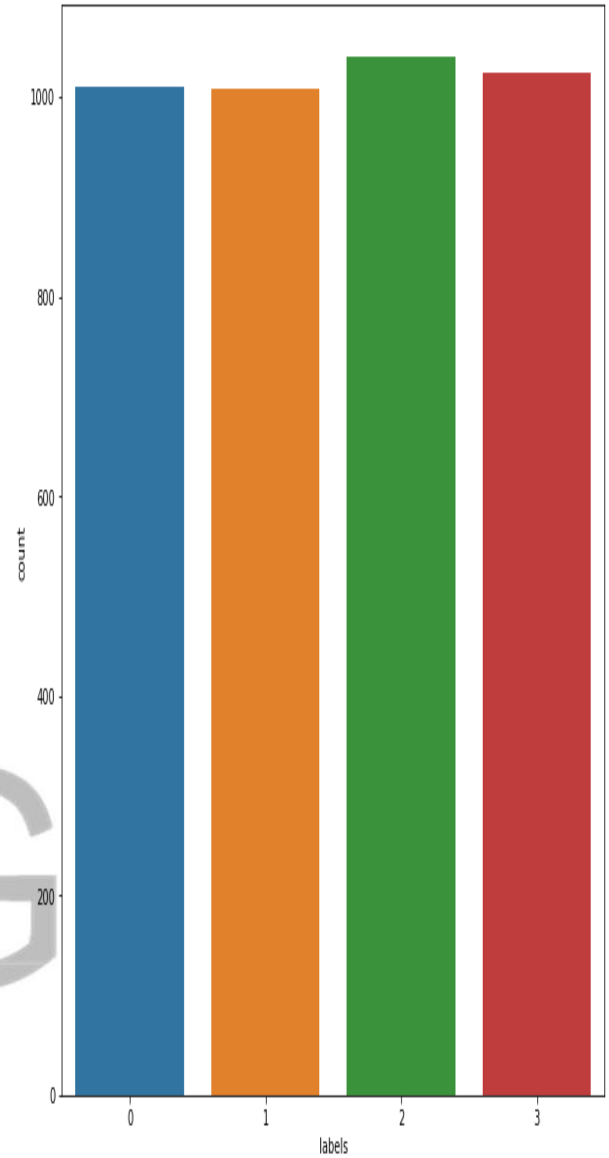
DRUSEN



DME



The picture below shows a count of the number of data instances in each category.



K-Means was able to find four centers from the data.

GSJ: Volume 11, Issue 4, April 2023
ISSN 2320-9186

EXPERIMENT

The experiment started by using an unsupervised learning approach to attempt to classify the images. We try to look for centroids and cluster images based on their distance to their closest centroid. The original labels of the data were not used in this approach since it is unsupervised.

Raw pixels of the data converted into two-dimensional array was passed into a K-Means algorithm already implemented in SciKit-Learn Library in Python.

The K-Means algorithm using Euclidean distance can be summarized as follows:

1. Initialize **cluster centroids** $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$ rand
2. Repeat until convergence: {

For every i, set

$$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

For each j, set

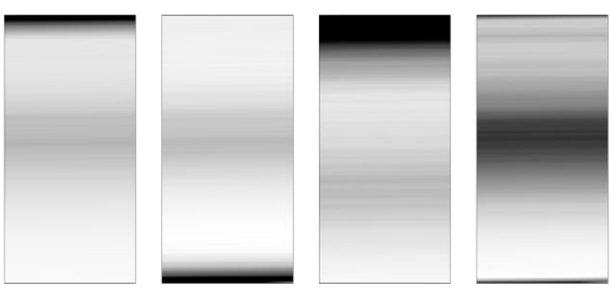
$$\mu_j := \frac{\sum_{i=1}^m 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)} = j\}}.$$

}

The t-distributed stochastic neighbor embedding (t-SNE) algorithm was also explored for pre-processing the data before performing *k*-means. t-SNE is a

statistical nonlinear embedding algorithm adept at preserving points within k means clusters [7].

K-MEANS CENTROIDS FOUND



SUPPORT VECTOR CLASSIFICATION

Rather than using each pixel value as a feature in our SVM model, preprocessing is done to extract related features. Principal component analysis is used to extract 150 components from input into our support vector machine classifier. The classifier and the PCA features are fed into a pipeline during classification.

[7].

A grid based cross-validation search for parameters.

DEEP CONVOLUTED NEURAL NETWORK

A Convolutional Neural Network (CNN) uses convolution, also known as filtering of the input, which in the case of CNNs is an image, in the feature extraction step of the algorithm. In our experiment, a kernel, also known as a filter, slides over the image performing a convolution
at each location. The result is a new image which contains the characteristics of the image defined by the kernel at all locations in the original image.

After a convolution has been performed an activation function is applied to the output. Multiple convolutions followed by an activation layer can be added after each other. There are many types of activation
functions which may be used, including sigma

oids, step-functions or linear rectifying units. We used the rectifying linear unit in this experiment. The task of these activation functions is to add non-linearity to the network. Without these functions the entire network would be a series of linear operations, which could be replaced by a single linear operation, i.e., one layer.

The next step in CNN is a pooling layer. Often used is the max-pooling operation.

Following these operations is a fully connected neural network. It is network which connects each output in the following layer to all the input of the previous layer. The idea is that once the local features in the image have been extracted in the first part, the fully connected network is trained to recognize which features are present and how they are orientated, both spatially in the image and in relations to each other, for each type of image it is being trained to recognize. This fully connected layer is in practice a series of weighted inner products. The weights of all these inner product operations are what changes during training of a CNN.

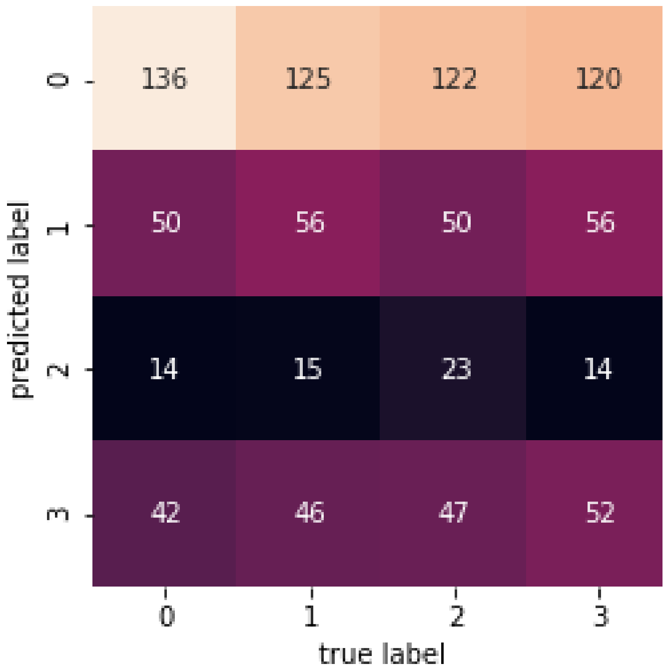
The model is a Convolutional Neural Network consisting of 5 layers. The first 3 layers are convolutional,
followed by 2 fully connected layers as shown below.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_16 (Conv2D)           (None, 222, 222, 32)      896
_________________________________________________________________
max_pooling2d_16 (MaxPooling (None, 111, 111, 32)      0
_________________________________________________________________
conv2d_17 (Conv2D)           (None, 109, 109, 64)      18496
_________________________________________________________________
max_pooling2d_17 (MaxPooling (None, 54, 54, 64)        0
_________________________________________________________________
conv2d_18 (Conv2D)           (None, 52, 52, 128)       73856
_________________________________________________________________
max_pooling2d_18 (MaxPooling (None, 26, 26, 128)       0
_________________________________________________________________
conv2d_19 (Conv2D)           (None, 24, 24, 128)       147584
_________________________________________________________________
max_pooling2d_19 (MaxPooling (None, 12, 12, 128)       0
_________________________________________________________________
flatten_4 (Flatten)          (None, 18432)             0
_________________________________________________________________
dropout_4 (Dropout)          (None, 18432)             0
_________________________________________________________________
dense_8 (Dense)              (None, 512)               9437696
_________________________________________________________________
dense_9 (Dense)              (None, 4)                 2052
=================================================================
Total params: 9,680,580
Trainable params: 9,680,580
Non-trainable params: 0
```

Training was carried out for roughly 50 iterations, and roughly 30 bathes per iteration. The training loss and the test accuracy were saved during training to be viewed once the training had finished.

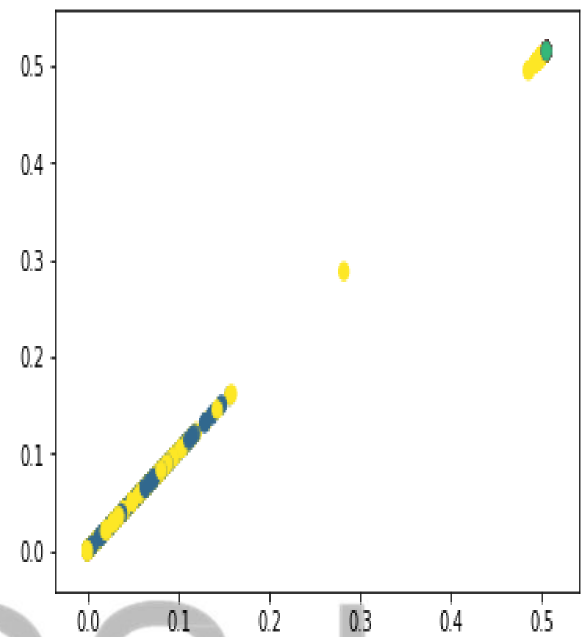
RESULTS

K-Means algorithm performed poorly in classifying the images with an accuracy of 27%. Clear clusters cannot be identified using this algorithm.



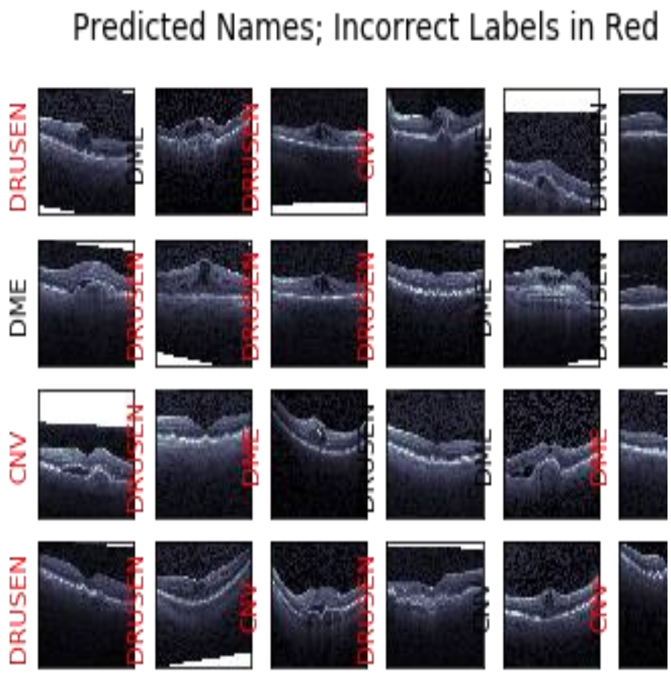
From the confusion matrix, t is obvious many of the labels are predicted wrongly, which clearly shows that K-Means clustering may not be suitable enough for this kind of classification task. Many of the Retina OCT images are classified as Normal, indicating that K-Means cannot distinguish these features from the images. It doesn't have enough feature extracting power to pick up the defective part of the images therefore most images are classified as normal.

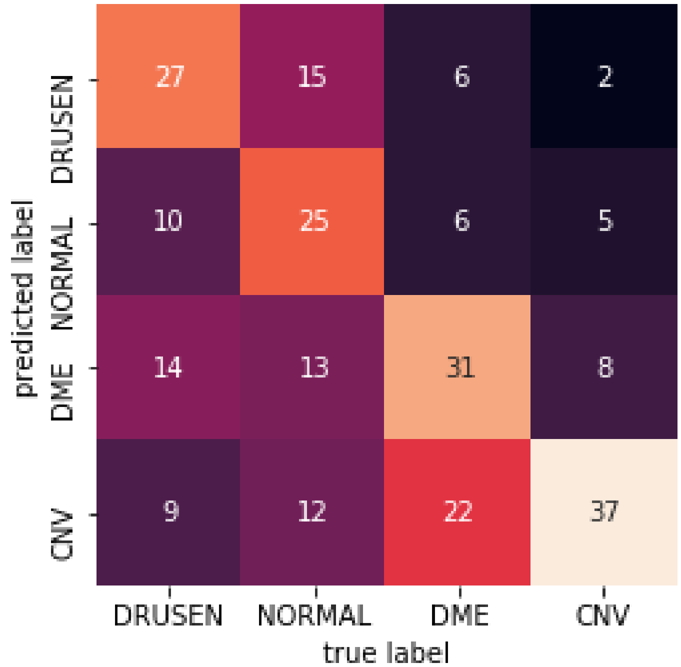
PLOT SHOWING THE RESULT OF IMAGE CLUSTERIG USING K-MEANS



Support Vector machine Plus PCA

The combination of these algorithms performed better than K-Means clustering with a precision of about 51%. The picture below shows the prediction of the images using support vector. Almost half of the images were classified wrong.





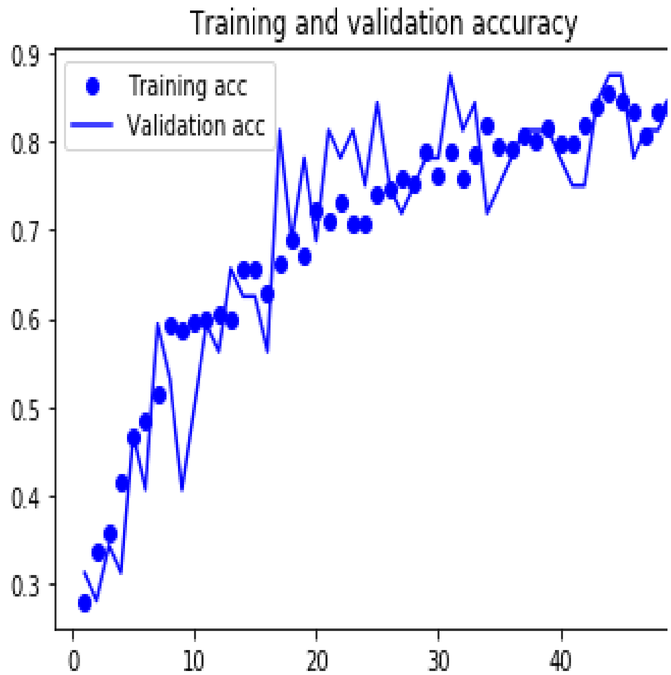
This table shows the precision table for SVM.

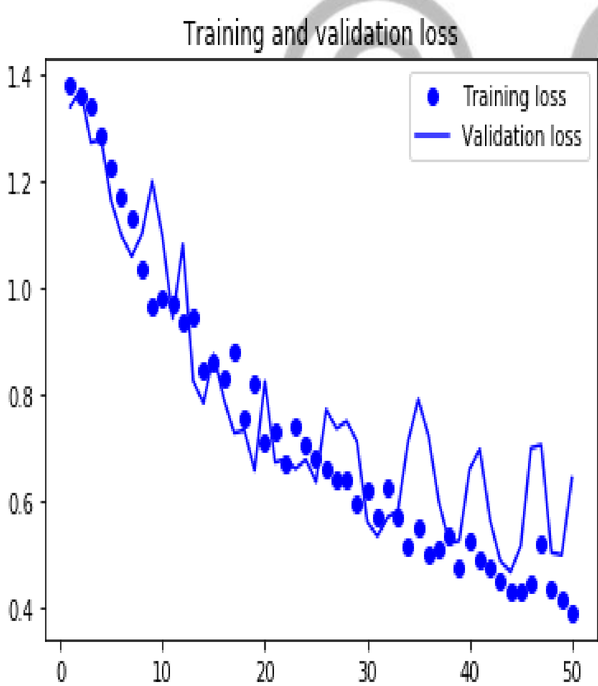
	precision	recall	f1-score	support
DME	0.54	0.45	0.49	60
CNV	0.54	0.38	0.45	65
DRUSEN	0.47	0.48	0.47	65
DRUSEN	0.46	0.71	0.56	52
avg / total	0.51	0.50	0.49	242

The confusion matrix also shows that approximately the prediction is around 50%.

DEEP NEURAL NETWORK

This model outperformed other models and it is the best model in terms of accuracy and robustness. It handled more data instances better and gave a better accuracy score.





With just 50 iterations over the data and 20 batches of examples we are able to get an accuracy of 73% on the test set. This accuracy can also increase significantly with more training examples and increased iteration (epoch) and lower batch number, the accuracy can jump up significantly.

From the figure above, it is obvious that the training loss is decreasing as epoch increases in interval, this is good for the model because it shows that the data is not overfitted.

Conclusion

We can infer from this experiment that deep convoluted neural network outperformed support vector machine and K-means clustering in gray scale image classification due to its ability to extract deeper and useful features from the data into the model. It also shows the inability of SVM to handle larger data instances.

Future work will deal with moving data to the cloud for easier and more efficient analysis, increasing the number of data points, and increasing the number of iterations on the data.

REFERENCES

[1] Leon Bottou 2009, Large-Scale Machine Learning with Stochastic Gradient Descent. NEC Labs America, Princeton, NJ 08542, USA.

[2] Shanmukhi M, et. al. 2018, Convolutional Neural Network for Supervised Image Classification. International Journal of Pure and Applied Mathematics Volume 119 No 14.

[3] Qing Li, Weidong Cai, Xiaogang Wang, Yun Zhou, David Dagan Feng and Mei Chen, Medical Image Classification with Convolutional Neural Network, 13th International Conference on Control Automation Robotics and Vision at Singapore on 10-12 December 2014.

[4] Gidudu A., Classification of Images using Support Vector Machines, Department of Electrical and Information Engineering, University of the Witwatersrand, South Africa.

[5] Vapnik, V.N. 1995. The Nature of Statistical Learning Theory, (New York: Springer-Verlag).

[6] Priyanka A. Abhang et al, Introduction to EEG- and Speech-Based Emotion Recognition

[7] Vanderplas J, Python Data Science Handbook, O'Reilly media, 2017

[8] Lisa Lab, Deep Learning Tutorials, University of Montreal, 2015.

[9] Francois Chollet, Deep Learning with Python, Oreilly Media, 2019