DETECTION OF DIABETIC RETINOPATHY USING ALEXNET AND LENET CNN MODELS

A PROJECT REPORT

Submitted By

NEYA BABU T (180701152)

NIKGHAMANTH S S (180701153)

In partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE,

ANNA UNIVERSITY::CHENNAI - 600 025

MAY 2022

RAJALAKSHMI ENGINEERING COLLEGE

BONAFIDE CERTIFICATE

Certified that this project report **"DETECTION OF DIABETIC RETINOPATHY USING ALEXNET AND LENET CNN Models**" is the bonafide work of "**NEYA BABU T. (180701152) & NIKGHAMANTH S.S. (180701153)** "who carried out the project under my supervision.

SIGNATURE
Dr. P. REVATHY
HEAD OF THE DEPARTMENT
Department of Computer Science
and Engineering
Rajalakshmi Engineering College
Chennai - 602 105

SIGNATURE
Mrs. V. JANANEE
SUPERVISOR
Assistant Professor
Department of Computer
Science and Engineering
Rajalakshmi Engineering College
Chennai - 602 105

Submitted to Project and Viva Examination held on ____________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

This study aims to develop a system to distinguish Diabetic retinopathy disease from fundus images. In this work, we extracted different Diabetic retinopathy features such as Diabetic retinopathy, optic disc and lesions, applied convolutional neural network based models for the detection of multiple Diabetic retinopathy diseases with fundus photographs involved in structured analysis of the retinal (STARE - Structured Analysis of the Retina) database. A variety of neuron-wise and layer-wise visualization methods were applied using CNN and trained with a publicly available Diabetic retinopathy disease image dataset. It is observed that neural networks can capture the colors and textures of lesions specific to respective diseases upon diagnosis, which resembles human decision-making.Diabetic Retinopathy is a condition that is caused by excessive glycemia. It can often be tough to tell the variation among DR and fundus photographs. To avoid difficulties, it is crucial to acknowledge the difference before concluding the disease and grafting the treatment plan. We can detect many Diabetic Eye Disease illnesses using Cnns. In addition to this, Cnns also detects the colors and patterns of sores and matches them to relevant conditions during medical diagnosis, which is similar to human decision-making. The Django web framework showcases the best output that is derived after comparing 3 models of Cnn. To determine the most efficient and accurate categorization of Diagnosed images, researchers use many related images as input into this convolutional semantic networks. The output derived shows whether the given input image is affected with DR or not. This also an easy process which is easily done by CNN models.

ACKNOWLEDGEMENT

First, we thank the almighty god for the successful completion of the project. Our sincere thanks to our chairman **Mr. S. Meganathan, B.E., F.I.E.**, for his sincere endeavor in educating us in his premier institution. We would like to express our deep gratitude to our beloved Chair Person **Dr. Thangam Meganathan, Ph.D.**, for her enthusiastic motivation which inspired us a lot in completing this project and Vice Chairman **Mr. Abhay Shankar Meganathan, B.E., M.S.**, for providing us with the requisite infrastructure.

We also express our sincere gratitude to our college principle, **Dr. S. N. Murugesan**, **M.E., Ph.D.**, and **Dr. P. Revathy M.E., Ph.D.**, Professor and Head of the Department of Computer Science and Engineering and our project guide **Mrs. V. Jananee, M.E.**, Assistant Professor (SG) for her encouragement and guiding us throughout the project. We would like to thank our project coordinator **Dr. S. Prithi, M.E., Ph.D**, Associate Professor for her encouragement and guidance towards successful completion of this process and to our parents, friends, all faculty members and supporting staffs for their direct and indirect involvement in successful completion of the project for their encouragement and support.

NEYA BABU. T

NIKGHAMANTH. S. S

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

To wage the therapy, it is important to identify the disease. In simplified words, complications lead to problems in finding the actually present disease from the given datasets and separating other diseases from the Diabetic Retinopathy (DR) disease is essential. Since DR diseases are often confused with fundus photographs, these are the major complications found while finding the datasets for analyzing this disease. So, it is important to separate them before analysis. This task is proficiently done using Convolutional Neural Network (CNN).

1.1 MOTIVATION

Diabetic Retinopathy is a deadly disease with no cure or prevention. Early detection and medications are the only way to stall it from complicating or becoming deadly. When a person is affected by DR, they slowly lose their eye sight leading to permanent blindness. This has lead us to take up this project and search for already existing methods used for early detection. We found certain flaws in those methods and thus decided to improve them with our proposed system for the betterment in treating people affected with this illness.

1.2 DIABETIC RETINOPATHY

It is a diabetes complication that affects eyes. It's caused by damage to the blood vessels of the light-sensitive tissue at the back of the eye (retina).

1.3 UNDERSTANDING DR

Diabetic retinopathy is caused by changes in the blood vessels of the retina, the light-sensitive layer of tissue at the back of the inner eye. In some people with diabetic retinopathy, the blood vessels in the retina may swell and leak fluid. In others, abnormal new blood vessels grow on the surface of the retina. There is no cure for diabetic retinopathy. But treatment works very well to prevent, delay, or reduce vision loss. The sooner the condition is found, the easier it is to treat. One of such treatment is laser treatment, which is used predominantly while tackling this disease. There are some early symptoms of DR such as spots or dark strings floating in your vision (floaters), Blurred vision, Fluctuating vision, Dark or empty areas in your vision & Vision loss. There are also 4 types of DR namely, Mild Nonproliferative Retinopathy, Moderate Nonproliferative Retinopathy, Severe Nonproliferative Retinopathy & Proliferative Diabetic Retinopathy (PDR).

1.4 DETECTION OF DR USING IMAGE CLASSICIATION

Machines can be taught to mimic the same way human brains interpret and analyze images and its features. Artificial Intelligence with image Processing has many application and lay foundation to modern technologies such as face recognition in mobile phones and other applications to ensure high level security, detecting and recognizing objects and patterns in images and videos, and so on. Today, image processing is widely used in medical visualization, biometrics, self-driving vehicles, gaming, surveillance, law enforcement, and many other domains. Some of the image processing application and where it is used widely are, Visualization which represents processed data in a human understandable way, giving visual form of objects that are easy to explain and helps to make key decision, Image sharpening and restoration which improve the quality of images that are processed, image retrieval which helps to search with image. Object detection which is used to detect objects in an image, Pattern recognition classifies objects, their positions and understand the hidden pattern in an image.

There are millions of diseases that already exist or are originated in this world. Thanks to the advanced technologies and image processing, these diseases can be very accurately detected and treated upon the affected. The computed tomography

(CT) scans, ultrasounds, magnetic resonance imaging (MRI)—to help the health care practitioner identify the cause of disease. A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an image, assign parameters (learnable weights and biases) to objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics of an image automatically. The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlaps to cover the entire visual area.

The objective is to develop a design for Diabetic eye disease or DR, by classification algorithms for possibly arriving the cause and the kind of supreme precision by contrasting the original style.

Thus, the CNN is used to analyze the given retinal image and first helps in differentiating DR images from fundus photographs. Then, the deep learning module helps to understand whether the disease is present in the given input image from the datasets or not.

1.5 CONVOLUTIONAL NEURAL NETWORK ARCHITECTURE

A CNN has three layers: a convolutional layer, a pooling layer, and a fully connected layer.

1.5.1 Convolution Layer

The convolution layer is the core building block of the CNN. It carries the main portion of the network's computational load. This layer performs a dot product between two matrices, where one matrix is the set of learnable parameters also known as a kernel, and the other matrix is the restricted portion of the receptive field. The kernel is spatially smaller than an image but is more in-depth in feature extraction, for example if the image is composed of three (RGB) channels, the kernel height and width will be spatially small, but the depth extends up to all three channels.

During the forward propagation, the kernel slides across the height and width of the image-producing the image representation of that receptive region. This produces a 2D representation of the image known as an activation map that gives the response of the kernel at each spatial position of the image. The sliding size of the kernel is called a stride.

If we have an input of size W x W x D and D out number of kernels with a spatial size of F with stride S and amount of padding P, then the size of output volume can be determined by the following formula:

$$W_{out} = \frac{W - F + 2P}{S} + 1$$

Figure 1.1 Formula -Output Volume Size

This results an output volume of size W out x W out x D out.

Motivation behind Convolution

Convolution leverages three important purpose that motivated computer vision researchers are sparse interaction, parameter sharing, and equivariant representation.

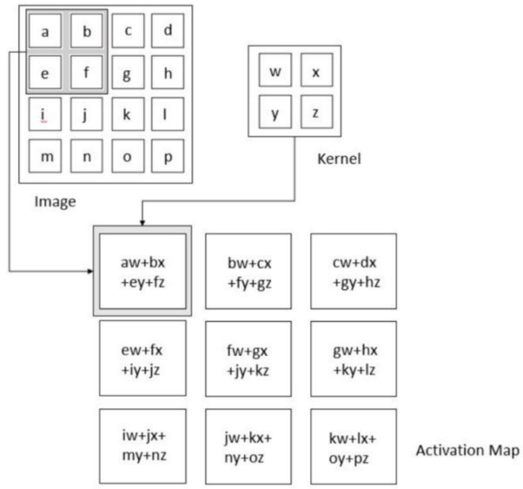


Figure 1.2 Activation Map

Neural network layers use matrix multiplication by a matrix of parameters describing the interaction between the input and output. This means that every output unit interacts with every input unit. Convolution neural networks have sparse interaction which is achieved by making kernel smaller than the input e.g., an image can have millions of pixels, but while processing it using kernel, we can detect features that is of fewer pixels which also implies that we need to store fewer parameters that reduces the memory requirement of the model and also improves the efficiency of the model.

Computing one feature at a spatial point (x1, y1) is useful then other spatial point say (x2, y2) is equivalently useful too. It implies that for a single 2D slice, for one activation map creation, neurons are constrained to use the same set of weights. In traditional neural network, each element of the weight matrix is used only once, while convolution network has shared parameters that includes both trainable and non-trainable parameters for outputs. Because of parameter sharing, the layers of convolution neural network will have a property of equivariance to translation, which implies that change in the input results in the output.

Pooling Layer

The pooling layer replaces the output of the network at certain locations by deriving a summary statistic of the nearby outputs. This helps in reducing the spatial size of the representation, and also decreases the computation cost and weights. The pooling operation is applied on every slice of the representation individually.

There are several pooling functions such as the average of the rectangular neighbourhood, L2 norm of the rectangular neighbourhood, and a weighted average based on the distance from the central pixel. However, the most popular process is max pooling, which reports the maximum output from the neighbourhood.

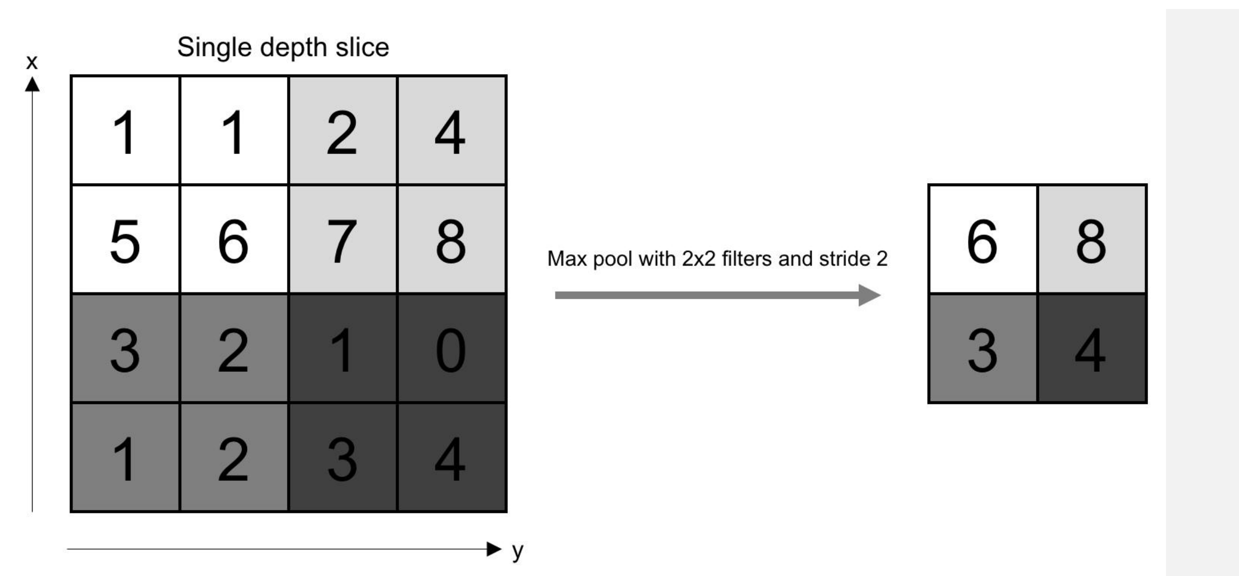


Figure 1.3 Pooling Layer Activation Map

If we have an activation map of size W x W x D, a pooling kernel of spatial size F, and stride S, then the size of output volume can be determined by the following formula:

$$W_{out} = \frac{W - F}{S} + 1$$

Figure 1.4 Formula -Output Size for padding Layer

Formula for Padding Layer

This will yield an output volume of size W out x W out x D. In all cases, pooling provides some translation invariance which means that an object would be recognizable regardless of where it appears on the frame.

Fully Connected Layer

Neurons in this layer have full connectivity with all neurons in the preceding and succeeding layer as seen in regular FCNN. This is why it can be computed as usual by a matrix multiplication followed by a bias effect. The FC layer helps to map the representation between the input and the output.

Activation Functions

Since convolution is a linear operation and images are far from linear, non-linearity layers are often placed directly after the convolutional layer to introduce non-linearity to the activation map. There are several types of non-linear operations, the popular ones being:

Sigmoid

The sigmoid non-linearity has the mathematical form σ(κ) = 1/(1+e¯κ). It takes a real-valued number and "squashes" it into a range between 0 and 1. However, a very undesirable property of sigmoid is that when the activation is at either tail, the gradient becomes almost zero. If the local gradient becomes very small, then in backpropagation it will effectively "kill" the gradient. Also, if the data coming into the neuron is always positive, then the output of sigmoid will be either all positives or all negatives, resulting in a zig-zag dynamic of gradient updates for weight.

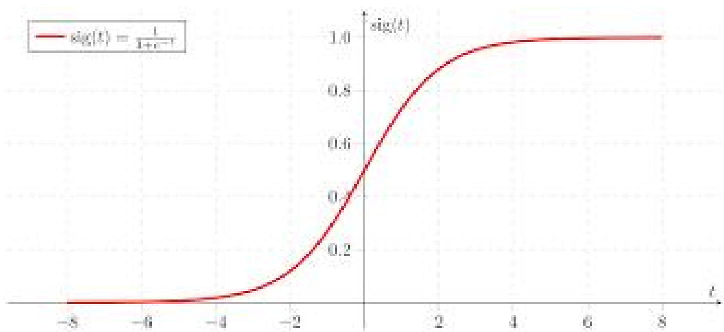


Figure 1.5 Activation Function – Sigmoid

Tanh

Tanh squashes a real-valued number to the range [-1, 1]. Like sigmoid, the activation saturates, but — unlike the sigmoid neurons — its output is zero centred.

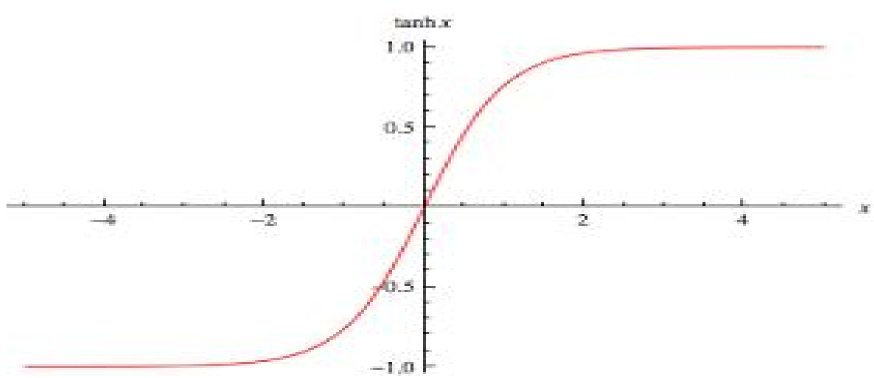


Figure 1.6 Activation Function – TanH

ReLU

The Rectified Linear Unit (ReLU) has become very popular in the last few years. It computes the function $f(\kappa)$=max (0, κ). In other words, the activation is simply threshold at zero. In comparison to sigmoid and tanh, ReLU is more reliable and accelerates the convergence by six times.

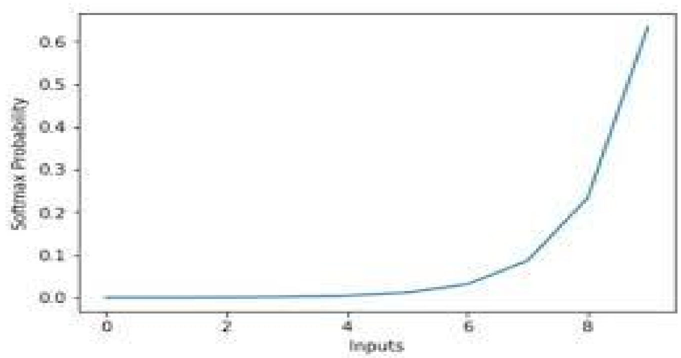


Figure 1.7 Activation Function – ReLU

Unfortunately, a con is that ReLU can be fragile during training. A large gradient flowing through it can update it in such a way that the neuron will never get further updated. However, we can work with this by setting a proper learning rate.

1.6 MANUALNET ARCHITECTURE

ManualNet architecture is our proposed architecture created by having the already existing AlexNet and LeNet architectures as the base. The architecture has 3 layers and thus has less errors and takes lesser time to complete its training and testing of datasets. This architecture yields an output of 98.44% max_accuracy consistently.

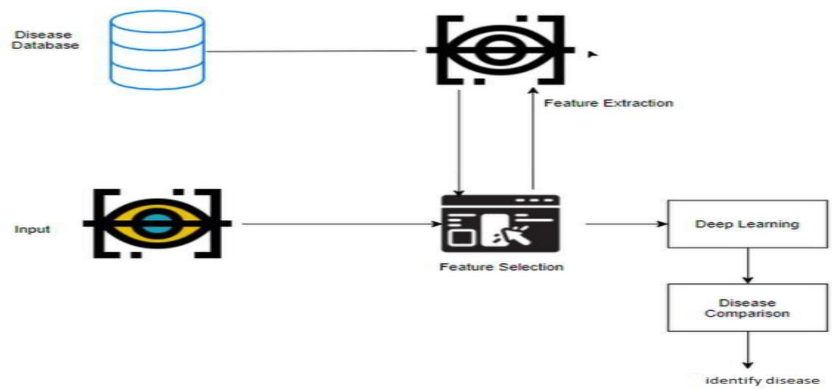


Figure 1.8 ManualNet Architecture

1.7 ALEXNET ARCHITECTURE

The Alexnet has eight layers with learnable parameters. The model consists of five layers with a combination of max pooling followed by 3 fully connected layers and they use Relu activation in each of these layers except the output layer. This architecture also yields error-free results without complications or time consumption compared to other bigger architectures like RESNET-50 as it has lesser layers with an output of 95% max_accuracy.

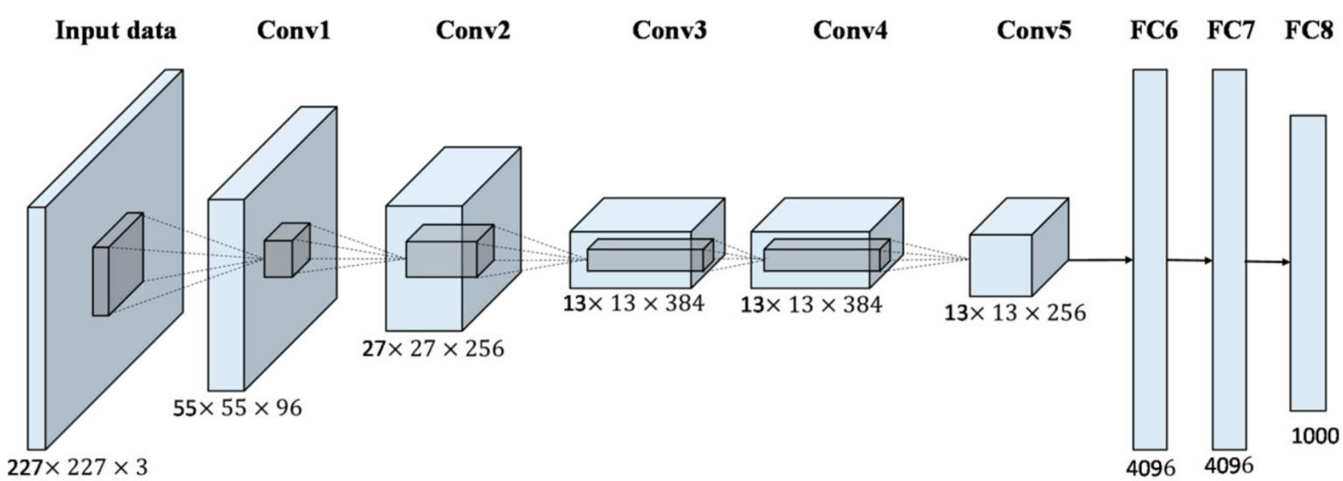


Figure 1.9 AlexNet Architecture

1.8 LENET ARCHITECTURE

The Lenet has 5 layers with learnable parameters and hence named Lenet-5. It has three sets of convolution layers with a combination of average pooling. After the convolution and average pooling layers, we have two fully connected layers. It is the most successful and consistent architecture of all the existing CNN architectures and yields an output of 100% max_accuracy and with an average accuracy of more than 98% consistently.

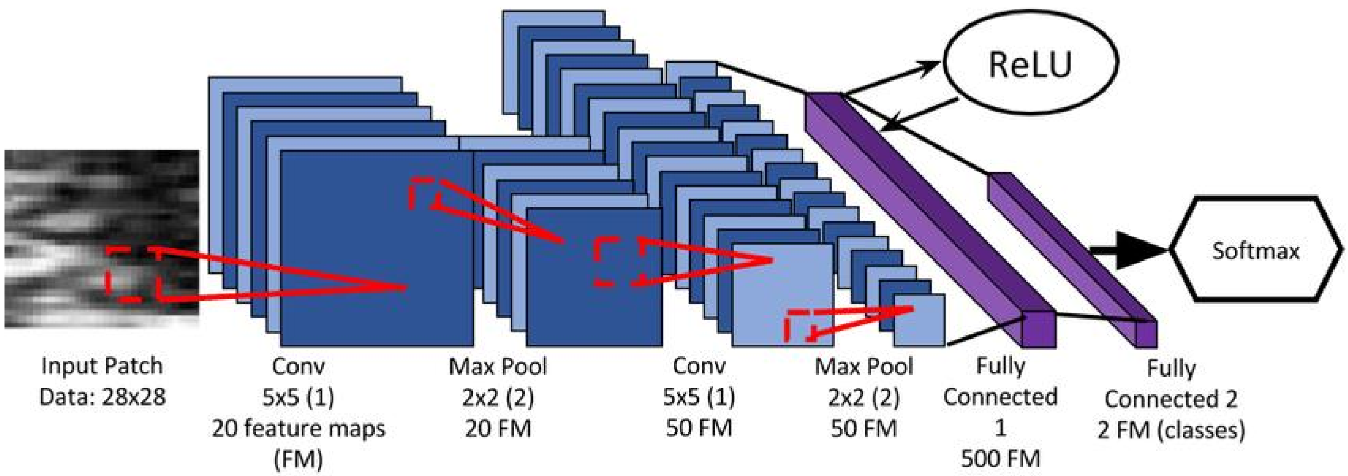


Figure 1.10 LeNet Architecture

CHAPTER 2
LITERATURE SURVEY

A literature survey or a literature review in a project report is that section which shows the various analyses and research made in the field of your interest and the results already published, taking into account the various parameters of the project and the extent of the project.

The following papers are studied in the following survey:

1. A Deep Learning Ensemble Approach for Diabetic Retinopathy Detection.(2020)

In this research, they used the publicly available Kaggle dataset of retina images to train an ensemble of five deep Convolution Neural Network (CNN) models (Resnet50, Inceptionv3, Xception, Dense121, Dense169) to encode the rich features and improve the classification for different stages of DR. The experimental results show that the proposed model detects all the stages of DR unlike the current methods and performs better compared to state-of-the-art methods on the same Kaggle dataset.

Sehrish Qummar, Fiaz Gul Khan, Sajid Shah, Ahmad Khan, Shahaboddin Shamshirband, Zia Ur Rehman, Iftikhar Ahmed Khan & Waqas Jadoon are the authors of this paper.

Advantages : Very high accuracy

Disadvantages : The method proposed, although finds PDR(final stage) cannot find the earlier symptoms or stages of DR.13

2. Improved and robust deep learning agent for preliminary detection of Diabetic Retinopathy using public datasets. (2020)

This work is an attempt to speed up preliminary screening of DR to cater to the future requirement of such a huge amount of diabetic patients. They have trained and validated robust classification models on publicly available datasets for early detection of DR. We have applied state-of-the-art deep learning models based on Convolutional Neural Networks (CNN), to exploit data-driven machine learning methods for the purpose. We framed the problem as a binary classification for the detection of DR of any grade (Grade 1–4) vs No-DR (Grade 0). We used 56,839 fundus images from the EyePACS dataset for training the models. The models were tested on a test set from EyePACS (14,210 images), benchmark test datasets Messidor-2 (1748 images) and Messidor-1 (1200 images).

Gaurav Saxena, Dhirendra Kumar Verma, Amit Paraye, Alpana Rajan & Anil Rawat proposed this paper.

Advantages : Fast coverage of data sets having image with similar feature size.

Disadvantages : Processing time is large because of the huge dataset that is used here.

3. Detection of Diabetic Retinopathy Using Bichannel Convolutional Neural Network. (2020)

In this paper, the entropy image computed by using the green component of fundus photograph is proposed. In addition, image enhancement by unsharp masking (UM) 14 is utilized for preprocessing before calculating the entropy images. The bichannel CNN incorporating the features of both the entropy images of the gray level and the green component preprocessed by UM is also proposed to improve the detection performance of referable DR by deep learning.

Shu-I Pao, Hong-Zin Lin, Ke-Hung Chien, Ming-Cheng Tai, Jiann-Torng Chen & Gen-Min Lin proposed this paper.

Advantages : Improves the detection performance by deep learning

Disadvantages : Given dataset is relatively smaller.

4. Diabetic Retinopathy Classification Using ResNet50 and VGG-16 Pretrained Networks. (2019)

This work represents an intelligent system of DR classification based on deep learning (DL) tools, especially convolutional neural networks (CNN). Proposed system can assist ophthalmologists to make a preliminary decision, it allows a DR classification considering normal eyes, mild DR, Moderate DR, Severe DR and Proliferative DR.

Mustapha AATILA, Mohamed LACHGAR, Hamid HRIMECH & Ali KARTIT proposed this paper.

Advantages : High accuracy.

Disadvantages : The architectures require large training datasets and powerful machines to work as the time taken is very high. Even though the output yielded by these architectures are excellent, smaller architectures like LeNet and AlexNet are more successful as they are easily trainable and executable due to their small architecture.

5. Advances in the treatment of diabetic retinopathy. (2020)

Using the Early Treatment Diabetic Retinopathy Study Diabetic Retinopathy Severity Scale, in which an improvement of ≥2 steps is considered clinically significant, approximately one-third of patients with DR and macular edema experience this level of improvement after 1 year of treatment with either ranibizumab or aflibercept. The rates of clinically significant DR improvement with ranibizumab could be twice that in the subgroup of patients with moderately severe or severe nonproliferative DR and macular edema. These clinical trial data indicate that intraocular inhibition of VEGF is a rational approach for the management of DR.

Rishi P Singh 1, Michael J Elman 2, Simran K Singh 3, Anne E Fung 4, Ivaylo Stoilov proposed this paper.

Advantages : Well classified and high accuracy

Disadvantages : Not using simpler architectures like Alexnet and Lenet for image classification.

6. Classification of Diabetic Retinopathy Images by Using Deep Learning Models. (2019)

The idea behind this paper is to propose an automated knowledge model to identify the key antecedents of DR. Proposed Model have been trained with three types, back propagation NN, Deep Neural Network (DNN) and Convolutional Neural Network (CNN) after testing models with CPU trained Neural network gives lowest accuracy because of one hidden layers whereas the deep learning models are out performing NN. The Deep Learning models are capable of quantifying the features as blood vessels, fluid drip, exudates, hemorrhages and micro aneurysms into different classes.

Suvajit Dutta, Bonthala C S Manideep, Muzamil Basha & Ronnie D. Caytiles proposed this paper.

Advantages : pre-processing of the images is done before training stage. Disadvantages : Image size is so high (2000x2000) which might not be idle for all the images from the available datasets and might also affect the clarity of the images which in-turn affects its accuracy.

7. Detection of Diabetic Retinopathy and Maculopathy in Eye Fundus Images Using Deep Learning and Image Augmentation. (2019)

This paper presents a novel diabetic retinopathy automatic detection in retinal images by implementing efficient image processing and deep learning techniques. Besides diabetic retinopathy detection, the developed system integrates a novel detection of maculopathy into one detection system. Maculopathy is the damage to the macula, the eye part that is responsible for central vision.

Sarni Suhaila Rahim, Vasile Palade, Ibrahim Almakky & Andreas Holzinger proposed this paper.

Advantages : Using data augmentation and image classification technique together to yield high accuracy.

Disadvantages : No actual implementation, thus no actual fact to backup the consistency of accuracy that is yielded.

8. Issues in Training a Convolutional Neural Network Model for Image Classification. (2019)

The paper summarizes the results of training the deep learning model using CNN on publicly available datasets of cats and dogs. Finally the paper discusses various methods such as data augmentation, regularization, dropout, etc to prevent the CNN model from overfitting problem. The paper will also help beginners to have a broad comprehension of CNN and motivate them to venture in this field.

Soumya Joshi, Dhirendra Kumar Verma, Gaurav Saxena & Amit Payaye propsed this paper.

Advantages : uses CNN to effectively classify two huge datasets.

Disadvantages : Not related to DR.

9. Deep Learning Methods for Underwater Target Feature Extraction and Recognition. (2020)

In this paper, a method for feature extraction and identification of underwater noise data based on CNN and ELM is proposed. An automatic feature extraction method of underwater acoustic signals is proposed using depth convolution network. An underwater target recognition classifier is based on extreme learning machine. 18 Although convolution neural networks can execute both feature extraction and classification, their function mainly relies on a full connection layer, which is trained by gradient descent-based; the generalization ability is limited and suboptimal, so an extreme learning machine (ELM) was used in classification stage.

Gang Hu, Kejun Wang, Yuan Peng, Mengran Qiu, Jianfei Shi & Liangliang Liu proposed this paper.

Advantages : very high accuracy and best method for classifying images. Disadvantages : Not related to disease identification.

10. Diabetic kidney disease and diabetic retinopathy: the ominous duo. (2020)

The association of diabetic microvascular complications such as diabetic retinopathy (DR) and diabetic kidney disease (DKD) with mortality in populations is not clear. To examine the association of DR and DKD separately and jointly with all cause and cardiovascular disease (CVD) mortality in a multiethnic Asian population.

Vijay Viswanathan proposed this paper.

Advantages : Provides good accuracy in finding the diseases and clearly explains the relationship between the kidney and retinopathy diseases.

Disadvantages : Does not provide an actual solution for classifying diabetic retinopathy disease from other eye diseases

CHAPTER 3
SYSTEM ARCHITECTURE

System architecture is the conceptual model that defines the structure, behavior and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. A system architecture can comprise system components, the expand systems developed, that will work together to implement the overall system.

3.1 EXISTING SYSTEM

The drawbacks are:

- Lacks accuracy
- Performance is less

3.2 PROPOSED SYSTEM

Features of the CNN

CNN extract these photos from information sets and arrange them to proportions before codification, as mentioned. Two-step process of extraction and organization along with background subtraction portion is handled by Con2D and max pooling. This both uncovers and eliminates the functionalities which are visible. Every filter is performed through an associated layer. All of them are segregated based on the tasks found.

This then offers a theoretical study of Photographs of various types of DED, collected as samples. The form and texture-oriented attributes are the item's crucial elements. Feature learning with Dnns and rapid state recognition have shown tremendous success in the diagnosis of numerous Neuropathy situations.

System Style

Firstly, we must enter a photograph into the system. Following that, the item selection occurs. Afterwards, the algorithm eliminates the traits and assesses which ailments are existent in the sickness database. Ultimately, it returns to the feature option component to indulge in more in-depth study. Further comparison of the ailments obtained shows the disease that best matches the parameters. That's how a Dcnn operates.

3.3 WORKING STEPS

1. Image to be processed are selected from the disease database.

2. The images are sent for feature extraction.

3. An input image is given for feature selection.

4. Cross referring to the extraction process, the image is sent for processing further.

5. Deep learning module is executed.

6. Disease comparison is done to confirm whether the input image has the disease or not.

7. Output is shown.

3.4 PROPOSED SYSTEM ARCHITECTURE

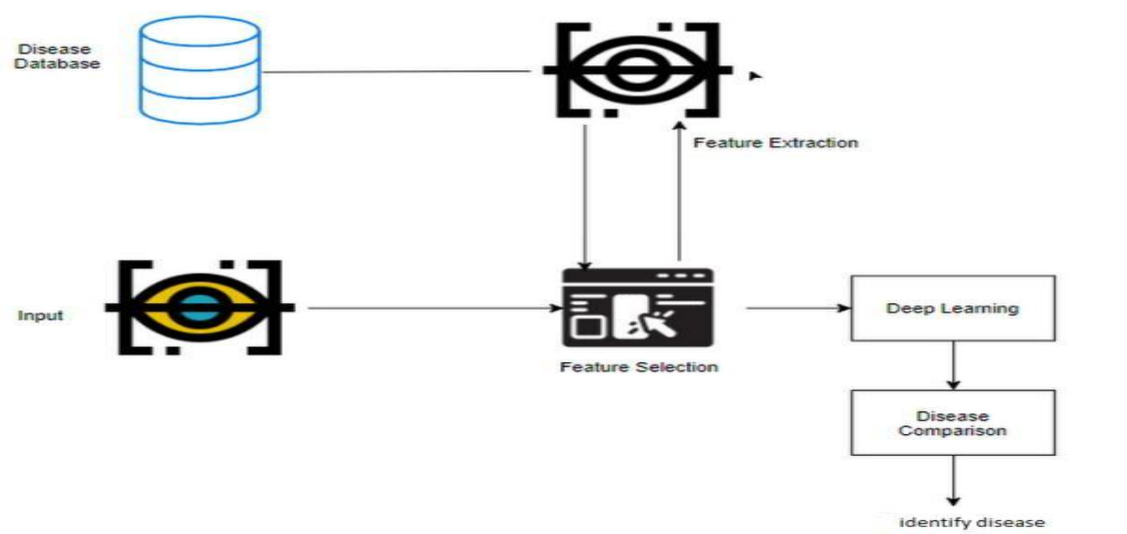


Fig 3.1 System Architecture

3.5 SYSTEM REQUIREMENTS

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified software or project. These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications.

3.5.1 SOFTWARE REQUIREMENTS

The software requirements are the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team's progress throughout the development activity.

- Operating system : Windows 10
- Languages used : Python
- Software Package : Anaconda 3

3.5.2 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the systems do and not how it should be implemented.

Table 3.1 Hardware Requirements

- Hard disk : 1 TB (minimum)
- Platform : IA-32(Windows Package Only)
- Ram : 8 GB RAM
- Processor : Pentium IV/III
- Processor speed : Minimum 1.99 GHZ

CHAPTER 4
PROJECT DESCRIPTION

Implementation is the phase where vision and plans become reality. This is the logical conclusion after evaluating, deciding, planning and finding the financial resources of a project. Technical implementation is one part of executing a project.

4.1 MODULES

- Module 1 - ManualNet architecture.
- Module 2 - AlexNet architecture.
- Module 3 - LeNet architecture.

4.1.1 MANUALNET ARCHITECTURE

This module is designed manually to train and test the diabetic retinopathy'sretinal datasets that are taken from the retinal database 'STARE'.The module is called the ManualNet and it gives the accuracy of 98.44%. ManualNet is an attempt to create our own image classification architecture with AlexNet and LeNet as its base.This module is also compared with the same existing image classification architectures that are coded in next modules.The system architecture in the above slide is the flow diagram for this module.

4.1.2 ALEXNET ARCHITECTURE

This module comprises the AlexNet architecture.AlexNet architecture consists of 8 layers.The module is trained and tested with the same dataset used in module 1 and the output of its accuracy is compared with ManualNet in module 1 and LeNet in module 3.The accuracy of this module is 89.70% which is less than ManualNet and LeNet. The architecture of AlexNet is shown in the next slide.

4.1.3 LENET ARCHITECTURE

This module comprises the LeNet architecture.LeNet architecture consists of 5 layers. The module is trained and tested with the same dataset as used in module 1 and the output of its accuracy is compared with ManualNet in module 1 and AlexNet in module 2.The accuracy of this module is 96.80% which is the highest accuracy when compared with ManualNet and AlexNet.The architecture of LeNet is shown in the next slide.

CHAPTER 5
RESULTS AND DISCUSSION

In the proposed system, we train all the three architectures with the same given datasets and compare them in terms of their respective accuracies and losses. The architecture with maximum average accuracy with least loss is considered the best architecture to yield the output of whether diabetic retinopathy disease is present in the given input image or not.

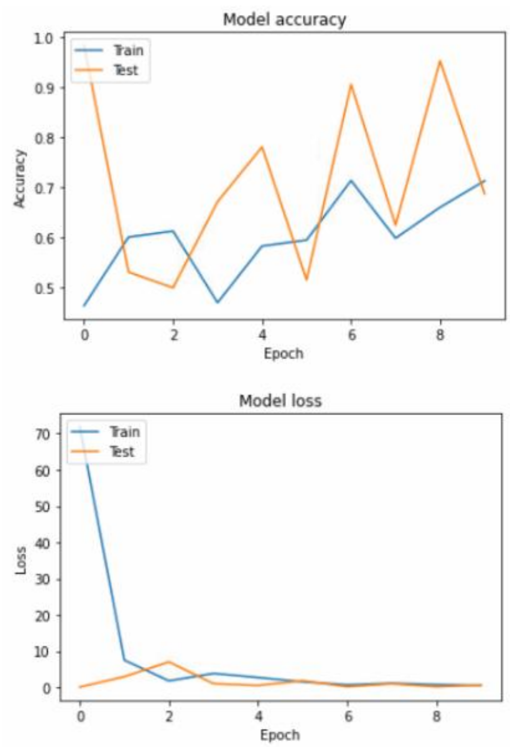


Figure 5.1 Graph Results of ManualNet

The figure 5.1 discuss the graph results of ManualNet architecture. There are two graphs in which x-axis is constant (epoch) while the y-axis is accuracy in the first graph and loss in the second graph. Both these graphs indicate the accuracy and loss variations of each train and test datasets that are given as inputs to the ManualNet architecture during each iteration (epoch), thus finding out the pattern of accuracy and loss throughout the process.

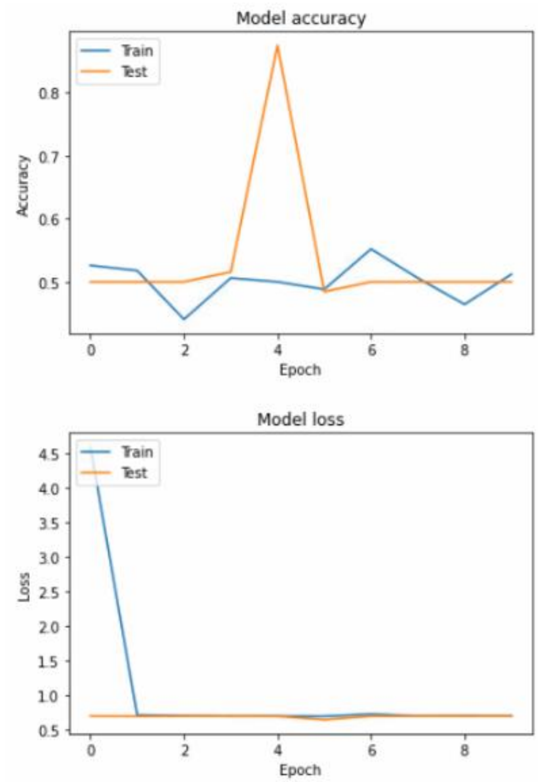


Figure 5.2 Graph Results of AlexNet

The figure 5.2 discuss the graph results of AlexNet architecture. There are two graphs in which x-axis is constant (epoch) while the y-axis is accuracy in the first graph and loss in the second graph. Both these graphs indicate the accuracy and loss variations of each train and test datasets that are given as inputs to the AlexNet architecture during each iteration (epoch), thus finding out the pattern of accuracy and loss throughout the process.

CHAPTER 6
CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

Our work focused on how the images from given data set (trained dataset) and past data sets are used to predict the pattern of diabetic retinopathy diseases using CNN model. This brings some of the following insights about diabetic retinopathy disease prediction. The major benefit of the CNN classification framework is the ability to classify images automatically.

In this study, we have discussed the overview of methodologies for detecting the abnormalities in diabetic retinopathy images which includes collection of retinopathy image data set, preprocessing techniques, feature extraction techniques and classification schemes.

6.2 FUTURE WORK

In the future works, we will satisfy the medical department requirements, that are necessary to automate the detecting of diabetic retinopathy disease for eligibility process (real time). We will also deploy our working model in cloud for better access to any medical department at anytime. The viability of the project can also be increased by this process.

By comparing the specificity of the 3 architectures used, it can be more clearly defined that one architecture stands out from others.

APPENDICES

APPENDIX 1

SAMPLE CODE : MANUALNET ARCHITECTURE

```
import os
import numpy as np # linear algebra
import matplotlib.pyplot as plt
# Dl framwork - tensorflow, keras a backend
import tensorflow as tf
import tensorflow.keras.backend as K
from tensorflow.keras.models import Model, Sequential
from tensorflow.keras.layers import Input, Dense, Flatten, Dropout, BatchNormalization
from tensorflow.keras.layers import Conv2D, SeparableConv2D, MaxPool2D, LeakyReLU, Activation
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.callbacks import ModelCheckpoint, ReduceLROnPlateau, EarlyStopping
from IPython.display import display
from os import listdir
from os.path import isfile, join
from PIL import Image
import glob
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.layers import Convolution2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Flatten
```

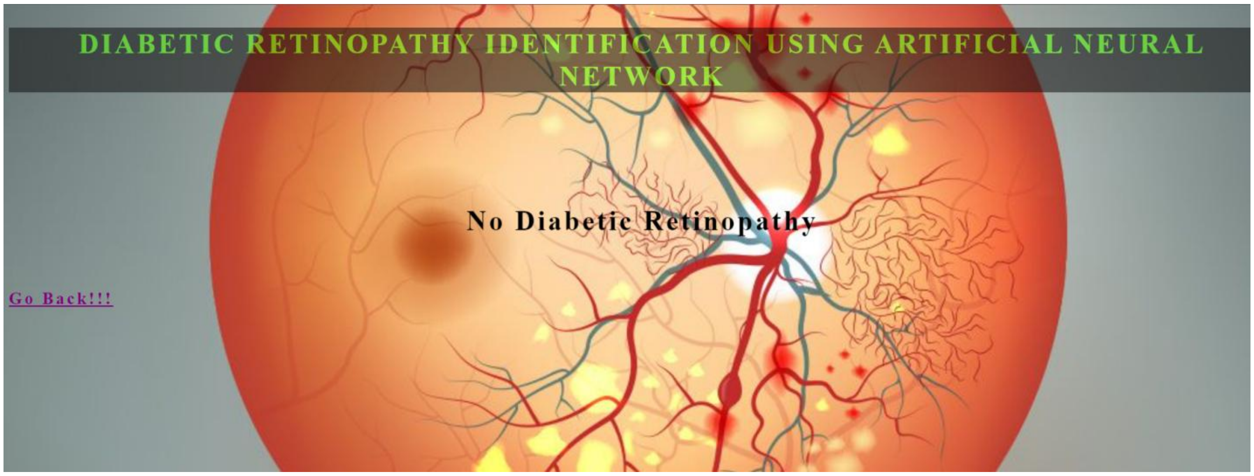
```
from tensorflow.keras.layers import Dense
import warnings
warnings.filterwarnings('ignore')
dir_name_train_No_DR = 'dataset/Train/No_DR'
dir_name_train_Severe = 'dataset/Train/Severe'
def plot_images(item_dir, n=6):
    all_item_dir = os.listdir(item_dir)
    item_files = [os.path.join(item_dir, file) for file in all_item_dir][:n]

    plt.figure(figsize=(80, 40))
    for idx, img_path in enumerate(item_files):
        plt.subplot(3, n, idx+1)
        img = plt.imread(img_path)
        plt.imshow(img, cmap='gray')
        plt.axis('off')

    plt.tight_layout()
    def Images_details_Print_data(data, path):
        print(" ====== Images in: ", path)
        for k, v in data.items():
            print("%s:\t%s" % (k, v))

    def Images_details(path):
        files = [f for f in glob.glob(path + "**/*.*", recursive=True)]
        data = {}
        data['images_count'] = len(files)
        data['min_width'] = 10**100  # No image will be bigger than that
```

```
    data['max_width'] = 0
    data['min_height'] = 10**100  # No image will be bigger than that
    data['max_height'] = 0


    for f in files:
        im = Image.open(f)
        width, height = im.size
        data['min_width'] = min(width, data['min_width'])
        data['max_width'] = max(width, data['max_height'])
        data['min_height'] = min(height, data['min_height'])
        data['max_height'] = max(height, data['max_height'])

    Images_details_Print_data(data, path)
print("")
print("Trainned data for No_DR:")
print("")
Images_details(dir_name_train_No_DR)
print("")
plot_images(dir_name_train_No_DR, 10)
print("")
print("Trainned data for Severe:")
print("")
Images_details(dir_name_train_Severe)
print("")
plot_images(dir_name_train_Severe, 10)
Classifier=Sequential()
```

```
Classifier.add(Convolution2D(32,(3,3),input_shape=(512,512,3),activation='relu'))
Classifier.add(MaxPooling2D(pool_size=(2,2)))
Classifier.add(Flatten())
Classifier.add(Dense(38, activation='relu'))
Classifier.add(Dense(2, activation='softmax'))
Classifier.compile(optimizer='rmsprop',loss='categorical_crossentropy',metrics=['accuracy'])
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)
training_set=train_datagen.flow_from_directory('dataset/Train',target_size=(512,512),batch_size=32,class_mode='categorical')
test_set=test_datagen.flow_from_directory('dataset/Test',target_size=(512,512),batch_size=32,class_mode='categorical')
img_dims = 150
epochs = 10
batch_size = 32
#### Fitting the model
history = Classifier.fit_generator(
        training_set, steps_per_epoch=training_set.samples // batch_size,
        epochs=epochs,
        validation_data=test_set,validation_steps=test_set.samples // batch_size)
def graph():
    #Plot training & validation accuracy values
    plt.plot(history.history['accuracy'])
    plt.plot(history.history['val_accuracy'])
```

```
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()

# Plot training & validation loss values
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```

SCREENSHOT: (from Django Platform were we deployed our model)



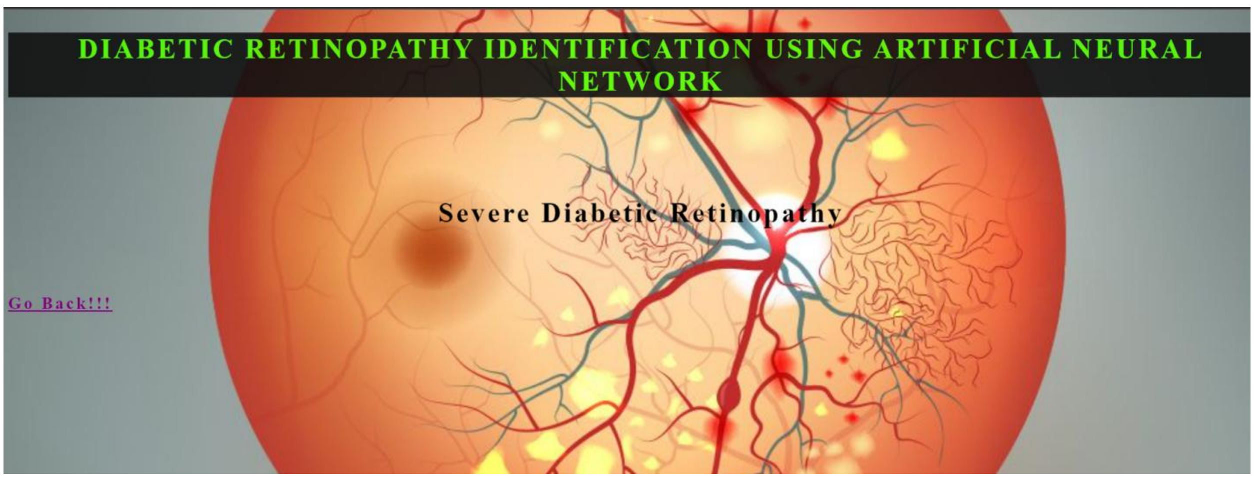


Figure 7.1 Pycharm Output Screenshots

APPENDIX 2

PAPER PUBLICATION

Detection Of Diabetic Retinopathy Using Alexnet and Lenet CNN Models

Jananee V(Assistant Professor)[1], Shanthalakshmi M(Assistant Professor)[2], Sherine Glory J(Assistant Professor)[3], Vaidhya G K (Research scholar)[4], Nikghamanth Seshadri Narayanan(student)[5] and Neya Babu Thayil(student)[6].

[1]Rajalakshmi Engineering College, Rajalakshmi nagar, Thandalam, Chennai-602105, Tamil Nadu.

[a)]jananee.v@rajalakshmi.edu.in

[b)]shanthalakshmi.m@rajalakshmi.edu.in

[c)]sherinegloryj@gmail.com

[d)]vaidhyagk@gmail.com

[e)]nikghamanth.ss.2018.cse@rajalakshmi.edu.in
[f)]neyababu.t.2018.cse@rajalakshmi.edu.in

Abstract-- Diabetic Retinopathy is a condition that is caused by excessive glycemia. It can often be tough to tell the variation among both DR and fundus photographs. To avoid difficulties, it is crucial to acknowledge. We can detect many Diabetic Eye Disease illnesse using Cnns. It also detects the colors and patterns of sores and matches them to relevant conditions during medical diagnosis, which is similar to human decision-making. The Django web framework showcases the output. To determine the most efficient and accurate categorization of Diagnosed images, researchers use many related images as input into convolutional semantic networks.

INTRODUCTION

TO WAGE THE THERAPY, IT IS IMPORTANT TO IDENTIFY THE ILLNESS. IN SIMPLIFIED WORDS, COMPLICATION RESULTS IN PROBLEMS. IT LIKEWISE APPLIES TO DIABETIC RETINOPATHY DISEASE. COMPLICATED DIABETIC PERSON RETINOPATHY PHOTOS WITH FUNDUS PICTURES IS An OPPORTUNITY. SO IT'S IMPORTANT TO SEPARATE DIABETIC RETINOPATHY IMAGES FROM FUNDUS PICTURES. WE CAN EFFICIENTLY FULL THIS CRUCIAL TASK BY APPLYING A CONVOLUTIONAL NEURAL NETWORK

A CONVOLUTIONAL SEMANTIC NETWORK CAN TAKE IMAGES AS WELL AS SET APART ONE FROM THE OTHER. THE CNN CANISTER THEN BE USED TO FIND LOTS OF DIABETIC RETINOPATHY ILLNESSES. THE REALITY THAT ITS CONTAINER

CAPTURES THE COLORS AND STRUCTURES OF LESIONS CERTAIN TO RESPECTIVE DISEASES DURING MEDICAL DIAGNOSIS IS AN FRINGE BENEFIT. IT STANDS FOR THE CAPABILITY TO MAKE DECISIONS IN HUMAN BEINGS.

A CNN IS AN A1 WHICH MIMICS THE HUMAN KN0WLEDGE. BUT HUMAN INTELLECT ALS0 HAS ITS LIMITATI0NS. THUS, WE CAN C0NCLUDE THAT THE EXPERT SYSTEM IS REFINED AND UPGRADED BRAINPOWER. IT IS CONVINCING TO C0UNT ON THE AI-- A CSN.

THE OBJECTIVE IS TO DEVEL0P A DESIGN FOR DIABETIC EYE DISEASE (DED) PICTURE ANALYSIS BY CLASSIFICATI0N ALGORITHMS FOR P0SSIBLY ARRIVING THE CAUSE AND THE KIND OF SUPREME PRECISI0N BY CONTRASTING THE 0RIGINAL STYLE.

Associated Works

0phthalmoscope

An 0phthalmoscope can obtain, store and analyze images of the occulations. Based on analog pictures, retinal photo handling takes place as well as concerned with the 0bservation of vessels in subcapsular images with flu0rescein [1]. This effulgent improves clarity of repository in the pic. They assist physicians to identify as well as determine. Yet, flu0rescein angi0graphy is an intrusive, thus a taxing procedure.

Technique Specific Attention Network

Technique/Modality-specific attention network (MSAN) makes use of fornix and also 0CT photos all at once to achieve 0phthalmology. The sound in them may disturb the R01 extraction. So, Gaussian filter with bit dimensions are applied for deblurring, as well as presented in-depth understanding methods for flavimactulas picture categorization established upon DED neurilemma discovery. Nevertheless, AlexNet CNN and LeNet models are not implemented for yielding efficient results.

Subcapsular Pics

Supplies shades or clear images of the retina. They are mostly automated, have several benefits in contrast to its precursor. Computerized retinal images offer structured, well-organized and retrievable outputs that are accessible as well as complaisant for picture amplification[2]. Nonetheless, occasionally distorted images may occur.

PROPOSED STRUCTURE

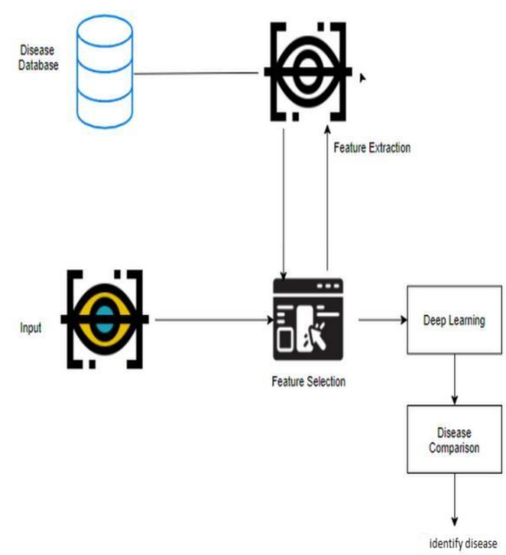
Features of the CSN

CSN extract these photos from information sets and arrange them to proportions before codification, as mentioned. Two-step process of extraction and organization along with background subtraction portion is handled by Con2D and max pooling. This both uncovers and eliminates the functionalities which are visible. Every filter is performed through an associated layer. All of them are segregated based on the tasks found.

This then offers a theoretical study of Photographs of various types of DED, collected as samples. The form and texture-oriented attributes are the item's crucial elements. Feature learning with Dnns and rapid state recognition have shown tremendous success in the diagnosis of numerous Neuropathy situations.

System Style

As shown in "Fig 1." Firstly, we must enter a photograph into the system. Following that, the item selection occurs. Afterwards, the algorithm eliminates the traits and assesses which ailments are existent in the sickness database. Ultimately, it returns to the feature option component to indulge in more in-depth study. Further comparison of the ailments obtained shows the disease that best matches the parameters. That's how a Dcnn operates.



(A)

Development environment

In terms of hardware, it requires a Pentium IV/III processor. It needs a minimum 80 GB hard disk and a minimum of 2 GB RAM.

In terms of software, it needs Windows/Linux Operating system as well as Anaconda with Jupyter Notebook simulation tool.

This twofold requirement is needed to develop the environment for Dcnn.

Figures

```
WARNING:tensorflow:From C:\Users\nytma\AppData\Local\Temp/ipykernel_11420/112350279.py:5: Model.fit_generator (from tensorflow.
python.keras.engine.training) is deprecated and will be removed in a future version.
Instructions for updating:
Please use Model.fit, which supports generators.
Epoch 1/10
6/6 [==============================] - 20s 3s/step - loss: 71.8827 - accuracy: 0.4643 - val_loss: 0.1003 - val_accuracy: 0.9844
Epoch 2/10
6/6 [==============================] - 19s 3s/step - loss: 7.5098 - accuracy: 0.6012 - val_loss: 2.9889 - val_accuracy: 0.5312
Epoch 3/10
6/6 [==============================] - 20s 3s/step - loss: 1.8257 - accuracy: 0.6131 - val_loss: 7.0325 - val_accuracy: 0.5000
Epoch 4/10
6/6 [==============================] - 17s 3s/step - loss: 3.8638 - accuracy: 0.4702 - val_loss: 1.0434 - val_accuracy: 0.6719
Epoch 5/10
6/6 [==============================] - 16s 3s/step - loss: 2.7475 - accuracy: 0.5833 - val_loss: 0.5480 - val_accuracy: 0.7812
Epoch 6/10
6/6 [==============================] - 18s 3s/step - loss: 1.5556 - accuracy: 0.5952 - val_loss: 1.8557 - val_accuracy: 0.5156
Epoch 7/10
6/6 [==============================] - 17s 3s/step - loss: 0.7143 - accuracy: 0.7143 - val_loss: 0.1904 - val_accuracy: 0.9062
Epoch 8/10
6/6 [==============================] - 19s 3s/step - loss: 1.1384 - accuracy: 0.5990 - val_loss: 1.0173 - val_accuracy: 0.6250
Epoch 9/10
6/6 [==============================] - 26s 4s/step - loss: 0.7847 - accuracy: 0.6607 - val_loss: 0.1887 - val_accuracy: 0.9531
Epoch 10/10
6/6 [==============================] - 54s 9s/step - loss: 0.5585 - accuracy: 0.7135 - val_loss: 0.6685 - val_accuracy: 0.6875
```

FIGURE 1. ManualNet image processing

```
Epoch 1/10
6/6 [==============================] - 71s 12s/step - loss: 4.6040 - accuracy: 0.5260 - val_loss: 0.6937 - val_accuracy: 0.5000
Epoch 2/10
6/6 [==============================] - 46s 8s/step - loss: 0.7095 - accuracy: 0.5179 - val_loss: 0.6915 - val_accuracy: 0.5000
Epoch 3/10
6/6 [==============================] - 49s 8s/step - loss: 0.6986 - accuracy: 0.4405 - val_loss: 0.6942 - val_accuracy: 0.5000
Epoch 4/10
6/6 [==============================] - 49s 8s/step - loss: 0.6929 - accuracy: 0.5060 - val_loss: 0.6927 - val_accuracy: 0.5156
Epoch 5/10
6/6 [==============================] - 54s 9s/step - loss: 0.6935 - accuracy: 0.5000 - val_loss: 0.6922 - val_accuracy: 0.8750
Epoch 6/10
6/6 [==============================] - 67s 11s/step - loss: 0.6911 - accuracy: 0.4881 - val_loss: 0.6403 - val_accuracy: 0.4844
Epoch 7/10
6/6 [==============================] - 71s 12s/step - loss: 0.7241 - accuracy: 0.5521 - val_loss: 0.6936 - val_accuracy: 0.5000
Epoch 8/10
6/6 [==============================] - 65s 11s/step - loss: 0.6934 - accuracy: 0.5060 - val_loss: 0.6950 - val_accuracy: 0.5000
Epoch 9/10
6/6 [==============================] - 63s 10s/step - loss: 0.6979 - accuracy: 0.4643 - val_loss: 0.6935 - val_accuracy: 0.5000
Epoch 10/10
6/6 [==============================] - 64s 11s/step - loss: 0.6931 - accuracy: 0.5119 - val_loss: 0.6939 - val_accuracy: 0.5000
```

FIGURE 2. AlexNet image processing

```
WARNING:tensorflow:From C:\Users\nytma\AppData\Local\Temp/ipykernel_6312/3080110501.py:3: Model.fit_generator (from tensorflow.python.keras.engine.training) is deprecated and will be removed in a future version.
Instructions for updating:
Please use Model.fit, which supports generators.
Epoch 1/50
6/6 [==============================] - 11s 2s/step - loss: 1.1050 - accuracy: 0.5119 - val_loss: 0.3912 - val_accuracy: 0.9844
Epoch 2/50
6/6 [==============================] - 9s 1s/step - loss: 0.7287 - accuracy: 0.5893 - val_loss: 0.5508 - val_accuracy: 0.6719
Epoch 3/50
6/6 [==============================] - 9s 1s/step - loss: 0.6549 - accuracy: 0.6131 - val_loss: 0.3146 - val_accuracy: 0.9219
Epoch 4/50
6/6 [==============================] - 10s 2s/step - loss: 0.6849 - accuracy: 0.6310 - val_loss: 0.2940 - val_accuracy: 0.9531
Epoch 5/50
6/6 [==============================] - 9s 2s/step - loss: 0.6459 - accuracy: 0.5952 - val_loss: 0.3513 - val_accuracy: 0.8125
Epoch 6/50
6/6 [==============================] - 10s 2s/step - loss: 0.5999 - accuracy: 0.6406 - val_loss: 0.2502 - val_accuracy: 0.9688
Epoch 7/50
6/6 [==============================] - 10s 2s/step - loss: 0.5811 - accuracy: 0.6667 - val_loss: 0.2660 - val_accuracy: 0.9375
Epoch 8/50
6/6 [==============================] - 10s 2s/step - loss: 0.5660 - accuracy: 0.7262 - val_loss: 0.2485 - val_accuracy: 0.8594
Epoch 9/50
6/6 [==============================] - 9s 2s/step - loss: 0.5743 - accuracy: 0.6726 - val_loss: 0.1957 - val_accuracy: 0.9688
Epoch 10/50
6/6 [==============================] - 9s 2s/step - loss: 0.5774 - accuracy: 0.6845 - val_loss: 0.2540 - val_accuracy: 0.9375
Epoch 11/50
6/6 [==============================] - 10s 2s/step - loss: 0.6081 - accuracy: 0.6726 - val_loss: 0.2838 - val_accuracy: 0.9531
Epoch 12/50
6/6 [==============================] - 9s 2s/step - loss: 0.5383 - accuracy: 0.6964 - val_loss: 0.2177 - val_accuracy: 0.9219
Epoch 13/50
6/6 [==============================] - 9s 2s/step - loss: 0.5710 - accuracy: 0.7083 - val_loss: 0.2391 - val_accuracy: 0.9219
Epoch 14/50
6/6 [==============================] - 9s 1s/step - loss: 0.7086 - accuracy: 0.6845 - val_loss: 0.2712 - val_accuracy: 0.9219
Epoch 15/50
6/6 [==============================] - 11s 2s/step - loss: 0.5804 - accuracy: 0.6964 - val_loss: 0.1827 - val_accuracy: 0.9688
Epoch 16/50
6/6 [==============================] - 9s 2s/step - loss: 0.5246 - accuracy: 0.7262 - val_loss: 0.1727 - val_accuracy: 0.9688
Epoch 17/50
6/6 [==============================] - 9s 2s/step - loss: 0.5164 - accuracy: 0.7679 - val_loss: 0.2828 - val_accuracy: 0.8750
Epoch 18/50
6/6 [==============================] - 10s 2s/step - loss: 0.5270 - accuracy: 0.7031 - val_loss: 0.3370 - val_accuracy: 0.8438
Epoch 19/50
6/6 [==============================] - 9s 2s/step - loss: 0.5348 - accuracy: 0.7143 - val_loss: 0.0477 - val_accuracy: 0.9844
Epoch 20/50
6/6 [==============================] - 9s 1s/step - loss: 0.5902 - accuracy: 0.7024 - val_loss: 0.2586 - val_accuracy: 0.9219
Epoch 21/50
6/6 [==============================] - 9s 1s/step - loss: 0.5177 - accuracy: 0.7440 - val_loss: 0.1109 - val_accuracy: 0.9844
Epoch 22/50
6/6 [==============================] - 9s 1s/step - loss: 0.5230 - accuracy: 0.7381 - val_loss: 0.1501 - val_accuracy: 0.9688
Epoch 23/50
6/6 [==============================] - 9s 2s/step - loss: 0.5323 - accuracy: 0.7619 - val_loss: 0.2533 - val_accuracy: 0.9375
Epoch 24/50
6/6 [==============================] - 9s 2s/step - loss: 0.4920 - accuracy: 0.7381 - val_loss: 0.1152 - val_accuracy: 0.9844
Epoch 25/50
6/6 [==============================] - 10s 2s/step - loss: 0.5370 - accuracy: 0.7292 - val_loss: 0.1752 - val_accuracy: 0.9531
Epoch 26/50
6/6 [==============================] - 11s 2s/step - loss: 0.5269 - accuracy: 0.6964 - val_loss: 0.2245 - val_accuracy: 0.9688
Epoch 27/50
6/6 [==============================] - 11s 2s/step - loss: 0.5386 - accuracy: 0.7262 - val_loss: 0.1781 - val_accuracy: 0.9531
Epoch 28/50
```

(a)

```
Epoch 29/50
6/6 [==============================] - 9s 2s/step - loss: 0.4967 - accuracy: 0.7440 - val_loss: 0.0928 - val_accuracy: 0.9688
Epoch 30/50
6/6 [==============================] - 9s 2s/step - loss: 0.5108 - accuracy: 0.7500 - val_loss: 0.1180 - val_accuracy: 0.9531
Epoch 31/50
6/6 [==============================] - 10s 2s/step - loss: 0.5239 - accuracy: 0.7381 - val_loss: 0.1477 - val_accuracy: 0.9688
Epoch 32/50
6/6 [==============================] - 9s 2s/step - loss: 0.4956 - accuracy: 0.7560 - val_loss: 0.2350 - val_accuracy: 0.8906
Epoch 33/50
6/6 [==============================] - 9s 1s/step - loss: 0.4916 - accuracy: 0.7798 - val_loss: 0.1323 - val_accuracy: 0.9531
Epoch 34/50
6/6 [==============================] - 10s 2s/step - loss: 0.4435 - accuracy: 0.8073 - val_loss: 0.3497 - val_accuracy: 0.7969
Epoch 35/50
6/6 [==============================] - 9s 1s/step - loss: 0.4847 - accuracy: 0.8036 - val_loss: 0.1460 - val_accuracy: 0.9375
Epoch 36/50
6/6 [==============================] - 9s 2s/step - loss: 0.4860 - accuracy: 0.7619 - val_loss: 0.1226 - val_accuracy: 0.9531
Epoch 37/50
6/6 [==============================] - 9s 1s/step - loss: 0.4647 - accuracy: 0.7619 - val_loss: 0.5700 - val_accuracy: 0.6562
Epoch 38/50
6/6 [==============================] - 9s 1s/step - loss: 0.4763 - accuracy: 0.7798 - val_loss: 0.1439 - val_accuracy: 0.9375
Epoch 39/50
6/6 [==============================] - 11s 2s/step - loss: 0.4925 - accuracy: 0.6964 - val_loss: 0.2031 - val_accuracy: 0.9062
Epoch 40/50
6/6 [==============================] - 11s 2s/step - loss: 0.4438 - accuracy: 0.7917 - val_loss: 0.1498 - val_accuracy: 0.9531
Epoch 41/50
6/6 [==============================] - 11s 2s/step - loss: 0.4237 - accuracy: 0.8155 - val_loss: 0.1367 - val_accuracy: 0.9375
Epoch 42/50
6/6 [==============================] - 9s 2s/step - loss: 0.4928 - accuracy: 0.7500 - val_loss: 0.1579 - val_accuracy: 0.9219
Epoch 43/50
6/6 [==============================] - 9s 2s/step - loss: 0.4212 - accuracy: 0.8155 - val_loss: 0.0999 - val_accuracy: 0.9531
Epoch 44/50
6/6 [==============================] - 9s 2s/step - loss: 0.4245 - accuracy: 0.8155 - val_loss: 0.1742 - val_accuracy: 0.9375
Epoch 45/50
6/6 [==============================] - 10s 2s/step - loss: 0.4253 - accuracy: 0.8333 - val_loss: 0.1510 - val_accuracy: 0.9375
Epoch 46/50
6/6 [==============================] - 9s 2s/step - loss: 0.4300 - accuracy: 0.7917 - val_loss: 0.1665 - val_accuracy: 0.9375
Epoch 47/50
6/6 [==============================] - 11s 2s/step - loss: 0.4381 - accuracy: 0.7976 - val_loss: 0.0878 - val_accuracy: 0.9531
Epoch 48/50
6/6 [==============================] - 11s 2s/step - loss: 0.4377 - accuracy: 0.8281 - val_loss: 0.2219 - val_accuracy: 0.9062
Epoch 49/50
6/6 [==============================] - 11s 2s/step - loss: 0.3884 - accuracy: 0.8155 - val_loss: 0.1471 - val_accuracy: 0.9531
Epoch 50/50
6/6 [==============================] - 10s 2s/step - loss: 0.3994 - accuracy: 0.8021 - val_loss: 0.1860 - val_accuracy: 0.9062
```

(b)

FIGURE 3. LeNet Image processing

Color Figures

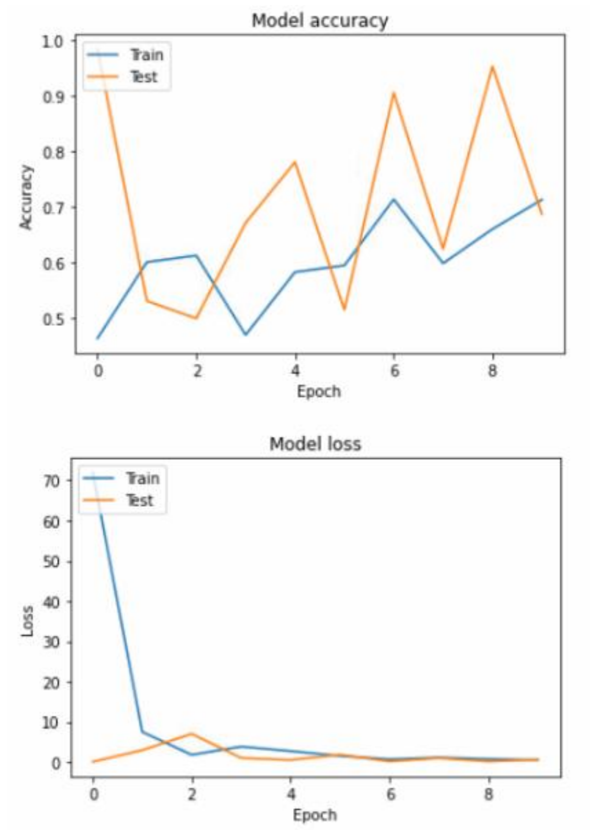


FIGURE 4. ManualNet Graphical Output

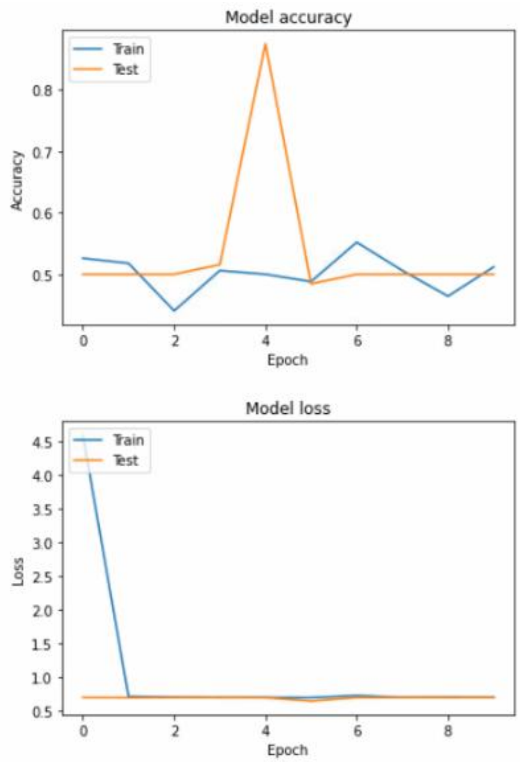
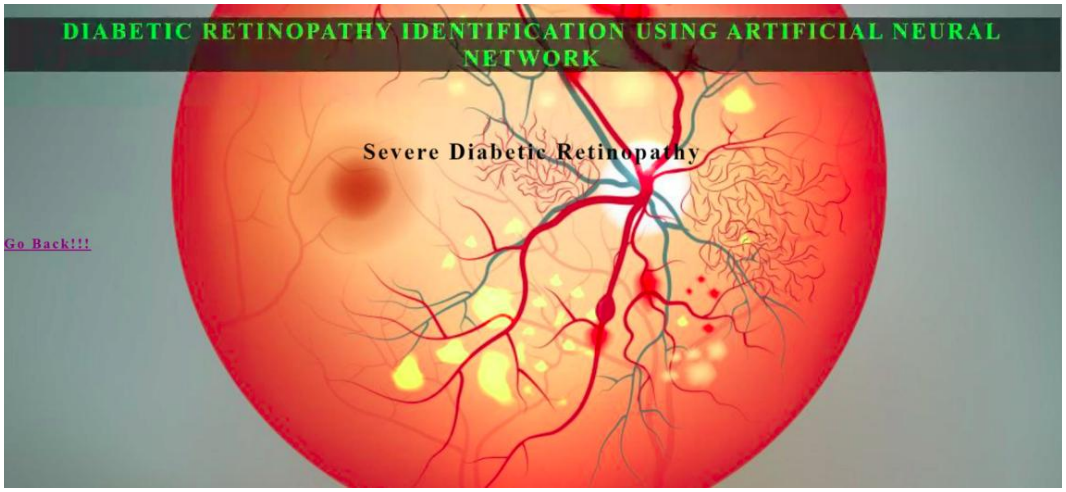
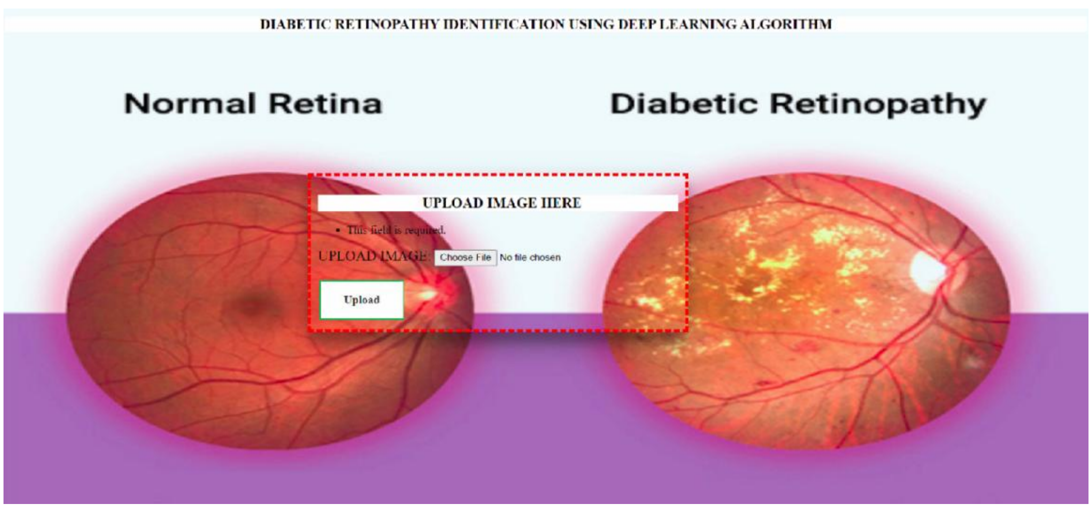


FIGURE 5. AlexNet Graphical Output

(a)



(b)



(c)

FIGURE 6. pycharm Output Screenshots

CONCLUSION

THE PROPOSED NETWORK IS CUSTOMIZED TO FORETELL THE TREND OF DR DISEASES BY USING IMAGES FROM ACQUIRED DATA (TRA1NED DATASET) AND ALSO PRIOR DATA SETS. SEVERAL OF THE DR STATE ASSUMPTIONS AND INTERPRETATIONS ARE PROVIDED THROUGH THIS. A MAJOR BENEFIT OF THE AFF0REMENTIONED METH0D IS THE ABILITY TO 1NSTANTLY DISTINGU1SH 1MAGES.

WE C0VERED AN 0VERALL V1EW OF METH0DOLOGIES REGARDING THE ABILITY TO DETECT ISSUES IN DR PH0TOS, INCLUDING PR0LIFERAT1VE DR DATA SET C0LLECT1ON, DATA TRANSF0RMATION STRATEG1ES, COMP0NENT SEGMENTATION TECHN1QUES, AND CATEG0RISATION PROPOSALS IN THIS PAPER.

WE W1LL PLEASE THE MED1CAL DEPARTMENT REQU1REMENTS F0R THE QUALIF1CATION PR0CEDURE BY AUT0MATING THE DISC0VERY OF DIABETIC RETIN0PATHY D1SEASE IN THE FUTURE WORKS (REAL-T1ME).
WE CAN ALSO USE IT ON THE NET.

ACKN0WLEDGMENTS

WE W0ULD LIKE TO CONVEY OUR HEARTFELT GRAT1TUDE TO OUR ADVISOR AND TEACHER MRS. JANANEE AND H0D MAM TOGETHER WITH OUR TASK C00RDINATORS WH0 SUSTA1NED 0UR SUGGESTION AND PERSPECT1VE THR0UGHOUT THE TASK, SUGGEST1NG ENHANCEMENTS WHEREVER NECESSARY.

REFERENCES

1. Sehrish Qummar, Fiaz Gul Khan, Sajid Shah, Ahmad Khan, Shahaboddin Shamshirband, Zia Ur Rehman & Iftikhar Ahmed Khan, "A Deep Learning Ensemble Approach for Diabetic Retinopathy Detection"

   IEEE (2019)

2. GauravSaxena, Dhirendra KumarVerma, Amit Paraye, AlpanaRajan & AnilRawat, "Improved and robust deep learning agent for preliminary detection of Diabetic Retinopathy using public datasets" (Intelligence-Based-Medicine, Volumes 3-4, 2020)

3. Shu-I Pao, Hong-Zin Lin, Ke-Hung Chien, Ming-Cheng Tai, Jiann-Torng Chen & Gen-Min Lin, "Detection of Diabetic Retinopathy Using Bichannel Convolutional Neural Network" (Enrico Peiretti, 2020)

4. Mustapha Aatil, Mohamed Lachgar, Hamid Hrimech & Ali Kartit, "Diabetic Retinopathy Classification Using ResNet50 and VGG-16 Pretrained Networks" (IJCEDS, volume 1, 2021)

5. Rishi P.Singh, Michael J.Elman, Simran K.Singh, Anne E.Fung & Ivaylo Stoilov, "Advances in the treatment of diabetic retinopathy" *PubMed (*2019)

6. Suvajit Dutta, Bonthala CS Manideep, Syed Muzamil Basha, Ronnie D. Caytiles & N. Ch. S. N. Iyengar, "Classification of Diabetic Retinopathy Images by Using Deep Learning Models" *International Journal of Grid and Distributed Computing (*2019)

7. Sarni Suhaila Rahim, Vasile Palade, Ibrahim Almakky & Andreas Holzinger, "Detection of Diabetic Retinopathy and Maculopathy in Eye Fundus Images Using Deep Learning and Image Augmentation" *International cross-domain conference*, *UK* (2019)

8. Soumya Joshi, Dhirendra Kumar Verma, Gaurav Saxena & Amit Paraye, "Issues in Training a Convolutional Neural Network Model for Image Classification" *International Conference on Advances in Computing and Data Science* (2019).

9. Gang Hu, Kejun Wang, Yuan Peng, Mengran Qiu, Jianfei Shi & Liangliang Liu, "Deep Learning Methods for Underwater Target Feature Extraction and Recognition" (Rasit Koker, 2019)

10. Yann LeCun, Yoshua Bengio & Geoffrey Hinton, "Deep Learning Concepts", (Nature, 2020)

11. Vijay Viswanathan, "Diabetic kidney disease and diabetic retinopathy: the ominous duo" *International Journal of Diabetics in Developing Countries* (2020)

12. K Sathyavani, H Kothandan, M Jayaraman & V Viswanathan, "Direct costs associated with chronic kidney disease among type 2 diabetic patients in India" (indiajnephrol.org, 2019)

13. Amol Prataprao Bhatkar & G.U. Kharat, "Detection of Diabetic Retinopathy in Retinal Images using MLP classifier" *IEEE* (2019)

14. Yashal Shakti Kanungo, Bhargav Srinivasan & Savitha Choudhary, “Detecting Diabetic Retinopathy using Deep Learning” *IEEE* (2020)

15. Xianlong Zeng, Haiquan Chen, Yuan Luo & Wenbin Ye, “Automated Diabetic Retinopathy Detection Based on Binocular Siamese-like Convolutional Neural Network” *IEEE* (volume 7, 2019)

16. C Jayakumari, Vidhya Lavanya & E P Sumesh, “Automated Diabetic Retinopathy Detection and classification using ImageNet Convolutional Neural Network using Fundus Images” *IEEE* (2020)

17. Supriya Mishra, Seema Hanchate & Zia Saquib, “Diabetic Retinopathy Detection using Deep Learning” *IEEE* (2020)

18. Sara Hosseinzadeh Kassani, Peyman Hosseinzadeh Kassani, Michal J. Wesolowski, Kevin A. Schneider & Ralph Deters, “Breast Cancer Diagnosis with Transfer Learning and Global Pooling” *IEEE* (2019)

19. Seema Garg & Richard M. Davis, “Diabetic Retinopathy Screening Update” (American Diabetics Association, 2020)

20. Ramachandran Rajalakshmi, Vijayaraghavan Prathiba, Subramanian Arulmalar & Manoharan Usha, “Review of retinal cameras for global coverage of diabetic retinopathy screening” (Nature-eye, 2020)

CERTIFICATES

PANIMALAR ENGINEERING COLLEGE

A Christian Minority Institution-Jaisakthi Educational Trust
An Autonomous Institution & Affiliated to Anna University, Chennai
Approved by AICTE, New Delhi, Accredited by National Board of Accreditation(NBA)
Approved by UGC for 2(f) & 12(B) Status
Bangalore Trunk Road, Poonamallee, Chennai-600123.

Certificate of Participation

PECTEAM 2K22

This is to certify that Mr./Ms......Nikghamanth Seshadri Narayanan.................of ....CSE...............from Rajalakshmi Engineering College....has presented paper entitled ..Detection of Diabetic Retinopathy using Alexnet and Lenet CNN models........................... in the AICTE Sponsored 5th International Conference on Intelligent Computing (IConIC) held on 25th & 26th March,2022.

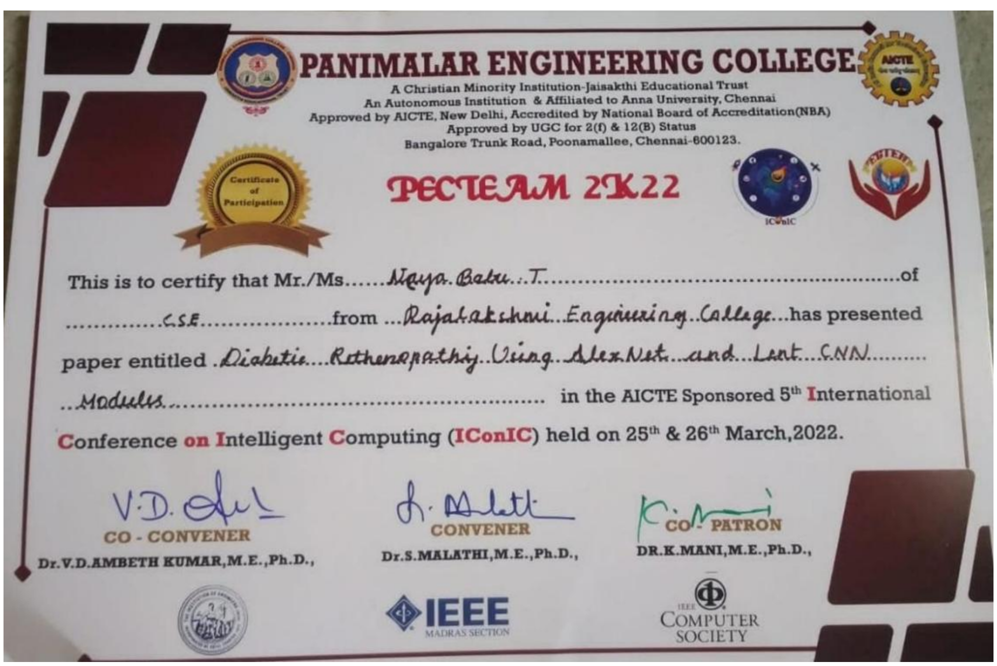
CO - CONVENER
Dr.V.D.AMBETH KUMAR,M.E.,Ph.D.,

CONVENER
Dr.S.MALATHI,M.E.,Ph.D.,

CO - PATRON
DR.K.MANI,M.E.,Ph.D.,

IEEE MADRAS SECTION

IEEE COMPUTER SOCIETY



PANIMALAR ENGINEERING COLLEGE

A Christian Minority Institution-Jaisakthi Educational Trust
An Autonomous Institution & Affiliated to Anna University, Chennai
Approved by AICTE, New Delhi, Accredited by National Board of Accreditation(NBA)
Approved by UGC for 2(f) & 12(B) Status
Bangalore Trunk Road, Poonamallee, Chennai-600123.

Certificate of Participation

PECTEAM 2K22

This is to certify that Mr./Ms......Naya Babu T...of ....CSE...................from ...Rajalakshmi Engineering College...has presented paper entitled .Diabetic Rethenopathy Using Alex Net and Lent CNN.........
..Modules... in the AICTE Sponsored 5th International Conference on Intelligent Computing (IConIC) held on 25th & 26th March,2022.

CO - CONVENER
Dr.V.D.AMBETH KUMAR,M.E.,Ph.D.,

CONVENER
Dr.S.MALATHI,M.E.,Ph.D.,

CO - PATRON
DR.K.MANI,M.E.,Ph.D.,

IEEE MADRAS SECTION

IEEE COMPUTER SOCIETY

APPENDIX III
CO-PO MAPPING

PROGRAMME EDUCATIONAL OBJECTIVES(PEOs)

PEO I To equip students with essential background in computer science, basic electronics and applied mathematics.

PEO II To prepare students with fundamental knowledge in programming languages and tools and enable them to develop applications.

PEO III To encourage the research abilities and innovative project development in the field of networking, security, data mining, web technology, mobile communication and also emerging technologies for the cause of social benefit.

PEO IV To develop professionally ethical individuals enhanced with analytical skills, communication skills and organizing ability to meet industry requirements.

PROGRAM OUTCOMES (POs)

A graduate of the Computer Science and Engineering Program will demonstrate:
PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.42

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and teamwork: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.43

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

A graduate of the Computer Science and Engineering Program will demonstrate:
PSO1:Foundation Skills: Ability to understand, analyze and develop computer programs in the areas related to algorithms, system software, web design, machine learning, data analytics, and networking for efficient design of computer-based systems of varying complexity. Familiarity and practical competence with a broad range of programming language and open-source platforms.

PSO2:Problem-Solving Skills: Ability to apply mathematical methodologies to solve computational task, model real world problem using appropriate data structure and suitable algorithm. To understand the Standard practices and strategies in software project development using open-ended programming environments to deliver a quality product.

PSO3:Successful Progression: Ability to apply knowledge in various domains to identify research gaps and to provide solution to new ideas, inculcate passion towards higher studies, creating innovative career paths to be an entrepreneur and evolve as an ethically social responsible computer science professional.

Project Work Course Outcome (CO):

1. On completion the students capable of execute the proposed plan and become aware of and overcome the bottlenecks throughout every stage.

2. On completion of the project work students could be in a role to take in any difficult sensible issues and locate answer through formulating right methodology.

3. Students will attain a hands-on revel in in changing a small novel idea / method right into an operating model / prototype related to multi-disciplinary abilities and / or understanding and operating in at team.

4. Students will be able to interpret the outcome of their project. Students will take on the challenges of teamwork, prepare a presentation in a professional manner, and document all aspects of design work.

5. Students will be able to publish or release the project to society.

PO/PSO CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
CO 1	3	3	3	2	1	-	1	-	3	3	2	1	3	3	3
CO 2	3	3	3	3	-	-	-	-	3	3	2	2	3	3	3
CO 3	2	2	2	2	-	-	-	-	2	2	2	2	2	2	2
CO 4	2	2	3	3	3	2	1	2	3	3	2	2	3	3	3
CO 5	3	3	3	3	1	1	1	1	3	3	1	1	3	3	3
Average	2.6	2.6	2.8	2.6	1	0.6	0.6	0.6	2.8	2.8	1.8	1.6	2.8	2.8	2.8

REFERENCES

1. Ahmad Khan, Fiaz Gul Khan, Iftikhar Ahmed Khan, Sajid Shah, Sehrish Qummar, , Shahaboddin Shamshirband & Zia Ur Rehman , "A Deep Learning Ensemble Approach for Diabetic Retinopathy Detection" *IEEE* (2019).

2. Amit Paraye, Alpana Rajan, Anil Rawat, Dhirendra KumarVerma, & Gaurav Saxena, "Improved and robust deep learning agent for preliminary detection of Diabetic Retinopathy using public datasets" (Intelligence-Based- Medicine, Volumes 3-4, 2020).

3. Shu-I Pao, Hong-Zin Lin, Ke-Hung Chien, Ming-Cheng Tai, Jiann-Torng Chen & Gen-Min Lin, "Detection of Diabetic Retinopathy Using Bichannel Convolutional Neural Network" (Enrico Peiretti, 2020).

4. Ali Kartit, Hamid Hrimech, Mustapha Aatil & Mohamed Lachgar, "Diabetic Retinopathy Classification Using ResNet50 and VGG-16 Pretrained Networks" (IJCEDS, volume 1, 2021).

5. Anne E.Fung, Ivaylo Stoilov, Michael J.Elman, Rishi P.Singh, & Simran K.Singh , "Advances in the treatment of diabetic retinopathy" *PubMed (*2019).

6. Bonthala CS Manideep, N. Ch. S. N. Iyengar, Ronnie D. Caytiles, & Suvajit Dutta, Syed Muzamil Basha, "Classification of Diabetic Retinopathy Images by Using Deep Learning Models" *International Journal of Grid and Distributed Computing (*2019).

7. Andreas Holzinger, Ibrahim Almakky, Sarni Suhaila Rahim, & Vasile Palade, "Detection of Diabetic Retinopathy and Maculopathy in Eye

Fundus Images Using Deep Learning and Image Augmentation" *International cross-domain conference*, *UK* (2019).

8. Amit Paraye, Dhirendra Kumar Verma, Gaurav Saxena, & Soumya Joshi, "Issues in Training a Convolutional Neural Network Model for Image Classification" *International Conference on Advances in Computing and Data Science* (2019).

9. Gang Hu, Kejun Wang, Yuan Peng, Mengran Qiu, Jianfei Shi & Liangliang Liu, "Deep Learning Methods for Underwater Target Feature Extraction and Recognition" (Rasit Koker, 2019).

10. Geoffrey Hinton, Yann LeCun, & Yoshua Bengio , "Deep Learning Concepts", (Nature, 2020).

11. Vijay Viswanathan, "Diabetic kidney disease and diabetic retinopathy: the ominous duo" *International Journal of Diabetics in Developing Countries* (2020).

12. H Kothandan, K Sathyavani, M Jayaraman & V Viswanathan, "Direct costs associated with chronic kidney disease among type 2 diabetic patients in India" (indiajnephrol.org, 2019).

13. Amol Prataprao Bhatkar & G.U. Kharat, "Detection of Diabetic Retinopathy in Retinal Images using MLP classifier" *IEEE* (2019).

14. Bhargav Srinivasan, Savitha Choudhary, & Yashal Shakti Kanungo, "Detecting Diabetic Retinopathy using Deep Learning" *IEEE* (2020).

15. Haiquan Chen, Wenbin Ye, Xianlong Zeng, & Yuan Luo, "Automated Diabetic Retinopathy Detection Based on Binocular Siamese-like Convolutional Neural Network" *IEEE* (volume 7, 2019).

16. C Jayakumari, E P Sumesh, & Vidhya Lavanya, "Automated Diabetic Retinopathy Detection and classification using ImageNet Convolutional Neural Network using Fundus Images" *IEEE* (2020).

17. Seema Hanchate, Supriya Mishra, & Zia Saquib, "Diabetic Retinopathy Detection using Deep Learning" *IEEE* (2020).

18. Kevin A. Schneider, Michal J. Wesolowski, Peyman Hosseinzadeh Kassani, Ralph Deters, & Sara Hosseinzadeh Kassani, "Breast Cancer Diagnosis with Transfer Learning and Global Pooling" *IEEE* (2019).

19. Richard M. Davis, & Seema Garg, "Diabetic Retinopathy Screening Update" (American Diabetics Association, 2020).

20. Manoharan Usha, Ramachandran Rajalakshmi, Subramanian Arulmalar & Vijayaraghavan Prathiba, "Review of retinal cameras for global coverage of diabetic retinopathy screening" (Nature-eye, 2020).