DESIRE: Deep learning Enabled System Integration for Retinopathy Evaluation

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Technology

In

Computer Science and Engineering

Submitted by

Karan Kalra

Roll No. 801632020

Under the supervision of:

Dr. Sanmeet Kaur

Assistant Professor, CSED Department

Dr. Parteek Bhatia

Associate Professor, CSED Department

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY
PATIALA – 147004

June 2018

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "*DESIRE: Deep learning Enabled System Integration for Retinopathy Evaluation*", in partial fulfillment of the requirements for the award of degree of Master of Technology in *Computer Science and Engineering* submitted in Computer Science and Engineering Department of Thapar Institute of Engineering and Technology, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Parteek Bhatia and Dr. Sanmeet Kaur* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

(Karan Kalra)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr. Sanmeet Kaur)

Assistant Professor, CSED

(Dr. Parteek Bhatia)

Associate Professor, CSED

ACKNOWLEGEMENT

First of all I would like to thank the Almighty, who has always guided me to work on the right path of the life.

This work would not have been possible without the encouragement and able guidance of my supervisor **Dr. Sanmeet Kaur.** I thank my supervisor for their time, patience, discussions and valuable comments. Their enthusiasm and optimism made this experience both rewarding and enjoyable. Their discipline and sincerity towards work, teaches sincerity is more important than seriousness in life.

I am equally grateful to **Dr. Parteek Bhatia**, Associate Professor, Computer Science & Engineering Department, a nice person, an excellent teacher and a well – credited researcher, who always advised me with his valuable suggestions.

I will be failing in my duty if I don't express my gratitude to **Dr. S.S. Bhatia,** Senior Professor and Dean of Academic Affairs, TIET, for making provisions of infrastructure such as library facilities, computer labs equipped with net facilities, immensely useful for the learners to equip themselves with the latest in the field.

I am also thankful to the entire faculty and staff members of Computer Science and Engineering Department for their direct-indirect help, cooperation, love and affection, which made my stay at TIET memorable.

Last but not least, I would like to thank my family whom I dearly miss and without whose blessings none of this would have been possible. To my parents, I own thanks for their wonderful love and encouragement. I would also like to thank my sister, since she insisted me in doing so. I would also like to thank my close friends for their constant support.

Date: 29 June, 2017

Place: TIET, Patiala

(Karan Kalra)

ABSTRACT

Diabetic Retinopathy (DR) is one of the significant reasons for visual impedance in the middle age population. An automated approach for detection of DR has an extraordinary social effect on rural population due to lack of expert ophthalmologists. The pre-screening of DR is exceedingly viable in averting the irreversible vision loss however its detection is tedious and an impeded process. Several strategies have been proposed in the literature, the greater part of them depend on the morphological component extractions. Deep learning conversely enhances the results when trained on the annotated data. In this paper, an automated DR grading algorithm is exhibited that examines patients' fundus images having a distinctive field of view and light intensity. It then classifies the images to their particular severity grade using deep learning.

The main focus is to diminish the false normal (Fnorm) recognition rate of the framework to maintain a strategic distance from the characterization of DR influenced eye as normal. The vast majority of the therapeutic recognition approaches center around expanding the precision as opposed to stifling the Fnorm rate which influences its appropriateness in a genuine. The paper initially compares different preprocessing techniques and their impact on enhancing the image quality followed by the medical data balancing approaches. The proposed deep learning based algorithms are trained over a publicly accessible EyePACS dataset of 35126 images and give a high accuracy of around 98.98% with a reasonable Fnorm value. DR identification approach can be utilized to supplant the manual screening procedures and help the ophthalmologist with precise primary screening.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

When the glucose level in the blood vessel of the retina is increased, the vision ends up noticeably obscured and without appropriate treatment, it can lead to complete blindness, this process of damage within the retina is called diabetic retinopathy [1, 2].

The progression of Diabetic Retinopathy is at different rates in different persons because of the two important vision pressuring difficulties: diabetic macular edema (DME) and proliferative retinopathy. That is the reason there is a tremendous interest in the most recent advancements and methodologies to analyze DR effectively. Researches show that progression to vision impairment can be slowed or averted if it is detected in the early stage of the disease.

Diabetic Retinopathy had influenced 3.4 percent of the aggregate population (4.1 million individuals) with nearly one-fourth of the general population affected by DR has the vision-debilitating disease. It contributes around 5% of total cases of blindness. As indicated by World Health Organization (WHO) estimation 347 million [3] of the world population is having the disease diabetes and about 40-45% of them have some stage of the DR out of which 75% of the general population with diabetic retinopathy lives in the developing countries. The manifestations of diabetic retinopathy don't appear in the early stages, which makes it much harder to keep the patient from being visually impaired. To prevent the delayed detection of DR, people affected by diabetes are usually prescribed for general examinations. Ascend in a number of patients subsequently increase the workload for the ophthalmologist for the early detection of diabetic retinopathy. Thus, there is high consensus on the need for an automated, time and cost effective algorithm which can help in the primary screening of the DR [4].

Deep Learning [5, 6, 7, 8, 9, 10] has recently emerged as an effective approach to settle a substantial assortment of image recognition problems. Such approach can automate the feature extraction [11, 12] process from adequately substantial dataset without requiring any manual impedance. Deep learning approaches are known to

outperform other physically designed methodologies on a huge assortment of utilization. Their segregation capacity is typically influenced by the measure of the accessible dataset. Deeper architectures learn more discriminative features, generally at the cost of requiring sufficiently large dataset to guarantee an appropriate generalization error and avert overfitting [13].

There are certain issues related to an automated grading of DR effective fundus image. Accomplishing a high accuracy and low false normal rate is essentially harder for multi-class dataset comprising of five classes namely normal, mild DR, moderate DR, severe DR, and PDR. Moreover, large datasets are generally skewed which produces a biased classifier that considers minority class instances as outliers and classifies all instances to the most conspicuous class of dataset. In the undertaken dataset less than 2.4% of the aggregate images are provided for severe DR and a PDR class which relates that major alterations are required in the dataset for a classifier to learn minority features effectively.

Diabetic retinopathy is categorized into two types that are NPDR (non-proliferative diabetic retinopathy) and PDR (proliferative diabetic retinopathy) where NPDR (non-proliferative diabetic retinopathy) can be subdivided into mild NPDR, moderate NPDR, severe NPDR. The stages of diabetic retinopathy are described below:

- **NPDR (Non-Proliferative Diabetic Retinopathy):** It occurs when retinal capillaries are damaged due to hyperglycemia and weakens the capillary walls and a small outpouching eventually causes the rupture. It can be further divided into the following categories:

- **Mild NPDR**: It occurs when small areas of retinal blood vessels swell. This swelling is known as microaneurysms which may further leak, increases the severity of the disease. There are approximately 40% of the people with diabetes have signs of mild NPDR [5].
- **Moderate NPDR**: With the progression of the disease, the distortion and swelling of the retinal blood vessel become acuter. This affects the blood transportation ability of the veins altering the appearance of the retina. This condition generally contributes to Diabetic Macular Edema (DME). 16% of the people who has moderate NPDR will show a tendency to develop PDR in about a time span of one year.

- **Severe NPDR**: In this case, severe forms of intraretinal microvascular abnormalities along with the cotton wool spots are found that blockade the blood supply to the retinal areas. These retinal areas signal the retina for creating new blood vessels.

➢ **PDR (Proliferative Diabetic Retinopathy):** At this stage, there is circulation problem depriving the retina of oxygen and hence in PDR small abnormal blood vessel starts to grow along the retinal wall. Like a film of a camera, the retina sits at the back of the eye and because of the abnormal blood vessel growth, a gel-like fluid is filled at the back of the eye which makes the vision blurry and in extreme cases complete blindness is also possible as the light rays cannot be received by the optical nerve. Fig. 1 shows the images of different stages of DR.

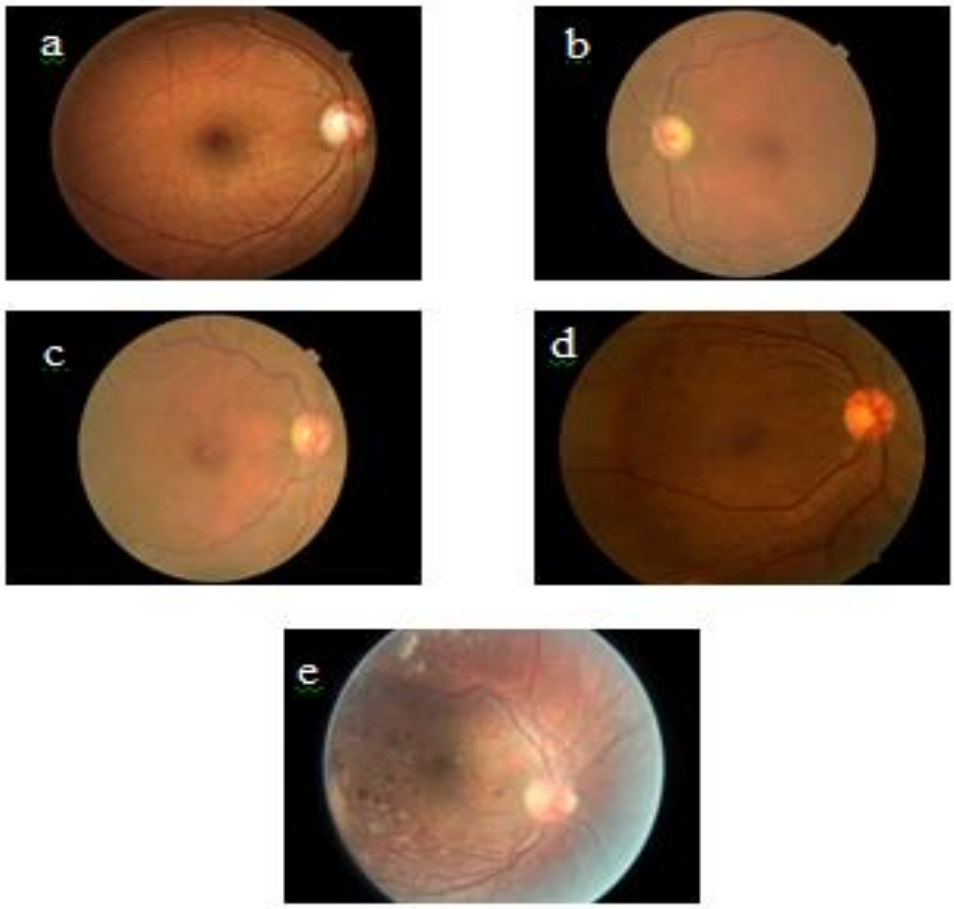


Fig.1.1. (a) Normal (b) Mild DR (c) Moderate DR (d) Severe DR (e) PDR

1.1 Contribution

The novel contribution of this research is first to compare the preprocessing techniques for enhancing imperative features required for further processing, then proposing distinctive architectures for profoundly effective and automated balancing of medical data. Finally, the classification of normal and abnormal retinal images is done by using the well trained convolutional neural network. This approach is assessed on a huge dataset of colored fundus images and will spare the manual work and time of the ophthalmologist.

1.2 Thesis Outline

This thesis includes seven chapters. The Chapter 1 introduces the subject area of the thesis research and provides an overview of the motivation and contribution in this work. The Chapter 2 covers a detailed literature review of the different methodologies followed till date to identify the diabetic retinopathy. The algorithms to classify the normal and abnormal retinal images are also reviewed in detail. Chapter 3 defines the importance of using deep learning over machine learning. It gives the different parameters and layers of CNN are also explained in detail. Chapter 4 explains the problem statement and the objectives of the proposed approach. Chapter 5 gives an insight into the methodology followed in this research work. It includes data balancing, Preprocessing, and model building. The Chapter 6 provides and discusses the experimentation results. The Chapter 7 concludes the work done in this research and provides the recommendations for the future work.

Chapter Summary

This chapter gives an overview of the thesis. It explains what the problem is and the need to find that problem. It explains the term Diabetic Retinopathy, how it caused, the real statistics of the effected people by this disease and the different stages of DR. Also, it discusses the motivation behind this thesis and the contribution to make such a system which can detect the DR and reduce the time of primary screening.

CHAPTER 2
LITERATURE SURVEY

A huge amount of work has been done in the field of detecting Diabetic Retinopathy. There are multiple ways of identifying DR like detecting microaneurysms, hemorrhages, exudates etc. Variation in size and shape of blood vessels is also a good indicator for DR detection. Similarly presence of different lesions aids in DR detection.

2.1 Unsupervised Methods

A fuzzy-based segmentation was proposed by Kande et al [14] that used green and red channels of an image to correct the non-uniform illumination from fundus image. The matched filter was applied to enhance the difference between background and blood vessels. Finally, fuzzy C Mean clustering and Connected labeling of components are performed to identify vascular tree structure. The author used publically available DRIVE and STARE datasets. Yao and Chen [15] used 2-dimensional Gaussian filters to perform DR classification. Experimentation was performed on STARE datasets. Another approach that used matched filter along with ant colony algorithm was proposed by Cinsidicsi and Aydin [16] that made use of DRIVE dataset to perform their experimentation and achieved AUC of 0.94 with an accuracy of 0.92.

A method for high-speed detection of blood vessels known as phase congruency was proposed by Amin and Yan [17]. A unique concatenation of centerlines detection of blood vessels and morphological bit plane was proposed by Fraz et al. [18]. Multiple morphological approaches were performed for purpose of pre-processing which are reviewed by Krizhevsky et al. [19]. Odstcilik et al [20] deduced a method for segmentation of retinal vessels that help in glaucoma, arteriolar narrowing and act as a preprocessing step for various image analysis. Segmentation of retinal vessels helps in detection of diabetic retinopathy by visualizing features such as hemorrhages, neovascular nets, microaneurysms etc. A matched filter coupled with minimum error threshold is used in the precise segmentation of retinal blood vessels. A new publically available dataset HRF is created which contains high-resolution fundus images. Author compared

performance of their approach on a new dataset with publically available DRIVE and STARE datasets.

2.2 Supervised Methods

Supervised learning algorithms use datasets annotated manually by ophthalmologists. These annotations are known as ground truth which is used to train the classifier. Different Supervised machine learning classifiers such as Artificial Neural Networks(ANN), K-Nearest Neighbor (KNN) [21], Gaussian Model [22], Support Vector Machine (SVM)[23], PCA coupled with Gabor Filters, Adaboost classifier [24].

Gardner et al. [25] utilized Neural Network and the intensity value of pixels to classify a fundus image as DR affected or not and achieved a sensitivity of 88.4%. Dataset comprising of 200 retinal images was used in which each retinal image was split into patches. Each patch was evaluated by a clinician for feature detection before implementation of SVM.

Another approach by Nayak et al. [26] used a neural network for classifying fundus images into three classes Normal, NPDR and PDR. Feature extraction was applied to extract features like exudates area, a region of blood vessels along with other parameters. The neural network was implemented on the dataset of 140 images to classify images into any of the three classes. Classified images were then validated by an ophthalmologist. The approach exhibited 90% sensitivity and 93% accuracy. Both training and testing phases required feature detection that made the detection process tedious.

Thomas and Mahesh [27] detected the retinal changes in diabetes patient's eye such as microaneurysms, hard exudates, soft exudates, hemorrhage etc. The authors intended to monitor the changes in retinal images to detect the presence of DR. Gandhi and Dhanasekaran [28] detected exudates in color fundus as well as classified the severity of the lesions using SVM classifier. Shahin et al. [29] also did the same type of work as that of [28] but in their work, they used ANN classifier. Prenta et al. [30] performed

segmentation of retinal vasculature. They localized macula and fovea that aids in the segmentation of diabetic retinopathy.

An approach to automating the detection of DR in five distinct classes by Acharya et al. [31] used spectra method for feature extraction. These features were fed to SVM for classification of fundus image and achieved an accuracy of 82%. Acharya et al. [32] also utilized image preprocessing to extract the most crucial features namely hemorrhages, microaneurysms, blood vessels, and exudates. SVM was applied to these features and the accuracy was increased to 85.9%. Image preprocessing was also applied by Adarsh et al. [33] for automatic grading of fundus images. The feature vector was created by using texture features and classification was done by SVM using two datasets of 130 images and 89 images. This approach achieved an accuracy of 94.6% and 96% respectively for each dataset.

Sadek et al. [34] followed an innovative approach to classify the DR and ARMD in the colored retinal images. The images were preprocessed with conventional intensity normalization. The features were extracted from images using HOG and SURF techniques and the final classification was done by implementing SVM. The author validated the method on 5 publically available datasets with the mean accuracies of 97.22% and 99.77% for single and multiple based dictionaries approaches.

2.3 Deep Neural Networks

ANN are widely applied in medical image classification problems but Deep learning neural networks like Convolutional Neural Networks (CNN) are generating exceedingly impressive results with classification and automatic feature extraction. Deep neural networks [35] use features such as drop out that help them generate accurate results.

Rectified Linear Unit (ReLU) nonlinearity is applied for effective training deep convolutional networks as they don’t vanish at extremes tangent and sigmoid functions. CNN due to their multilayer architecture are easily parallelized using GPUs. CNN [36][37] are a type of feedforward networks which are trained using backpropagation and

performs automatic feature extraction. Automatic feature extraction produces both high level and low-level features. The architecture of CNN comprises of Convolutional and pooling layer along with fully connected layers. Convolutional layers act as main feature extraction layers that map various features with the input image. The feature extraction process uses a window that slides on entire image to search for unique features. For multiple features, the network generates multiple feature maps. Every feature map has similar weights and maps similar features at multiple positions. Weight sharing feature reduces the number of weights which are learned. Feature selection process helps in dimensions reduction and spatial resolution of features and is carried out by Pooling layers of the network.

A hybrid approach was proposed by Wang et al. [38] that used Random Forest and CNN models. The author performed a huge amount of preprocessing for improving quality of the image. Hierarchical features that are insensitive towards distortion, translation etc. are extracted from the image by using CNN. The author used STARE and DRIVE datasets for purpose of experimentation and achieved AUC and accuracy of 0.98/0.97 and 0.97/0.94 respectively for each dataset. Another deep learning based supervised approach was proposed by Liskowski et al. [39]. They applied multiple preprocessing methods on the image to increase the performance of deep learning classifier. Augmentation of images was done by performing operations like flipping, scaling etc. to increase the STARE and DRIVE dataset. To perform cross-validation CHASE dataset was used. A M x M patch is applied on every pixel to detect whether it is a vessel or not. Value of M was made 27 and for each channel of the image i.e. R, G, and B a patch was created resulting in size 3*27*27=2187. A sample of 20000 patches was created for STARE and DRIVE datasets. Two CNNs were used in experimentation in which one consisted of the max-pooling layer and 3 fully connected layers and other consisted of only 3 fully connected layers. End layers used sigmoid function whereas hidden layers used ReLU nonlinearity function. Multiple variants of their algorithms resulted in AUV of 0.99 achieving accuracy of 0.97.

Pratt et al. [40] recently approached the problem of detecting the diabetic retinopathy by applying deep learning over a relatively very large dataset of the retinal

images. Before feeding the images into CNN architecture the images were resized and preprocessed but the method achieved an accuracy of 75% on a dataset of 5000 validation images.

Most of the five class classification approaches utilize the tedious handcrafted feature extraction process and then apply SVM to detect the severity grade of the retinal images. This delays the automated process and affects its pertinence. These approaches are validated on the very small dataset and do not demonstrate the precise accuracy of the system. None of them concentrate on diminishing the false normal rate of the system. Thus they are less effective in real world.

Chapter Summary

In this chapter, a detailed literature survey is carried out on the ongoing research and new technologies which have been used by different authors in the previous years. The chapter is divided into three major areas which are; Unsupervised Methods, Supervised Methods and Deep Neural Networks. Each of these sections contains detailed description of the related papers. Also, the analysis has been done on the outcomes and findings of these papers to better understand the need of the time and the research gaps left.

CHAPTER 3
BACKGROUND

3.1 Deep Learning

Machine learning has a subfield known as deep learning which inspired by functioning and architecture of the brain. Deep learning gained interest due to the availability of high-speed computers and a huge amount of data to efficiently train a deep network.

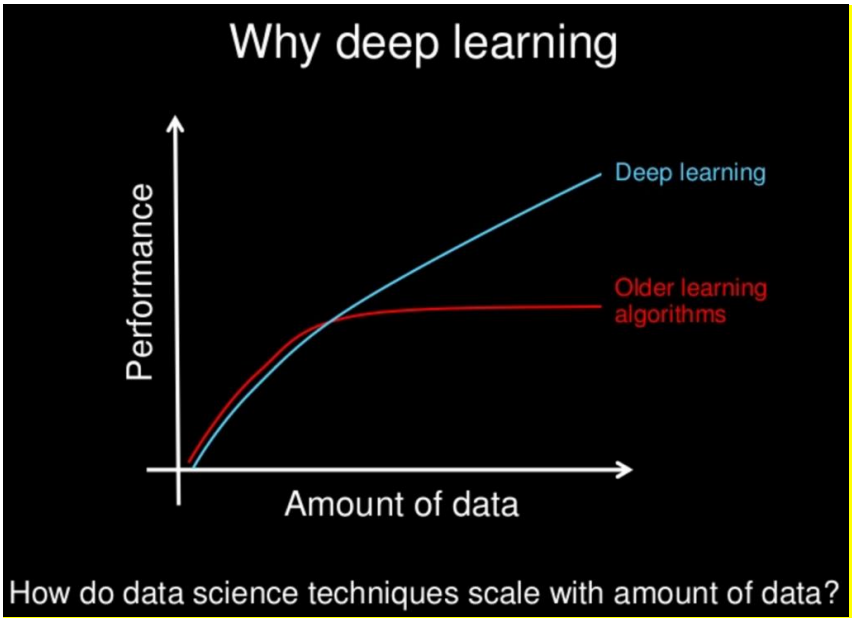


Fig.3.1.Performance of Older and deep learning algorithm w.r.t. Quantity of Data [41]

Deep learning network is highly scalable with an additional feature of automatic feature extraction that helps the network learn features from raw data. Deep learning models are capable of learning complex patterns which small networks are often unable to learn. Deep networks are a part of Artificial intelligence which emulates human-like perception to get certain kinds of knowledge. In simple term, deep learning is a process of automating predictive analysis as shown in Fig.3.1.

Deep learning networks have a stacked architecture with increasing abstraction and complexity hierarchy. To understand the working of these networks, imagine a toddler whose first word is a cat. He learns about a cat by pointing at object saying word cat.

Parents give their input by saying “Yes or No”. With time toddler gets aware of unique features that cat possesses. Toddler unintentionally learns complex patterns by building a layer of abstraction by gaining knowledge from the previous layer of hierarchy.

3.1.1 Working of deep learning

Every deep learning model goes through the similar process of applying non-linear transformations on input and outputs the model. This process continues until the model achieves high accuracy. Deep learning gets its name from the depth of network.

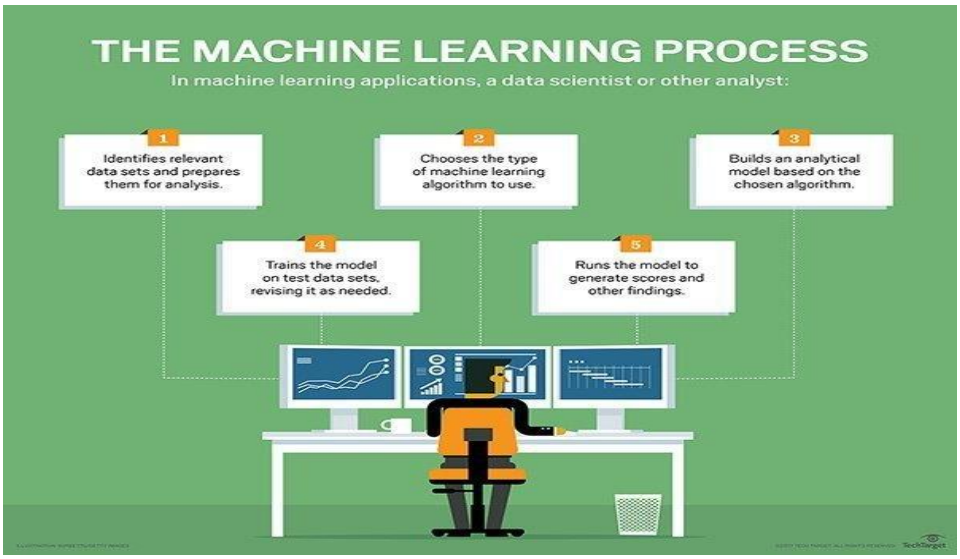


Fig.3.2.The machine learning process [42]

Machine learning algorithms use supervised learning in which classifier in told what to look for in an image during image classification as explained in Fig.3.2. These algorithms use manual feature extraction in which the accuracy of the classifier completely depends on the quality of feature set used. Here the deep learning algorithm doesn’t require any supervision for building feature set. This type of learning is more accurate and takes lesser time. A Classifier may get annotated training data with labels “cat” or “not a cat” for each image given by a human expert. The classifier uses the training data and creates a feature map. Initially, classifier might only be able to learn simple features like “Tail” and “four

legs". But with every iteration, the classifier starts to learn more complex features and becomes more accurate.

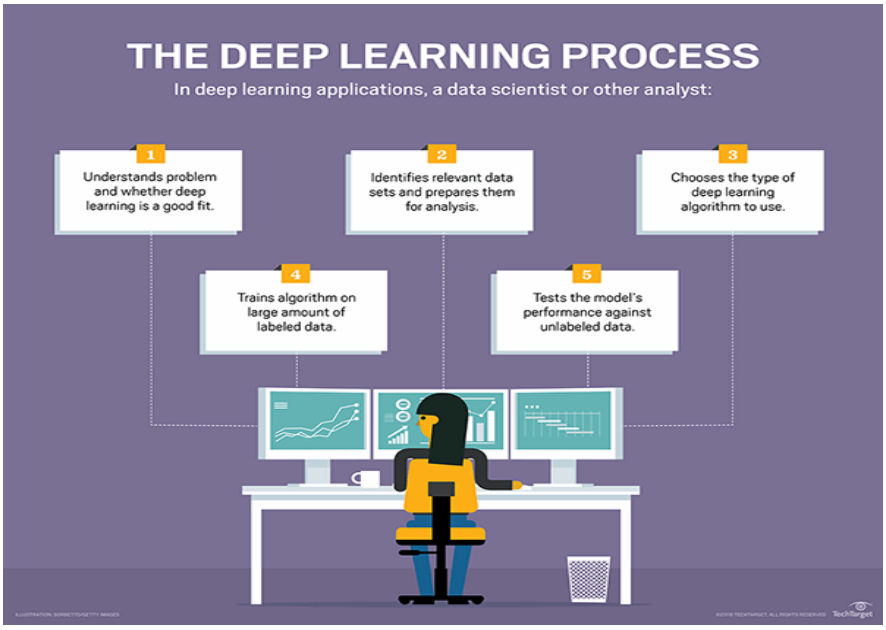


Fig.3.3.The Deep Learning Process [41]

Though a toddler takes several months to learn the concept of "Cat", deep learning networks with the use of high-performance computers when given large dataset may classify millions of images correctly in few minutes. The ability of deep learning networks to find meaningful insights from unstructured and unlabeled data increases its applications as most of the data gathered using IoT and other sources are unstructured as shown in Fig.3.3. Deep learning can be applied to a wide range of applications such as natural language processing, disease diagnosis, stock market prediction, image classification, intrusion detection as well as applications involving big data analytics.

3.2 Convolutional Neural Network (CNN)

Convolutional Neural Network (**CNNs**) is a category of Neural Networks which has proven very effective in areas such as image classification. CNN has been successfully applied in multiple fields like in identifying faces, self-driving cars. Convolutional Neural Networks (CNN) are everywhere. It is arguably the most popular deep learning

architecture. The recent surge of interest in deep learning is due to the immense popularity and effectiveness of CNNs. The interest in CNN started with AlexNet in 2012 and it has grown exponentially ever since. In just three years, researchers progressed from 8 layers AlexNet to 152 layer ResNet.

CNN is now the go-to model for every image related problem. It is also successfully applied to recommender systems, natural language processing and more. The main advantage of CNN compared to its predecessors is that it automatically detects the important features without any human supervision. For example, given many pictures of cats and dogs, it learns distinctive features for each class by itself. CNN is also computationally efficient. It uses special convolution and pooling operations and performs parameter sharing. This enables CNN models to run on any device, making them universally attractive. It is a very powerful and efficient model which performs automatic feature extraction to achieve superhuman accuracy. All CNN models follow a similar architecture, as shown in the Fig.3.4.

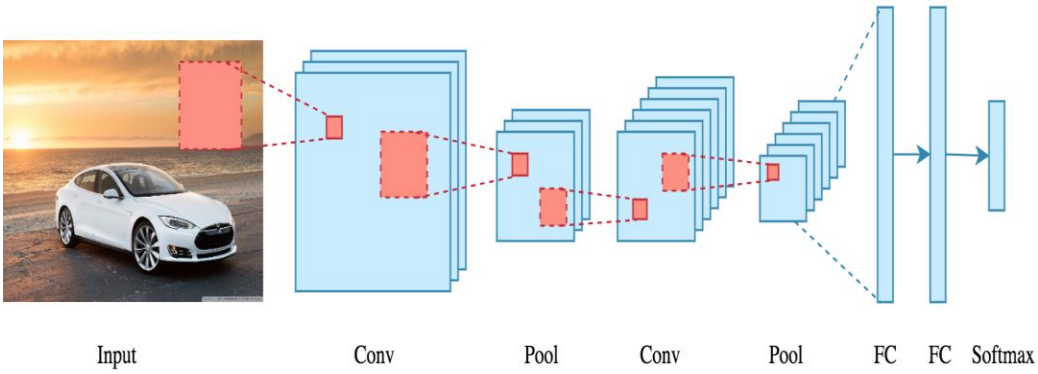


Fig.3.4. Architecture of CNN [43]

There is an input image on which CNN is applied. CNN perform a series convolution + pooling operations, followed by a number of fully connected layers. While performing multiclass classification the output activation function is chosen as softmax. The detailed description of each layer is explained below.

3.2.1 Convolution

The main building block of CNN is the convolutional layer. Convolution is a mathematical operation to merge two sets of information. In our case, the convolution is applied on the input data using a convolution filter to produce a feature map.

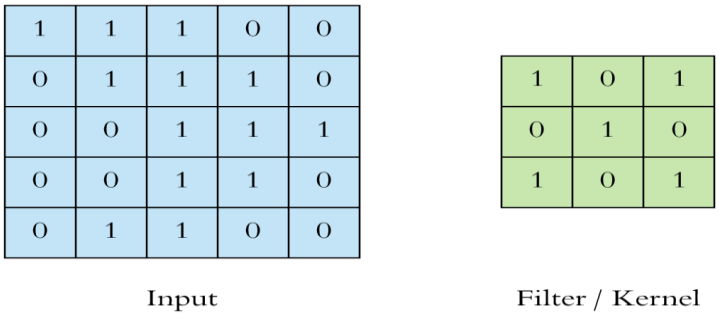


Fig.3.5.The input matrix and filter/kernel matrix [43]

In Fig. 3.5 the left side is the input to the convolution layer, for example, the input image. On the right is the convolution *filter*, also called the *kernel*, we will use these terms interchangeably. This is called a *3x3 convolution* due to the shape of the filter. We perform the convolution operation by sliding this filter over the input as shown in Fig. 3.6. At every location, we do element-wise matrix multiplication and sum the result. This sum goes into the feature map. The green area where the convolution operation takes place is called the *receptive field*. Due to the size of the filter the receptive field is also 3x3.

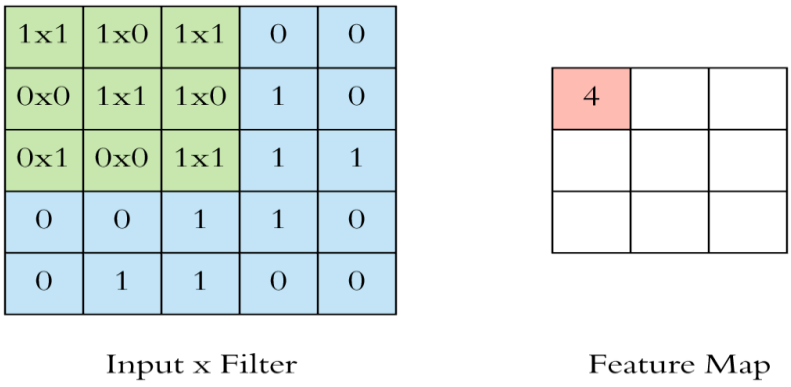


Fig.3.6.Working of convolutional layer [43]

Here, the filter is at the top left, the output of the convolution operation “4” is shown in the resulting feature map. We then slide the filter to the right and perform the same operation, adding that result to the feature map as well. We continue like this and aggregate the convolution results in the feature map. Here’s an animation that shows the entire convolution operation.

This was an example convolution operation shown in 2D using a 3x3 filter. But in reality, these convolutions are performed in 3D. In reality, an image is represented as a 3D matrix with dimensions of height, width, and depth, where depth corresponds to color channels (RGB). A convolution filter has a specific height and width, like 3x3 or 5x5, and by design, it covers the entire depth of its input so it needs to be 3D as well.

One more important point before we visualize the actual convolution operation. We perform multiple convolutions on an input, each using a different filter and resulting in a distinct feature map. We then stack all these feature maps together and that becomes the final output of the convolution layer. But first, let’s start simple and visualize a convolution using a single filter.

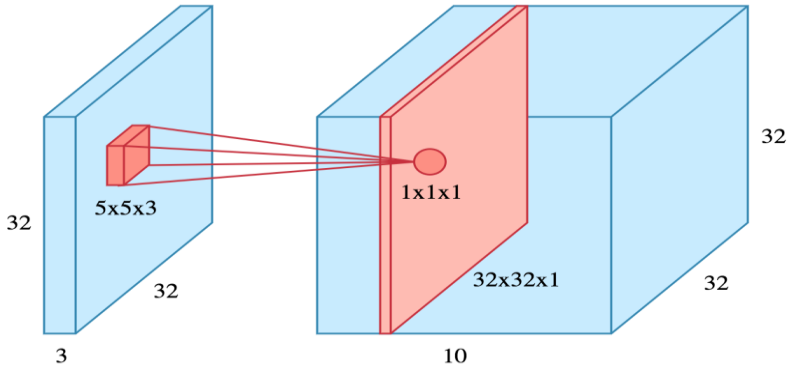


Fig.3.7.Convolutional operation using a single filter [43]

In Fig. 3.7 we have a 32x32x3 image and we use a filter of size 5x5x3 (note that the depth of the convolution filter matches the depth of the image, both being 3). When the filter is at a particular location it covers a small volume of the input, and we perform the convolution operation described above. The only difference is this time we do the sum of matrix

multiply in 3D instead of 2D, but the result is still a scalar. We slide the filter over the input like above and perform the convolution at every location aggregating the result in a feature map. This feature map is of size 32x32x1, shown as the red slice on the right.

If we used 10 different filters we would have 10 feature maps of size 32x32x1 and stacking them along the depth dimension would give us the final output of the convolution layer: a volume of size 32x32x10, shown as the large blue box on the right. Note that the height and width of the feature map are unchanged and still 32, it's due to padding and we will elaborate on that shortly.

To help with visualization, we slide the filter over the input as follows. At each location, we get a scalar and we collect them in the feature map. The animation shows the sliding operation at 4 locations, but in reality, it's performed over the entire input. In Fig. 3.8 we can see how two feature maps are stacked along the depth dimension. The convolution operation for each filter is performed independently and the resulting feature maps are disjoint.

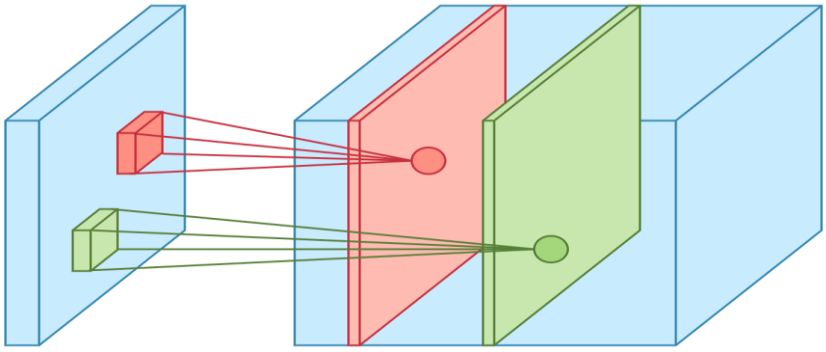


Fig.3.8.Convolutional operation using a double filter [43]

3.2.2 Non-linearity

For any kind of neural network to be powerful, it needs to contain non-linearity. Both the ANN and autoencoder we saw before achieved this by passing the weighted sum of its inputs through an activation function, and CNN is no different. We again pass the result of the convolution operation through *relu* activation function. So the values in the final feature

maps are not actually the sums, but the relu function applied to them. We have omitted this in the figures above for simplicity. But keep in mind that any type of convolution involves a relu operation, without that the network won't achieve its true potential.

3.2.3 Stride and Padding

Stride specifies how much we move the convolution filter at each step. By default, the value is 1, as you can see in the Fig. 3.9.

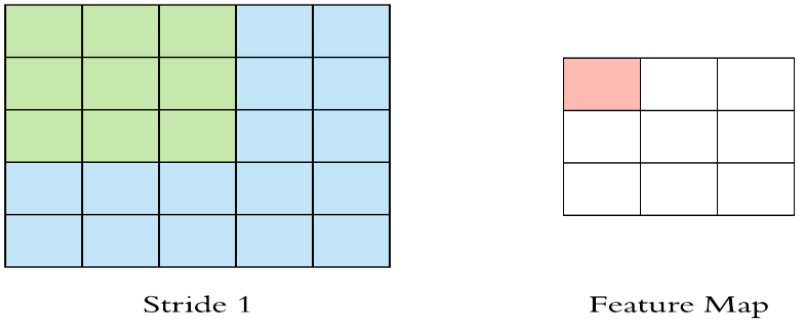


Fig.3.9.Sliding of the convolutional filter using 1*1 stride [43]

We can have bigger strides if we want less overlap between the receptive fields. This also makes the resulting feature map smaller since we are skipping over potential locations. Fig. 3.9 demonstrates a stride of 2. Note that the feature map got smaller. We see that the size of the feature map is smaller than the input because the convolution filter needs to be contained in the input. If we want to maintain the same dimensionality, we can use *padding* to surround the input with zeros.

We can have bigger strides if we want less overlap between the receptive fields. This also makes the resulting feature map smaller since we are skipping over potential locations. Fig. 3.9 demonstrates a stride of 2. Note that the feature map got smaller. We see that the size of the feature map is smaller than the input because the convolution filter needs to be contained in the input. If we want to maintain the same dimensionality, we can use *padding* to surround the input with zeros.

3.2.4 Pooling

After a convolution operation, we usually perform *pooling* to reduce the dimensionality. This enables us to reduce the number of parameters, which both shortens the training time and combats overfitting. Pooling layers downsample each feature map independently, reducing the height and width, keeping the depth intact. The most common type of pooling is *max pooling* which just takes the max value in the pooling window. Contrary to the convolution operation, pooling has no parameters. It slides a window over its input, and simply takes the max value in the window. Similar to a convolution, we specify the window size and stride. Fig. 3.10 shows the result of max pooling using a 2x2 window and stride 2. Each color denotes a different window. Since both the window size and stride are 2, the windows are not overlapping.

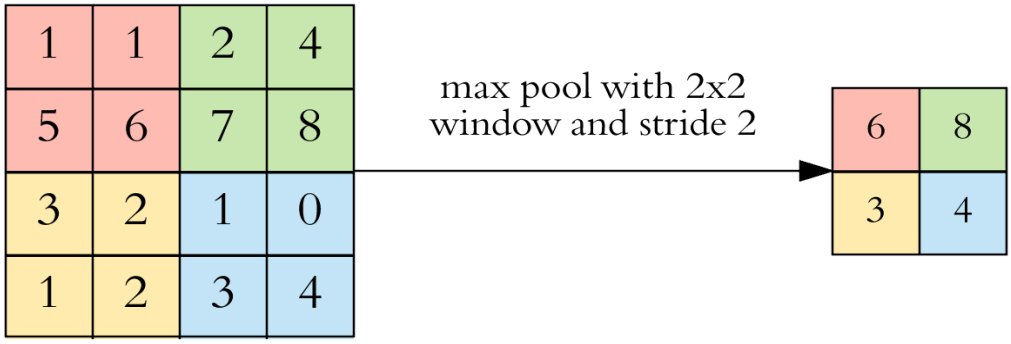


Fig.3.10.Working of MaxPooling [43]

Note that this window and stride configuration halves the size of the feature map. This is the main use case of pooling, downsampling the feature map while keeping the important information. Fig. 3.11 shows the feature map dimensions before and after pooling. If the input to the pooling layer has the dimensionality 32x32x10, using the same pooling parameters described above, the result will be a 16x16x10 feature map. Both the height and width of the feature map are halved, but the depth doesn't change because pooling works independently on each depth slice the input.

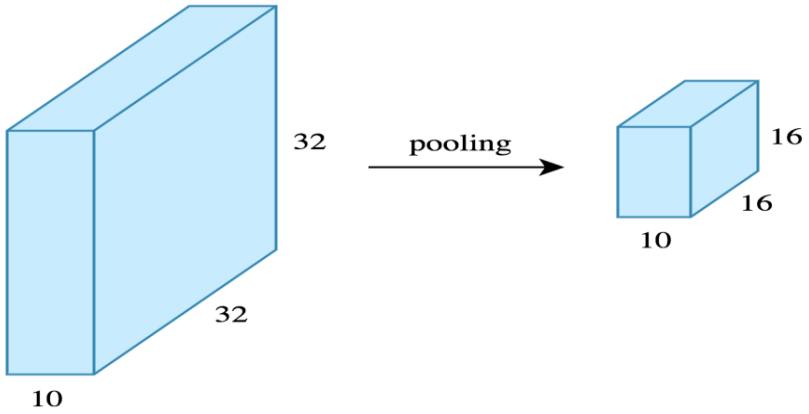


Fig 3.11.Dimension reduction using polling [43]

By having the height and the width, we reduced the number of weights to 1/4 of the input. Considering that we typically deal with millions of weights in CNN architectures, this reduction is a pretty big deal. In CNN architectures, pooling is typically performed with 2x2 windows, stride 2 and no padding. While convolution is done with 3x3 windows, stride 1 and with padding.

3.2.5 Hyper-parameters

We have 4 important hyper-parameters to decide on:

- Filter size: we typically use 3x3 filters, but 5x5 or 7x7 are also used depending on the application. There are also 1x1 filters which we will explore in another article, at first sight, it might look strange but they have interesting applications. Remember that these filters are 3D and have a depth dimension as well, but since the depth of a filter at a given layer is equal to the depth of its input, we omit that.
- Filter count: this is the most variable parameter, it's a power of two anywhere between 32 and 1024. Using more filters results in a more powerful model, but we risk overfitting due to increased parameter count. Usually, we start with a small number of filters at the initial layers and progressively increase the count as we go deeper into the network.
- Stride: we keep it at the default value 1.

- Padding: we usually use padding.

3.2.6 Fully Connected

After the convolution + pooling layers we add a couple of fully connected layers to wrap up the CNN architecture. The output of both convolution and pooling layers are 3D volumes, but a fully connected layer expects a 1D vector of numbers. So we *flatten* the output of the final pooling layer to a vector and that becomes the input to the fully connected layer. Flattening is simply arranging the 3D volume of numbers into a 1D vector, nothing fancy happens here.

Chapter Summary

This chapter is mainly divided into two subsections namely Deep Learning and Convolutional Neural Network. In the first subsection, the detailed working of deep learning algorithms is explained. Here, the comparison is drawn between the traditional machine learning algorithm and deep learning algorithm. In the second subsection, the architecture of CNN is explained in detail along with different parameters.

CHAPTER 4
PROBLEM STATEMENT

From literature surveys, it was observed that most of the researchers used techniques like SVM, AdaBoost and Random forest to perform DR detection. These techniques require manual feature extraction so there was a scope of identifying more relevant features to improve the performance of classifiers. This can be achieved by applying better feature selection methods to identify more significant features in a fundus image. Deep learning algorithms can improve the segmentation of blood vessels and Identification of multiple exudates and lesions using automatic feature extraction. Deep learning networks have replaced the feature selection and extraction but they require efficient pre-processing of fundus images to remove the variations due to different lighting conditions and enhance the quality of the image. Most of the medical data related databases are highly unbalanced and are unsuitable for deep learning application. Thus there is a need to create data balancing algorithm that is specially designed for medical data applications.

4.1 Objectives

The objectives of the current research work can be listed as:

1. To study, analyze and explore the already existing detection methods for diabetic retinopathy.
2. To propose an algorithm that can balance medical data efficiently to make it suitable for deep learning applications.
3. To propose an automated detection system to identify the lesions in the retina of a diabetic eye by employing deep learning.
4. To test and validate the proposed technique on publically available datasets of the fundus.

CHAPTER 5

PROPOSED SYSTEM FOR RETINOPATHY DETECTION

The proposed framework is composed of three noteworthy modules that associate in the characterization of the retinal image. The image augmentation module aids in preprocessing [44] of the image preparing it for model implementation. Next module helps in data transformation and balancing to reduce the disproportion among different classes of the dataset. The exceedingly skewed dataset is approached by applying two distinct architectures for data balancing. In first architecture, the entire data is divided into multiple sub-datasets followed by the application of Deep Learning model.

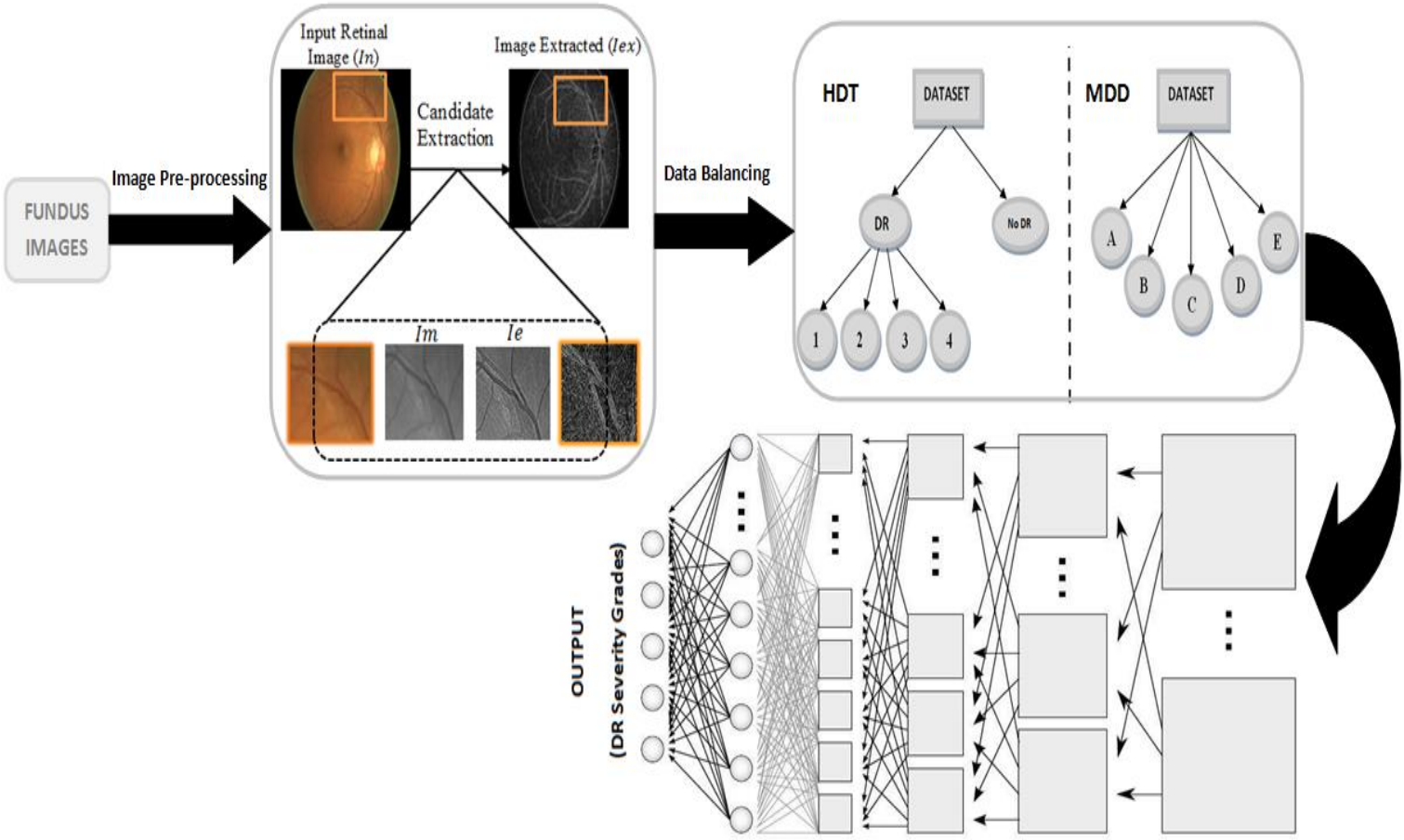


Fig.5.1.The architecture of the Proposed approach

To obtain the last outcome, results from each trained model are bagged together and ensembled. The second approach deals with multiclass classification problem in a

different way, by first converting it into a binary classification problem of characterizing image as affected by DR or not. Further another classifier is trained to characterize the DR affected image into 4 distinctive severity grades. Each of the data transformation approach uses a custom 21 layer deep CNN trained on the large annotated dataset. Fig.5.1 shows the multistage algorithm to automate the detection of severity grade of DR.

5.1 Image Augmentation

In stage 1 diverse transformation are made on the image to enhance the highlights and stifle the variation due to various illuminations in the fundus image. Image (In) provided in the dataset has vaa ried field of view, illumination and size.

Initial preprocessing is done to resize the images into a standard 128 x 128 pixels RGB image (Ir). To reduce the further complexities and the variation due to different illumination conditions, it is converted into monoa chrome image (Im)shown in Fig.5.2. The conversion is done using ITU-R 601-2 luma transformation shown in (1).

$$Im = Ir * \frac{299}{1000} + Ig * \frac{587}{1000} + Ib * \frac{114}{1000} \quad (1)$$

where:

Ir representsred spectrum of theimage.

Ig representsgreen spectrum of theimage.

Ib representsblue spectrum of theimage.

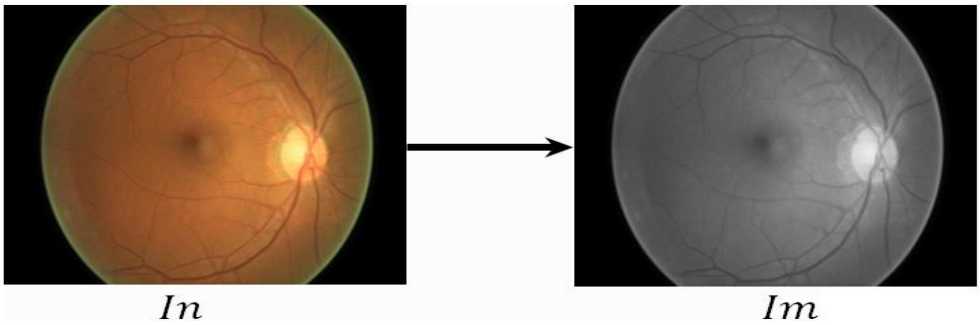


Fig.5.2. Conversion of a normal retinal image $In$to monochrome image Im

In stage 2 *Im* needs to be further transformed to enhance the structure of exudates and veins in rethe tina. For this, edge enhancement is applied that increases the contrast of pixels around specific edges so that they are visible conspicuously. This is an image processing channel that improves the edge intricacy of an image or video endeavoring to enhance its acutance (evident sharpness). The channel works by recognizing sharp edge limits in the image, for example, the edge between a subject and a background of a differentiating color, and expanding the image differentiate in the region instantly around the edge. This has the impact of making unobtrusive, astonishing and dim features on either side of any edge in the image, called overshoot and undershoots, driving the edge to look more characterized when seen from a standard site. For this a 3*3 kernelis used which is shown below:

$$\begin{vmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{vmatrix}$$

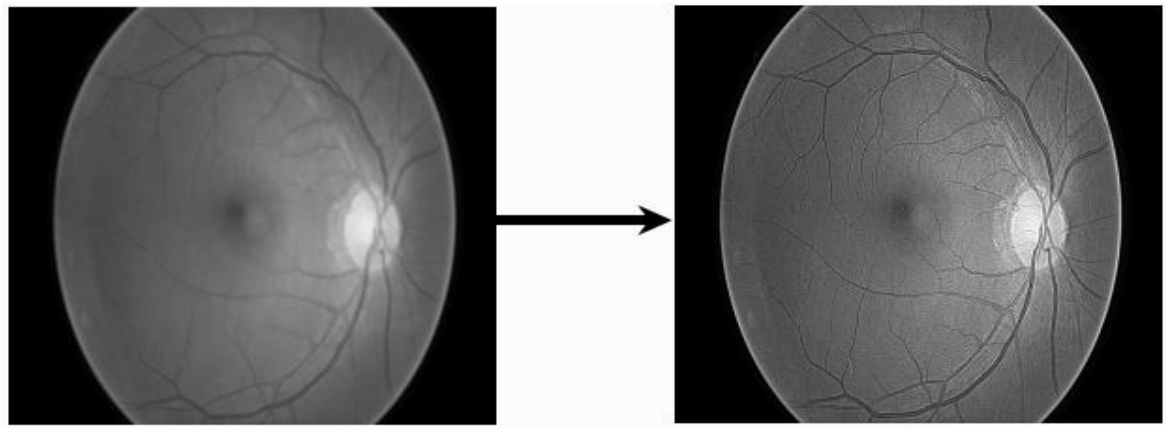


Fig.5.3.Conversion of a monochrome retinal image *Im*to Edge enhanced image *Ie*

This kernel is convolutional to enhance the edge of the image.*Im*gets transformed into *Ie*as shown in Fig.5.3. The conversion helps in enhancing the structural edges of the image. The kernel slides on the whole image multiplying each pixel by the corresponding

value of the matrix and amplify the higher contrast regions of the image from the rest of the image. This transformation is shown in Fig.5.4.

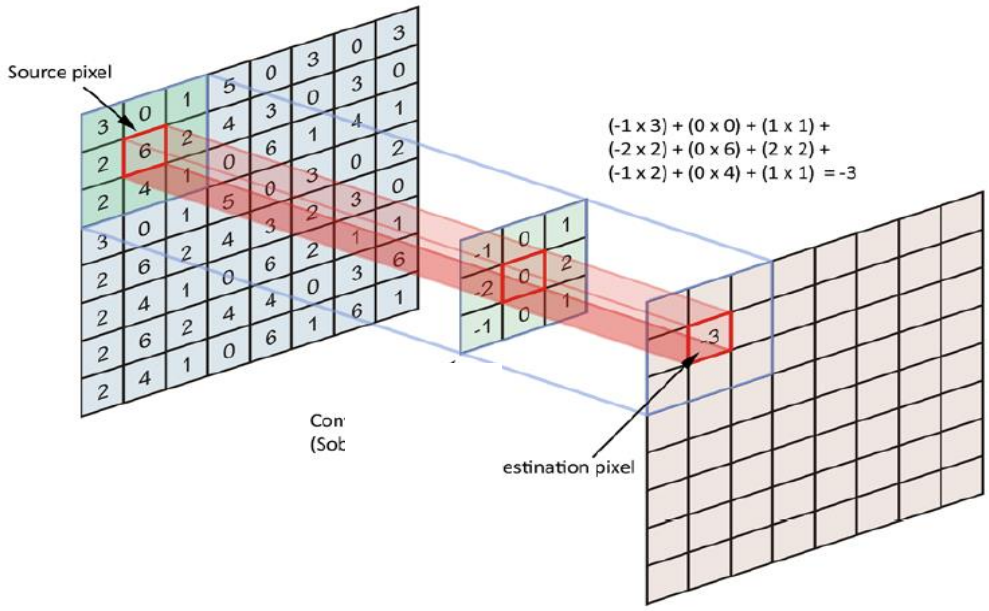


Fig.5.4. Working of convolution kernel on image

After sharpening the edges of auxiliary highlights the next transformation in stage 3 is to extract only the structural details of the image containing all the features creating imageIexshown in Fig.5.5.

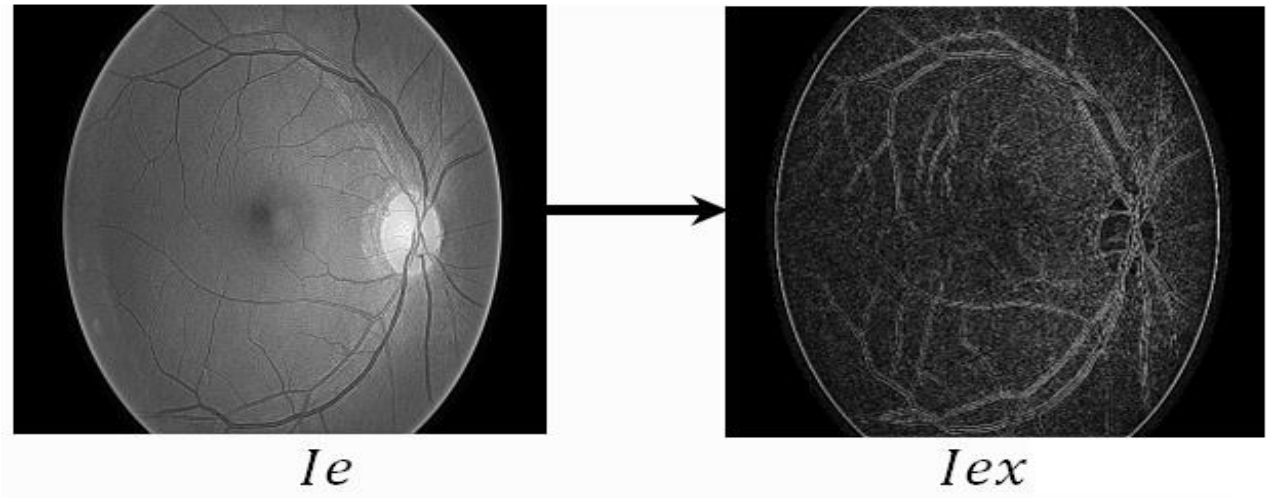


Fig.5.5. Conversion of an Edge enhanced retinal image $Ie$to Edge extracted image Iex

Edge detection incorporates an assortment of numerical techniques that aim at identifying the points in a digital image at which the image luster changes strongly or, more formally, has discontinuities. The concentrations at which picture brightness changes forcefully are regularly sorted out into an arrangement of bent line fragments named edges.

A comparable issue of finding discontinuities in one-dimensional signs is known as step revelation and the issue of finding signal discontinuities after some time is known as change distinguishing proof and this is finished utilizing 3*3 kernel which is demonstrated as follows: The working of the following kernel is similar to that shown in Fig.5.4.

$$\begin{vmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{vmatrix}$$

5.2 Model Selection

Post preprocessing deep learning classifier has been applied for the evaluation of fundus image. The Convolutional Neural Network (CNN) [45, 46, 47] was designed specifically for the purpose of analyzing visual imagery. It is a class of feed-forward deep artificial neural networks which performs the task of image classification more accurately and efficiently as compared to ANN.

CNN uses kernel or filter for extracting features from an image. The filter is a small size vector of weights, which generates a feature map as output. This filter mask is made to move along different pixels of the input image gradually covering the entire image. As a result of the nature of the convolution operation, the network units belonging to a single layer are forced to share weights among them. This aspect of CNN gives the network the ability to detect features irrespective of its location in the image. The sharing of weights among neurons in a layer helps to significantly reduce the number of parameters in the network.

Table 5.1.Convolutional Neural Network (CNN) architecture used in our experiment

#	NAME	SIZE
0	Input	1x128x128
1	Conv1	32x126x126
2	Pool1	32x63x63
3	Dropout1	32x63x63
4	Conv2	64x62x62
5	Pool2	64x31x31
6	Dropout2	64x31x31
7	Conv3	128x30x30
8	Pool3	128x15x15
9	Dropout3	128x15x15
10	Conv4	256x14x14
11	Pool4	256x7x7
12	Dropout4	256x7x7
13	Conv5	512x6x6
14	Pool5	512x3x3
15	Dropout5	512x3x3
16	Hidden6	500
17	Dropout6	500
18	Hidden7	1000
19	Hidden8	1000
20	Hidden9	1000
21	Output	5

Image of size 128x128 pixel is used for feeding the network. This resizing was done by center cropping the input images to get the desired pixel sizes for the experiments. The retinal images are processed through a pile of convolutional layers where a small 3x3filter size was used and the custom network comprised of 21 layers. A fixed convolutional stride of 1 pixel with a padding of 1 pixel was utilized for the convolutional layer. For

dimension reduction, a 2x2 pixel window is used for applying Max-pooling. Addition of multiple dropout layers decreases the overfitting of the system and aid in improving the accuracy of the system. Table 5.1 shows the layer information of the custom model.

5.3 Dataset Transformation

The EyePACS dataset contains *35,126 images* for training the network and *53,576 images* for testing the network. Table 5.2 shows the number of images in the dataset. There are five fairly unbalanced classes of images which are taken in various lighting condition. These five training labels [0,1,2,3,4] are named as normal, mild DR, moderate DR, severe DR and PDR respectively having a unique patient-id with a substring "left" for the left eye and "right" for the right eye.

Table5.2 Dataset Distribution

Class	Name	Training Images	Testing Images
0	Normal	25,810	39,533
1	Mild DR	2,443	3,762
2	Moderate DR	5,292	7,861
3	Severe DR	873	1,214
4	PDR	708	1,206

The initial approach towards addressing the problem of Diabetic Retinopathy through automatic detection exploits deep learning algorithms which when applied to the dataset through rigorous training are able to predict the categorization of the patient's fundus image within the given severity grades. The deep neural network is an exceptionally viable and productive model that is widely used for the image classification. This CNN is a supervised machine learning model that takes in input well-annotated data as a training set and learns the pattern among the classes generally the more the number of layers in the CNN more the embedded and complex features it can learn. For our motivation of

building a mechanized framework, 21 layered CNN is used which is highly trained over a large number of labeled images. Most of the approaches in the past have used a small dataset of about 100-5000 images.

This approach creates a more robust and better-trained network by using around 35000 images. This image dataset is highly skewed towards class representing non-diabetic patients. The underlying methodology was to implement this 21 layer CNN over the entire dataset. This approach when actualized restored a one-sided classifier which when applied to any fundus image categorize it to the normal image. Despite the fact that the model accuracy was about 73%, this model was very inadmissible for any application for clinical purposes. The principal issue towards tending to any problem that deals with the categorization of medical data require utmost priority for diminishing the misclassification of a disease affected patient to normal patient. For this, deep learning therapeutic classification models ought to be brilliantly trained for stifling such anomalies. The lesser this misclassification happens the more practical is the application of the model. Thus, for further evaluation of any approach, the false normal rate is an imperative parameter that makes a model constant material.

5.3.1 Existing Technology Data Balancing

There are many approaches proposed to solve the problem of the unbalanced learning. Each of this approach tries to balance the dataset by using either undersampling or oversampling. To have a deep understanding, these approaches can be generalized as follows:

Random Sampling. This data balancing technique uses non-heuristic functions to balance data which can be further classified into two categories namely ROS and RUS.

Random oversampling. Random oversampling is a method that tries to balance disproportion among the classes by using a non-heuristic function to randomly replicate instances of a minority class. The main drawback of random oversampling is that it

replicates instances of minority class without any alteration which increases the chances of occurrence of overfitting.

Random undersampling. Random undersampling is a method that tries to balance disproportion among the classes by using a non-heuristic function to randomly eliminate instances of majority class. The main shortcoming of random undersampling is that it affects the induction process by potentially discarding valuable data.

Tomek links. It is an approach used for Undersampling in which Tomek links[15] belonging to majority class are eliminated. let Ia, Ib are instances that belong to two different classes, then (Ia, Ib) is a Tomek Link if there doesn't exist any instance Ic for which D(Ia, Ic) < D(Ia, Ib) or D(Ib, Ic) < D(Ia, Ib) where D(x, y) represents distance between instances x and y. The disadvantage of this approach is that if there exists a majority class that represents more than 70 percent of data than Tomeks links are not an effective solution as they will not able to undersample the majority class to an extent where classes become balanced.

Neighborhood Cleaning Rule. It is an Under sampling technique used to remove majority class instances by using Edited Nearest Neighbor Rule (ENN) [16] proposed by Wilson. The principle behind the working of ENN is that it removes an instance Ia from the dataset if two of its three nearest neighbor instances differ from its class label. Neighborhood Cleaning Rule modifies ENN which can be explained as follows:

Given a binary classification problem where M and N represent majority class and minority class respectively. For each instance Ia, three nearest neighbors are identified. If Ia Є M is an instance of majority class and any of its three nearest neighbors belong to class N then Ia is deleted from the dataset. If Ia Є N and any of its three nearest neighbors belong to class M than those neighbors of Ia are deleted from the dataset.

Synthetic Instance Creation. Synthetic data creation is the production of new instances by using the already given instances. The three widely used approaches under this are as follows.

SMOTE. Synthetic Minority Oversampling Technique is one of the most widely used oversampling methods. In this approach synthetic instances of a minority, the class is created to increase oversample the minority class. Instance creation in SMOTE uses a simple approach of taking k nearest instances of a minority class and interpolating them to create a new minority class instance. This approach helps spread boundaries of a minority class and also avoids overfitting.

Borderline SMOTE. Borderline SMOTE [17] method is also a type of Over-sampling method based on SMOTE. The main idea behind Borderline SMOTE is to create instances of using the borderline instances of a minority class. This approach achieves better true positive rates than SMOTE.

ADASYN. Adaptive synthetic sampling [18] approach is an Over-sampling Techniques which uses weights to increase the learning from unbalanced datasets. The main method behind ADASYN is to assign more weightage to the instances of minority class that are difficult to learn by the classifier than those instances of minority class that was learned more easily. So, more synthetic instances are created from data points that were harder to be classified by the classifier.

All the data balancing techniques discussed above have drawbacks that prevent them to be used in balancing Medical Data. Undersampling techniques don't take into account the cost of collecting medical data and try to eliminate valuable medical data instances. Oversampling techniques create synthetic minority data which otherwise should include sample data points having significant information about disease affected patients. So oversampling, the minority data affects the authenticity of data as well as the classifier.

5.3.2 Multiple Data Division (MDD)

The medical datasets are generally skewed towards the class representing the people not affected by the disease. Certain data balancing approaches are required to apply deep learning to these datasets. For this reason, there are numerous methodologies accessible that expect to balance the unbalance dataset by either using oversampling approach to

increase the instances of the minority class or undersampling approach which tries to remove some instances from the majority class. Both these methodologies are not appropriate for the medical data balancing as the oversampling approach challenges the credibility of the data created while the under-sampling approach neglects the endeavors and the cost required for gathering the dataset. Due to this fact, there is a need to handle the medical data balancing in a unique way to preserve both authenticity and value of the data. For this, the MDD algorithm is developed which creates multiple sub-datasets from one parent dataset where each of these sub-datasets contains an acceptable proportion of instances.

These sub-datasets can be further trained concurrently using parallel processing approaches to reduce the training time of the system. The output from each of these sub-datasets can be bagged together using voting techniques to create a much steady and unprejudiced classifier that can classify the input fundus image into its particular severity grade effectively. Fig.5.6 shows the block diagram of MDD technique.

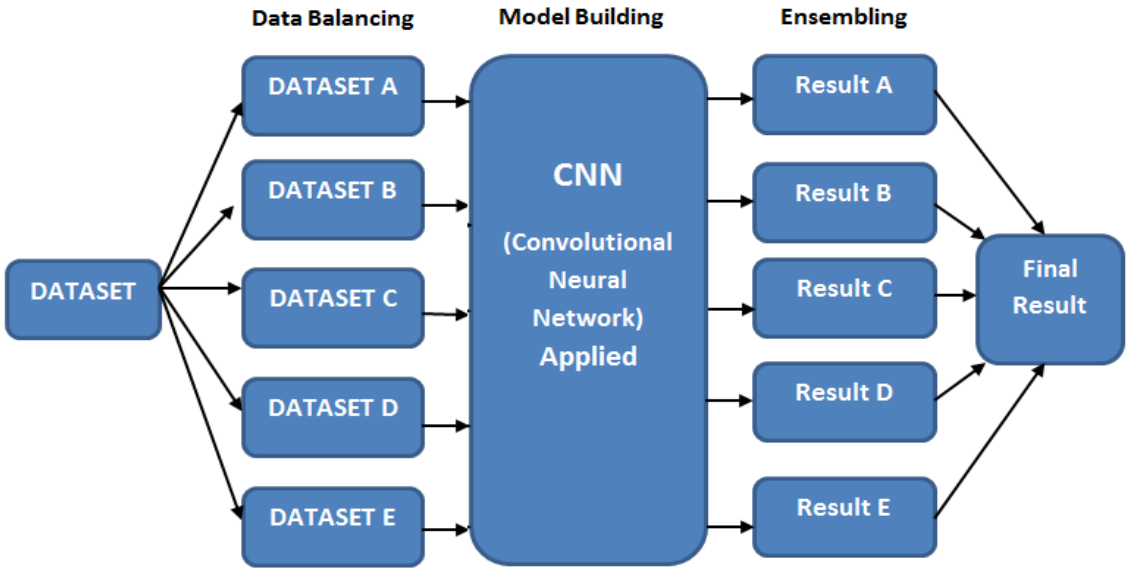


Fig.5.6.The architecture of Multiple data division

Let a dataset (M) contains three different classes i.e. $C1, C2, C3$ where, $C1 > C2 > C3$ in order of their number of instances available in each class. The majority class C1 and C2 are divided into "n" multiple datasets as given in (2), (3), (4) and (5).

$$C1i \; \forall \; i \in \{1, 2, 3 \ldots, n\} \quad (2)$$

$$C2i \; \forall \; i \in \{1, 2, 3 \ldots, n\} \quad (3)$$

where,

$$C1i \; \subseteq \; C1 \quad (4)$$

$$C2i \; \subseteq \; C2 \quad (5)$$

Each of these subsets is in acceptable proportion to$C3$. So, if only two subsets of each subclass is created than two datasets will be created having coa nfiguration as in (6) and (7).

$$M1 \; (C11, C21, C3) \quad (6)$$

$$M2 \; (C12, C22, C3) \quad (7)$$

Therefore, $M_1$and M_2 are balanced datasets and can afterward in the application of deep neural network for further feature extraction, detection, and evaluation. The data division is done using Algorithm1where, Maj (), Min() functions return the majority class and minority class of the given dataset. Count ( ) function is used for counting the number of instances in the given class. Round_Of ( ) function returns the nearest integer value of Z. Divide (a,b) divides every class in equal proportion to b and ↘is the threshold value for maximum disproportion.

Algorithm 1: Multiple Data Division

.***Algorithm*** $MDD \; (S, D)$

1. $S \leftarrow Maj \; (D)$
2. $D \leftarrow D - S$
3. $X \leftarrow Maj \; (D)$
4. $Y \leftarrow Maj \; (S)$
5. $Z \leftarrow Min \; (D)$

6. $\boldsymbol{If}\ \frac{Count(Y)}{Count(Z)} \geq \lambda\ \boldsymbol{then}$
7. $P = Round\_Of\ (\frac{Count(Y)}{Count(X)})$
8. $Sub\_Data \leftarrow Divide(S, P)$
9. $\boldsymbol{foreach}\ class\ A\ \in\ Sub\_Data\ \boldsymbol{do}$
10. $MDD(A, Di)$
11. $\boldsymbol{end}$
12. $\boldsymbol{else}$
13. $\boldsymbol{return}\ D$
14. $\boldsymbol{end}$

The parallel architecture defined in Algorithm 1 uses ensemble voting for bagging the output from every sub-dataset. The value of λ has been set to 5 upon testing on the given dataset the embedding of this ensemble voting in our approach helps in suppressing the false normal rate of the system by taking in consideration the output from multiple classifiers. For instance, let $Ci\ \forall\ i \in (1, 2, 3, 4, 5)$ are models trained on dataset $Dj\ \forall\ i(1, 2, 3, 4, 5)$ respectively and input fundus image (Im) needs to be categorized into its particular severity grade. The result of each classifier on this image (Im) can be shown as in Table 5.3.

Table.5.3. Classification results of the image

Classifier	**Actual Class**	**Predicted Class**
C1	1	1
C2		0
C3		0
C4		1
C5		1

Due to the implementation of ensemble voting, priority is given to the most predicted class and the image is classified into "1" inspite of the fact that two models categorized it into class "0". The division of Majority class "Normal" into multiple sub-datasets trains the

models on the different set of images for “Normal” and a similar set of images for rest of the classes. Each model learns a different set of features for class “Normal” and makes categorization into this class highly tough. An image is not categorized into the class “Normal” until the majority classifiers are in favor of it. Rest of the classes are not divided so each model train and extract similar features for rest of the classes. This approach turns out to be highly efficient in diminishing the false normal rate of the system.

5.3.3 Hierarchical Data Transformation (HDT)

MDD is highly efficient in balancing of the medical dataset but suffers the shortcoming i.e. if there exists a high difference in the proportion of instances in the minority and the rest of the classes of the dataset then, very large number of datasets will be created. This affects the resource acquisition and increases the system requirements for applying parallel processing techniques.

For this purpose, another algorithm is proposed in which the multiclass classification problem is transformed in terms of binary categorization. This approach is not at par with MDD in terms of balancing but it uses a hierarchical construct to deal with the skewness of the dataset.

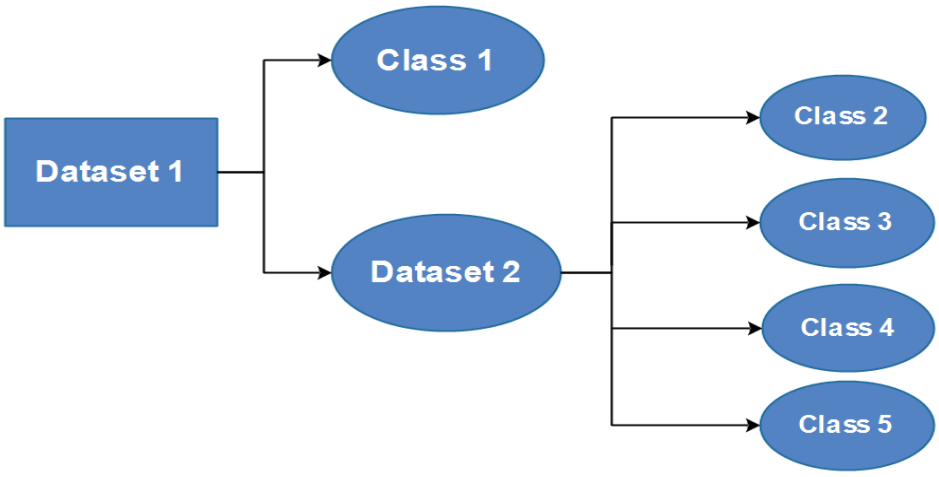


Fig.5.7. Hierarchical Approach to data transformation

In Fig.5.7 initially, a two-level construct is created. Each level of this construct has a different transformation of the database. At level 1 the complete dataset having multiple classes is transformed into a binary dataset containing only two classes. This binary

dataset contains the exact copy of the majority class as class 1 and the class 2 contains rest of the classes clubbed together having their class value normalized to a class value of second majority class.

Let D be the dataset having class, $Ci \; \forall \; i \; \in \; (1, 2, 3)$ where, $Ci > \; Ci + 1$ in order of their number of instances available in each class. This dataset is converted into dataset D1 whose class distribution is given in (8), (9)

$$D1 = \{C2, Z2\} \tag{8}$$

$$C2 \cup C3 \rightarrow Z2 \tag{9}$$

Level 2 contains all the classes of the dataset D except the majority class C1 and can be represented as D2 having only classes C2, C3. If the dataset contains a large number of classes where the proportion of instances among the classes is highly varied, then the levels of the HDT can be increased in accordance to the proportion amongst rest of the classes in the dataset excluding the majority class. This approach can be explained in pseudo code below. Let, D be the dataset containing class $Ci \; \forall \; i \in \{1, 2, 3 \ldots, n\}$ and $Ci > Ci + 1$ according to number oa f instance available in both classes. The data division is done by using Algorithm 2 where, Maj ( ) gives the majority class of the given dataset. Com ( ) is used to combine all the classes of dataset into a sthe ingle class. Min ( ) returns the minority class of dataset of A. CC( ) function counts the number of classes in the dataset and ↘ is the threshold value for maximum disproportion.

Algorithm 2: Hierarchical Data Transformation

1. $\boldsymbol{Algorithm}\; HDT\; (D)$
2. $i \leftarrow i + 1$
3. $X \leftarrow Maj\; (D)$
4. $D \leftarrow (D - X)$
5. $Di \leftarrow Com\; (D)$
6. $Di \leftarrow X \; \cup \; Di$
7. $Y \leftarrow Maj(D)$

8. $Z \leftarrow CC(D)$
9. $M \leftarrow Min\ (D)$
10. $S = S\ \cup\ (D\ \cup\ Di)$
11. $\boldsymbol{If}\ Y/M > \lambda \boldsymbol{and}\ Z > 2\ \boldsymbol{then}$
12. $HDT\ (D)$
13. $\boldsymbol{else}$
14. $return\ S$
15. $\boldsymbol{end}$

The hierarchical multi-stage algorithm used above approaches the multi-classification DR categorization problem in a binary fashion. The value of threshold varies with different datasets, upon testing the λ has been set to 5 for the given dataset. In the given dataset two levels are created both of which are fed to custom CNN of 21 layers to create two trained classifiers. The primarily preferred standpoint of this approach is that it balances the disproportionate weightage problem in multi-classification medical datasets. The weightage among the given dataset for the correct categorization of the image is different for different classes. Generally, every multi-class image classification problem is approached in a similar fashion. Therapeutic recognition approaches are exceptionally distinct on the grounds that, the misclassification of a disease affected patient to a class of non-affected patients holds higher weight age than the misclassification among different stages of the disease. This issue is not considered while approaching any medical classification problem. Dividing dataset into multistage approach helps in dealing with weightage problem by first training the model for classifying an input image as affected by DR or not.

This binary classification provides equal weightage to both the classes and also helps in balancing the skewness among the given data by clubbing all the classes except the majority class into a single class. Next level categorizes the DR affected image into its particular severity grades by training another model over only the dataset containing the DR affected images. This helps in diminishing the false normal rate of the system by providing equal weightage to the DR affected and non-affected images and makes the

system more reliable and robust for automated detection. Moreover, this approach consumes less time, for instance, if a patient is not affected by DR then no further processing is done and diagnosis takes lesser time. But if the patient is having DR then the next step of categorizing the severity grades is executed.

Chapter Summary

In this chapter, an automated DR detection framework is proposed. This framework consists of mainly three modules, i.e., Image Augmentation, Model Selection and Data Transformation. The image augmentation section compares different preprocessing techniques to find the best performing method for our system. The model selection explains the architecture of deep learning network along with different parameters used to optimize its performance. Lastly, the data transformation explains the problems related to skewed datasets and discusses existing data balancing methods. This section also contains two proposed data balancing methods explained with their algorithms which are used for further experimentation.

CHAPTER 6
RESULTS AND DISCUSSION

This research introduced an automated DR grading framework that categorizes the patients' fundus images into five unique classes. Past methodologies in the field of diabetic retinopathy have a maximum of two to three classes for classification purposes. The categorization of retinal images into five classes makes the problem acuter because greater variability in the dataset challenges the predefined methodologies. For the above problem, two diverse approaches are proposed to handle the multiclass problem in their own unique way.

The first approach balances the unbalanced class distribution by creating multiple datasets and bagged their output to create an efficient classifier while the second approach transforms the entire dataset to resolve the multiclass problem in a binary fashion. The images are tested on the handcrafted deep neural network and the class score is allotted to all the test images. With the assistance of these class scores, the framework is then able to distinguish between the five distinct classes of diabetic retinopathy. Each of these approaches has their own pros and cons and are examined on multiple states of the art evaluation metrics to check their performance.

6.1 Comparison of Different Preprocessing Methods

The EyePACS dataset, used for evaluation of proposed approach, contains retinal images taken from different models and types of cameras having varied size and illumination. To make these images usable and enhance their structural features, three distinctive incremental preprocessing approaches are applied and compared intrinsically to assess the best appropriate approach for the framework.

Each image is resized and the left eye images are flipped to apply the same model to both left and right eye. The first preprocessing is done to convert the image into a

monochrome image which is represented as Im. In the second approach, Im is further processed to enhance the structural edges of the retinal image which is represented as Ie. Finally, the edge enhanced image Ie is additionally upgraded to evacuate all the complexities by extricating only the edges of the image and is represented as Iex. Im, Ie,andIex are compared over the common 21 layer network and the best performing processing strategy can be shown in Fig.6.1 which gives the individual accuracies of the different preprocessing techniques.

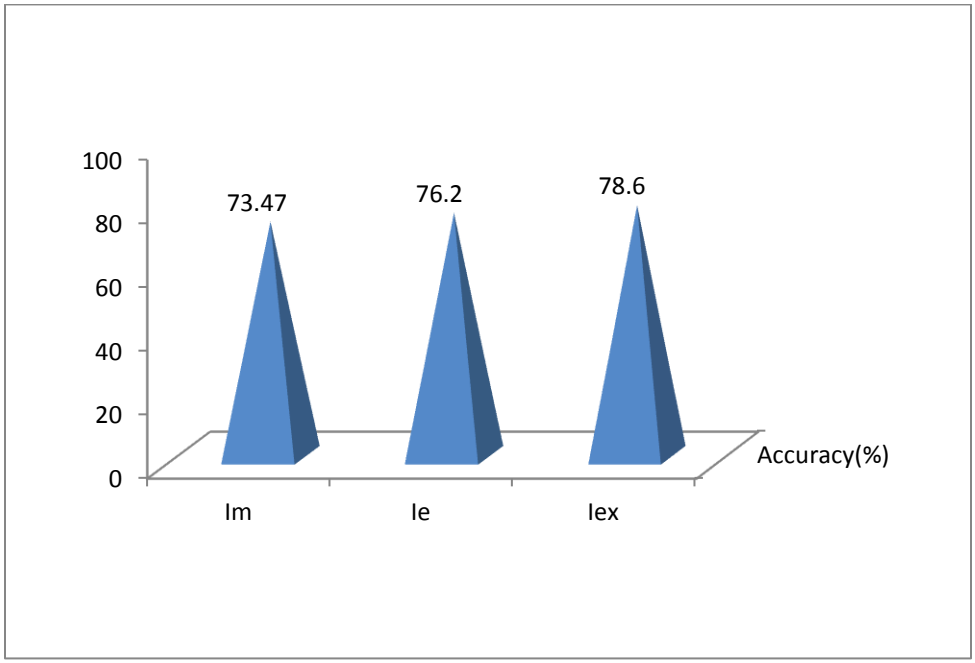


Fig 6.1.The Accuracy of different Pre- processing measures

Each incremental processing enhances the structural features of the image and aides in better execution of the model. The results demonstrate that Iex outperforms Im and Ie and accordingly, is chosen as the base for further model comparisons. This outcome is due to the fact that the edge extracted image Iex initially expels all the illumination constraints by only extracting the edges or the structural features of the retina. Edge enhancement and Monochrome image contribute to it by upgrading the edges and normalizing the intensity. This combination improves the feature extraction and in the meantime expels additional complexities of the image.

6.2 Performance evaluation of classifiers

For evaluating the different classes of the retinal image, a standard 21 layer custom CNN classifier is applied to the whole dataset as well as on the two proposed architecture. This 21 layer CNN consists of 5 sets of convolutional and max-pooling layers with additional hidden and dropout layers. Initially, this model is applied to the complete unbalanced dataset comprising of 35126 images. Table 6.1 shows the confusion matrix of the unbalanced dataset.

Table 6.1.Confusion matrix of the Unbalanced dataset

Actual \ Predicted	Normal	Mild	Moderate	Severe	PDR
Normal	25810	0	0	0	0
Mild	1963	475	3	2	0
Moderate	4370	2	915	4	1
Severe	545	3	1	322	2
PDR	620	1	0	0	87

As no balancing technique was applied to the dataset the trained classifier classifies all the images of majority class which is “Normal” and all other classes are treated as noise and are ignored. The Confusion Matrix in Table 6.1 shows that all the minority classes are classified in “Normal” class. So, the model is highly partial towards a single class due to highly unbalanced data. For enhancing the model performance, two distinctive architectures are proposed that enhanced the performance of classifier and in the meantime reduces time constraints. The first approach MDD uses multiple data division techniques

To balance out the disproportion among different classes. In our approach, MDD creates 5 parallel sub-datasets on each of these sub-datasets our custom CNN is applied. The outcome from each of these datasets is then ensembled. Fig.6.2 shows the confusion matrixes of each sub-dataset of the MDD approach.

	Normal	Mild	Moderate	Severe	PDR
Normal	4101	275	738	20	25
Mild	230	1882	302	16	13
Moderate	446	231	4538	37	40
Severe	58	26	117	665	7
PDR	37	12	97	7	555

	Normal	Mild	Moderate	Severe	PDR
Normal	4426	213	427	62	34
Mild	222	1987	194	25	15
Moderate	446	202	4527	71	46
Severe	44	32	74	713	10
PDR	42	19	57	12	578

	Normal	Mild	Moderate	Severe	PDR
Normal	4471	208	463	6	14
Mild	311	1889	228	7	8
Moderate	568	191	4483	25	25
Severe	83	21	117	641	11
PDR	70	17	108	3	510

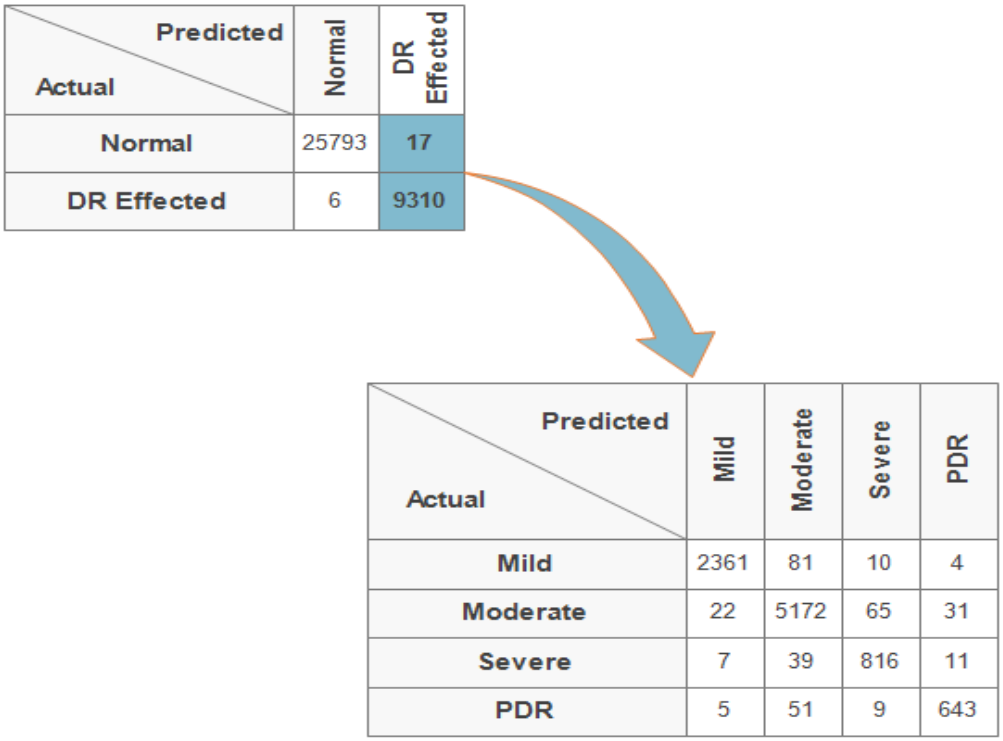
	Normal	Mild	Moderate	Severe	PDR
Normal	4233	332	538	17	42
Mild	204	2012	209	11	7
Moderate	435	297	4458	48	54
Severe	54	45	91	672	11
PDR	47	16	58	10	577

	Normal	Mild	Moderate	Severe	PDR
Normal	4410	203	498	28	23
Mild	271	1938	205	13	16
Moderate	559	184	4494	28	27
Severe	83	31	109	642	8
PDR	47	26	71	9	555

	Normal	Mild	Moderate	Severe	PDR
Normal	25478	218	97	6	11
Mild	91	2048	290	10	4
Moderate	18	132	5054	57	31
Severe	5	7	1340	716	11
PDR	1	5	80	9	613

Fig.6.2.Confusion matrices of MDD approach

The final confusion matrix from the above approach demonstrates an exponential performance augment over the base unbalanced datasets. Balanced multiple datasets help in training unprejudiced classifiers that when club together perform exceedingly superior to the normal unbalanced approach. The second framework is HDT that approaches the problem in a different pattern by converting it into two levels. The first level uses binary classification to classify the image as DR affected or not and the second level uses multiclass classification to further classify its DR affected image into its particular severity grades.



Actual \ Predicted	Normal	DR Effected
Normal	25793	17
DR Effected	6	9310

Actual \ Predicted	Mild	Moderate	Severe	PDR
Mild	2361	81	10	4
Moderate	22	5172	65	31
Severe	7	39	816	11
PDR	5	51	9	643

Fig 6.3.Confusion matrices of HDT approach

The confusion matrix of both levels of HDT is shown in Fig.6.3. These matrices reach the conclusion that the conversion of the multiclass problem into binary enables the model train better and furthermore provides equal weightage of 50% to both normal and DR affected classes which in case of multiclass classification were distributed with the value of 20% among five different classes. This change consequently, helps in making a

medically evident classifier which gives more inclination towards categorizing an image as disease affected or not rather than classifying them into 5 distinct classes normally.

6.3 Experimental results of the proposed approach

Several parameters like Precision, recall, False Normal (FNORM), kappa statics and accuracy are used to evaluate the performance of the classifier by using confusion matrix. The generalization to multi-class problems is to sum over rows/columns of the confusion matrix. Given the confusion matrix with values represented as Mij where 'i' represents no of columns and 'j' represents no of rows of the matrix. Then the precision for each class can be calculated as in (10).

$$\textbf{Precision}_{i} = \frac{Mii}{\sum_{j=1}^{n} Mji} \qquad \textbf{(10)}$$

That is, precision is the fraction of events where we correctly declared 'i' out of all instances where the algorithm declared 'i'. It refers to the closeness of two or more measurements to each other. For instance, if you weigh a given item 5 times and get the same result each time that measurement is very precise. Precision is generally independent of accuracy, you can be very precise but inaccurate and vice versa. Table 6.2 shows the precision value of each approach corresponding to 5 different classes in the dataset.

Table 6.2.Per class Precision and Recall of Unbalanced data, MDD and HDT

Classes	Precision			Recall		
	Unbalanced Data	MDD	HDT	Unbalanced Data	MDD	HDT
Normal	100	98.71	99.93	77.48	99.55	99.97
Mild	19.44	83.83	95.94	98.75	84.97	97.87
Moderate	17.29	95.53	97.73	99.56	73.66	96.8
Severe	36.88	34.44	93.47	98.17	89.72	90.66

PDR	12.288	86.58	90.81	96.66	91.49	93.32

Unbalanced data approach shows the least average precision value of 37.18 whereas MDD performed exceedingly better with the average precision rate of 79.81. HDT performed the best out of the three approaches with the average precision rate of 95.58. Conversely, recall is the fraction of events where we correctly declared i out of all of the cases where the true of the state of the world is 'i'. The recall for each class shown in Table 5 is calculated as (11).

$$\textbf{Recall}_{\,i} = \frac{Mii}{\sum_{j=1}^{n} Mij} \qquad \textbf{(11)}$$

Recall distribution in Table 5 shows the lowest average recall for an unbalanced dataset with a value of 94.12 and the highest recall was recorded for HDT with an average value of 95.72. Both precision and recall give a better view of class proportion. Similar to these matrices, accuracy also helps in evaluating the performance of the framework. It is the measure of rightly classified instances among all the instances. The higher the accuracy the higher is the number of instances correctly classified into their classes and thus the better is the model. This can be calculated as in (12).

$$\textbf{Accuracy} = \frac{\sum_{i} Mii}{\sum_{i=1}^{n}\sum_{j=1}^{n} Mij} \qquad \textbf{(12)}$$

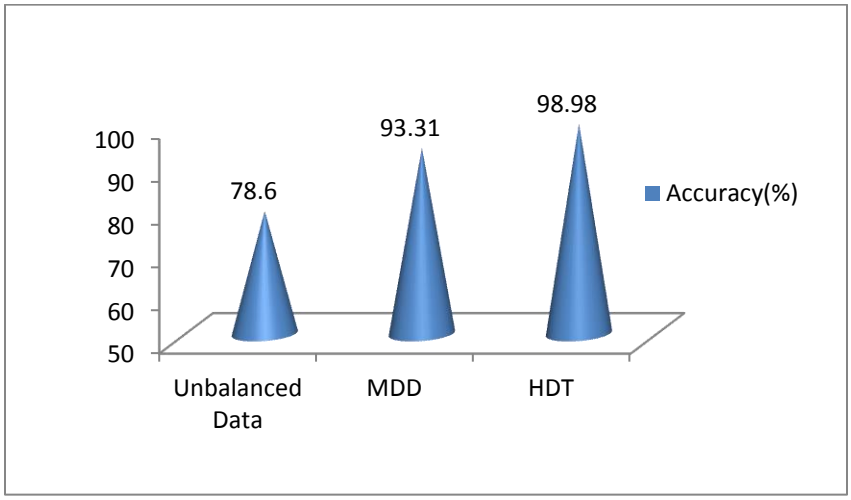


Fig.6.4. Accuracy measure of Unbalanced data, MDD, HDT approach

The Fig.6.4 shows the accuracy of all the algorithms used in the proposed work and gives the outcome that HDT performs better with an accuracy of 98.98% in contrasts with the other two algorithms. All the three metrics are not sufficient to depict the performance of the model in the real world. For this a new metrics False Normal (Fnorm) rate is used which helps in detecting the validity of a model in terms of whether a disease affected is categorized rightly or not. Thus, the value of the false normal should be kept as low as possible. Efforts have been made in this research to diminish the*F*norm rate which can be calculated as in (13).

$$\boldsymbol{Fnorm} = \frac{(\boldsymbol{Number\ of\ images\ wrongly\ classified\ as\ Normal})}{(\boldsymbol{Total\ no\ of\ images})} \qquad \textbf{(13)}$$

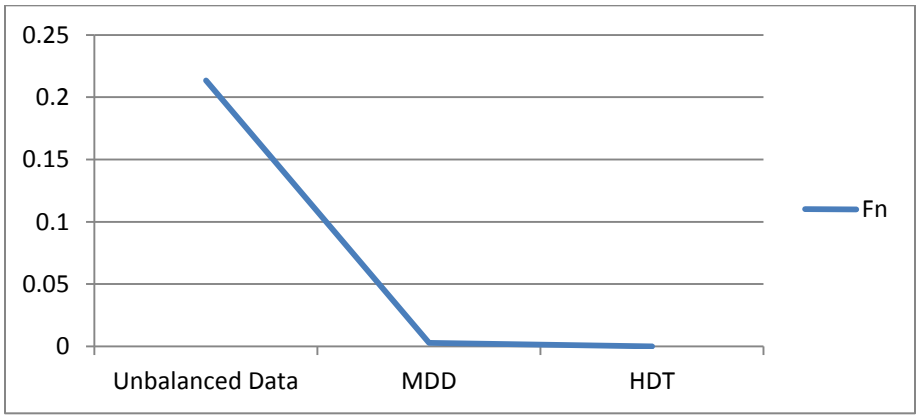


Fig 6.5.False Normal (Fnorm) rate of the three algorithms

The false normal rate of each algorithm shown as a line graph in Fig.6.5 represents that MDD and HDT both highly optimizes the performance of the model by compressing the Fnorm value in the range of 0.003 - 0.0001. Another metrics Kappa is also considered as a true metrics in evaluating the performance in case of multiclass data.

The Kappa statistic is essentially a measure of how well the classifier performed as compared to how well it would have performed simply by chance. In other words, a model will have a high Kappa score if there is a big difference between the accuracy and the null error rate. It is a matrix that compares an *Observed Accuracy* with an *Expected Accuracy*. It can be evaluated as:

$$Kappa = \frac{observed\ accuracy - expected\ accuracy}{1 - expected\ accuracy} \quad (14)$$

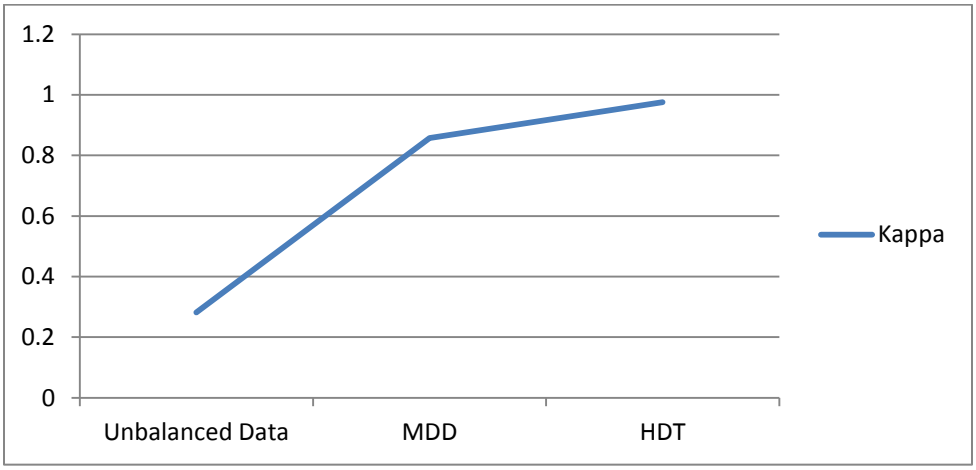


Fig.6.6.Kappa statistic of the proposed approaches

The kappa statistic is used not only to evaluate a single classifier but also to evaluate classifiers amongst themselves. In addition, it takes into account random which generally means it is less misleading than simply using accuracy as a metric. The higher the value of kappa, more efficient is the model. Fig.6.6 shows that HDT has the highest value of Kappa.

6.4 Comparison with the Existing work

The performance of the proposed DR grading system is compared with the existing frameworks in Table 6.3. Majority of the pre-existing systems uses a very small dataset and accomplishes a lesser sensitivity, specificity, and accuracy than the proposed approach. Pratt et al. [24] utilizes EyePACS dataset to prepare deep learning models but achieves an accuracy of 75% with a sensitivity of 95%. Sadek et al. [23] use publically accessible STARE, DRIVE, DRIDB, HEI-MED, MESSIDOR, and HRF datasets having 430 images in total for classifying normal, drusen and exudates and achieves an accuracy of 97%. Many other approaches used local dataset comprising of very less number of images and thus were not able to achieve a high accuracy. Our system outperforms all the existing methods and categorizes the DR affected images with an accuracy of 98.98%. It

is also observed that the proposed system accomplishes a high sensitivity and very low false normal rate which implies that none of DR affected patient is classified as normal.

Table 6.3.Comparison of the proposed system with the existing system

Method	**Approach**	**No. of images**	**SEN (%)**	**SPEC (%)**	**AUC (%)**
Gardner et al. (1996) [25]	SVM	200	88.4	83.5	-
Nayak et al. (2008) [26]	Neural Network	140	90	100	93
Shahin et al. (2012) [29]	ANN	340	87.5	90.9	88.9
Acharya et al. (2008) [31]	SVM	300	82	88	82
Acharya et al. (2009) [32]	SVM	331	82	86	85.9
Lupascu et al. (2010) [24]	Adaboost	40	.72	-	.9597
Fraz et al. (2012) [18, 19]	Ensemble Classifier	40 and 20	.740/.754	.980/.976	.948/.953
Odstrcilik et al. (2013) [20]	Improved Matched Filtering	45	.746/.706	.961/.969	.944/.934
Adarsh et al. (2013) [33]	SVM	130 and 89	90.6	93.65	95.3
Sadek et al. (2015) [34]	SVM	430	-	-	97.22 - 99.77
Wang et al. (2015) [38]	CNN and RF	40 and 20	.817/.810	.973/.979	.976/.981
Liskowski et al. (2016) [39]	Variants of CNN	40 / 20 / 28	-	-	>.97
Pratt et al. (2016) [40]	CNN	35126	95	-	75
Unbalanced data	CNN	35126	94.12	-	78.6
MDD	CNN	35126	87.87	-	93.31
HDT	CNN	35126	95.72	-	98.98

Chapter Summary

This chapter is divided into four subsections. Firstly, Comparison of different preprocessing methods is done over the EyePACS dataset. Accuracy comparison between each successive preprocessing method is done to find the best performing technique. Secondly, the performance evaluation of classifiers is carried out in which the confusion matrix is drawn for different data balancing techniques. The next subsection discusses the experimentation results of the proposed approach in which different evaluation metrics are used to show the performance of three proposed approaches. Finally, a comparison between the proposed approaches and the existing work is shown in the tabular form.

CHAPTER 7
CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

We have proposed a novel method for DR detection in fundus images using deep learning based approach. The contribution of this research is two-fold: firstly, we proposed a special deep neural network architecture for the diabetic retinopathy image classification task, which demonstrates superior performance over conventional feature extraction based methods. Moreover, two data augmentation method MDD and HDT were introduced for the proposed algorithm, which also improves the performance of the algorithm. The existing supervised algorithms for detection of DR used multiple pre-processing and post-processing stages. In addition, the vast majority of algorithms require feature extraction to classify fundus images. The use of handcrafted 21 layers deep learning network automates the feature extraction phase reducing the human intervention for DR characterization. The EyePACS dataset undertaken in this approach is highly skewed and results in a highly biased classifier.

Thus, the network struggled in identifying the minority classes that resulted in misclassification of DR affected eye into Normal class causing a high Fnorm rate. The disproportions among the instances of different classes lead to the creation of two different approaches. Extensive experimentation was performed over these proposed algorithms using state of the art matrices. Results on benchmark EyePACS dataset observationally exhibit that HDT achieves a new state of the art in this domain followed by MDD. The main insight provided by HDT is that the distribution of the dataset into two-tier architecture aides in learning of the classifier. At the same time, HDT reduces the Fnorm rate by approaching the problem in a more medical sense.

7.2 Future Scope

The main drawback of this system is the complex network architecture which is computation-intensive and requires a graphics processing unit to process the high-resolution images which constrained us to use 128-pixel images that affected the accuracy

of our system. A custom dataset consisting of large number of high resolution fundus images need to be built for better training of classifier. The results from our network are very promising from an orthodox network topology. To conclude, we have shown that proposed architectures have the potential to extract complex features even from highly skewed data to identify Diabetic Retinopathy efficiently offering a real-time classification.

REFERENCES

1. R. Frank, "Diabetic Retinopathy", New England Journal of Medicine, vol. 350, no. 1, pp. 48 58, 2004.

2. Ng, E. Y. K., Acharya, U. R., Rangayyan, R. M., & Suri, J. S. (Eds.). (2014). *Ophthalmological Imaging and Applications*. CRC Press.

3. World Diabetes, A newsletter from the World Health Organization, 4, 1998.

4. Ong, G. L., Ripley, L. G., Newsom, R. S., Cooper, M., &Casswell, A. G. (2004). Screening for sight-threatening diabetic retinopathy: comparison of fundus photography with automated color contrast threshold test. American journal of ophthalmology, 137(3), 445-452.

5. Sharma, H., Zerbe, N., Klempert, I., Hellwich, O., & Hufnagl, P. (2017). Deep convolutional neural networks for automatic classification of gastric carcinoma using whole slide images in digital histopathology. *Computerized Medical Imaging and Graphics*, *61*, 2-13.

6. Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, *61*, 85-117.

7. Glorot, X., Bordes, A., & Bengio, Y. (2011, June). Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*(pp. 315-323).

8. Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., ... & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, *29*(6), 82-97.

9. Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013, February). On the importance of initialization and momentum in deep learning. In *International conference on machine learning* (pp. 1139-1147).

10. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

11. Zhu, C., Zou, B., Zhao, R., Cui, J., Duan, X., Chen, Z., & Liang, Y. (2017). Retinal vessel segmentation in color fundus images using Extreme Learning Machine. *Computerized Medical Imaging and Graphics*, *55*, 68-77.

12. Haleem, M. S., Han, L., van Hemert, J., & Li, B. (2013). Automatic extraction of retinal features from colour retinal images for glaucoma diagnosis: a review. *Computerized medical imaging and graphics*, *37*(7), 581-596.

13. Dietterich, T. (1995). Overfitting and under computing in machine learning. *ACM computing surveys (CSUR), 27*(3), 326-327.

14. Kande, Giri Babu, P. Venkata Subbaiah, and T. Satya Savithri. (2010) "Unsupervised fuzzy based vessel segmentation in pathological digital fundus images." Journal of medical systems 34(5): 849-858.

15.Yao, Chang, and Hou-jin Chen. (2009) "Automated retinal blood vessels segmentation based on simplified PCNN and fast 2DOtsu algorithm." Journal of Central South University of Technology 16(4): 640-646.

16. Cinsdikici, Muhammed Gökhan, and Doğan Aydın. (2009) "Detection of blood vessels in ophthalmoscope images using MF/ant (matched filter/ant colony) algorithm." Computer methods and programs in biomedicine 96(2): 85-95.

17. Amin, M. Ashraful, and Hong Yan. (2011) "High-speed detection of retinal blood vessels in fundus image using phase congruency." Soft Computing 15(6): 1217-1230.

18. Fraz, Muhammad Moazam, et al. (2012) "An approach to localize the retinal blood vessels using bit planes and centerline detection." Computer methods and programs in biomedicine108(2): 600-616.

19. Fraz, Muhammad Moazam, et al. (2012) "Blood vessel segmentation methodologies in retinal images–a survey." Computer methods and programs in biomedicine 108(1): 407-433.

20. Odstrcilik, Jan, et al. (2013) "Retinal vessel segmentation by improved matched filtering: evaluation on a new high-resolution fundus image database." IET Image Processing 7(4): 373-383.

21. Sinthanayothin, Chanjira, et al. (1999) "Automated localization of the optic disc, fovea, and retinal blood vessels from digital color fundus images." British Journal of Ophthalmology 83(8): 902-910.

22. Soares, João VB, et al. (2006) "Retinal vessel segmentation using the 2-D Gabor wavelet and supervised classification." IEEE Transactions on medical Imaging 25(9): 1214-1222.

23. Ricci, Elisa, and Renzo Perfetti. (2007) "Retinal blood vessel segmentation using line operators and support vector classification." IEEE transactions on medical imaging 26(10): 1357-1365.

24. Lupascu, Carmen Alina, Domenico Tegolo, and Emanuele Trucco. (2010) "FABC: retinal vessel segmentation using AdaBoost." IEEE Transactions on Information Technology in Biomedicine14(5): 1267-1274.

25. Gardner, G. G., Keating, D., Williamson, T. H., & Elliott, A. T. (1996). Automatic detection of diabetic retinopathy using an artificial neural network: a screening tool. *British Journal of Ophthalmology*, *80*(11), 940-944.

26. Nayak, J., Bhat, P. S., Acharya, R., Lim, C. M., & Kagathi, M. (2008). Automated identification of diabetic retinopathy stages using digital fundus images. *Journal of medical systems*, *32*(2), 107-115.

27. Thomas, N., & Mahesh, T. (2014). Detecting Clinical Features of Diabetic Retinopathy using Image Processing. International Journal of Engineering Research & Technology.

28. Gandhi, M., & Dhanasekaran, R. (2013, April). Diagnosis of diabetic retinopathy using morphological process and SVM classifier. In *Communications and Signal Processing (ICCSP), 2013 International Conference on* (pp. 873-877). IEEE.

29. Shahin, E. M., Taha, T. E., Al-Nuaimy, W., El Rabaie, S., Zahran, O. F., & El-Samie, F. E. A. (2012, December). Automated detection of diabetic retinopathy in blurred digital fundus images. In *Computer Engineering Conference (ICENCO), 2012 8th International* (pp. 20-25). IEEE.

30. Prenta, P. (2013). Detection of Diabetic Retinopathy in Fundus Photographs. *University of Zagreb, Faculty of Electrical Engineering and Computing, Unska*, *3*, 10000.

31. Acharya, R., Chua, C. K., Ng, E. Y. K., Yu, W., & Chee, C. (2008). Application of higher order spectra for the identification of diabetes retinopathy stages. *Journal of Medical Systems*, *32*(6), 481-488.

32. Acharya, U. R., Lim, C. M., Ng, E. Y. K., Chee, C., & Tamura, T. (2009). Computer-based detection of diabetes retinopathy stages using digital fundus images. *Proceedings of the institution of mechanical engineers, part H: journal of engineering in medicine*, *223*(5), 545-553.

33. Adarsh, P., & Jeyakumari, D. (2013, April). Multiclass SVM-based automated diagnosis of diabetic retinopathy. In *Communications and Signal Processing (ICCSP), 2013 International Conference on* (pp. 206-210). IEEE.

34. Sadek, I., Sidibé, D., & Meriaudeau, F. (2015, March). Automatic discrimination of color retinal images using the bag of words approach. In *Medical Imaging 2015: Computer-Aided Diagnosis*(Vol. 9414, p. 94141J). International Society for Optics and Photonics.

35. Bengio, Yoshua, et al. (2007) "Greedy layer-wise training of deep networks." Advances in neural information processing systems: 153-160.

36. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. (2012) "Image net classification with deep convolutional neural networks." Advances in neural information processing systems: 1097-1105.

37. LeCun, Yann, and Yoshua Bengio. (1995) "Convolutional networks for images, speech, and time series." The handbook of brain theory and neural networks 3361(10): 1995.

38. Wang, Shuangling, et al. (2015) "Hierarchical retinal blood vessel segmentation based on feature and ensemble learning." Neurocomputing 149: 708-717.

39. Liskowski, Paweł, and Krzysztof Krawiec.(2016) "Segmenting retinal blood vessels with deep neural networks." IEEE transactions on medical imaging 35(11): 2369-2380.

40. Pratt, H., Coenen, F., Broadbent, D. M., Harding, S. P., & Zheng, Y. (2016). Convolutional neural networks for diabetic retinopathy. *Procedia Computer Science*, *90*, 200-205.

41.https://www.analyticsvidhya.com/blog/2017/04/comparison-between-deep-learning-machine-learning/

42. https://searchenterpriseai.techtarget.com/definition/machine-learning-ML.

43.https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

44. Gross, R., & Brajovic, V. (2003, June). An image preprocessing algorithm for illumination invariant face recognition. In *International Conference on Audio-and Video-Based Biometric Person Authentication* (pp. 10-18). Springer, Berlin, Heidelberg.

45. Shin, H. C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., ... & Summers, R. M. (2016). Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, *35*(5), 1285-1298.

46. Chitra, R., & Seenivasagam, V. (2013). Heart disease prediction system using supervised learning classifier. *Bonfring International Journal of Software Engineering and Soft Computing*, *3*(1), 1.

47. Orlando, J. I., Prokofyeva, E., del Fresno, M., & Blaschko, M. B. (2018). An ensemble deep learning based approach for red lesion detection in fundus images. *Computer methods and programs in biomedicine*, *153*, 115-127.

LIST OF PUBLICATIONS

1. K. Kalra, R. Goyal, S. Kaur, P. Bhatia, "PDD Algorithm for Balancing Medical Data" in 2nd International Conference on Advances in Computing and Data Sciences (ICACDS 2018)

   [Accepted]

2. R. Goyal, K. Kalra, P. Bhatia, S. Kaur, "Intelligent Face Recognition System For Visually Impaired" in 2nd International Conference on Advances in Computing and Data Sciences (ICACDS 2018).

   [Accepted]

3. K. Kalra, R. Goyal, S. Kaur, P. Bhatia, "Automated real time hand gesture recognition interface" in Journal of multimode user interface (Springer).

   [With Editor]

4. R. Goyal, K. Kalra, P. Bhatia, S. Kaur, "UMEED: Deep Learning based complete walking and reading solution for visually impaired people" in Universal Access in the information Society (Springer).

   [Communicated]

5. K. Kalra, R. Goyal, S. Kaur, P. Bhatia, "DESIRE: Deep Learning Enabled System Integration for Retinopathy Evaluation" in International Journal of pattern recognition and artificial intelligence (World Scientific).

   [With Editor]

VIDEO LINK

Karan Kalra, “Deep Learning based Diabetic Retinopathy Detection”, https://youtu.be/l8fH2EO1Y5s