

Classification of Retinal Pathology from OCT images using a Parametric Tuned CNN



Dublin Business School
excellence through learning

Shalini Singh

Dissertation submitted for the Degree of Master of Science in Data
Analytics

At Dublin Business School

11th Jan 2021

Supervisor: Ms. Terri Hoare

DEDICATION

This work is dedicated to my family and friends to make this possible as well as memorable. I would also dedicate this to Year 2020, which was unexpected and difficult but still left us with so many things that I did not foresee two years back.

DECLARATION

I, Shalini Singh, declare that this research is completed by myself for the Degree of Master of Science in Data Analytics and contain my personal work and efforts. It is never presented in any other Institution for the grant of a Degree.

I would also like to declare that all the work done has been done ethically and it does not contain any information that may harm anyone, as best of my knowledge. This research takes care of all the copyrights, licenses, ethical concerns, and citations. This work follows the academic honesty policy provided by Dublin Business School.

Shalini Singh

ACKNOWLEDGEMENT

I am grateful to accept immense help from my supervisor Ms. Terry Hoare, who guided me throughout the research and provided constructive feedback. I am also thankful for her patient and support that she has shown me during the research.

I am thankful to Dublin Business School to provide me the help and resources required to make this research possible. I would also like to thank the faculty for providing us the material for studies and being available whenever required.

In the end, I would like to thank my colleagues and friends for the support they have shown.

ABSTRACT

Optometrists nowadays use optical biopsy to get cross sectional images of the retina infected by pathologies. This is also known as Optical Coherence Tomography (OCT). It is important to identify the retinal diseases at an early stage to prevent damage to the vision. There is a lot of research to be done to find a suitable method which can automatically detect retinal diseases. Therefore, we propose this research for automatic detection of retinal diseases by using a novel method of hyperparameter tuning instead of manually detecting the parameters of our Convolutional Neural Network (CNN). The Model is tested on metrics such as F1-score, precision, specificity, sensitivity, loss graph and accuracy. We also compare it with pretrained state-of-the-art model of Inception V3 and result shows that hyperparameter-tuned CNN gets better results. Being reliable, this proposed model can be used by optometrists to detect retinal disorders at an early stage.

TABLE OF CONTENTS

DEDICATION.....	2
DECLARATION	3
ACKNOWLEDGEMENT.....	4
ABSTRACT	5
LIST OF FIGURES	8
LIST OF TABLES.....	9
LIST OF ALGORITHMS	10
LIST OF IMPORTANT ABBRIVIATIONS	11
CHAPTER 1 – INTRODUCTION.....	12
1.1 Background.....	12
1.2 Business Background	13
1.3 Research Problem.....	16
<i>Research Question:</i>	16
<i>Aim:</i>	16
<i>Objective:</i>	16
<i>Hypothesis:</i>	16
1.4 Scope	17
1.5 Limitations	17
1.6 Roadmap.....	18
CHAPTER 2- LITERATURE REVIEW	19
CHAPTER 3- METHODOLOGY	27
3.1 Business Understanding	28
3.2 Data Understanding and Data Preparation	30

3.2.1 Data Understanding	30
a) Data Collection.....	30
b) Data Visualization.....	32
3.2.2 Data Preparation	33
a) Dataset Preparation:	33
b) Data Transformation:	34
c) Data Imbalance Handling:	35
3.3 Modeling.....	35
3.3.1 Technical Requirements.....	36
a) Model Selection:	36
b) Environment Specification	38
c) Programming Language Specification:	39
d) Hyperparameter Tuning:	39
3.3.2 Model Implementation.....	40
a) Convolutional Neural Network (CNN)	40
b) Inception V3	46
3.4 Evaluation.....	49
3.5 Deployment.....	51
CHAPTER 4- RESULTS ANALYSIS	52
4.1 CNN Model.....	52
4.2 Inception V3.....	56
CHAPTER 5- CONSCLUSION AND FUTURE WORK	60
REFERENCES	61
APPENDICIES	64

LIST OF FIGURES

Figure 1. 1 Comparison of eye retinas with and without pathologies.....	13
Figure 1. 2 Roadmap of research	18
Figure 3. 1: Implemented CRISP-DM	28
Figure 3. 2: Process Diagram	29
Figure 3. 3: Retinal Pathologies and healthy retina.	31
Figure 3. 4: Data Representation	33
Figure 3. 5: Design of the Model.....	36
Figure 3. 6: Rapid Miner Auto Model Result.....	38
Figure 3. 7: CNN Architecture	41
Figure 3. 8: Convolved Features.....	41
Figure 3. 9: Max pooling of features.....	42
Figure 3. 10: Layers of CNN according to research requirement.....	45
Figure 3.11: Inception V3 Model Architecture	46
Figure 3. 12: Transfer Learning.....	47
Figure 3.13: Process of Imagegenerator	48
Figure 4. 1: Training Vs Validation Graph	53
Figure 4. 2: Confusion Matrix	53
Figure 4. 3: Visual representation of Confusion Matrix for Hyperparameter-Tuned CNN.....	54
Figure 4. 4: Confusion matrix for Inception V3	56
Figure 4. 5: Visual representation of Confusion matrix for Inception V3	57

LIST OF TABLES

Table 2.1: Literature Review	26
Table 3. 1: Dataset Image availability.....	32
Table 3. 2: Rapid Miner auto model results	37
Table 4. 1: Output of CNN	55
Table 4.2: Output of Inception V3.....	58
Table 4.3: Consolidated table of all the results.....	59

LIST OF ALGORITHMS

Deep Learning

Convolutional Neural Network

Inception V3

Transfer Learning

LIST OF IMPORTANT ABBRIVIATIONS

Age related macular degeneration (AMD)

Bruch's membrane (BrM)

Choroidal Neovascularization (CNV)

Convolutional Neural Network (CNN)

Cross-Industry Process for Data Mining (CRISP-DM)

Diabetic macular edema (DME)

Histogram of Oriented Gradient (HOG)

Inner limiting membrane (ILM)

Label smoothing generative adversarial network (LSGAN)

Layer guided Convolution Neural Network (LGCNN)

Local Binary Pattern (LBP),

Multi-Scale Deep Feature Fusion (MDFF)

Optical coherence tomography (OCT)

Rectified Linear Activation Function (ReLU)

Retinal Image Analysis (RIA)

Retinal pigment epithelium (RPE)

CHAPTER 1 – INTRODUCTION

The rareness of blindness can never be understood by the people carrying the gift of sight. In this era of rapidly improving technology, it is possible to understand as well as predict the outcome using few algorithms. It is critical to analyze the possibility of retinal pathologies as early as possible, so it does not permanently damage the eye and cause incurable blindness.

1.1 Background

Human eye consists of many important components and one of them is the retina. It is a layer of tissues lining the back of the eye and is present close to the optic nerve. The most critical function of the retina is to convert the received light into neural signals. Neural signals are then sent to the brain to create a vision. Like every other organ, retina too can experience some diseases overtime due to the process of aging and many more factors.

Three of these pathologies are Choroidal Neovascularization (CNV), Diabetic macular edema (DME) and Drusen.

- ***Choroidal neovascularization (CNV)*** is caused by the generation of some new and abnormal blood vessels which formalize in the choroid, which is a layer under the retina that contains vessels. These new blood vessels create fluids that sometimes leaks into the retina along with blood and forms a blister that later causes blindness as discussed by (Moreiraet al. 2009).
- ***Diabetic macular edema(DME)*** is caused due to the visual complications that can arise in a diabetic person. In this the retina is damaged because of frequent leakage of fluids and swelling of area containing small blood vessels which can later cause

an incurable blindness (Sun et al.; 2017). It is also expected that DME cases is going to affect over 300 million people in the world in the next few years (Kanagasingam et al. 2014; Lemaitre et al. 2016).

- **Drusen** are extracellular deposits of different sizes that are present between a Bruch's layer and a Retinal pigmented epithelium layer (RPE) and normally causes abnormalities in the photoreceptors. This is generally caused use to high risk presented because of aging (Chiu et al. 2012).

In the Figure 1.1, we can easily compare the difference between the normal eye retinas with respect to all the pathologies (Johnson et al. 2003).

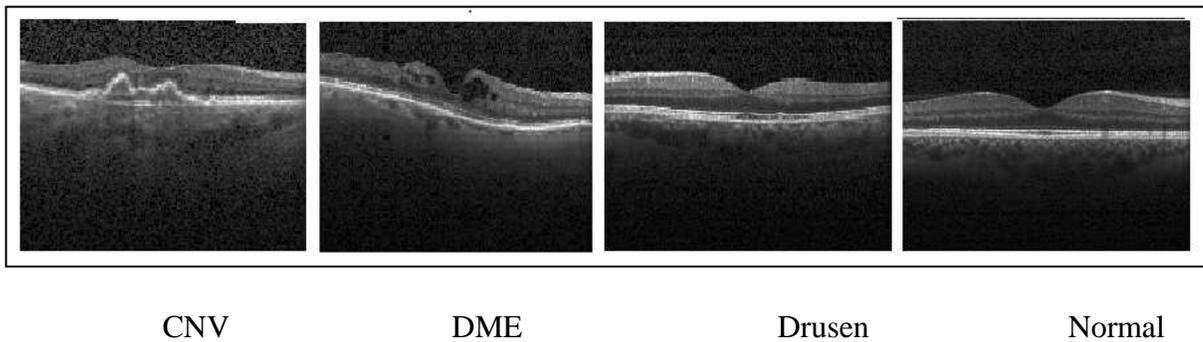


Figure 1.1 Comparison of eye retinas with and without pathologies.

1.2 Business Background

Diagnosing the pathologies at an early stage is critical, as it may help in making treatment more successful for the patients as well as it can help them in recovering properly. As all the conditions are degenerative and seriousness of issues increase with time, delayed detection can cause blindness that cannot be reversed. Factors like this have increased the necessity of using technology to identify the symptoms as early as possible as well as with

high accuracy. With the advent of technology, Retinal Image Analysis (RIA) has become widely automated. One of the earliest implementations of RIA was documented by (Baudoin et al.; 1984) on the detection of micro aneurysms in OCT images. In the 1990s, there was a significant shift with the introduction of digital retinal imaging and the growth of electronic filter-based imaging techniques. These innovations gave rise to an ever-increasing number of articles published in this area. Optical coherence tomography (OCT) is an image-based technique in which image of the retina is taken with the help of low-coherent light to capture smaller details with high resolution. The light used for the images is having a long wavelength due to which it can easily obtain high resolution images. Optical Techniques like this are proving to be extremely important in the medical field nowadays as it is considered safe and affordable while providing accurate results (Huang et al. 1991; Bayleyegn et al. 2012).

Digital RIA has enormous potential as there is the demand for a retinal assessment to be more accurate, reliable, and more measurable. The daily costs of eye care clinicians' assessments and the ever-increasing number of retinal images to be examined are primary factors for this project research. With this application, clinical professionals will have access to quicker analysis of OCT images with reduced cost as opposed to manual identification of eye defects. The project uses Hyperparameter tuned CNN to detect and predict CVN, DME and Drusen in OCT images and compares the results with Inception V3 model which uses transfer learning and is a state-of-the-art model.

Recently, researchers have come up with a lot of ideas about applying the deep learning algorithms on medical field for classification and segmentation of the images as suggested by (Alsaih et al.2017) and (Farsiu et al. 2014).It is proved that Deep learning is very helpful in deciphering a complicated system that contains a large set of data (Xiang et al. 2018). With the help of back propagation algorithm, any deep learning model can learn to change the weights and learn on its own. As discussed by (LeCun et al. 2015),It learns about the different parameters by refereeing to parameters present in all the previous layers.

The focus of this research is to use a novel method to provide us better results with respect to a state-of-the-art method, for the segmentation of retinal pathologies. It was noticed that CNN was widely used in predicting the retinal pathologies, but because of manual parameter configurations the results were not as promising as they can be. Previous works done related to the same domain were studied and it was found that no automated hyperparameter tuning was done to achieve a list of parameters that are positively impacting the CNN model's results. To address this problem, we introduced a hyperparameter tuning using Keras tuner.

The dataset that consisted of almost 80k images provided us with a challenge, that all the images having the pathologies of different types, only showed a minor deviation from the normal retinal images. This was because the dataset consisted of images having different sizes and pixel configurations, due to which the model was not able to perform a precise feature extraction. To rectify this, we divided the dataset into multiple sections, all sections

containing different images which were independent of each other i.e., no duplicate images were used. This dataset was then pre-processed to reduce the noise and to even out the images so that less bias is introduced in the CNN model. A comparative study of the models was also done to verify the results and it was observed that our novel model outperformed a pre-trained model.

1.3 Research Problem

The focus of this research is to detect the possible retinal pathologies in the eye at an early stage so that the damage done by them can be cured. Our research question is based on finding the best suitable method for this.

Research Question: How well can Parametric-Tuned CNN improve efficiency in detecting macular-pathologies using OCT when compared to Inception-V3?

Aim: Classifying the retinal pathologies falling under four major categories from OCT images with the help of parametric tuned CNN.

Objective: To use Hyperparameter tuning with CNN to find parameters that provide best results with highest predictive accuracy and to understand how this step can impact the overall results of our model when compared with State-of-the-art models.

Hypothesis: The model will prove to be more efficient than other models due to hyperparameter tuning along with CNN.

1.4 Scope

- To write python code into current IDE i.e.Jupyter Notebook which transforms images into numpy arrays of smaller size.
- To save those numpy arrays into a CSV file for loading data into Rapid Miner.
- To apply Automodel in Rapid Miner using the CSV and compare different model's accuracy and other performance matrices.
- To Find the best suitable model for our dataset using the comparison in Rapid Miner.
- To Tune the hyperparameters of CNN for better results.
- To apply CNN model on the dataset and check for various metrics of the model on the previously unseen dataset i.e. test dataset.
- To apply one more pre-trained (Inception V3) model on the same dataset so as to compare it with our novel approach.
- Compare the results provided by Hyperparameter tuned CNN and State-of-the-art pre-trained Inception V3.

1.5 Limitations

The research is taking into consideration a huge amount of dataset with class imbalance, so it is necessary to even out the classes to reduce bias; this reduces the dataset and issues a limitation of amount of training data. The computational power of systems used is also a limitation as it is difficult to get a system with enough computational power and working on Google Colab has issue such as connection breakage after every 12 hours.

1.6 Roadmap

The Roadmap of the dissertation is as shown in Figure 1.2:

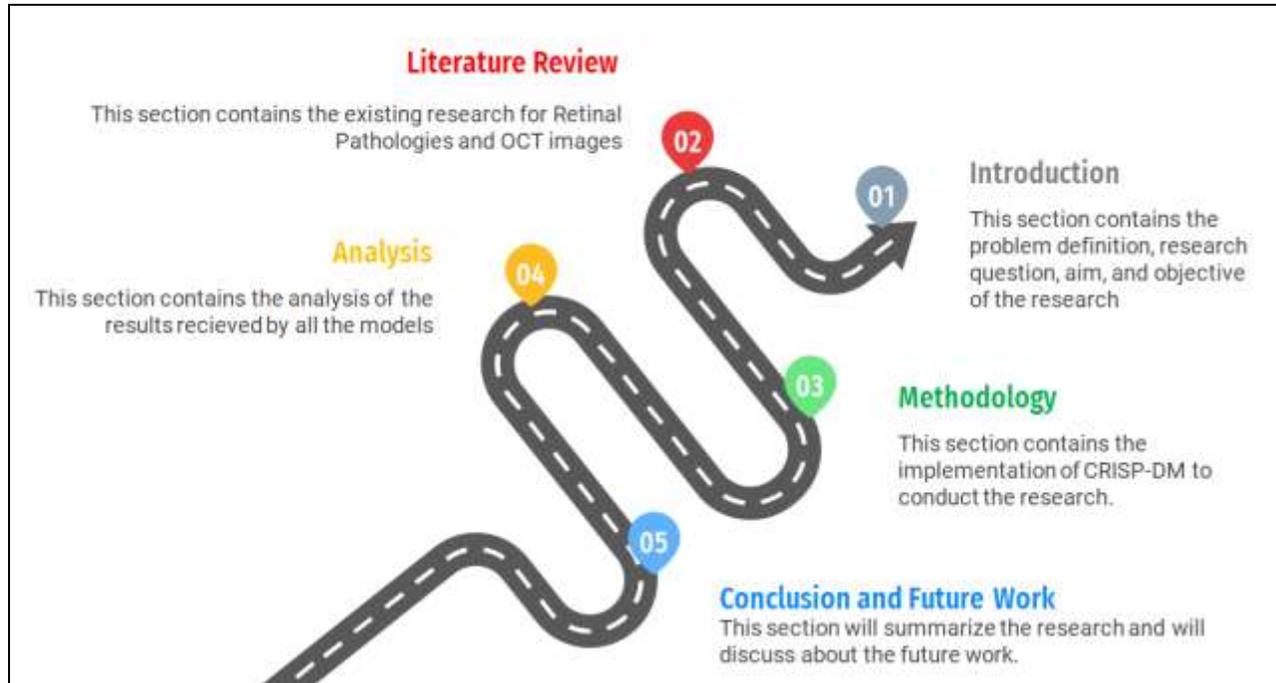


Figure 1.2 Roadmap of research

In upcoming section i.e., section 2, we have discussed more on the previous work done on this research topic, which will be followed by the explanation of methodology used for the implementation of our model in section 3. The discussion of results is done in section 4, where we evaluate the outcomes of the model. Section 5 contains the conclusion of the research.

CHAPTER 2- LITERATURE REVIEW

Machine Learning has been widely used in medical field. Numerous algorithms and models have been developed so far for grouping small features of the OCT images. Yet the desired outcome has not been achieved without any errors. Hence, there is still a lot of research to be done on the latest technologies coming in Artificial Intelligence to develop algorithms that can easily classify images and provide satisfying results. In this section the work done so far in the Optimal Coherence Tomography field will be discussed and it will help us in driving our research forward.

(Bhowmik et al. 2019) used the state-of-the-art CNN along with VGG16 and Inception V3 model. The team used 224*224 RGB images for research with relatively small dataset. With the help of pre-trained dataset, they have applied transfer learning to the models to reduce the overall training time of the model (Li et al. 2019). All the initial weights are kept as is and changes are made in the outermost layer based on their dataset. The aim of their research is to achieve a higher accuracy with significant smaller part of the dataset. All the layers of the original model were frozen, and layers introduced by them were trained, and regularization was done using a drop-out layer. With the help of this approach, they achieved the accuracy of 94% with all the data.

(Das et al. 2019) mentioned the use of a novel approach of multi-scale deep feature fusion (MDFF) along with CNN. They have introduced this method to classify Age related macular degeneration (AMD), Drusen and CNV from Normal retina. The team have emphasized that most of the hand-crafted features are highly dependent on the overall knowledge and understanding of the developers and do not prove to be optimal most of the time. They agree

with (Bhowmik et al. 2019) with regards to the pre-trained models but have critical views on limited effectiveness of diagnosis made depending on the manual setting of all the hyperparameters as they take a long time to configure. The requirement of hyperparameter tuning is removed with the help of MDFF as it provides good results as the fusion of features can help in making easier classifications due to the higher dimensions of the fused feature vectors. The accuracy received using this method was 95.5%.

Layer guided Convolution Neural Network (LGCNN) is based on the principle that says, the issues caused in the retina are mostly because there are certain lesion present in some particular layer of the retina. This information can be used to make predictions as well (Marginean et al. 2019). It needs to be understood that almost all the retinal pathologies are related to the layers present in the retina. This fact is used by (Huang et al. 2019) while introducing a novel Layer guided convolutional neural network (LGCNN). In this paper the author is suggesting that LGCNN can easily and efficiently concentrate on the layers that can be related to the lesion generation such as Inner limiting membrane (ILM), retinal pigment epithelium (RPE) and Bruch's membrane (BrM), hence they will help in identifying the presence and location of the lesion. The team has used ReLayNet for the generation of a segmentation map for the retina layer, which extracts two layers related to lesion. ReLayNet is mainly used for feature extraction from images as used by (Roy et al. 2017). LGCNN is used to extract the information related to the lesion structure from the segmented layers for the purpose of identification. This data is later used to classify the disease. This process was applied on two datasets which were taken from University of California San Diego (UCSD) and Hunan University of Chinese Medicine (HUCM). Results experienced were very promising as the accuracy of detecting retinal pathology on HUCM dataset was 96% and for UCSD was around 94%.

A comparative study of deep neural networks and handcrafted feature extraction is done by (Nugroho; 2018) and he presented that non-automatic or handcrafted methods for feature extraction as not as effective as automatic methods. This study was done using handcrafted methods such as Histogram of Oriented Gradient (HOG) and Local Binary Pattern (LBP), whereas the deep learning models used were Residual Network (ResNet50) and (DenseNet-169). According to the study it is suggested that the use of handcrafted feature extraction should be done with a limited dataset and automated feature extraction should be applied to a large dataset as it provides the chance to learn some complex patterns existing in the data. This study was done on similar dataset as ours and image configuration was of 224*224 pixels. After completing all process such as data pre-processing, feature extraction and classification using both techniques, it was concluded that automatic methods are more success with precision up to 91%.

(He et al. 2020) proposed a different way to diagnose the retinal pathologies when the data is limited. They used Label smoothing generative adversarial network (LSGAN) with CNN. The LSGAN is having three main processes and components, the very first process is of generating data that is synthetic and unlabeled, and this data is bit similar to the real images getting used in the diagnosis. The next process is to differentiate between real and synthetic images which is done by a component named discriminator and the last process is to classify the real labeled images and pseudo labeled images. The team has raised a real concern regarding the availability of limited labeled data and how it can cause overfitting and have tried to provide some favorable solutions to avoid it. As it is also treated like a comparative study, they have worked on two datasets and have applied three different architectures such as HOG-SVM, FC finetuning and Inception V3 to understand what works best for the small classification dataset. The training data is divided between the range of 1 to 30% to understand the pattern of model performance, it is

noted that as the data increases the precision and sensitivity also increases. The result represents that the proposed model works the best with 87% sensitivity for UCSD dataset and 66% for HUCM dataset.

CNN, when used for the purpose of classification, hides that there could be a positional relation in the pooling layers as discussed by (Tsuji et al. 2020; Alqudah; 2020). Hence, they chose capsule network for removing that as it can learn positional details present in the features of different images. Also, this network works on those datasets which has fewer images. In this process they downgraded the pixel value of image from $1536*512$ to $512*512$. The number of epochs normally affect the accuracy value, Higher the epochs higher the accuracy. This is same with stride parameter but for learning rate, lower the learning rate, higher the learning speed of the model (Kharisudin et al. 2020). Then to come up with shortcomings of less dataset, they performed data augmentation. Here they have used four convolution layers of the discussed network. The accuracy achieved was not satisfactory hence they could have increased the layers present to fetch smaller details in the image and come up with good accuracy.

A comparative study of different image preprocessing factors and their impacts on results is done by (Chan et al. 2017). The paper aims to detect DME and understand what kind of image preprocessing methods produce most efficient results. The images are subjected to different types of process such as noise removal, image boundary extraction, cropping, and feature extraction, before they are classified with the help of SVM. During denoising of the images, all the unclear edges are removed as they are considered as noise. This is done using BM3D, which is believed to have higher and better effects than any other filtering method. In further process,

the first and final layer of the retina is identified, and rest of the image is cropped. All the pixels which are not following characteristic related to DME are removed during image cropping. CNN is then used to extract the features from these images. The classification is done using SVM, where multiple scenarios are taken into consideration. It is observed that the fully connected layers of CNN generate the best output with accuracy of 96% when noise removal and image cropping is done. It shows that preprocessing an image can easily help in increasing the overall accuracy of the model. Including Hyperparameter tuning to this will again increase the efficiency as the best parameter set will be provided to us on which our model can train later and provide higher accuracy.

We need to do image flattening before feeding the image into the model. This was performed by (Chiu et al. 2010). They explained a flattening method in which they were pointing at the brightest pixel to estimate the RPE in each column on the scanned image. The image that is formed from this process is then fitted to second order polynomial then shifting each column according to that to achieve flat RPE. Then each scanned image is cropped above and below horizontally by 160 and 90 pixels in order to get rid of unwanted portions of the image. After that the image was resized to 128*256 and made available to use in the model as also discussed by (Islam et al. 2019).

Many scientists have used transfer learning to detect retinal diseases previously, as done by (Kermany et al. 2018). It was noticed by them that the image analysis methods, handcrafted object segregation, were getting outdated as they were not very efficient. Also, these methods required higher cost, higher time and skilled professionals. There is a need to come up with a

technology that produces better and smarter results, so they suggested Artificial Intelligence can help in achieving that.

(Sertkaya et al. 2019) used LeNet, Vgg16 and AlexNet for the diagnosis of all four retinal pathologies. Although, information is provided related to the pretrained models used in the diagnosis, but no information is provided as to exactly why these models were chosen and what they did best that others failed to do. The AlexNet was used with a dropout layer so that the memorization of the dataset can be prevented, but no other significant use of VGG16 and LeNet is mentioned. The author mentioned the use of AlexNet separately as well, which indicates that two types of AlexNets were used for the diagnosis, one with a dropout layer and one without a dropout layer. The lack of many other processes such as data preprocessing and Hyperparameter Tuning has impacted the results in multiple ways, the overall accuracy for LeNet was just 86.86%, for AlexNet it was 94.48% and for VGG16 it was 95.74%. It was also observed that the models were more efficient in identifying CNV with all the results exceeding 90%, but rest of the pathologies are not identified as efficiently as we would hope.

A research carried out in 2018 by (Xiang et al. 2018) proposed a method to detect CNV in 42 spectral-domain OCT images. The model utilized 7 grey-level features, 3 spatial features, and 14 layer-like features to train the neural network classifier. As a process to ease identification of the layers in the retinal images based on the different thicknesses, the authors introduced multi-scale bright and dark layer detection filters. Finally, they detected retinal surfaces using a constrained graph search algorithm, resulting in a method that outperformed other methods in similar domain (Xiang et al. 2018). Their implementation of Non-negative Conjugate Gradient (NNCG)

algorithm surpassed the performance of the Non-networked Microgrids (NNMG) algorithm resulting from the enhancement of the ellipsoid zone using bright layer detection filter. One limitation encountered in this project was the restriction of surface detection accuracy in layers resulting from the low contrast between the multiple layers making their boundaries nearly invisible. Another limitation listed was the high computing time required to implement the algorithm.

Following back on all the previous research mentioned in this segment, it is evident that a lot of trial and errors are done to generate high accuracy results for Retinal pathologies. Some of the researchers have worked with different layers of the retina and have tried to figure out the presence of diseases, while others have taken the help of statistics, deep learning as well as Machine Learning algorithms to predict the retinal pathologies. It is also noticed that none of the research follows any automated hyperparameter tuning methods. Therefore, in this research we will be working on not relying on manual hyperparameter tuning and will be finding the best possible parameter with the help of automated hyperparameter tuning. We will be applying this data to CNN and will do a comparative study with a state-of-the-art method i.e. Inception V3 with Transfer Learning.

A summary of few of the above-mentioned previous works can be seen in the table 2.1.

Authors	Title	Source	Keywords	Findings
Bhowmik et al.; 2019	Eye Disease Prediction from Optical Coherence Tomography Images with Transfer Learning	Springer	Age-related macular degeneration, Transfer learning, Choroidal Neovascularization, Diabetic Retinopathy, Diabetic Macular Edema	Transfer learning with VGG16, Inception V3 and CNN works best for smaller datasets with smaller number of classes.
Das et al.; 2019	Multi-scale deep feature fusion for automated classification of macular pathologies from OCT images	Elsevier	Optical coherence tomography (OCT), Convolutional neural network (CNN), Feature fusion, Macular pathology	Multi-scale deep feature fusion architecture for the classification of the images can be use very well for detecting retinal pathologies with the help of CNN
Huang et al.; 2019	Automatic Classification of Retinal Optical Coherence Tomography Images With Layer Guided Convolutional Neural Network	IEEE	Biomedical optical imaging, medical image processing, Image segmentation, OCT classification, Optical coherence tomography (OCT), convolutional neural network (CNN)	All the lesions are generated between two layers and including this information can prove helpful in detecting eye diseases. Layer guided convolutional neural network (LGCNN) identifies the layers and finds the significant pathology with 94% accuracy.
Nugroho; 2018	A Comparison of Handcrafted and Deep Neural Network Feature Extraction for Classifying Optical Coherence Tomography (OCT) Images	IEEE	Optical Coherence Tomography, HOG, LBP, DenseNet, ResNet, Feature extraction	Both the handcrafted methods Histogram of Oriented Gradient (HOG) and Local Binary Pattern (LBP) proved to be much less accurate than deep learning methods Residual Network(ResNet50) and (DenseNet-169)
Sertkaya et al.; 2019	Diagnosis of Eye Retinal Diseases Based on Convolutional Neural Networks Using Optical Coherence Images	IEEE	Biomedical optical imaging, Convolutional neural networks, Medical diagnosis, Supervised learning	Out of all the methods (AlexNet, LeNet, VGG-16 architectures) used for the classification of retinal pathologies, VGG worked the best with 93% accuracy and using dropout layer with AlexNet helped in loss reduction.
He et al.; 2020	Retinal Optical Coherence Tomography Image Classification with Label Smoothing Generative Adversarial Network	Elsevier	Retinal Classification, Optical coherence tomography (OCT) Convolutional neural network (CNN)Generative Adversarial Network (GAN)	Label smoothing generative adversarial network (LSGAN) can work efficiently for detecting diseases in the eye retina as they work well for limited datasets by creating synthetic images and labels.
Chan et al.; 2017	Transfer Learning For Diabetic Macular Edema (DME) Detection On Optical Coherence Tomography (OCT) Images	IEEE	Diabetic Macular Edema (DME), Convolutional Neural Network (CNN), Spectral Domain Optical Coherence Tomography (SD-OCT)	Removing any noise from the images and emphasizing only on the required parts of the retina can help identify the retinal pathologies easily with the accuracy of 96%.
Li et al.; 2019	Fully automated detection of retinal disorders by image-based deep learning	Springer	Age-related macular degeneration, Diabetic macular edema, Optical coherence tomography, Deep transfer learning, Visual geometry group 16 network	Using deep transfer learning with VGG16 can easily help in increasing the efficiency in detecting AMD and DME diseases in the retina.
Tsuji et al.; 2020	Classification of optical coherence tomography images using a capsule network	BMC Ophthalmology	Capsule network, Choroidal neovascularization, Deep learning, Diabetic macular edema Drusen, Optical coherence tomography	Addition of a capsule layer with four CNN layers improved the accuracy of the model and helped in early detection of the issues.
Kharisudin et al.; 2020	Deep convolutional neural networks for the detection of macular diseases from optical coherence tomography images	IOP Publishing Ltd	Optical Coherence Tomography (OCT), Convolutional neural network (CNN)	The number of epochs normally affect the accuracy value, Higher the epochs higher the accuracy. This is same with stride parameter but for learning rate, lower the learning rate, higher the learning speed of the model.

Table 2.1: Summary of few Literature Reviews

CHAPTER 3- METHODOLOGY

This qualitative comparison research is conducted with the help of Cross-Industry Process for Data Mining which also stands for CRISP-DM. This methodology provides the flexibility of modifying the stages according to domain and its requirements, due to which it is very efficient in planning and executing research. Some of the most important stages of CRISP-DM are as follows:

Business Understanding

Data Understanding

Data Preparation

Modeling

Evaluation

Deployment

According to the requirements we will be modifying few stages, we will be starting with Business Understanding, and moving on with Data Understanding and Preparation, we will next work on modeling which will contain design specification as well as implementation and will further move to Evaluation and Deployment. This flow can be seen in the figure 3.1.



Figure 3. 1: Implemented CRISP-DM

3.1 Business Understanding

Manual segmentation of the retinal diseases takes a lot of time and efforts, even then it may not be correct all the time. This requires an artificially intelligent system to work and come up with a better solution within time. This research is completely focused on using the niche technologies to implement the automated system that will help doctors to reduce efforts while coming up with faster results with better accuracy. Deep learning always comes into consideration when we think of creating such automated system.

In this paper, our novel approach is based on considering the new technologies and using them for hyperparameter tuning and then applying CNN. We also implement the baseline approach for comparing it with our novel approach. This approach is using pre-trained models based on transfer learning which is Inception-V3. Below at figure 3.2, we represent a process diagram which is followed in our research. As it is depicted in figure 3.2, there are certain steps carried out starting with the data acquisition. Then we have done some pre-processing on the data to

make it available for feeding into our model. There have been certain steps involved in it such as image resizing, rescaling, noise reduction, batch formation etc. Then we do hyperparameter tuning with some of the data to ensure better accuracy. Once the data is fed to the network and after training the model, we check the accuracy and other parameters to know more about our predictions. Later, we compare the result of our model with the state-of-the-art approach.

Our business understanding can be summarized with the help of below points:

- Reduction of overall time taken for the detection of retinal pathologies.
- Increment in the overall accuracy using an automated system.
- Introducing a novel technique that achieves both the points.

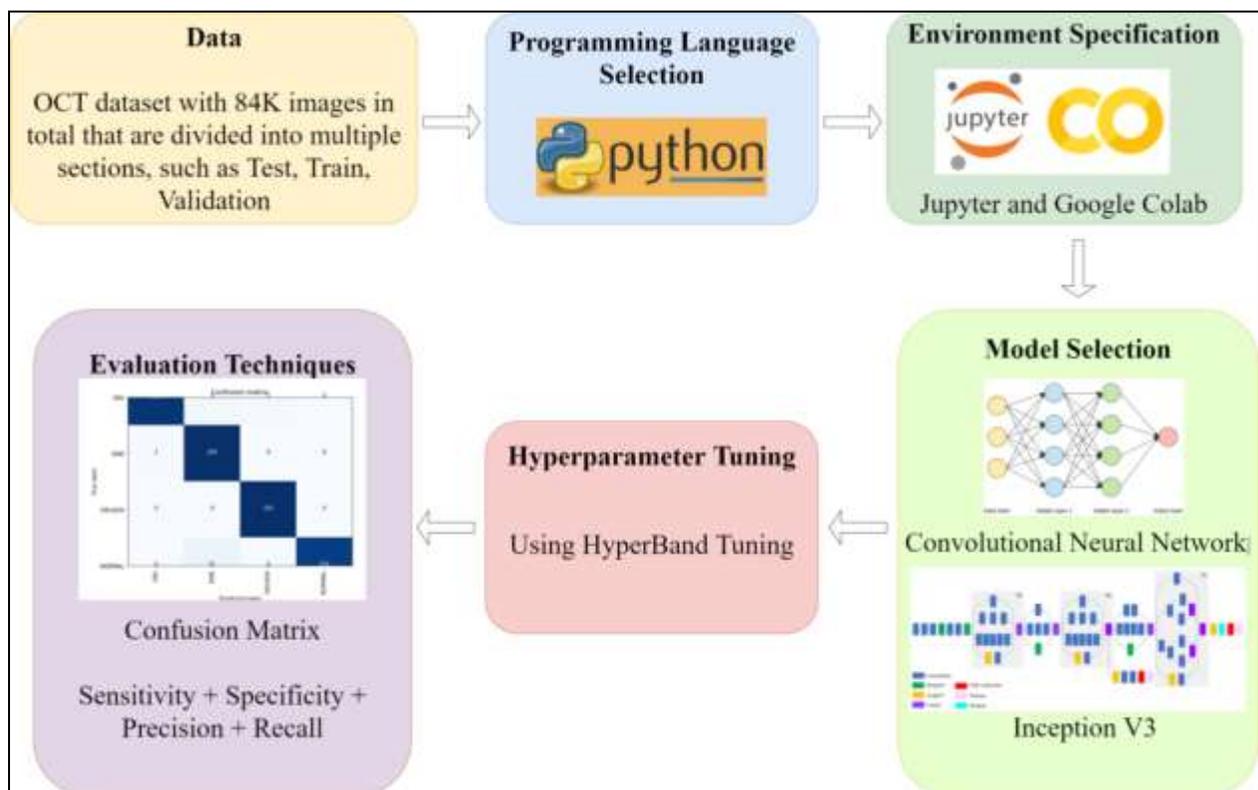


Figure 3. 2: Process Diagram

3.2 Data Understanding and Data Preparation

This section will explain few important aspects of the Data understanding and Preparation required for the research. We will be discussing few major points such as:

Data Understanding

- Data Collection
- Data Visualization

Data Preparation

- General Data Preparation
- Data Transformation
- Data Imbalance Handling

3.2.1 Data Understanding

While working with a huge dataset we need to take care of the cleanliness of the data and what kind of results it would generate in the end. We also need to understand what kind of preparation process can be applied on the dataset for it to provide the best outcomes possible. Along with this, we need to understand the possible ethical issues with the data and how they can be removed or addressed prior or during the study. All these points will be discussed below:

a) Data Collection

The data is collected from a public dataset which provides the images of eye retinas. This data is present on <https://data.mendeley.com/datasets/rscbjbr9sj/2>. This data has been made available by the combined effort of University of California San Diego and HUCM from Hunan University of Chinese Medicine. (Kermany et al. 2018). finally published it

for public use by hiding the patient's identity to ensure privacy. Merging these different datasets led us to have overall 80k+ images. The dataset consists of images with different retinal conditions namely Choroidal Neovascularisation (CNV), Diabetic Macular Edema (DME) and Drusen. Also, it contains images of normal retina. All the images present are in black and white color. As we can see from the figure 3.3 that all the images look like a normal eye and it needs a lot of effort to differentiate the diseased retina from a normal one.

As the dataset does not contain any identifying feature of the patients, the data is safe to use and follows all the ethical concerns.

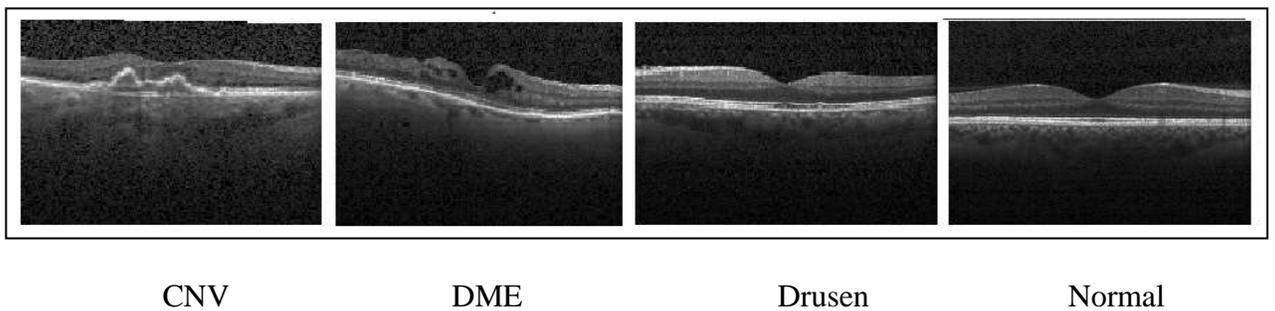


Figure 3. 3: Retinal Pathologies and healthy retina.

The dataset provided contained labeled images and two folders for testing and training datasets. We divided the whole dataset into few sections based on our requirement. We randomly selected images for Hyperparameter tuning and kept rest of the images for Modeling. The following table 3.1 shows the data segmentation done for the research.

Major Segments	Folder	Classes	Image Count
Hyperparameter Tuning	Train	CVN	1800
		DME	1656
		DRUSEN	1848
		NORMAL	1464
	Validate	CVN	8
		DME	8
		DRUSEN	8
		NORMAL	8
Model	Train	CVN	35405
		DME	9692
		DRUSEN	6768
		NORMAL	24851
	Test	CVN	250
		DME	250
		DRUSEN	250
		NORMAL	250
	Validate	CVN	8
		DME	8
		DRUSEN	8
		NORMAL	8
Total			84548

Table 3. 1: Dataset Image availability

b) Data Visualization

The dataset consists of total 84,452 images of four different classes; these classes are CNV, DME, DRUSEN and NORMAL. First three classes are based on the major retinal issues found in the eyes, whereas the last class only contains images of healthy eye retina. Based on our study, the dataset is divided into two main folders i.e., Hyperparameter tuning and modeling, that have random selections of the images from each category.

As shown in figure 3.4, the data present for hyperparameter tuning as well as modeling have class imbalance and the images for CNV are more compared to others, which can now be handled during modeling and tuning.

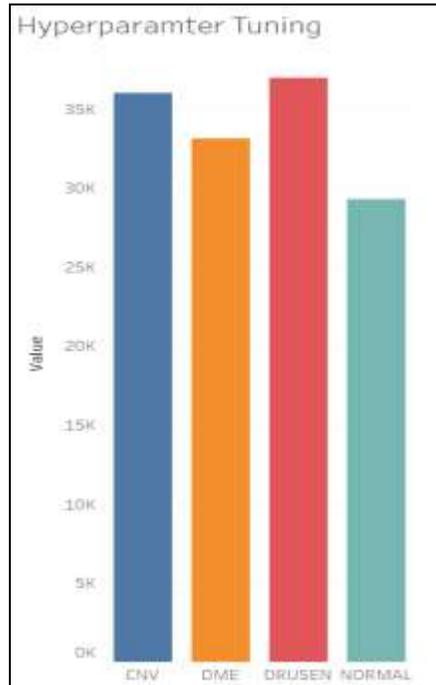


Figure 3. 4: Data Representation

3.2.2 Data Preparation

Preprocessing or preparing the data prior to modeling is critical as it ensures that the data will provide optimum results in the end. As our data consists of images of different sizes, we need to make sure that all the data is similar, to reduce the bias introduced due to it. The following processes are done for data preparation:

a) Dataset Preparation:

All our images have been made available in jpeg format. These are retinal images of dimension 512x496. The resolution of these images is very high hence we cannot feed these images directly for training our model as our computation power available to us is

not enough to store these images into the memory and process them fast. So, we had to reduce their dimension. We converted the images into NumPy arrays as well as CSV formats. The NumPy array is then converted into .npy files for feature extraction. The image size is kept as 28*28 for better processing.

b) Data Transformation:

In Data transformation process, we transform our data according to the need of our model. It may include removing some data or reducing dimensions of the data etc. Initially we had our data separated in two folders namely test and train with total 84k+ images. As we have a lot of actions to perform so we split the data into two subcategories namely Hyperparameter data and Model training data. Considering that no folder should contain the same images at the same time as it will introduce bias into the model, we randomly choose images for the two categories. The first folder consisted of approximately 6000+ images present in four different classes. We had fewer images in this folder as hyperparameter tuning does not require much data and the number of images we provide is suitable. Our hyperparameter folder contains two more subcategories i.e., train and validation images for tuning the hyperparameters. The second category i.e., Model training folder that contains 80000+ images, has three more subcategories namely test, train and validation. Here test set is required because we will need to test our model on previously unseen dataset. The split ratio is 80 percent training data, 15 percent test data and 5 percent validation data.

c) **Data Imbalance Handling:**

Our dataset contains images in different classes which are not equally distributed. Some classes have more data present in them than the other classes. This may lead in the model to become biased as it may overtrain on the class that has more data. Hence, we need to train our model in a dataset having almost equal images. We used RandomUnderSampler of python library which handles the class imbalance by removing random images from the class having more images which helps in removing model bias.

3.3 Modeling

Selecting and applying correct concepts while training and testing a model are some important aspects of the Deep learning algorithm implementation and so are the computational power as well the design, in this section we will be discussing all the mentioned segments of modeling:

Technical Requirements

- Model Selection
- Environment Specifications
- Programming Language Specification
- Hyperparameter Tuning:

Model Implementation

- CNN
- Inception V3

3.3.1 Technical Requirements

Training a deep learning model requires a high computation and it is not easily available that can fulfill the requirements of training the model in a very less time and with high accuracy. It is very important to choose the programming language as well because the libraries present in certain programming language are not up to the mark which may lead us to fewer options available considering the kind of training required for our model. Also choosing the best deep learning model is very important while considering the accuracy that is required. For this purpose, after going through certain state of the art approach we have come up with few solutions.

Below is the Figure 3.5, showing our choice for designing the model.

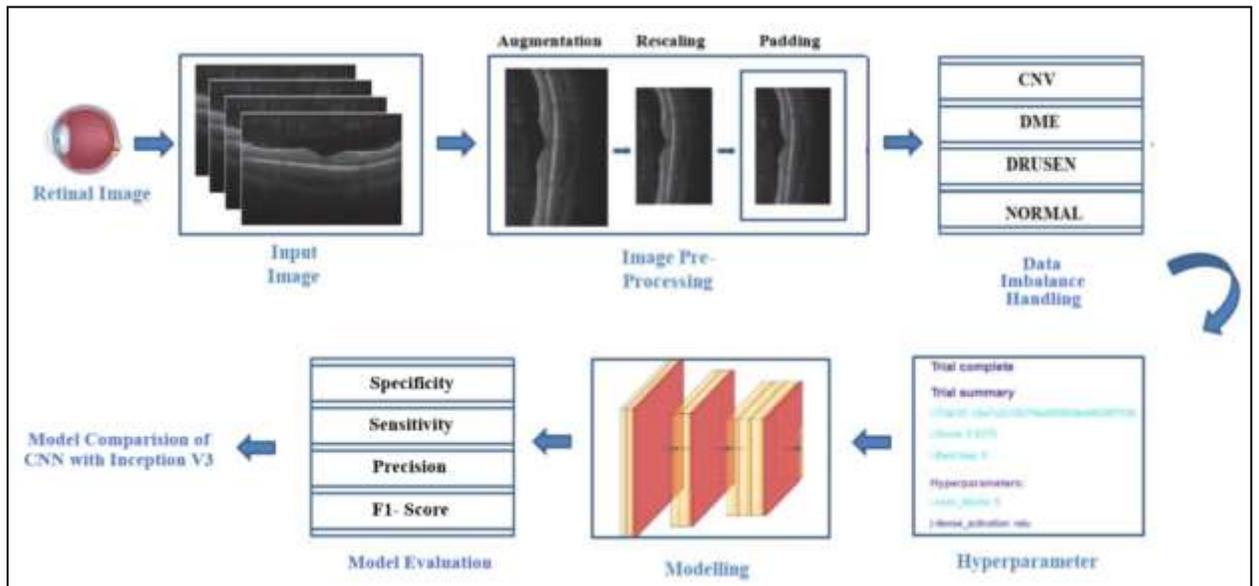


Figure 3. 5: Design of the Model

a) Model Selection:

The deep learning model to be used is chosen as per the requirements. There are numerous options available for choosing the deep learning model and it is very important to train our model and compare it with other models so that it gives more confidence on

our work. Here we have chosen Convolution Neural Network (CNN) based architecture. It is one of the classes of deep neural networks. We can consider CNN as regularized version of fully connected networks i.e., multilayer perceptron. As the layers are fully connected hence, there is less chance of overfitting the data. CNN make use of the hierarchical pattern of the data and hence this would be good for our image recognition as we must deal with the pattern that follows in the different classes of tomography images available to us. CNN is inspired by biological process. CNN is mostly comprised of three different layers i.e., input layer, multiple hidden layers, and output layer. The input that we are passing in our CNN is a tensor with of the image. Our Convolutional model has convolutional kernels, input channels and output channels.

With the help of rapid miner auto model, we also determined the general time taken by other methods that work well with this domain as shown in Table 3.2 and found that the deep learning models work best with this dataset.

*All values in %	Models	Fast Large Margin	Support Vector Machine	Deep Learning
Performance factors	Accuracy	41.1	27.3	63.4
	Classification Error	58.9	72.7	36.3
Class Precision	CNV	50.57	9.62	79.96
	DME	40.35	2.87	59
	DRUSEN	35.62	27.3	61.57
	NORMAL	35.48	5.21	51.44
Class Recall	CNV	60.51	6.35	76.85
	DME	24.26	3	67.23
	DRUSEN	53.13	100	56.44
	NORMAL	21	1.84	51.2

Table 3. 2: Rapid Miner auto model results

It can be easily noticed that the classification error is smallest for deep learning, whereas the Accuracy is highest for it. Deep learning algorithm took the least time in execution while both the other models were time consuming. It can be seen again in the figure 3.6. This helped us in understanding which model will work best for our research.



Figure 3. 6: Rapid Miner Auto Model Result.

b) Environment Specification

We used google-colab notebook to write python code. Google colab provides us with 16 GB of RAM and 12 GB of Graphics processing unit (GPU) memory. This makes image processing faster and hence resulting in taking less time to train the model. We can save the model training weights and continue working with the fastest GPU available to us. This reduces the overhead from our computer and helps in faster computation. It is free

for the users hence that make it more reliable. We uploaded all our dataset on the Google drive in order to mount it and later use it in our code. We also took the help of anaconda Jupyter for the coding as sometimes google colab faces issues related to session time out or the mounting expires. This enabled us to code at both the fronts and work faster.

c) Programming Language Specification:

We used python as our programming language as it provides libraries for machine learning which are easy to implement and are effective. This gives us the option of using TensorFlow which is very helpful while implementing any deep learning model. It gives us the ready to use libraries such as keras etc.

d) Hyperparameter Tuning:

In earlier days, due to less computational power and low storage capacity, people had less command over the flow of the data. Nowadays companies are mostly dependent on the inferences that come out of data and their business model revolves around it. So, there is a need to come up with analyzing the data more effectively using certain models. To get the desired result from any model, its hyperparameters needs to be considered. CNN has different convolutional layers. Each layer has certain convolutional filter which is pixel values, which needs to be tweaked during training. These hyperparameters are responsible for the behavior of the model. Training a model takes a lot of time and if we must try every combination of the hyperparameters to test our result then it may take a lot of time and still we may not end up with the desired output. To avoid this situation, there are different ways available to optimize the hyperparameters of the model automatically.

Here we are using Hyperband optimization technique. To implement this, we have used keras tuner. This tuner will help with choosing learning rate, number of layers, type of optimization function, activation function, pooling layers etc.

After working on the dataset and making it ready to use in our model, we have created a structure model for hyperparameter tuning of CNN. This model is validated on the validation dataset and multiple models are compared according to their accuracy and the best model is loaded to be used. Later, we analyze our model based on certain parameters and compare it with baseline approach.

3.3.2 Model Implementation

a) Convolutional Neural Network (CNN)

CNN is an algorithm that comes under deep learning, it generally takes the input and try to provide important weights and biases to the different aspects in the image so that it can easily differentiate between two images with the help of weights and biases. CNN consist of multiple layers which enables it to work faster and better with less pre-processing (Albawi et al. 2017). These layers are Input layer, Convolutional layer, Pooling layer, Fully connected layer, softmax and output layer. The presence of these layers can be seen in the Figure3.7.

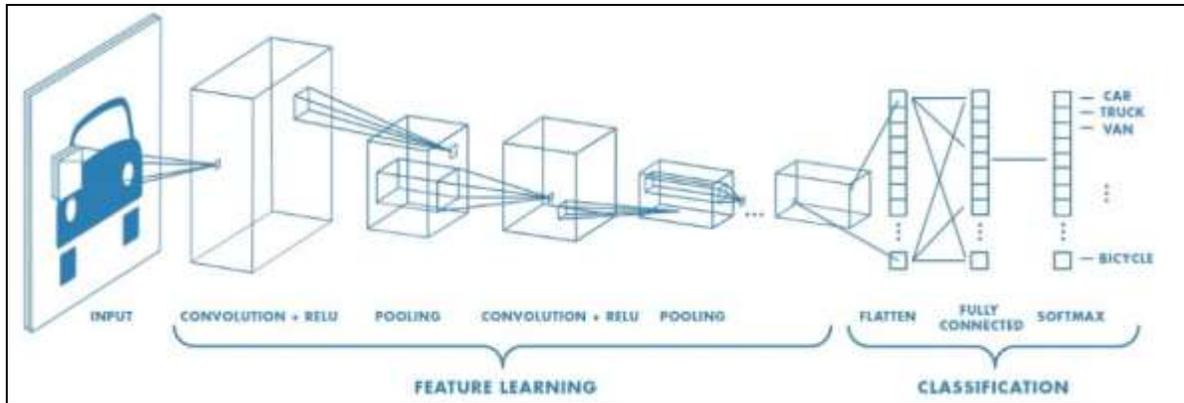


Figure 3. 7: CNN Arcitecture

Every layer of CNN works differently and provides different outcomes as discussed here:

Convolutional Layer: This layer is the most important layer of the network as it performs the function of dot product of two matrices. The first matrix is the kernel and the second matrix is a small portion of the image. This layer helps in reducing the image into a form that is easier to process. This is shown in Figure 3.8.

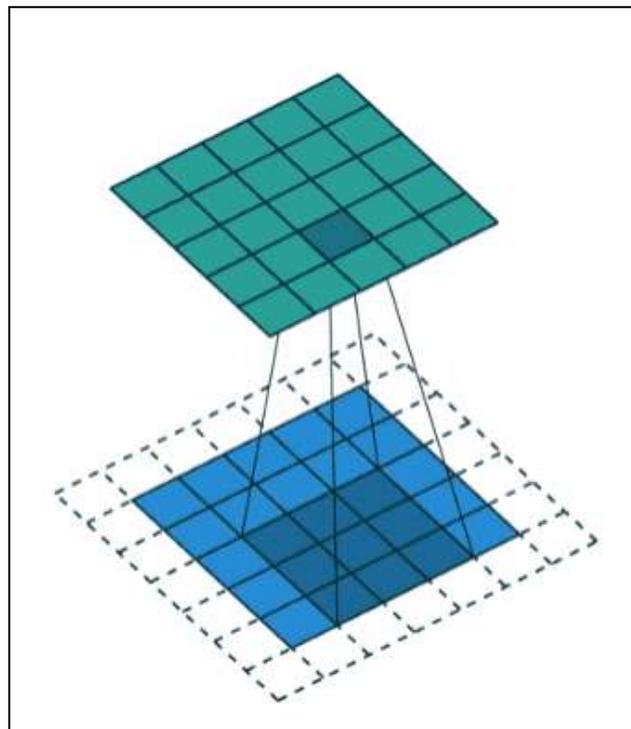


Figure 3. 8: Convolved Features

Pooling Layer:The output of convolutional layer is fed to pooling layer, which in turn reduces the size of the features so that less computation power is used for processing. This can be implemented using Max pooling, which is a way of taking the max value from the featured region, an example of this is shown in Figure 3.9.

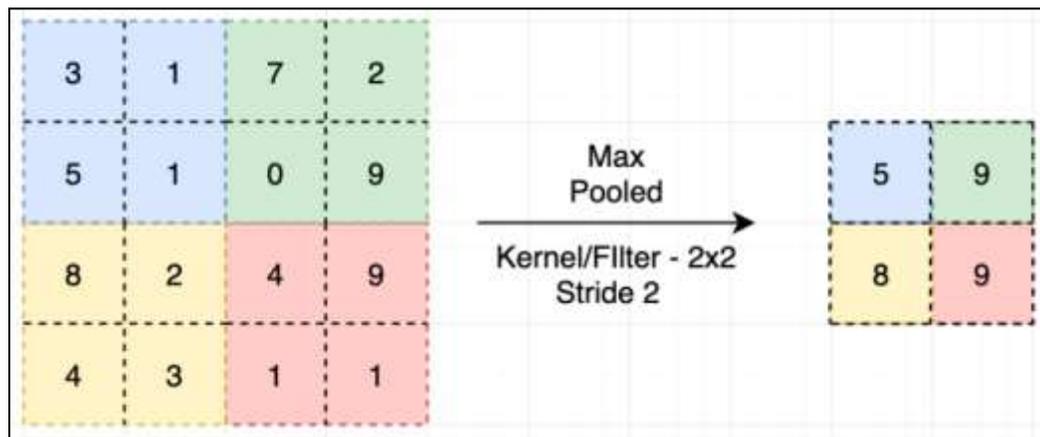


Figure 3. 9: Max pooling of features

Fully Connected Layer: The output of pooling layer is fed to fully connected layer, which then flattens it and creates a single vector, after which the classification starts with the help of weights and a probability is provided to the labels.

More details about the exact implementation of CNN are discussed here:

- The implementation started with data segmentation into different types i.e., hyperparameter tuning data and model training data. After dividing the respective datasets, we imported the whole dataset in our application.
- After importing the data, we decided to extract the features so that the training the model will take lesser time with better accuracy. So, we resized the images into 28*28 size and then converted all the images to a numpy array. We saved all the data into multiple “.npy” files to further use it.

- We started the hyperparameter tuning using keras tuner and the algorithm used is hyperband tuning. In this step, we defined convolution blocks between 3 to 5. Then filters between 32 to 256.
- Features are then learned from patterns in the image using the Rectified Linear Unit (ReLU) activation function. This function transforms the summed weighted input from the node to activate the said node or as a resultant output for that input. The ReLU function outputs the input directly if it is positive, otherwise, it sends an output of 0 (Agarap; 2018). A layer of batch normalization is applied to standardize the inputs to a layer for each mini batch. It is applied directly to the inputs from the layer and the activations from prior layers. The batch normalization layer reduces the number of epochs needed to train the model thus stabilizing the learning process and reducing the training time. The result from this is then passed into the pooling layer. The pooling layers offer an alternative to map sampling functions by summarizing the position of features in the feature map patches. The two pooling strategies applied during this project are the average pooling and the max pooling, which describe the average position of the feature and the most prominent position of the feature.
- After that we added dense layers with hidden size between 30 and 100. We choose dropout value between 0 and 0.5. Then the final output layer with activation function as SoftMax as we had multiclass classification problem here. Later we defined our learning rate to be between 0.01 and 0.0001. The choice of optimization algorithm was the Adam optimization algorithm. This was used over the traditional stochastic gradient descent approach to update the network weights in the training data. Using Adam optimizer, we compiled the model and chose our loss function as categorical cross entropy.

- After defining all the parameters, we used hyperband algorithm to search for the best set of hyperparameters by taking different combinations of the parameters. The maximum number of epochs selected is 8. Once the training was done; we got the best model out of multiple models.
- The next step for us to perform was model training on the dataset using the same model configurations obtained by the previous step. We performed this step on our local machine instead of google-colab as it was faster on the local machine. For that we needed to be sure that our model should not be biased for one class if some class has a greater number of images whereas the other has comparatively very less. So, we used RandomUnderSampler package from python to remove randomly certain images from the class having more images than the others and hence balancing the number of images in all the classes.
- After making data ready to be fed into model, we started training the model on the train data which was different from validation or testing data. The numbers of epochs taken were 5 and for the validation of the model purpose we used different set of images defined in validation data.
- Once the training was done, it was important to analyze our model and test it on the previously unseen dataset. We plotted training and validation loss graph and accuracy graph. After applying prediction on the test dataset, we plotted the confusion matrix and later checked for its specificity, sensitivity, precision, recall and f-1 score.
- Once our model was ready, we applied one more model to compare it with our model. We implemented the transfer learning model i.e., inceptionV3. We imported the model

and implemented it on our dataset and finally got the result using the process. Later its result was compared with that of our model.

The Figure 3.10 shows the layers present in our model.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 128, 128, 3)]	0
conv2d (Conv2D)	(None, 128, 128, 96)	4416
batch_normalization (BatchNo	(None, 128, 128, 96)	384
re_lu (ReLU)	(None, 128, 128, 96)	0
conv2d_1 (Conv2D)	(None, 128, 128, 96)	138336
batch_normalization_1 (Batch	(None, 128, 128, 96)	384
re_lu_1 (ReLU)	(None, 128, 128, 96)	0
average_pooling2d (AveragePo	(None, 64, 64, 96)	0
conv2d_2 (Conv2D)	(None, 64, 64, 96)	138336
batch_normalization_2 (Batch	(None, 64, 64, 96)	384
re_lu_2 (ReLU)	(None, 64, 64, 96)	0
conv2d_3 (Conv2D)	(None, 64, 64, 96)	138336
batch_normalization_3 (Batch	(None, 64, 64, 96)	384
re_lu_3 (ReLU)	(None, 64, 64, 96)	0
max_pooling2d (MaxPooling2D)	(None, 32, 32, 96)	0
conv2d_4 (Conv2D)	(None, 32, 32, 64)	92224
batch_normalization_4 (Batch	(None, 32, 32, 64)	256
re_lu_4 (ReLU)	(None, 32, 32, 64)	0
conv2d_5 (Conv2D)	(None, 32, 32, 64)	61504
batch_normalization_5 (Batch	(None, 32, 32, 64)	256
re_lu_5 (ReLU)	(None, 32, 32, 64)	0
max_pooling2d_1 (MaxPooling2	(None, 16, 16, 64)	0
conv2d_6 (Conv2D)	(None, 16, 16, 128)	123008
batch_normalization_6 (Batch	(None, 16, 16, 128)	512
re_lu_6 (ReLU)	(None, 16, 16, 128)	0
conv2d_7 (Conv2D)	(None, 16, 16, 128)	245888
batch_normalization_7 (Batch	(None, 16, 16, 128)	512
re_lu_7 (ReLU)	(None, 16, 16, 128)	0
max_pooling2d_2 (MaxPooling2	(None, 8, 8, 128)	0
conv2d_8 (Conv2D)	(None, 8, 8, 160)	307360
batch_normalization_8 (Batch	(None, 8, 8, 160)	640
re_lu_8 (ReLU)	(None, 8, 8, 160)	0
conv2d_9 (Conv2D)	(None, 8, 8, 160)	384160
batch_normalization_9 (Batch	(None, 8, 8, 160)	640
re_lu_9 (ReLU)	(None, 8, 8, 160)	0
max_pooling2d_3 (MaxPooling2	(None, 4, 4, 160)	0
global_average_pooling2d (Gl	(None, 160)	0
Flatten (Flatten)	(None, 160)	0
dense (Dense)	(None, 50)	8050
dropout (Dropout)	(None, 50)	0

Figure 3. 10: Layers of CNN according to research requirement

b) Inception V3

For this research, we are using Inception V3 as the State-of-the-art model. It is modified CNN architecture which uses Transfer learning for modeling. The model was pre-trained on ImageNet Dataset due to which it provides the possibility of containing information, which was not seen during normal CNN implementation, or can provide higher efficiency, better accuracy, and better results due to already being trained on a dataset. Inception V3 is also known to reduce the over fitting and the total computational power used for the modeling as discussed by (Szegedy et al. 2015). The architecture of Inception V3 can be seen in the Figure 3.11 below.

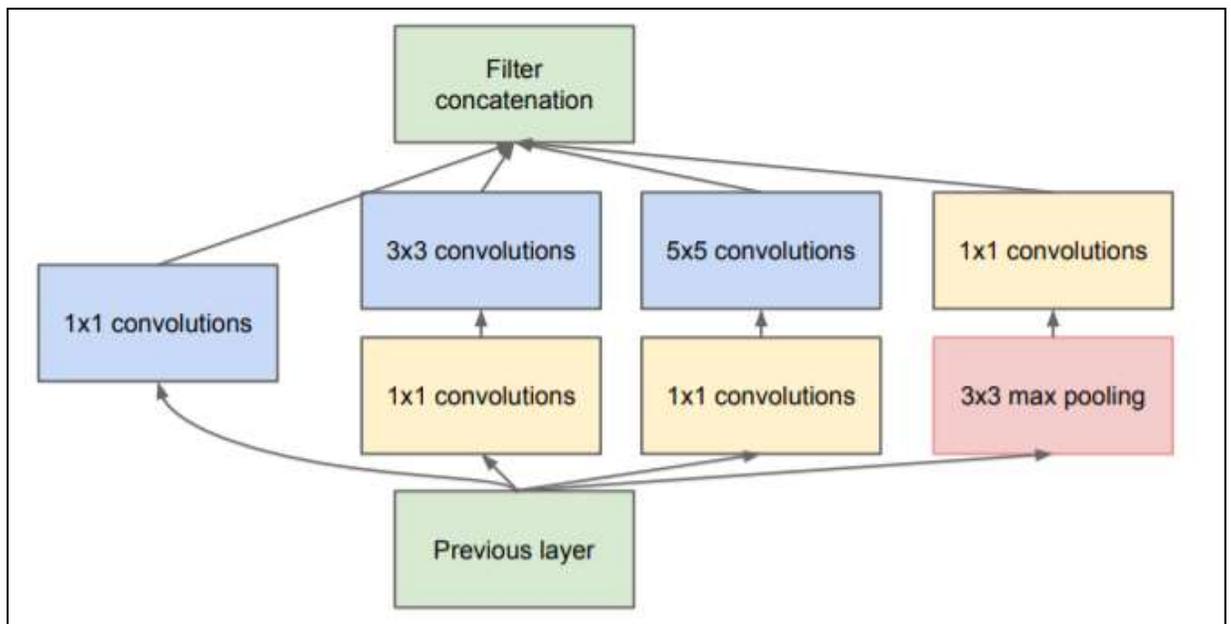


Figure 3.11: Inception V3 Model Architecture

Transfer learning helps in preparing a model quickly as rebuilding from scratch is not required. It takes the knowledge gained from performing one task and applies the information of perform similar tasks. It is easily explained with the help of Figure 3.12.

Here, Transfer learning is used for the comparison with CNN that is built from scratch to understand how the outcome is impacted.

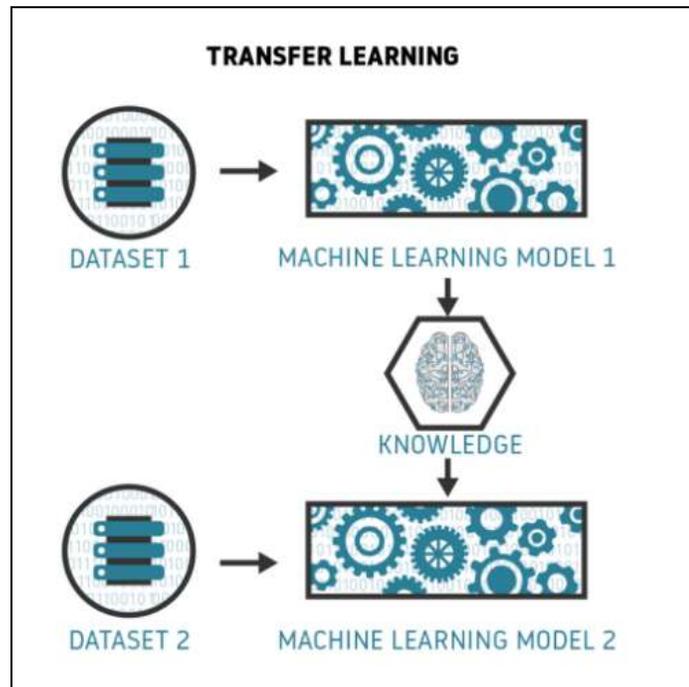


Figure 3. 12: Transfer Learning

The implementation of Inception V3 is as follows:

- The dataset was divided into three parts i.e., training, testing and validation.
- The “ImageDataGenerator” function in Keras was used to perform data augmentation on-the-fly during training. This works by first accepting a batch of images from the original dataset as the input, the function then transforms the batch of images using a number of randomized rotations while retaining the original class label. The returned value from this function is only the transformed batch of the images generated from the function. This is finally passed to the Inception V3 to train the model. This entire process is done at training time. The Figure 3.13 below shows an overview of the process carried out by the imagedatagenerator function.

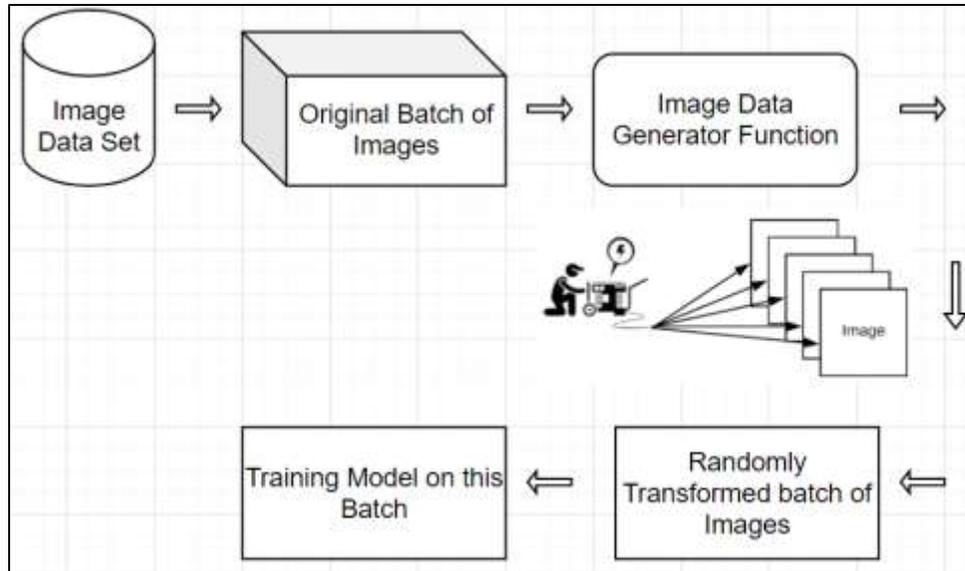


Figure 3.13: Process of Imagegenerator

- Training, test, and Validation generators were defined in order to import the dataset and perform other operations.
- Once it was done, we imported the inception model with output containing 4 classes and activation function as SoftMax.
- After compiling the model, we performed training on the training dataset with the help of validation images to validate the model in parallel.
- Training images contained a total of 61374 images whereas; validation contained 15342 images in total. The testing set also contained 15342 images in total.
- Prediction on test dataset was performed which received an accuracy of 92.73%, which can be considered good.
- Final interpretation of results was done with the help of confusion matrix, which explained how many of the images were predicted correctly.

- For further understanding, it was required to generate evolution factors such as specificity, precision, sensitivity, f-1 score to compare our model with the novel technique model.

The result of both the models will be discussed later.

3.4 Evaluation

To test and evaluate our model, it is very important to analyze how our model is performing on the previously unseen data. So, it is not only important to investigate the accuracy of the model but also other statistics such as F-1 score, specificity, sensitivity, confusion matrix, recall etc. As it is a medical field, we are more focused on false negatives because it affects more than any other matrix. So, sensitivity is the focus.

We will also look at validation accuracy to check if our model is not biased and to check for variance introduced, we will look how different our validation accuracy is from our test accuracy. Let us further discuss on what evaluation matrices our model will be tested: -

- **Precision** can be described as the ratio of true positives and total positives; this can be understood using an example of patients in a medical field, all the patients getting correctly diagnosed are true positives and all the patients having the disease is considered as total positives.

$$Precision = \frac{TP}{(TP + FP)}$$

The model presenting x% precision concludes that the model can predict correctly for x% times.

- **Sensitivity** is used to measure the proportion of a model correctly identifying true positives and is same as recall. It can also be understood as diagnosing an unhealthy person as unhealthy instead of diagnosing them as healthy. It tells us how accurately the model can detect the relevant data.

$$Sensitivity = \frac{TP}{(TP + FN)}$$

Any model with x% sensitivity will tell us that the model can identify actual positive x% times.

- **Specificity** can be described as the ratio of True Negative and total true negatives. For example, it is trying to determine how many people are not diagnosed with the disease out of everyone who is not having the disease.

$$Specificity = \frac{TN}{(TN + FP)}$$

Any model that is having the specificity of x%, will be able to detect true negatives correctly for x% times.

- **F1 Score** is a measure to understand the balance between precision and sensitivity, also known as recall. It a harmonic mean of the two values and the model that is having a good F1- Score will be working better than any other model having less F1-Score.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

3.5 Deployment

There are many possibilities in case of deployment of this study. The first and foremost is to make it available to Ophthalmologists as an application, so that they can use it to study the symptoms and pre diagnose the issues before providing a final diagnosis based on all the factors assembled. This can also be used in Medical universities for research purposes in form of a website or information manual. But as it is a research and more work needs to be done on before making it available to public, it is deployed in the form of this report.

CHAPTER 4- RESULTS ANALYSIS

Here we are going to discuss about a comprehensive analysis of the results obtained and their implications on using the model for future purpose. We also discuss about how our model is better than other pre-trained models. It is very important to analyze the results obtained after our model predicts for the test images. We also check about any bias or variance in the model present. We will plot the graph to see how our model improves its accuracy and loss is reduced with each epoch. We will also look at the other results such as specificity, sensitivity, accuracy, f-1 score etc. to understand more about our model.

4.1 CNN Model

While implementing CNN we provided images to it for model training, validation, and testing. Model training was done on 76716 images. Total numbers of epochs considered were 5. We achieved a high accuracy of about 97.7%. We achieved validation accuracy as 93%. This shows that there was very less variance introduced by our model. We can see from the graph below how our accuracy increased with each epoch and how we reduced the loss.

Here our training accuracy and validation accuracy increased for certain time reaching nearly about 100% and then the rate slowed down. We can also see from the Figure 4.1, that training loss and validation loss gradually decreased with every epoch. Validation loss increased suddenly at a point but again started descending.

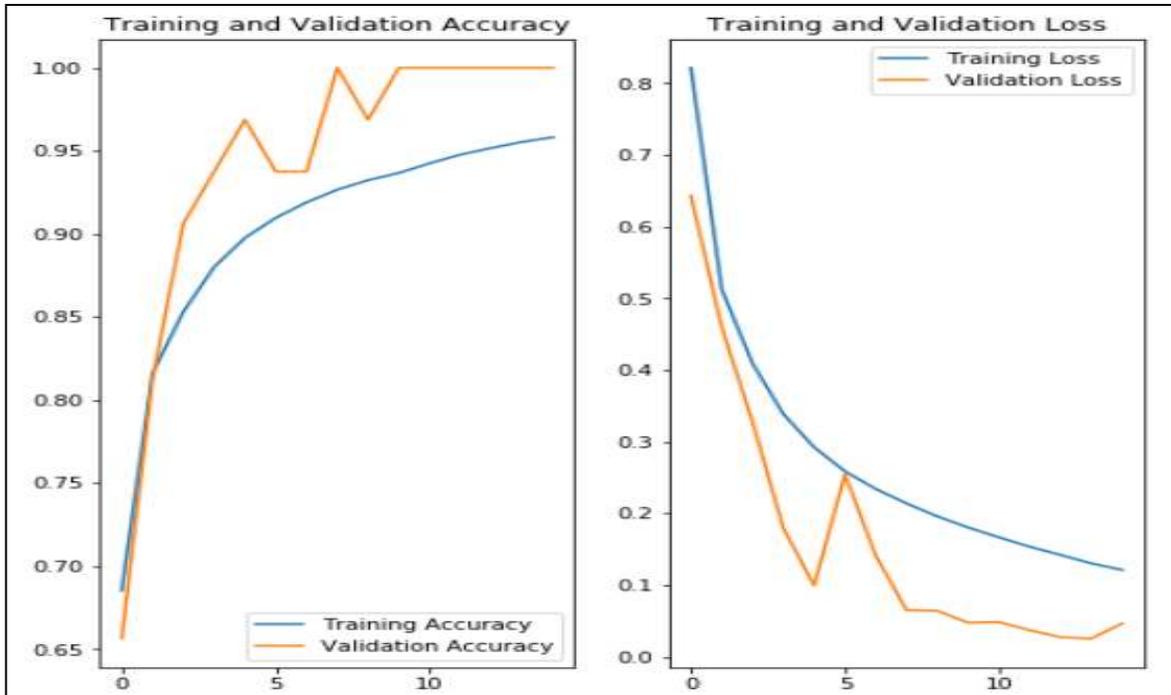


Figure 4. 1: Training Vs Validation Graph

To gain further insight into our model's performance, we created a confusion matrix as shown in Figure 4.2, which can be understood as per Figure 4.3.

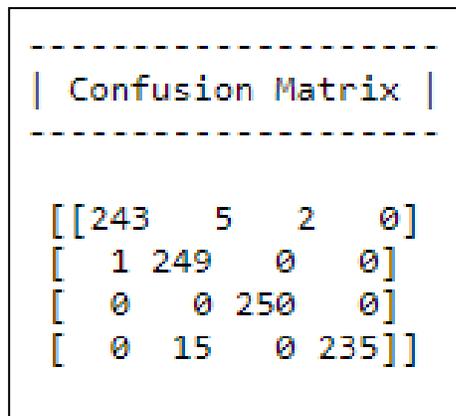


Figure 4. 2: Confusion Matrix

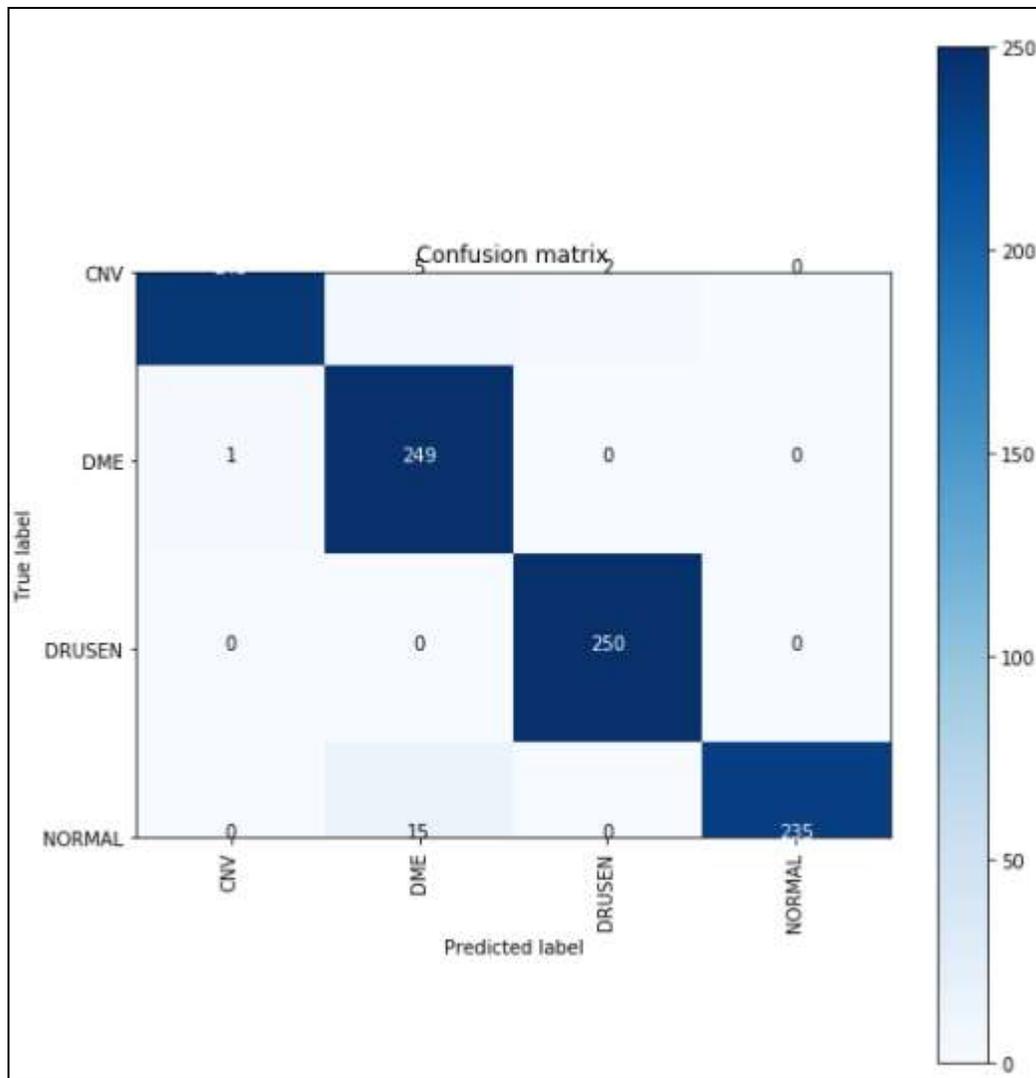


Figure 4. 3: Visual representation of Confusion Matrix for Hyperparameter-Tuned CNN

Here can see from the Figure 4.3, that out of 250 CNV images total 243 were identified correctly whereas 5 were identified as DME and 2 as Drusen which was incorrect. Again, in DME 249 out of 250 images were identified correctly excluding one image. In Drusen, all the images were identified correctly by our model and 15 normal images were classified as DME whereas rests of 235 images were identified correctly.

As our model is based on working on the medical field, it is very important to look at other parameters such as specificity, sensitivity, false negative rate etc. We have created a matrix as shown for the same in Table 4.1. Precision tells us about positive predictive value. Here we can see that the precision for CNV was 1 which means the ratio of true positive to the sum of true positive and false positive was 1. Also, it was 0.93 for DME, 0.99 for Drusen and 1 for normal images. So, for DME the no of true positives was less than the other categories. Though, overall results look satisfying. Similarly recall talks about the ratio of true positive and the sum of true positive and false negative. As the scores are high so we can say that there were less false negative cases which is a good result in terms of medical field.

Method	Class	Precision (%)	Sensitivity (%)	Specificity (%)	Accuracy (%)	Overall Accuracy (%)
CNN	CNV	99.5	97.2	99.8	99.2	97.7
	DME	92.5	99.6	97.3	97.9	
	DRUSEN	99.2	1	99.7	99.8	
	NORMAL	1	94	1	98.5	

Table 4. 1: Output of CNN

Here we can see that the results that we got for sensitivity for the different classes was near about 100% except for DME which was 0.97. Hence, there were very less cases in which there was a certain disease present, but we predicted it as some other or no disease, which is a very good result in medical field and worth considering. Specificity values are also close to 1 hence our true negative values are almost all correct. Overall accuracy for different categories is also near about 99% whereas the minimum value was 97%. The high accuracy achieved here is very helpful in ensuring that our model will perform better than other models available and this can be used optometrists confidently.

4.2 Inception V3

Inception V3 is applied on the same dataset to compare the results with CNN. While implementing this model we provided it 61374 images for training the model and another 15342 for validation. The value of epochs was kept as 31 and using this accuracy of the model was 92% was and for validation it was around 88.9%. This shows that there was very less variance introduced by this model.

To further understand the model's performance, we created a confusion matrix which is shown in Figure 4.4 and Figure 4.5.

Confusion Matrix				
[[1247	2129	53	257]
[287	651	11	70]
[162	471	23	44]
[539	1738	76	242]]

Figure 4. 4: Confusion matrix for Inception V3

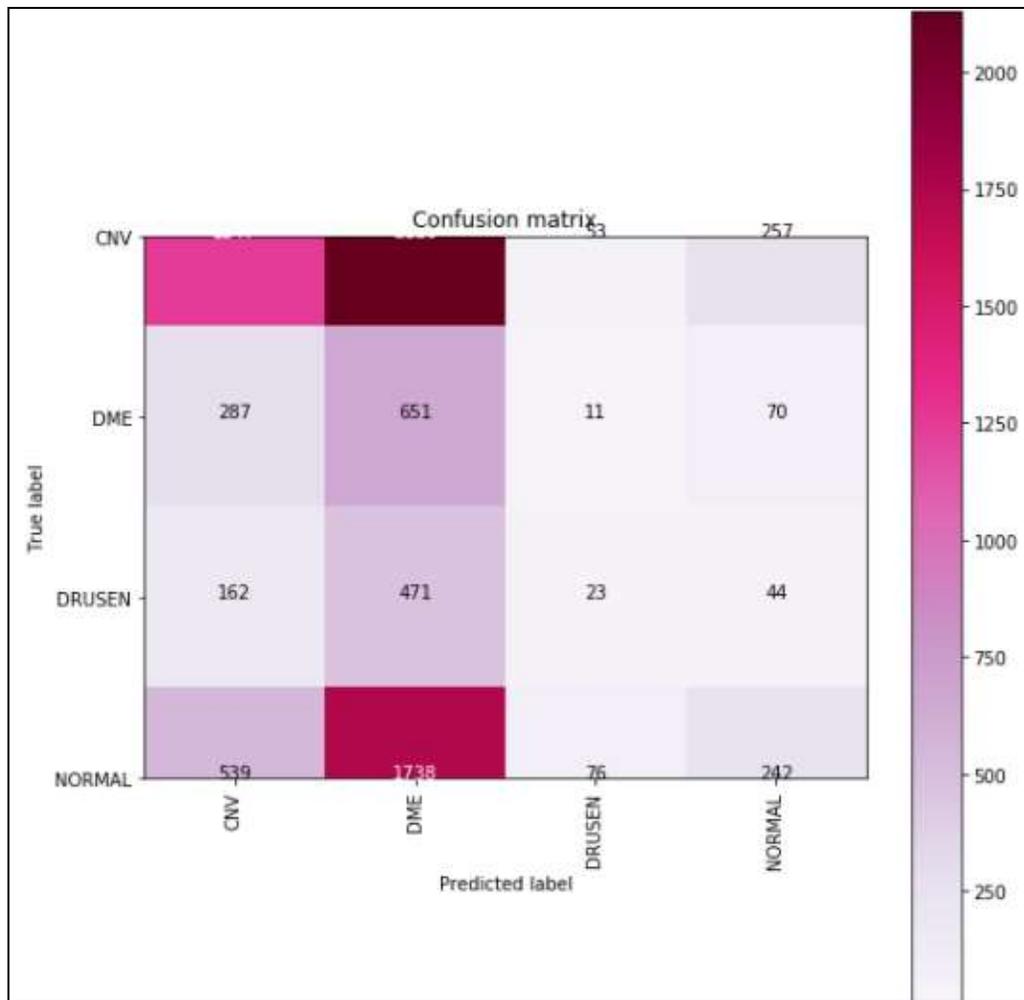


Figure 4. 5: Visual representation of Confusion matrix for Inception V3

It can easily be seen in Figure 4.5 that Inception V3 model was able to correctly classify 1247 CNV images, whereas 651 were correctly identified for DME pathology. For DRUSEN and Normal the model was able to correctly identify 23 and 242 images, respectively.

Inception V3	CNV	55.7	33.8	77	57.1	92.7
	DME	13	63.8	37.8	41.1	
	DRUSEN	14.1	3.2	98	89.7	
	NORMAL	39.4	9.3	93.1	65.9	

Table 4.2: Output of Inception V3

As our model is based on working on the medical field, it is very important to look at other parameters such as specificity, sensitivity, false negative rate etc. We have created a matrix as shown in Table 4.2. Precision tells us about positive predictive value. Here we can see that the precision for CNV was .55 which means the ratio of true positive to the sum of true positive and false positive was 0.55. Also, it was 0.13 for DME, 0.14 for Drusen and 10.39 for normal images. So, for DME the no of true positives was less than the other categories. Though overall accuracy looks satisfying and it can be noted that the model is able to identify DRUSEN the best with 89% accuracy.

Similarly recall talks about the ratio of true positive and the sum of true positive and false negative. As the scores are both high and low so we can say that there were less false negative cases for DME and Normal but for CNV and DME there were more false negatives which is not so good in terms of medical field.

As Discussed above, it can be easily compared that the model with hyperparameter tuning done by keras tuner worked better than the pre-trained model in terms of accuracy, sensitivity for each class, Precision, Specificity. We can see that in every aspect our CNN model surpassed the quality of pre-trained model.

Though our model was good enough to work in most of the situations but in case of DME its specificity is 97.3. This could have been improved better by taking more images for DME in the training set and by using feature extraction techniques.

A summary of all the results can be found in the table 4.3 below:

Method	Class	Precision (%)	Sensitivity (%)	Specificity (%)	Accuracy (%)	Overall Accuracy (%)
CNN	CNV	99.5	97.2	99.8	99.2	97.7
	DME	92.5	99.6	97.3	97.9	
	DRUSEN	99.2	1	99.7	99.8	
	NORMAL	1	94	1	98.5	
Inception V3	CNV	55.7	33.8	77	57.1	92.7
	DME	13	63.8	37.8	41.1	
	DRUSEN	14.1	3.2	98	89.7	
	NORMAL	39.4	9.3	93.1	65.9	

Table 4.3: Consolidated table of all the results

CHAPTER 5- CONCLUSION AND FUTURE WORK

The detection of retinal pathologies at an initial stage is critical for the treatment of blurriness and blindness in the eyes caused due to diseases. Automated detection systems can help ophthalmologists in diagnosing the issues much more accurately and quickly so that enough time can be given to other procedures. Our study is focused on detecting diseases present in the retina of the eye using Optical Coherence Tomography technique with the help of deep learning models and to understand what works best. Transfer learning technique such as Inception V3(He et al.; 2020) is applied for the purpose of comparing our proposed method's results. The dataset is a combined effort of UCSD and HUCM datasets consisting of 80K+ images and was converted in CSV format as well as. npy formal for better processing. The proposed method has provided promising results with overall accuracy of 97.7% and all the values of other comparative parameters such as specificity and sensitivity are also greater. This has been achieved due to no manual intervention in hyperparameter tuning. The only limitations of this study were the time-consuming nature of both hyperparameter tuning and training the model on such a large dataset.

For the future work, more eye related pathologies can be detected using this same process. This research work can also be presented as the starting point for many other studies in the medical field. As we have used CNN for the for the detection, other models such as Support Vector Machine (SVM), Inception recurrent residual convolutional neural network (IRRCNN) and random forest can be investigated for a comparative study. With the help of higher computational power and higher number of epochs, the accuracy can be further improved. In future we need to work on the question that Will Models such as SVM and Fast Large Margin be able to takes lesser time and provide better results?

REFERENCES

- Agarap, A.F., 2018. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.
- Albawi, S., Mohammed, T.A. and Al-Zawi, S., 2017, August. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)* (pp. 1-6). IEEE.
- Alqudah, A.M., 2020. AOCT-NET: a convolutional network automated classification of multiclass retinal diseases using spectral-domain optical coherence tomography images. *Medical & biological engineering & computing*, 58(1), pp.41-53.
- Alsaih, K., Lemaitre, G., Rastgoo, M., Massich, J., Sidibé, D. and Meriaudeau, F., 2017. Machine learning techniques for diabetic macular edema (DME) classification on SD-OCT images. *Biomedical engineering online*, 16(1), p.68.
- Baudoin, C.E., Lay, B.J. and Klein, J.C., 1984. Automatic detection of microaneurysms in diabetic fluorescein angiography. *Revue d'épidémiologie et de santé publique*, 32(3-4), pp.254-261.
- Bayleyegn, M.D., Makhlof, H., Crotti, C., Plamann, K. and Dubois, A., 2012. Ultrahigh resolution spectral-domain optical coherence tomography at 1.3 μm using a broadband superluminescent diode light source. *Optics Communications*, 285(24), pp.5564-5569.
- Bhowmik, A., Kumar, S. and Bhat, N., 2019, May. Eye disease prediction from optical coherence tomography images with transfer learning. In *International Conference on Engineering Applications of Neural Networks* (pp. 104-114). Springer, Cham.
- Chan, G.C., Muhammad, A., Shah, S.A., Tang, T.B., Lu, C.K. and Meriaudeau, F., 2017, September. Transfer learning for diabetic macular edema (dme) detection on optical coherence tomography (oct) images. In *2017 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)* (pp. 493-496). IEEE.
- Chiu, S.J., Li, X.T., Nicholas, P., Toth, C.A., Izatt, J.A. and Farsiu, S., 2010. Automatic segmentation of seven retinal layers in SDOCT images congruent with expert manual segmentation. *Optics express*, 18(18), pp.19413-19428.
- Das, V., Dandapat, S. and Bora, P.K., 2019. Multi-scale deep feature fusion for automated classification of macular pathologies from OCT images. *Biomedical Signal Processing and Control*, 54, p.101605.
- Farsiu, S., Chiu, S.J., O'Connell, R.V., Folgar, F.A., Yuan, E., Izatt, J.A., Toth, C.A. and Age-Related Eye Disease Study 2 Ancillary Spectral Domain Optical Coherence Tomography Study Group, 2014. Quantitative classification of eyes with and without intermediate age-related macular degeneration using optical coherence tomography. *Ophthalmology*, 121(1), pp.162-172.

He, X., Fang, L., Rabbani, H., Chen, X. and Liu, Z., 2020. Retinal Optical Coherence Tomography Image Classification with Label Smoothing Generative Adversarial Network. *Neurocomputing*.

Huang, D., Swanson, E.A., Lin, C.P., Schuman, J.S., Stinson, W.G., Chang, W., Hee, M.R., Flotte, T., Gregory, K. and Puliafito, C.A., 1991. Optical coherence tomography. *science*, 254(5035), pp.1178-1181.

Huang, L., He, X., Fang, L., Rabbani, H. and Chen, X., 2019. Automatic classification of retinal optical coherence tomography images with layer guided convolutional neural network. *IEEE Signal Processing Letters*, 26(7), pp.1026-1030.

Islam, K.T., Wijewickrema, S. and O'Leary, S., 2019, June. Identifying Diabetic Retinopathy from OCT Images using Deep Transfer Learning with Artificial Neural Networks. In *2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS)* (pp. 281-286). IEEE.

Johnson, P.T., Lewis, G.P., Talaga, K.C., Brown, M.N., Kappel, P.J., Fisher, S.K., Anderson, D.H. and Johnson, L.V., 2003. Drusen-associated degeneration in the retina. *Investigative ophthalmology & visual science*, 44(10), pp.4481-4488.

Kanagasingam, Y., Bhuiyan, A., Abramoff, M.D., Smith, R.T., Goldschmidt, L. and Wong, T.Y., 2014. Progress on retinal image analysis for age related macular degeneration. *Progress in retinal and eye research*, 38, pp.20-42.

Kermany, D.S., Goldbaum, M., Cai, W., Valentim, C.C., Liang, H., Baxter, S.L., McKeown, A., Yang, G., Wu, X., Yan, F. and Dong, J., 2018. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172(5), pp.1122-1131.

Kharisudin, I., Az-Zahra, M.F., Winarti, E.R. and Waluya, S.B., 2020, June. Deep convolutional neural networks for the detection of macular diseases from optical coherence tomography images. In *Journal of Physics: Conference Series* (Vol. 1567, No. 2, p. 022076). IOP Publishing.

LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. *nature*, 521(7553), pp.436-444.

Lemaître, G., Rastgoo, M., Massich, J., Cheung, C.Y., Wong, T.Y., Lamoureux, E., Milea, D., Mériaudeau, F. and Sidibé, D., 2016. Classification of SD-OCT volumes using local binary patterns: experimental validation for DME detection. *Journal of ophthalmology*, 2016.

Li, F., Chen, H., Liu, Z., Zhang, X. and Wu, Z., 2019. Fully automated detection of retinal disorders by image-based deep learning. *Graefes Archive for Clinical and Experimental Ophthalmology*, 257(3), pp.495-505.

Marginean, A., Groza, A., Nicoara, S.D., Muntean, G., Slavescu, R.R. and Letia, I.A., 2019, September. Towards Balancing the Complexity of Convolutional Neural Network with the Role of Optical Coherence Tomography in Retinal Conditions. In *2019 IEEE 15th International*

Conference on Intelligent Computer Communication and Processing (ICCP) (pp. 475-482). IEEE.

Moreira, E.F., Larrayoz, I.M., Lee, J.W. and Rodríguez, I.R., 2009. 7-Ketocholesterol is present in lipid deposits in the primate retina: potential implication in the induction of VEGF and CNV formation. *Investigative ophthalmology & visual science*, 50(2), pp.523-532.

Nugroho, K.A., 2018, October. A comparison of handcrafted and deep neural network feature extraction for classifying optical coherence tomography (OCT) images. In *2018 2nd International Conference on Informatics and Computational Sciences (ICICoS)* (pp. 1-6). IEEE.

Roy, A.G., Conjeti, S., Karri, S.P.K., Sheet, D., Katouzian, A., Wachinger, C. and Navab, N., 2017. ReLayNet: retinal layer and fluid segmentation of macular optical coherence tomography using fully convolutional networks. *Biomedical optics express*, 8(8), pp.3627-3642.

Sertkaya, M.E., Ergen, B. and Togacar, M., 2019, June. Diagnosis of eye retinal diseases based on convolutional neural networks using optical coherence images. In *2019 23rd International Conference Electronics* (pp. 1-5). IEEE.

Sun, Y., Li, S. and Sun, Z., 2017. Fully automated macular pathology detection in retina optical coherence tomography images using sparse coding and dictionary learning. *Journal of biomedical optics*, 22(1), p.016012.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

Tsuji, T., Hirose, Y., Fujimori, K., Hirose, T., Oyama, A., Saikawa, Y., Mimura, T., Shiraishi, K., Kobayashi, T., Mizota, A. and Kotoku, J.I., 2020. Classification of optical coherence tomography images using a capsule network. *BMC ophthalmology*, 20(1), pp.1-9.

Xiang, D., Tian, H., Yang, X., Shi, F., Zhu, W., Chen, H. and Chen, X., 2018. Automatic segmentation of retinal layer in OCT images with choroidal neovascularization. *IEEE Transactions on Image Processing*, 27(12), pp.5880-5891.

APPENDICIES

This document is made for the purpose of guiding you through the content of artifacts provided for the research title “**Classification of Retinal Pathology from OCT images using a Parametric Tuned CNN**”. This document contains the steps to run the python code in Google Colab or Jupyter Notebook.

Artifact contents:

1. Dataset

As the dataset is quite a large file, we have shared a text file with the link of the dataset as “**Dataset_link.txt**”.

2. CSV Files

We have used .csv and .npy files for the purpose of auto model and rest of the coding, which is shared in form of “**CSV Files**” folder, containing .csv files for all the folders of the dataset. ‘.npy’ files are not shared as they are very large and are not possible to upload.

3. Python Code.

The code of the project is provided in .ipynb file format with the name “**10543362-Shalini Singh_Code.ipynb**”. This file can be opened directly in Google colab or jupyter notebook by Uploading the file and using ‘Open file’ method.

4. Tableau Visualization file.

As the dataset just contains labels and values the visualization of classes is shared in tableau file “**10543362-Hyperparameter Data Visulization.twb**”.

5. Rapid Miner Auto Model Results

Auto model is applied on some part of the results and is shared in the artifacts with name “**10543362_Rapid Miner_AutoModel.rmp**”.