

SMART RETINAL VESSEL IDENTIFICATION AND PROJECTION FOR EYE DISEASE PREDICTION

Submitted in partial fulfillment of the requirements for the award of Bachelor of
Technology Degree in Biomedical Engineering

By

LOGAPRIYA S (Reg No. 37240047)

SANTHI P (Reg No. 37240065)



**DEPARTMENT OF BIOMEDICAL ENGINEERING
SCHOOL OF BIO AND CHEMICAL ENGINEERING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

MARCH 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC | 12B Status by UGC | Approved by AICTE

Jeppiar Nagar, Rajiv Gandhi Salai, Chennai – 600 1190

www.sathyabama.ac.in

DEPARTMENT OF BIOMEDICAL ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **LOGAPRIYA S (37240047)**, **SANTHI P (37240065)** who carried out the project entitled "**Smart retinal vessel identification and projection for eye disease prediction**" under our supervision from **October 2020 to March 2021**.

Mr. G. Umashankar, M.Tech., (Ph.D.),

Guide and Supervisor

Dr. T. SUDHAKAR, M.Sc., Ph.D.,

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

We, **Logapriya S (37240047)** and **Santhi P (37240065)** hereby declare that the Project Report entitled “**Smart retinal vessel identification and projection for eye disease prediction**” done by us under the guidance of **Mr. G. Umashankar**, Department of Biomedical Engineering is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Biomedical Engineering.

DATE:

PLACE: CHENNAI

1. *S. Logapriya*
2. *P. Santhi*

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. J. Premkumar**, M.Sc., Ph.D., and **Dr. T. Sudhakar** M.Sc., Ph.D., **Head of the Department, Dept. of Biomedical Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Mr. G. Umashankar**, M.Tech., (Ph.D)., **Department of Biomedical Engineering** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

We wish to express our thanks to all **Teaching and Non-teaching** staff members of the **Department of Biomedical Engineering** who were helpful in many ways

ABSTRACT

In India, among the 20 + population, there were an estimated 65 million people with DR and retinitis pigmentosa in 2016. Assuming 15%–20% of these individuals have any diabetic retinopathy (DR) & retinitis pigmentosa and 5%–7% have vision-threatening DR, 3.25–4.55 million people are at risk of visual loss or have already lost vision from DR. Diabetic retinopathy may be a diabetes complication that affects eyes. It's caused by damage to the blood vessels of the light-sensitive tissue at the rear of the attention (retina). At first, diabetic retinopathy may cause no symptoms or only mild vision problems. Eventually, it can cause blindness. This condition can develop in anyone who have type 1 or type 2 diabetes. The longer you've diabetes and thus the less controlled your blood sugar is, the more likely you're to develop this eye complication. Retinitis pigmentosa is an inherited degenerative disease. It slowly affects the retina and causes loss of night vision and side vision. In this project, we will be effectively classifying the retinal vessel in an eye based on the scanned images of the eye using deep learning algorithms to effectively identify the occurrence of diabetic retinopathy in a patient. A dataset consisting of the scanned images of the eye is trained using the resnet deep learning algorithm. Thus, this project gives an innovative solution to effectively identify the occurrence of diabetic retinopathy & retinitis pigmentosa in a patient based on the classification of the retinal vessels.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	I
	LIST OF ABBREVIATIONS	lii
	LIST OF FIGURES	iv
1	INTRODUCTION	1
	1.1 OUTLINE OF THE PROJECT	1
	1.2 TECHNOLOGIS USED	2
	1.2.1 DEEP LEARNING	2
	1.2.1.1 DEEP LEARNING PROCESS	4
	1.2.1.2 CLASSFICIATION OF NEURAL NETWORKS	4
	1.2.1.3 CONVOLUTION NEURAL NETWORKS	7
	1.2.2 REACTJS WEB APPLICATION DEVELOPMENT	8
2	LITERATURE SURVEY	10
3	AIM AND SCOPE	18
	3.1 AIM	18
	3.2 SCOPE	18
	3.3 EXISTING SYSTEM	19
	3.4 PROPOSED SYSTEM	20
4	METHODOLOGY	21
	4.1 ARCHITECTURE	21
	4.2 WORKING	22
	4.3 MODULE DESCRIPTION	23
	4.3.1 DATA COLLECTION	23

	4.3.2 DATA AUGMENTATION	25
	4.3.3 DISEASE PREDICTION USING THE STRUCTURE	27
	4.3.4 WEB APPLICATION DEVELOPMENT	28
	4.4 SOFTWARE DESCRIPTION	30
	4.4.1 VISUAL STUDIO	30
	4.4.2 PYTHON	32
5	RESULT AND DISCUSSION	33
	5.1 FINAL RESULTS OBTAINED	33
	5.2 DISCUSSION	40
6	SUMMARY AND CONCLUSION	41
	6.1 SUMMARY	41
	6.2 CONCLUSION	41
	REFERENCE	42
	APPENDIX	45

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
CNN	CONVOLUTION NEURAL NETWORK
RNN	RESIDUAL NEURAL NETWORK
JS	JAVA SCRIPT
HTML	HYPER TEXT MARKUP LANGUAGE
DOM	DOCUMENT OBJECT MODEL
XML	EXTENSIBLE MARKUP LANGUAGE
NPDR	NON-PROLIFERATIVE DIABETIC RETINOPATHY
MVC	MODEL VIEW CONTROLLER
OCTA	OPTICAL COHERENCE TOMOGRAPHY ANGIOGRAPHY
IDE	INTEGRATED DEVELOPMENT ENVIRONMENT

LIST OF FIGURES

FIG NO	TITLE	PAGE NO
1.1	DEEP LEARNING LAYERS	3
1.2	DEEP LEARNING PROCESS	4
1.3	TYPES OF DEEP LEARNING NETWORKS	5
1.4	CNN	7
1.5	TYPES OF CNN	8
1.6	REACTJS ARCHITECTURE	9
4.1	SYSTEM ARCHITECTURE	21
4.2	WEB APPLICATION ARCHITECTURE	22
4.3	DATA AUGMENTATION	27
4.4	RESNET ARCHITECTURE	28
5.1	DATASET COLLECTED	33
5.2	DATA SEPARATION	34
5..3	MOVING INSIDE THE CODING FOLDER	34
5.4	EXECUTING IN THE COMMAND PROMPT	35
5.5	TRAINING PROCESS	35
5.6	TRANING LOSS GRAPH	36
5.7	LOGIN PAGE OF WEB APPLICATION	37
5.8	HOME PAGE	37
5.9	RETINAL SCAN UPLOAD	38
5.10	NORMAL EYE SCAN DIAGNOSIS	39
5.11	DIABETIC RETINOPATHY DIAGNOSIS	39
5.12	RETINITIS PIGMENTOSA DIAGNOSIS	40

CHAPTER 1

INTRODUCTION

1.1 OUTLINE OF THE PROJECT

The Union Health Ministry's first National Diabetes and Diabetic Retinopathy Survey (2015-19) has revealed that the prevalence of Diabetic Retinopathy (DR) is 16.9 percent while the prevalence of sight-threatening DR is 3.6 percent.

It may be noted that diabetes and diabetic retinopathy are emerging as a huge non-communicable disease resulting in ocular morbidity.

The survey has also revealed the prevalence of mild retinopathy at about 11.8 percent.

The prevalence of diabetes within the surveyed population has been recorded 11.8 percent while the prevalence of Known Diabetes (KD) 8.0 percent.

Diabetic retinopathy could also be a diabetes complication that affects eyes. It's caused by damage to the blood vessels of the light-sensitive tissue at the rear of the eye (retina).

Early stage, diabetic retinopathy may cause no symptoms or only mild vision problems. Eventually, it can cause blindness.

The condition can develop in anyone who have type 1 or type 2 diabetes. The longer you've diabetes and thus the less controlled your blood sugar is, the more likely you're to develop this eye complication.

Early stages of diabetic retinopathy usually don't have any symptoms. Some people notice changes in their vision, like trouble of reading or seeing faraway objects. These changes may come and go.

In later stages of the disease, blood vessels within the retina start to bleed into the vitreous (gel-like fluid within the middle of the eye). If this happens, you will see dark, floating spots or streaks that appear as if cobwebs.

Sometimes, the spots clear on their own — but it's important to urge treatment directly. Without treatment, bleeding can happen again, get worse, or cause scarring.

Retinitis pigmentosa (RP) could also be a genetic disorder of the eyes that causes loss of vision. Symptoms include trouble seeing within the dark and decreased sight (side vision). As sight worsens, people may experience "tunnel vision". Complete blindness is rare.

According to a union government survey, criteria and thus the ophthalmoscopic fundus appearance, prevalence of retinitis pigmentosa was about 1:750 within the adult population of rural central India.

1.2 TECHNOLOGIES USED

1.2.1 Deep learning

Deep learning may be a computer software that mimics the network of neurons in a brain. It's a subset of machine learning and is named deep learning because it makes use of deep neural networks.

Deep learning algorithms are constructed with connected layers.

- The primary layer is Input Layer
- The last layer is Output Layer
- All layers in between are Hidden Layers. The word deep means the network

join neurons in additional than two layers.

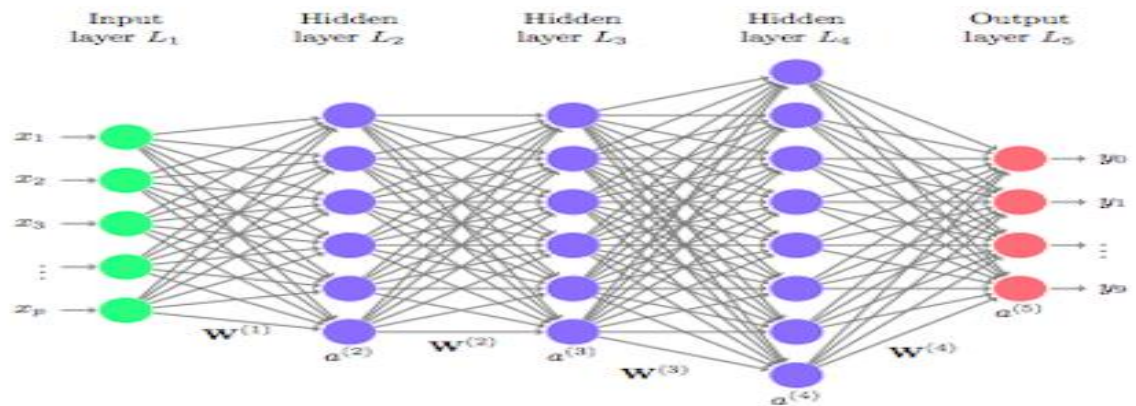


Figure 1.1 Deep Learning Layers

Each Hidden layer is composed of neurons. The neurons are connected to each other. The neuron will process then propagate the input it receives the layer above it.

The strength of the signal given the neuron within the next layer depends on the load, bias and activation function.

The network consumes large amounts of input file and operates them through multiple layers, the network can learn increasingly complex features of the info at each layer.

Deep learning may be a powerful tool to form prediction an actionable result. Deep learning excels in pattern discovery and knowledge-based prediction. Big data is the fuel for deep learning.

When both are combined, a corporation can reap unprecedented leads to term of productivity, sales, management, and innovation.

Deep learning can outperform traditional method. For instance, deep learning algorithms are 41% more accurate than machine learning algorithm in image classification, 27% more accurate in facial recognition and 25% in voice recognition.

1.2.1.1 Deep Learning Process

A deep neural network provide state-of-the-art accuracy in many tasks, from object detection to speech recognition. Automatically they can learn, without predefined knowledge explicitly coded by the programmers.

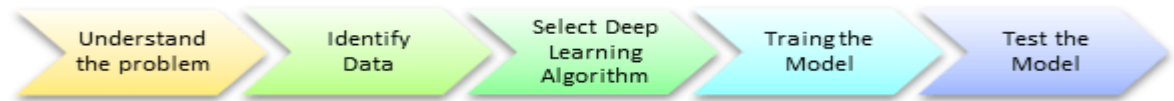


Figure 1.2 Deep Learning Process

Each layer represents a deeper level of data, i.e., the hierarchy of data. A neural network with four layers will learn more complex feature than with two layers.

The learning occurs in two phases.

- The first phase consists of applying a nonlinear transformation of the input and make a statistical model as output.
- The second phase aims at improving the model with a mathematical method referred to as derivative.

The neural network repeats these two phases hundred to thousand times, until it's reached a tolerable level of accuracy. The repeat of this two phase is named an iteration.

1.2.1.2 Classification of Neural Networks

Shallow neural network: The Shallow neural network has just one hidden layer between the input and output layer.

Deep neural network: The Deep neural networks have more than one layer. For instance, Google Net model for image recognition counts 22 layers.

Nowadays, deep learning neural network is employed in many ways like

driverless car, mobile, Google program, Fraud detection, TV, and so on.

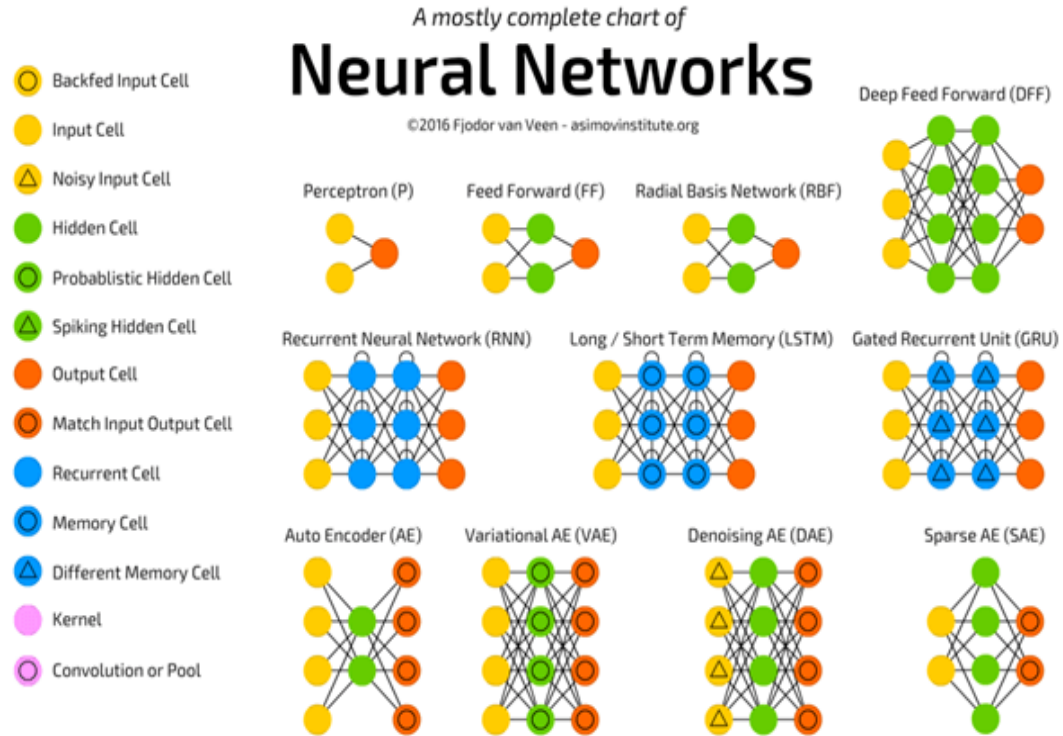


Figure 1.3 Types of Deep Learning Networks

Feed-Forward Neural Network

The simplest sort of artificial neural network. This type of architecture, information flows in just one direction, forward. This suggests, the information flows starts at the input layer, goes to the hidden layers, and end at the output layer. The network doesn't have a loop. Information stops at the output layers.

Recurrent Neural Network

RNN could also be a multi-layered neural network which can store information in context nodes, allowing it to learn data sequence and output variety or another sequence. In simple words it a man-made neural networks whose connections between neurons include loops. RNNs are compatible for processing sequence of inputs.

The manner of RNNs are defined below

- The RNN neurons will obtain a signal that point to the begin of the sentence.
- The network receives the phrase "Do" as an input and produces a vector of the number. This vector is fed again to the neuron to grant a memory to the network. This stage helps the network to take into account it received "Do" and it received in the first position.
- The network will in a similar fashion proceed to the next words. It takes the word "you" and "want." The state of the neurons is updated upon receiving each phrase.
- The final stage happens after receiving the word "a." The neural network will provide a probability for each English phrase that can be used to whole the sentence. A well-trained RNN probably assigns a high probability.

Common uses of RNN:

- Help securities merchants to get analytic reports.
- Detect abnormalities within the contract of monetary statement.
- Detect fraudulent credit-card transaction.
- Provide a caption for images.
- Power chatbots.

The preferred makes of RNN happen when the practitioners are working with time-series data or sequences.

1.2.1.3 Convolutional Neural Networks (CNN)

CNN is a multi-layered neural network with a special architecture designed to extract increasing number of complex features of the data at each layer to determine the output. CNN's are properly applicable for perceptual tasks.

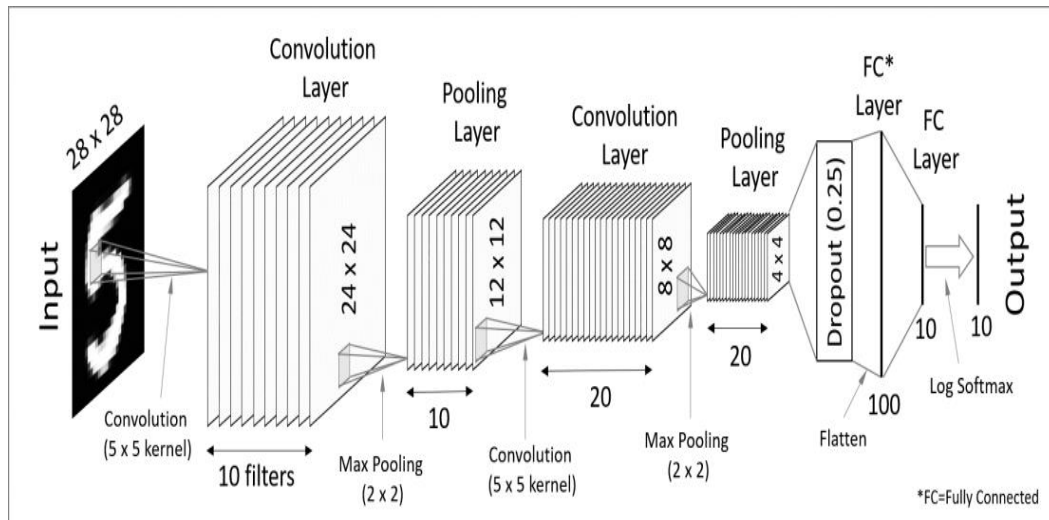


Figure 1.4 CNN

CNN is usually used when there's an unstructured data set (e.g., images) and therefore the practitioners got to extract information from it. For instance, if the task is to predict a picture caption:

- As an example, a picture of a cat is given as input, this image, in computer term, may be a collection of the pixel. Generally, one layer for the grayscale picture and three layers for a colour picture.
- During the feature going to know (i.e., hidden layers), the network will discover special features, for instance, the tail of the cat, the ear, etc.
- When the network entirely discovered the way to recognize an image, it can provide a probability for every image it knows. The label with the very best probability will emerge because the prediction of the network.

A Convolutional Neural Network (CNN, or ConvNet) are a special quite multi-layer neural networks, designed to acknowledge visual patterns directly from pixel images with minimal Pre-processing. The ImageNet project may be a large visual database designed to be used in visible visual perception software research. The ImageNet project runs an annual software contest, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), where

the software program packages compete to efficaciously classify and detect the objects and scenes.

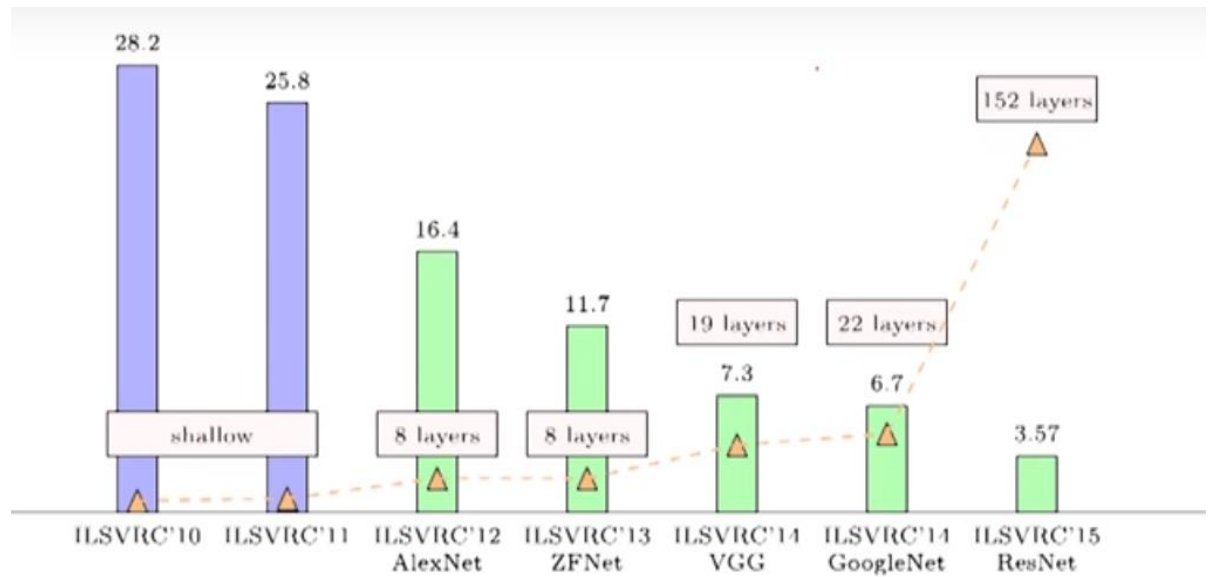


Figure 1.5 Types of CNN

1.2.2 ReactJS web application development

ReactJS basically is an open-source JavaScript library which is employed for building user interfaces specifically for single page applications. It's used for managing view layer for web and mobile apps. React also allows us to form reusable UI components. React was first created by using Jordan Walke, a programmer working for Facebook. React first categorized on Facebook's newsfeed in 2011 and on Instagram.com in 2012. React approves developers to create large web applications which may change data, except reloading the page. The principle purpose of React is to be fast, scalable, and straightforward. It works only on user interfaces in application. This corresponds to view in the MVC template. It are often used with a combination of other JavaScript libraries or frameworks, such as Angular JS in MVC.

Feature

JSX: JSX stands for JavaScript XML. Its an XML/ HTML like syntax employed by React. It extends the ECMAScript in order that XML/ HTML like text can co-exist along side JavaScript react code. This syntax is employed by the pre-processors like Babel to rework HTML like text found in JavaScript files into standard JavaScript objects. With JSX, we will go a step further by again embedding the HTML code inside the JavaScript. This makes HTML codes easy to know and boosts JavaScript's performance while making our application robust.

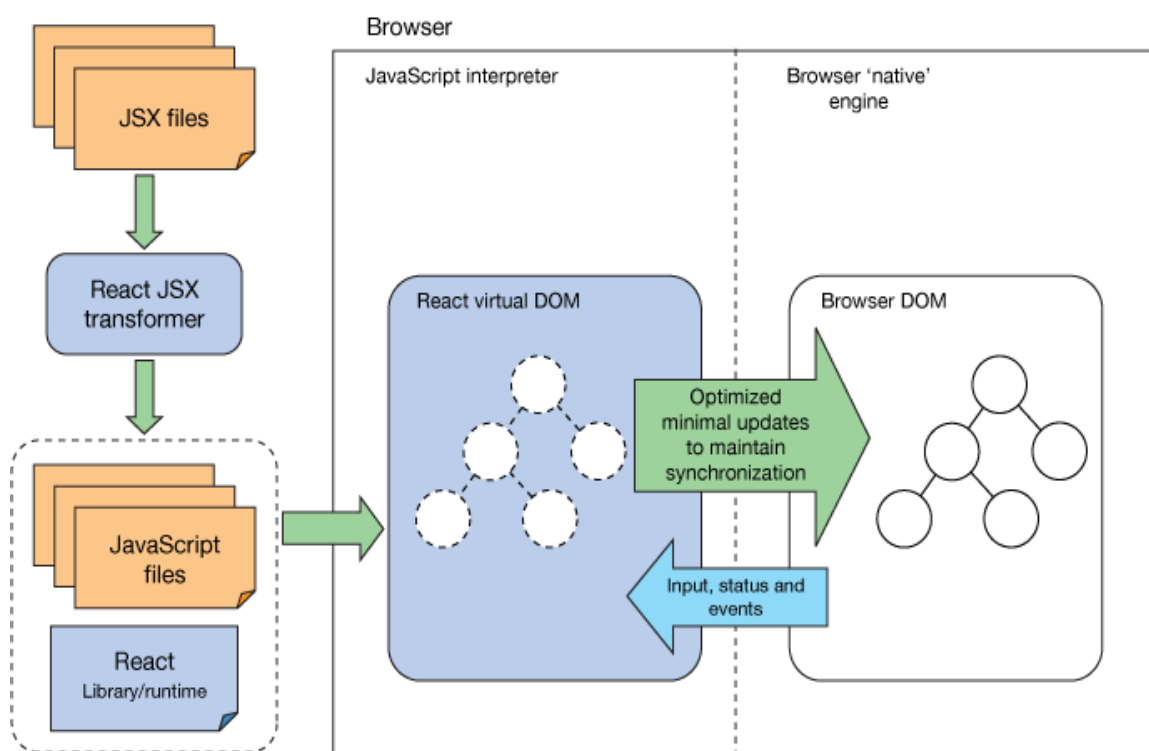


Figure 1.6 ReactJS architecture

CHAPTER -2

LITERATURE SURVEY

Bogunović, et al., (2019). Stating that, Retinal swelling is foremost vision-threatening retinal diseases. Optical coherence tomography (OCT) is helpful for in assessing the presence and quantity of retinal fluid and image-guided treatment. Deep learning methods have create their impact across medical imaging. The challenge have two tasks: fluid detection and fluid segmentation. It featured for the primary time: all three retinal fluid types, along with annotated images provided by the clinical centers. Which were provided three most common OCT device vendors from patients with two different retinal diseases. The analysis gives that in the detection task, the performance on the automated fluid detection was within the inter-grader variability. Within the segmentation task, fusing the automated methods p produced segmentations that were superior to individual methods, It indicating the necessity for further development within the segmentation.

Fu, et al., (2017). Stating that, Angle-closure glaucoma is a important reason for irreversible visual impairment. It can be identified by measuring the anterior chamber angle (ACA) of the eye. The ACA could be viewed clearly by anterior segment optical coherence tomography (AS-OCT). This complicating image analysis. We provide a data-driven technique for automatic AS-OCT structure segmentation, measurement, and screening. This technique foremost estimates initial markers in the eye by the help of label, This transfer from a hand-labeled exemplar data set, whose images are collected over different patients and AS-OCT modalities. These initial markers are then refined by usage of a graph-based smoothing method. That is guided by AS-OCT structural information. These markers promote segmentation of major clinical structures, which are helpful to recover standard clinical parameters. These parameters could be used not only to

support clinicians in making anatomical assessments. Experiments on Visante AS-OCT and Cirrus high-definition-OCT data sets demonstrate the effectiveness of our task.

Herliana, et al., (2018). Stating that, Currently, there are many medical experts facing difficulty in conducting early detection for diabetic retinopathy. This happens because it is difficult to recognize the early symptoms of DR disease. This disease to be detected early. An accurate classification method is needed. Data mining is the concept which one alternative in conducting classification. This method is conducting by the help of applying particle swarm optimization (PSO) method. Which able to select the best Diabetic Retinopathy feature, based upon diabetic retinopathy dataset. Then, the selected feature is further classified by the help of classification method of neural network. The result of the method indicate that there is an increase in result. By helping of applying neural network based particle swarm optimization (PSO) of 76.11%. This method also gives that there is an increase in classification result. By the usage of feature selection method of 4.35% from previous result of 71.76% by only applying neural network method.

Kepp, et al.,(2019).Stating that, Optical coherence tomography is the non-invasive imaging technique. Which provides cross-sectional 3D images of biological tissue. Especially in the ophthalmology department OCT is used for the diagnosis of various eye diseases. Automatic retinal layer segmentation algorithms, which are highly based upon deep learning techniques. It can support diagnostics. Topology properties, like the order of retinal layers, are no longer considered. In this methodology we provide an automatic segmentation approach based upon shape regression. Which is using convolutional neural networks (CNNs). Shapes are indicated by signed distance maps (SDMs). That will be able to assign the distance to the next object contour to each pixel. The spatial regularization is introduced and segmentations can be generated. The results gives that our method has

some outliers. Along with comparable segmentation performance. Additionally it have an improved topology preservation, which saves further post-processing.

Medvedev, et al., (2020). Stating that, we give a robust semi-analytical multi-factor model of OCT-scan formation. Along with this have. rigorous accounting for the beam-focusing effects. It is based upon numerical summation. it is the contributions of the localized sub-resolution scatterers. It take account of variations in the phase-amplitude parameters of the incident and backscattered optical waves. Along with a subsequent integration of the latter over the objective aperture. The main advantage of the model is convenience to account for scatterer motions. It have the difficulties in performing highly-controlled phantom experiments. Application of the numerical model opens flexible and convenient. It have possibilities for evaluating the influence of scatter motions with usage of numerically simulated OCT scans. This numerical model is readily permit one to introduce various types of scatterer motions.

Najeeb, et al., (2018). Stating that, Biomedical image classification for the diseases is a longer and manual process. Recent progresses in computer vision has authorized the detection and classification of medical images. With the usage of machine intelligence a more attainable solution. We examine the possibility of automated detection. And also classification of retinal abnormalities from retinal OCT scan images of patients. We can develop an algorithm to detect the region of interest from a retinal OCT scan. The use a computationally inexpensive single layer convolutional neural network structure helpful for the classification process. Our model is trained on an open source retinal OCT dataset. Which containing 83,484 images of various tunnel disease patients. It also provides a attainable classification accuracy.

Naz, et al., (2017). Stating that, in this method provide automatic classification of OCT images for identification of patients along with DME versus normal subjects. It gives relatively simple and practical approach. It is employed to exploit the information in OCT images for a robust classification of Diabetic Macular Edema (DME) with usage of coherent tensors. From the retinal OCT scan top and bottom layers which are helpful for extracted to find thickness profile. Cyst spaces are segmented out from the normal image and DME images. The features extracted from thickness profile. The cyst are tested on Duke Dataset. It having 55 diseased and 53 normal OCT scans. Results provide SVM with Leave-one-Out gives the maximum accuracy of 79.65% with 7.6 standard deviation. The experiments inform that for the identification of DME, nearly same accuracy of 78.7% can be achieved by usage of a simple threshold. Which could be calculated by the usage of thickness variation of OCT layers. Comparison of this algorithm on a standard dataset gives the best classification performance.

Perdomo, et al., (2018). Stating that, Diabetic macular edema (DME) is one of the most common eye complication induced by diabetes mellitus. Resulting in the partial or total loss of vision of the eye. Optical Coherence Tomography (OCT) have been widely used imaging technique to diagnose different eye diseases. It detect the sensitivity to represent small amounts of fluid, thickness between layers and swelling. The lack of tools for analysis the image automatic technique is still a problem. Convolutional neural networks (CNNs) have to provide good performance when applied to several medical images for analysis tasks. This method OCT-NET, based on a CNN for the automatic classification of OCT volumes. The model was evaluated on a dataset of OCT volumes for DME diagnosis. Usage of a leave-one-out cross-validation strategy obtaining an accuracy, sensitivity, and specificity of 93.75%.

Qiao, et al., (2020). Stating that, Predicting the presence of Micro aneurysms within the images and identification of diabetic retinopathy in early-stage has always been a challenge. Diabetic Retinopathy is suffering from prolonged high blood sugar level. Which ends up in microvascular difficulties and irreversible vision loss. The rapid development of deep learning makes efficient technique which give solution for medical image analysis problems. This system which analysis the presence of micro aneurysm in fundus image with help of convolutional neural network algorithms. Which is useful for embeds deep learning act as a core component accelerated with GPU (Graphics Processing Unit) which can perform medical image detection and segmentation alongside high-performance and low-latency inference. The semantic segmentation algorithm is used to spot the fundus images as normal or infected. Semantic segmentation divides the image into pixels supported their common semantic to identify the feature of micro aneurysm. This provides an automatic system which can assist ophthalmologists to grade the fundus images as early NPDR, moderate NPDR, and severe NPDR.

Qummar, et al., (2019). Stating that, Diabetic Retinopathy (DR) is an ophthalmic disease which is damages retinal blood vessels of the eye. DR causes impaired vision and blindness. If it does not diagnosed in early stages. It have five stages, that are normal, mild, moderate, severe and PDR (Proliferative Diabetic Retinopathy). Normally, highly trained clinician examine the colored fundus images to diagnose this fatal disease. Various computer vision-based methods are helpful for automatically detect DR and its different stages from retina images. In this research, we used the publicly accessible Kaggle dataset of retina images which is used to train an ensemble of 5 deep Convolution Neural Network (CNN) models (Resnet50, Inceptionv3, Xception, Dense121, Dense169). Wish used to encode the rich features and improve the classification for various stages of DR. The

experimental outputs show that the proposed model detects all the stages of DR.

Soltan, et al.,(2019).Stating that, The arrival of light-sensitive opsins made it possible to optically trigger the neurons into activation or inhibition by help of visible light. In this methodology we integrated virtual reality optical system. Along with a high radiance high density micro display. Which helpful to implement a wearable device for Retinitis Pigmentosa (RP) patients. We construct system to imitate the human eye and light attenuation. While it passes through the retina of eye. The moderate light intensity on the surface of the retina after the optical system is 1.35mW/mm^2 . Which is above the ChR-2 stimulation level (0.7mW/mm^2). The number of pixels which provide the light intensity above the stimulation threshold. This 7128 pixel out of 8100 pixel in the micro display. The average potential of the optical system is 5.75%. So, these wearable device will be able to restore vision. It helpful for RP patients without the need for invasive implantable device

Soltan, et al., (2019).Stating that, Optogenetics is a highly growing neuromodulation techniques. It could able to remotely stimulate neural activities of a genetically modified cells. The advantage of remotely controlling the neural activity triggered which helpful for the researchers. Implement a headset to externally stimulate retina cells for people along with retina pigmentosa disease. The wearable device needs an efficient optical system. Because of focus the transmitted light pattern into the retina surface. In this methodology, three different lenses are used. That are contact lens, folded prism and linear lenses. That are used to evaluate the headset performance. A $90\times 90\text{ }\mu\text{LED}$ display is act as a light source. And it have the optical efficiency for each lens is measured for different points over the lens area.

Vahadane, et al., (2018).Stating that, we are using a twostep framework. It used to automatically classify an OCT scan as indicative of Diabetic Macular Edema (DME). By the help of detecting abnormal pathologies in OCT frames.

The starting step involves detection of candidate patches. Which is for fluid filled regions and hard exudates usage of image processing techniques. The next step is to predict a label. Which is for these candidate patches usage of deep convolutional neural network. The last step is collation step. We collect the confidences of the CNN models. And it use a rule based methodology able to predict the presence of DME.

Vasilyeva, et al., (2020). Stating that, Retinal dystrophy is the highly heterogeneous. "Frequent among rare" inherited eye pathology which is results in the blindness. Some common terms of retinal dystrophy are associated with and years ago established genes. Hundreds of others are found and kept being accomplished to be involved in its pathogenesis. In this methodology for genetic testing, a whole-exome sequencing by next-generation sequencing (NGS) was implied, which also have bioinformatics analysis usage of open algorithms. And also recommendations on filtering. We reported 6 solved usage of NGS cases of hereditary retinitis pigmentosa. Which is associated along with pathogenic variants in the genes PRPF31, IFT140, RGS9, RPGR, RP1.

Xi, et al., (2018). Stating that, Convolutional Neural Networks (CNNs) is capacity of automatically learning the feature representations. CNN-based recognition algorithm has been a other method for human activity recognition. The general convolution operation followed by pooling can be expand the receptive fields for extracting features. It will propose about information loss in feature representation. Because of that dilated convolutions not only could expand receptive field. Exponentially without changing the size of field map or pooling. But it also will not produce information loss. Hence, we use D2CL, a novel deep learning framework. It is for human activity recognition by the usage of multi-model wearable sensors. This framework have dilated convolutional neural networks and recurrent neural networks. Given these learned feature. Two RNNs are applied to the model their latent temporal dependencies. At last, a softmax

classifier at the topmost layer is able to recognize activities. To assess the performance of D2CL on activity recognition, we are supposed to select two open datasets OPPORTUNITY and PAMAP2 for training and testing. Results indicate the methodology gains a higher classification performance than the state-of-the-art DeepConvLSTM.

CHAPTER – 3

AIM AND SCOPE

3.1 AIM

- A fluorescein angiography may be a procedure during which a fluorescein is injected into the bloodstream. The dye highlights the blood vessels within the back of the attention in order that they are often photographed.
- In this project, we'll be effectively detecting the retinal vessel in an eye fixed supported the scanned images of the attention using deep learning algorithms. A dataset consisting of the scanned images of the attention is trained using the resnet deep learning algorithm. (Qummar, et al., 2019)
- To provide a technical solution to effectively determine and predict retinopathy and retinal pigmentosa. To facilitate doctors to offer effective treatment.

3.2 SCOPE

In the coming future, we review the application of the retinitis pigmentosa and diabetic retinopathy determination technology in the healthcare field and it can promote for detecting various types of cancer with more accuracy. In medical field they have more chance to develop or convert this project in many ways. Thus, this project has an efficient scope in coming future where manual predicting can be converted to computerized production in a cheap way.

3.3 EXISTING SYSTEM

Optical coherence tomography (OCT) and its angiography (OCTA) have numerous advantages for the early detection and diagnosis of diabetic retinopathy (DR). However, automated, entire DR classification frameworks based on each OCT and OCTA data have no longer been proposed. In this study, a convolutional neural network (CNN) based technique is proposed to fulfill a DR classification framework the use of en face OCT and OCTA. (Herlixna, et al., 2018) Techniques: A densely and continuously connected neural network with adaptive rate dropout (DcardNet) is designed for the DR classification. Additionally, adaptive label smoothing was proposed and went to suppress overfitting. Three separate classification stages are generated for each case based on the International Clinical Diabetic Retinopathy scale. At the very best level the network classifies scans as referable or non-referable for DR. The second level classifies the attention as non-DR, non-proliferative DR (NPDR), or proliferative DR (PDR). The final stage classifies the case as no DR, slight and reasonable NPDR, severe NPDR, and PDR. (Zang, et al., 2020) Results: We used 10-fold cross-validation with 10% of the data to check the network's performance. The final classification accuracies of the three levels were 95.7%, 85.0%, and 71.0% respectively. Significance: A reliable, sensitive and specific automated classification framework for referral to an ophthalmologist can be a key technology for lowering vision loss related to DR.

Disadvantages of Existing System

- The system does not focus on identifying the retinitis pigmentosa in eye.
- The project does not give any solution to facilitate the doctors to give proper treatment to the patient.
- The project does not produce accurate results.

3.4 PROPOSED SYSTEM

Diabetic retinopathy may be a diabetes complication that affects eyes. It's caused by damage to the blood vessels of the light-sensitive tissue at the rear of the attention (retina).At first, diabetic retinopathy may cause no symptoms or only mild vision problems. Eventually, it can cause blindness. This condition can develop in anyone who have type 1 or type 2 diabetes. The longer you've diabetes and thus the less controlled your blood sugar is, the more likely you're to develop this eye complication. Retinitis pigmentosa is an inherited degenerative disease. It slowly affects the retina and causes loss of night vision and side vision. In this project, we will be effectively classifying the retinal vessel in an eye based on the scanned images of the eye using deep learning algorithms to effectively identify the occurrence of diabetic retinopathy in a patient. A dataset consisting of the scanned images of the eye is trained using the resnet deep learning algorithm.(Qummar, et al.,2019) Thus, this project gives an innovative solution to effectively identify the occurrence of diabetic retinopathy & retinitis pigmentosa in a patient based on the classification of the retinal vessels.

Advantages

- Provides a technological solution to effectively facilitating better identification of the occurrence of diabetic retinopathy as well as retinitis pigmentosa.
- Facilitates the doctors to provide proper treatment to the patient.
- The accuracy of detection is very high.

CHAPTER 4

METHODOLOGY

4.1 ARCHITECTURE

This chapter describes the overall and the detailed architectural design. It also describes each module that is to be implemented along with Data Flow diagram.

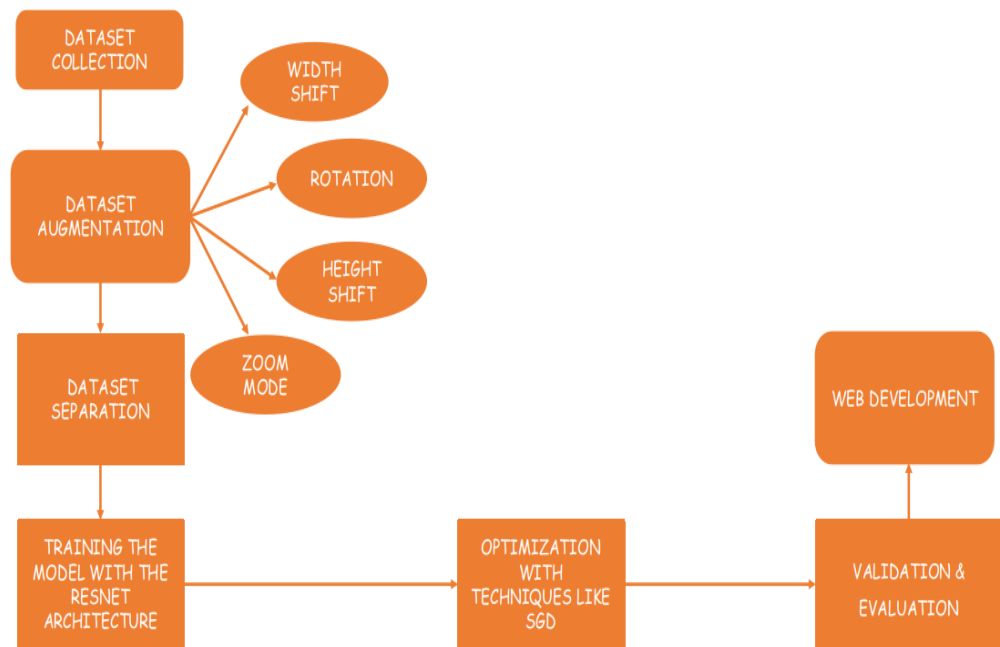


Figure 4.1 System Architecture

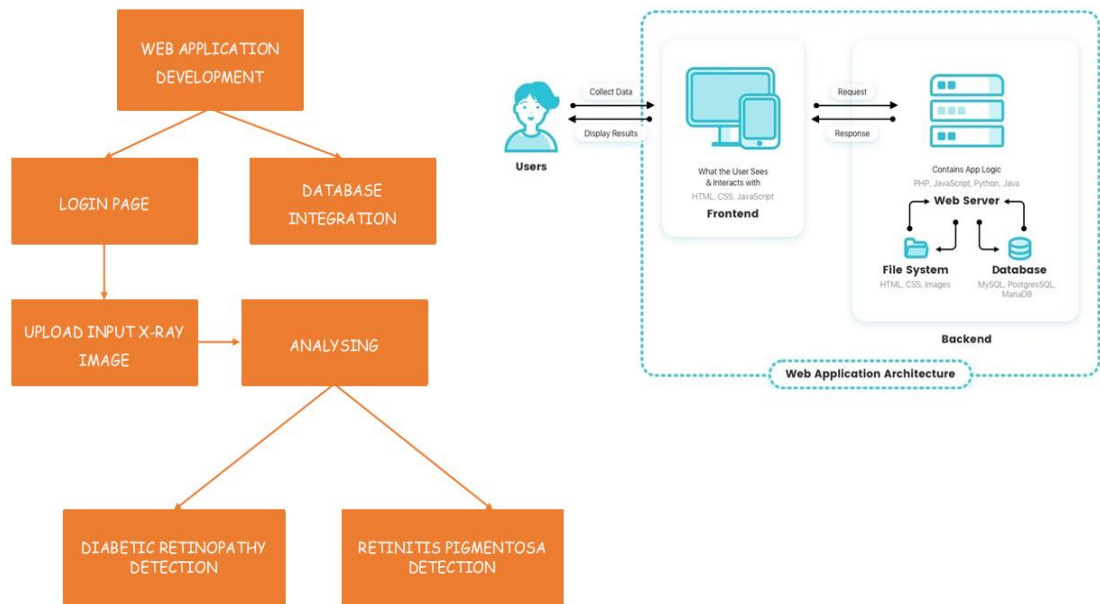


Figure 4.2 Web application Architecture

4.2 WORKING

In this project, we are going to determine occurrence of diabetic retinopathy and retinitis pigmentosa. By this project, we can able to determine the occurrence of diabetic retinopathy and retinitis pigmentosa from the eye scans. So, initially the first step will be dataset collection where we will be collecting dataset such as CT scans of eyes which are used by the laboratories to analyse the the occurrence of diabetic retinopathy and retinitis pigmentosa from various resources through internet. After that, we will be splitting those datasets into different categories that is we will be splitting the dataset into training and testing dataset. In training datasets, we will be using the dataset for training the module whereas testing dataset is used to evaluate the model when it is been completely ready. So training dataset first

undergoes the process called dataset augmentation, where the dataset is multiplied into many datasets then it will undergo the process called preprocessing, which is to make all sizes into single size. We train that datasets by extracting the features using an effective deep learning algorithm called the Resnet algorithm. It undergoes a process called optimization which will optimize the model and loss minimization which will reduce the noises generated during training. In the last it will be undergo a process is called model generation which will be evaluated after generating model using the testing dataset and predict the presence of diabetic retinopathy or retinitis pigmentosa. A web application using a javascript framework reactJS will be developed where an input scanned image will be given to predict which type of diabetic retinopathy or retinitis pigmentosa. Thus, this method provides an effective and cheap method to determine the presence of diabetic retinopathy or retinitis pigmentosa than the methodologies used nowadays.

4.3 MODULE DESCRIPTION

- Dataset Collection
- Dataset Augmentation
- Training using architecture
- Web Application Development

4.3.1 Data Collection

A data set is a collection of data. Deep Learning has become the go-to approach for solving many challenging real-world problems. (Perdomo, et al., 2018). It's truly by way of far and the simplest performing method for computer vision tasks. The image above showcases the facility of deep learning for computer vision. With adequate training, a deep network can segment and identify the “key points” of each and every person in the image. These deep learning machines that have been working so properly need fuel

loads of fuel, that fuel is data. The more labelled data available, the better our model performs. The thought of more data leading to better performance has even been explored at a large-scale by way of Google with a dataset of 300 Million images. When deploying a Deep Learning model in a real-world application, data need to be constantly fed to continue enhancing its performance. And, in the deep learning era, data is very well arguably the most valuable resource. There are three Steps of accumulating data.

Scraping from the Web

Manually discovering and downloading images takes a lengthy time virtually due to the amount of human work involved. The task probably has some sort of common objects are to be detected. And so that becomes the keyword for web-scraping. It also becomes the category name for that object. Every single pixel in the image is required. To get those, it's best to use some simply incredible image annotation tools that are already out there. Given a rough collection of polygon points around an object, could generate the pixel labels for segmentation. Deep intense reduce is also pretty comparable without they use only the four intense points round the object. This will then supply some fine bounding box and segmentation labels. Another alternative is to use a current image annotation GUIs.

Third-party

Since data has become such a valuable commodity in the deep learning era, many start-ups have started to offer their own image annotation services they'll gather and label the data. Given a description of what kind of data and annotations needed. Mighty is one that has been doing self-driving car image annotation and has become pretty big in the space were at CVPR 2018 too.

Payment AI are less specialized than Mighty AI, offering image annotation for any domain

4.3.2 Data Augmentation

The performance of deep learning neural networks often improves with the quantity of knowledge available.

Data augmentation may be a technique to artificially create new training data from existing training data.

This is often done by applying domain-specific techniques to examples from the training data that make new and different training examples.

Image data augmentation is probably the foremost well-known sort of data augmentation and involves creating transformed versions of images within the training dataset that belong to an equivalent class because the original image.

Transforms include a variety of operations from the sector of image manipulation, likes hifts, flips, zooms, and far more. The goal is to expand the training dataset with new, truthful examples.

This suggests, variations of the training set images that are likely to be seen by the model. For instance, a horizontal flip of an image of a cat may add up, because the photo could are taken from the left or right.

A vertical flip of the photo of a cat doesn't add up and would probably not be appropriate as long as the model is extremely unlikely to ascertain a photograph of an upside-down cat.

As such, it's clear that the selection of the precise data augmentation techniques used for a training dataset must be chosen carefully and within the context of the training dataset and knowledge of the matter domain.

Additionally , it are often useful to experiment with data augmentation methods in isolation and together to ascertain if they end in a measurable improvement to model performance, perhaps with a little prototype dataset, model, and training run. Modern deep learning algorithms, like convolutional neural network can learn features that are invariant to their location within the image.

Nevertheless, augmentation can further aid during this transform invariant approach to learning and may aid the model in learning features that also are invariant to transforms like left-to-right to top-to-bottom ordering, light levels in photographs, and more.

Image data augmentation is usually only applied to the training dataset, and to not the validation or test dataset. This is often different from data preparation like image resizing and pixel scaling; they need to be performed consistently across all datasets that interact with the model.

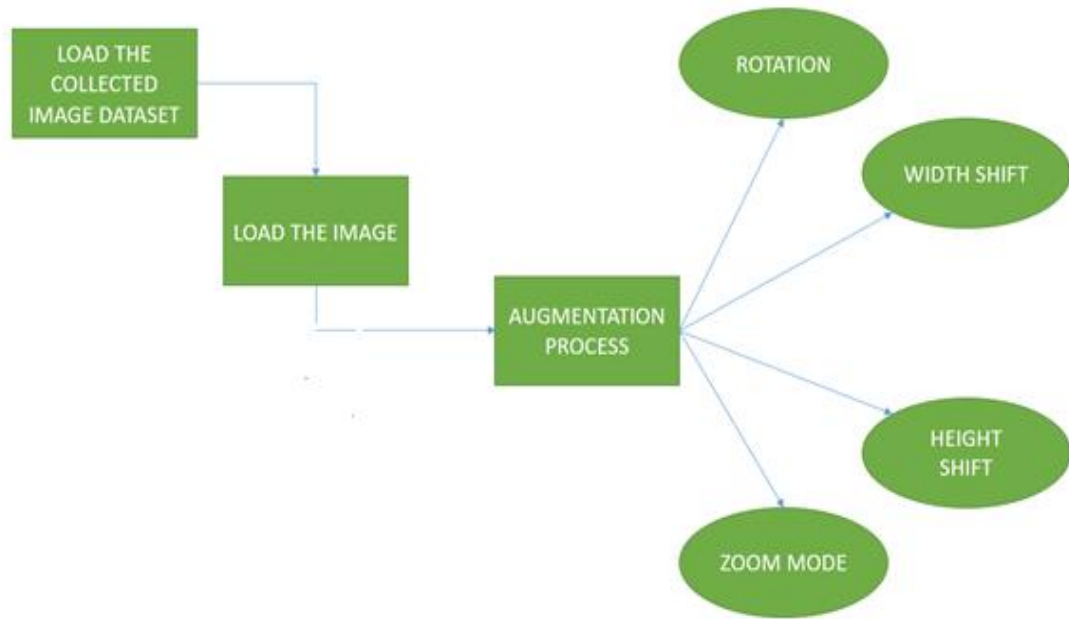


Figure 4.3 Data augmentation

4.3.3 Disease Prediction Using the Architecture

In this project, the Resnet algorithm is used for training the model. The architecture of ResNet50 has 4 stages as shown within the diagram below. The network can take the input image having height, width as multiples of 32 and three as channel width. For the sake of explanation, we'll consider the input size as 224 x 224 x 3. Every ResNet architecture achieve the initial convolution and max-pooling using 7×7 and 3×3 kernel sizes respectively. Afterward, Stage 1 of the network starts and it's 3 Residual blocks containing 3 layers each. The size of kernels wont to perform the convolution operation altogether 3 layers of the block of stage 1 are 64, 64 and 128 respectively. The curved arrows refer to the identity connection. The dashed connected arrow indicate that the convolution operation in the Residual Block is performed with stride 2, hence, the size of input will be reduced to half in terms of height and width but the channel width are going to be doubled. As we progress from one stage to a different, the channel

width is double and therefore the size of the input is reduced to half. For deeper networks like ResNet50, ResNet152, etc., bottleneck design is employed. For each residual function F , 3 layers are stacked one over the opposite. The three layers are 1×1 , 3×3 , 1×1 convolutions. The 1×1 convolution layers are liable for reducing then restoring the size. The 3×3 layer is left as a bottleneck with smaller input or output dimensions. At last, the network has a mean Pooling layer followed by a totally connected layer having 1000 neurons (ImageNet class output).

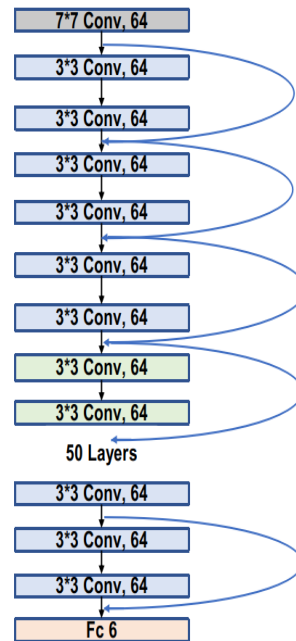


Figure 4.4 Resnet architecture

4.3.4 Web application development

ReactJS is JavaScript library used for constructing reusable UI components. Consistent with React official documentation, following is that the definition-React could also be a library for building composable user interfaces. It

promote the creation of reusable UI components, which current data that get modify over time. Many people use React because the V in MVC. React abstracts away the DOM from you, suggesting a simpler programming model and great performance. React also can render on the server usage of Node, and it can power native apps usage of React Native. React implements one-way reactive data flow, which decrease the boilerplate and is simpler to reason about than usual data binding.

JSX

JSX stands for JavaScript XML. JSX allows us to write down HTML in React. JSX makes it easier to write down and add HTML in React. It is faster than normal JavaScript because it performs optimizations while translating to regular JavaScript. It makes easier for us to make templates. Rather than separating the mark-up and logic in separated files, React uses components for this purpose. We'll study components in details in further articles.

Components

Components are independent and reusable bits of code. They serve an equivalent purpose as JavaScript functions, but add isolation and returns HTML via a render function. Components have two types, Class components and Function components, in this project we will concentrate on Class components.

Unidirectional data flow and Flux

React implements one-way data flow which makes it easy to reason about your app. Flux is a pattern that helps keeping your data unidirectional.

Advantages of React Js

- Uses virtual DOM which may be a JavaScript object. this may improve apps performance, since JavaScript virtual DOM is quicker than the regular DOM.
- Can be used on client and server side also like other frameworks.
- Component and data patterns improve readability, which helps to take care of larger apps.

4.4 SOFTWARE DESCRIPTION

4.4.1 Visual studio

In this project the Microsoft visual studio is used as an IDE. Visual Studio Code combines the simplicity of a ASCII text file editor along with powerful developer tooling, such as IntelliSense code completion and debugging.

First and foremost, it's an editor that receives out of your way. The delightfully frictionless edit-build-debug cycle known as much less time fiddling with your environment, and longer executing on your ideas. Visual Studio Code helps macOS, Linux, and Windows - so you can hit the bottom running, no be counted the platform. At its heart, Visual Studio Code features a lightning speed ASCII text file editor, perfect for everyday use. With assist help for many languages, VS Code helps you be immediately productive with syntax highlighting, bracket-matching, auto-indentation, box-selection, snippets, and more. Intuitive keyboard shortcuts, convenient customization and community-contributed keyboard shortcut mappings let you navigate your code with ease. For serious coding, you will often benefit from equipment with greater code understanding than simply blocks of text. Visual Studio Code consists of built-in assist for IntelliSense code completion, rich

semantic code understanding and navigation, and code refactoring. And when the coding receives tough, the tough get debugging. Debugging is often the one feature that developer omit most in a leaner coding experience, so we made it happen. Visual Studio Code includes an interactive debugger, so you can step via ASCII text file , inspect variables, view call stacks, and execute commands in the console.VS Code also integrates with build and scripting tools to achieve function frequent tasks making daily workflows faster. VS Code can has help for Git so you can work with source control except leaving the editor along with viewing pending changes diffs.Customize every feature to your liking and install wide number of third-party extensions. While most situation work "out of the box" with no configuration, VS Code additionally grows with you, and we inspire you to optimize your experience to suit your unique needs.VS Code includes enriched built-in support for Node.js development with JavaScript and TypeScript, powered by using the same underlying technologies that drive Visual Studio. VS Code also include notable tooling for web technologies such as JSX/React, HTML, CSS, SCSS, Less, and JSON. Visual Studio Code is the combination of web, native, and language-specific technologies. Usage of Electron and VS Code combines web technologies such as JavaScript and Node.js along with the speed and flexibility of native apps. VS Code uses a newer , quicker model of an same industrial-strength HTML-based editor this has powered the" Monaco" cloud editor, Internet Explorer F12 Tools, and other projects. Additionally, VS Code makes use of a tools service that enables it to combine with many of the same technologies that power Visual Studio, inclusive of Roslyn for .NET, TypeScript, Visual Studio debugging engine, and more. The visual Studio Code includes a public extensibility model that helpful for developers to build and use extensions, richly personalize their edit-build-debug experience.

4.4.2 Python

In this project, we used python as the programming language. Python is an object-oriented, high-level programming language. It integrated with dynamic semantics primarily for web and app development. It offers dynamic typing and dynamic binding option. It is extremely attractive field in the Rapid development.

Python is comparatively simple, so it is easy to find out since it requires a singular syntax that focuses on readability. Developers can able to read and translate Python code easier than other languages. In turn, this reduces the value of program maintenance and development because it allows teams to figure collaboratively without significant language and knowledge barriers. Python helps to utilization of modules and packages, which suggests that programs are often designed during a modular style and code are often reused across a spread of projects. One of the foremost promising benefits of Python is that both the quality library and therefore the interpreter are available freed from charge, in both binary and source form. There is no exclusivity either, as Python and every one the required tools are available on all major platforms. Therefore, it's an attractive option for developers who don't need to stress about paying high development costs.

CHAPTER 5

RESULT AND DISCUSSION

5.1 FINAL RESULTS OBTAINED

Dataset collection involves the process of collecting scanned images of the retinal eye scan dataset. Various datasets were collected and one example among the collected dataset can be found below.

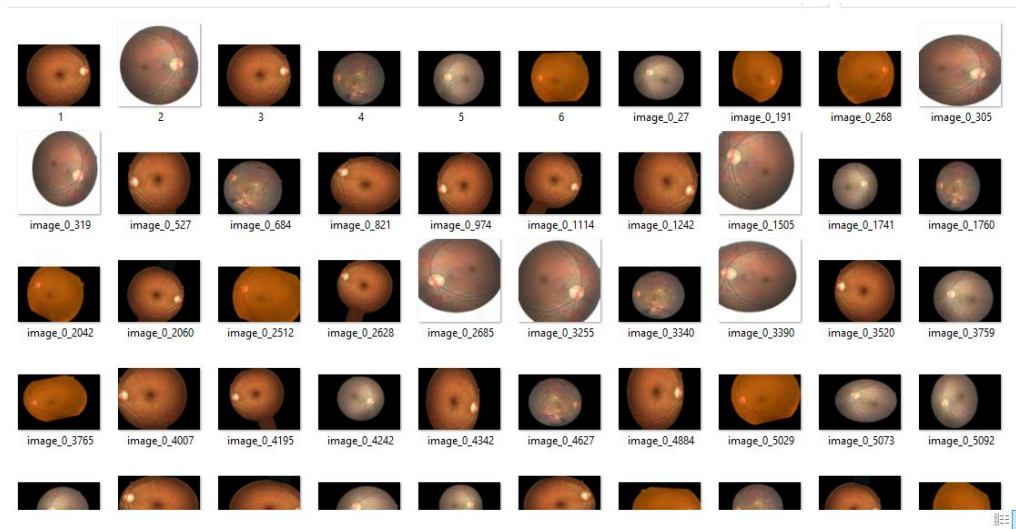


Figure 5.1 Dataset Collected

These datasets are then preprocessed to form an equal aspect ratio so that it can be made ready for training with the model. The below image shows the effective dataset separation of all the image dataset procured.

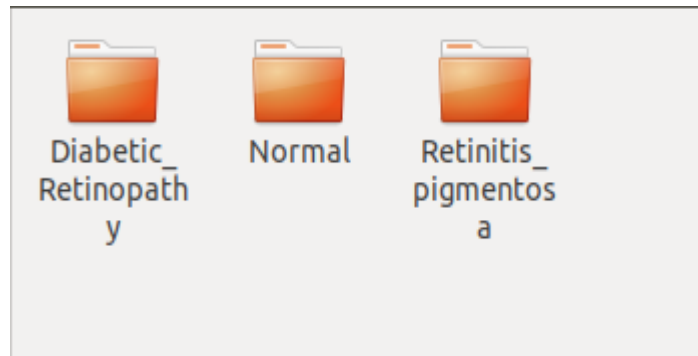


Figure 5.2 Data separation

After this the final implementation is done where the training process takes place and the results are obtained. Initially the anaconda navigator is started which is used to perform all the required execution of the project. This can be seen in the following figure:

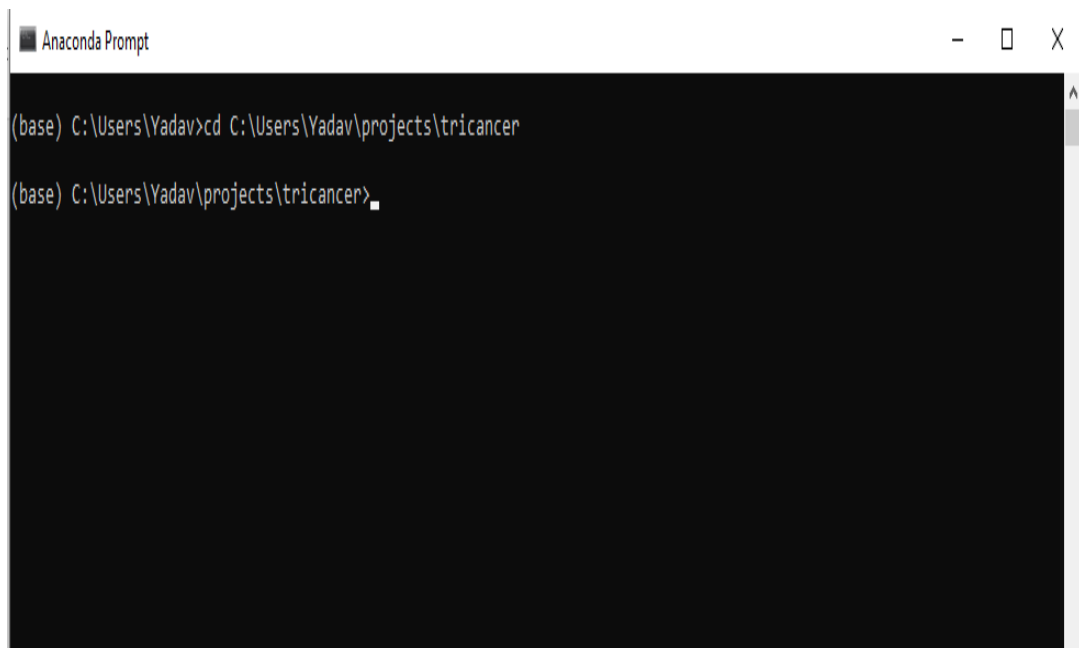


Figure 5.3 Moving Inside the coding folder

Then the code is run using certain commands in the command prompt which can be seen in the following figure:

```

(base) C:\Users\HP\project\tricancers\Dataset_Augmentation>python augmentation_demo.py --image 1.jpeg --output output
[INFO] loading images...
[INFO] processed 500/1256
[INFO] processed 1000/1256
WARNING:tensorflow:From C:\Users\HP\Anaconda3\lib\site-packages\tensorflow\python\ops\runtime_variant_ops.py:435: colocate_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
Downloading data from https://github.com/fchollet/deep-learning-models/releases/download/v0.1/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
Traceback (most recent call last):
  File "C:\Users\HP\Anaconda3\lib\urllib\request.py", line 1317, in do_open
    encode_chunked=req.has_header('Transfer-encoding'))
  File "C:\Users\HP\Anaconda3\lib\http\client.py", line 1244, in request
    self._send_request(method, url, body, headers, encode_chunked)
  File "C:\Users\HP\Anaconda3\lib\http\client.py", line 1290, in _send_request
    self.send_headers(body, encode_chunked=encode_chunked)
  File "C:\Users\HP\Anaconda3\lib\http\client.py", line 1239, in send_headers
    self.send_output(message_body, encode_chunked=encode_chunked)
  File "C:\Users\HP\Anaconda3\lib\http\client.py", line 1826, in send_output
    self.send(msg)
  File "C:\Users\HP\Anaconda3\lib\http\client.py", line 966, in send
    self.connect()
  File "C:\Users\HP\Anaconda3\lib\http\client.py", line 1406, in connect
    super().connect()
  File "C:\Users\HP\Anaconda3\lib\http\client.py", line 938, in connect
    (self.host,self.port), self.timeout, self.source_address)
  File "C:\Users\HP\Anaconda3\lib\socket.py", line 707, in create_connection
    for res in getaddrinfo(host, port, 0, SOCK_STREAM):
  File "C:\Users\HP\Anaconda3\lib\socket.py", line 748, in getaddrinfo
    for res in _socket.getaddrinfo(host, port, family, type, proto, flags):
socket.gaierror: [Errno 11001] getaddrinfo failed

During handling of the above exception, another exception occurred:

```

Figure 5.4 Executing In the Command Prompt

As the execution gets completed the accuracy is been plotted for the training process obtained for diabetic retinopathy and retinitis pigmentosa obtained and it can be seen in the following figure:

```
[INFO] evaluating after fine-tuning...
```

	precision	recall	f1-score	support
Diabetic_Retinopathy	1.00	1.00	1.00	13
Normal	1.00	1.00	1.00	22
Retinitis_pigmentosa	1.00	1.00	1.00	14
accuracy			1.00	49
macro avg	1.00	1.00	1.00	49
weighted avg	1.00	1.00	1.00	49

```
[INFO] serializing model...
```

Figure 5.5 Training process

The below image shows the training loss and accuracy graph obtained for retinitis pigmentosa and diabetic retinopathy prediction:

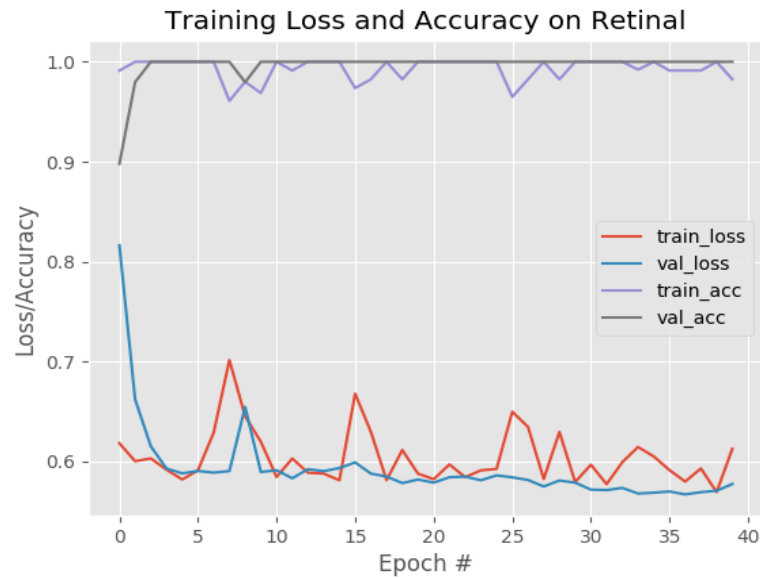


Figure 5.6 Training loss graph

A web application is developed using a JavaScript framework reactJS for interactively uploading retinal eye scan images and viewing the results. The below image shows the login page of the web application:

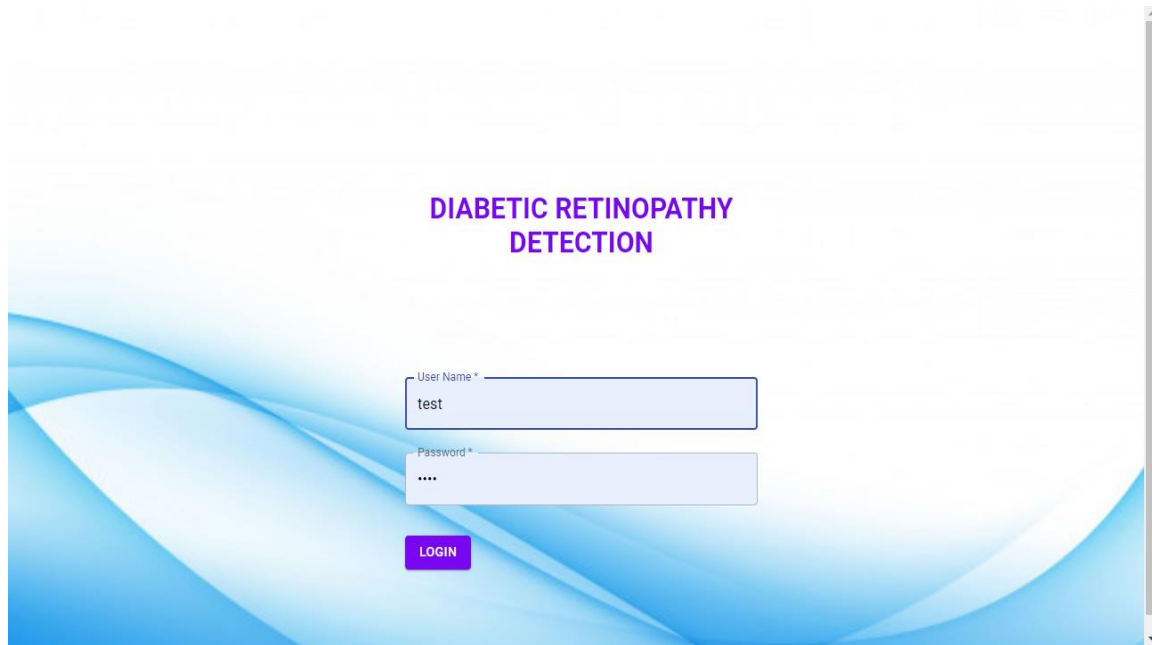


Figure 5.7 Login page of the web application

On successful login the home page is displayed to the user. The below image shows the user uploading the eye scan image for effective detection of retinitis pigmentosa and diabetic retinopathy.

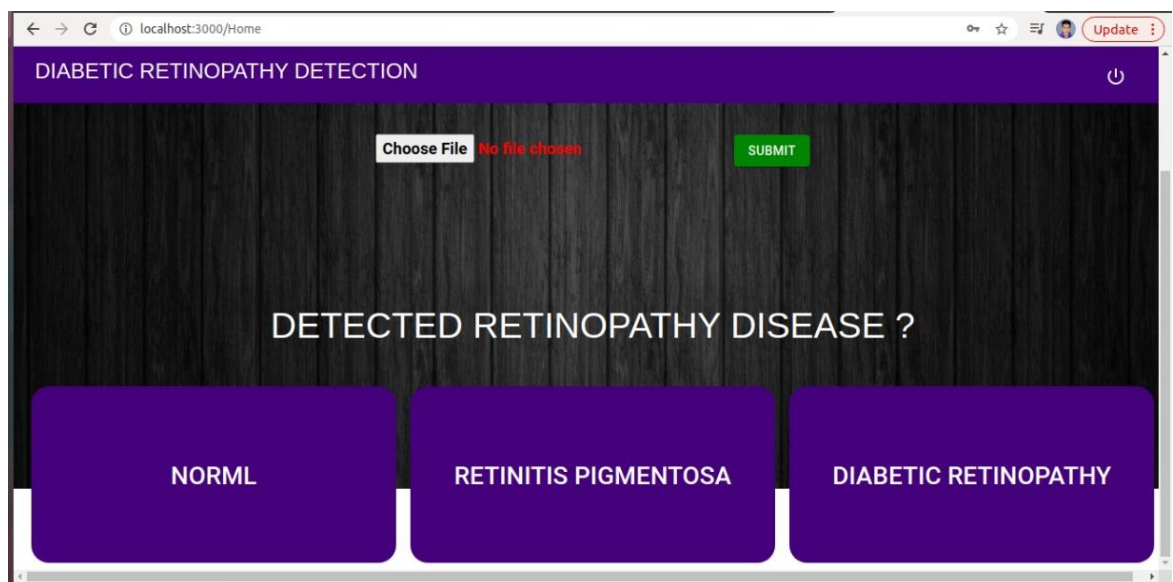


Figure 5.8 Home page

The user can upload eye scan images to check for retinitis pigmentosa and diabetic retinopathy diagnosis. The below image shows the user uploading eye scan image for effective diagnosis:

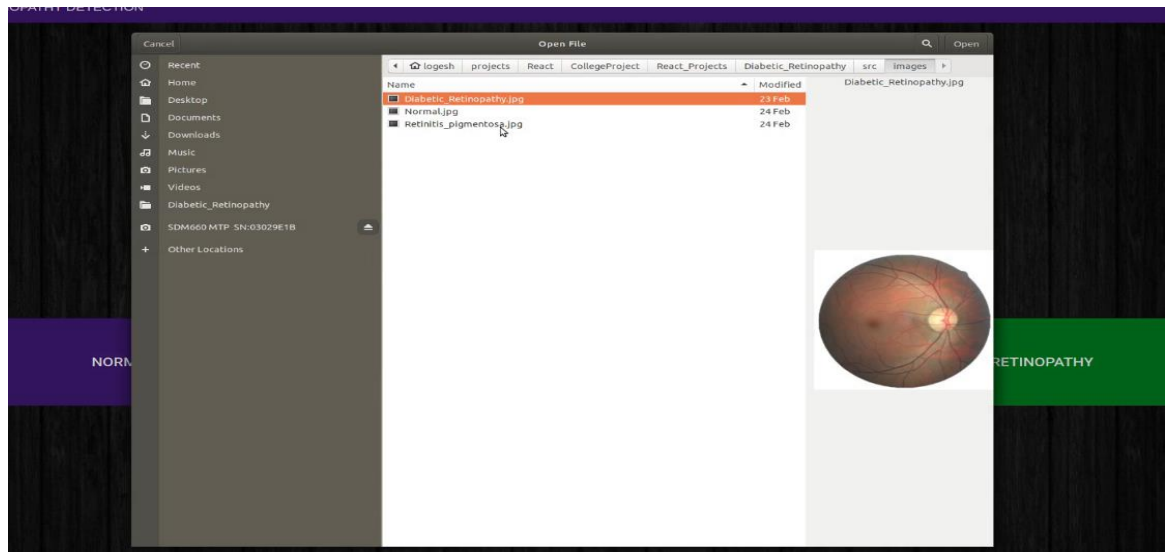


Figure 5.9 Retinal scan upload

The uploaded image is then sent to the database and the response is given back to the web page which are the results. The below image shows that the uploaded Image is of a normal eye scan.

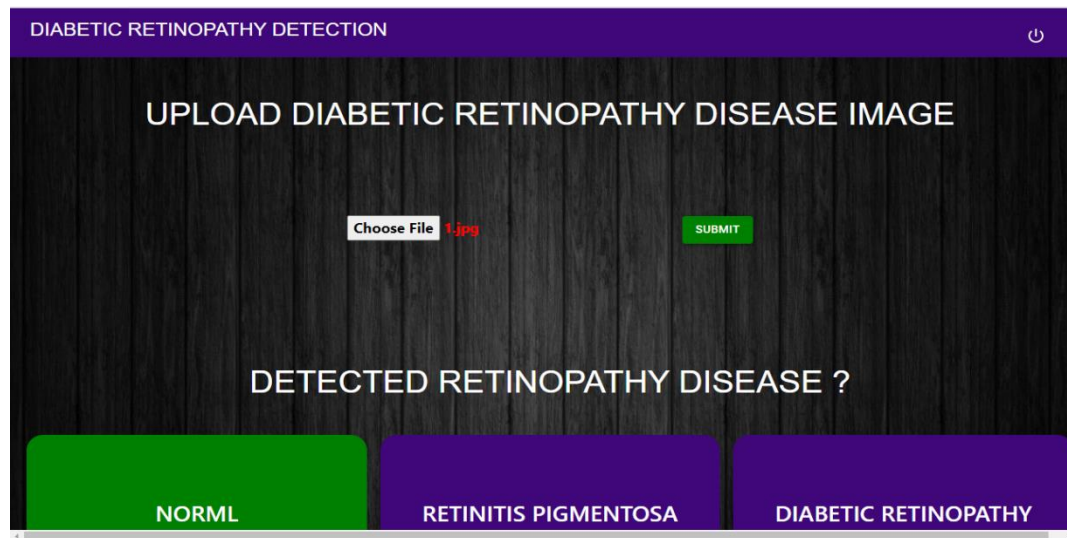


Figure 5.10 Normal eye scan diagnosis

If the uploaded image is that of an eye scan of a diabetic retinopathy is mentioned along with it. The below image shows that the uploaded image is that of a diabetic retinopathy.

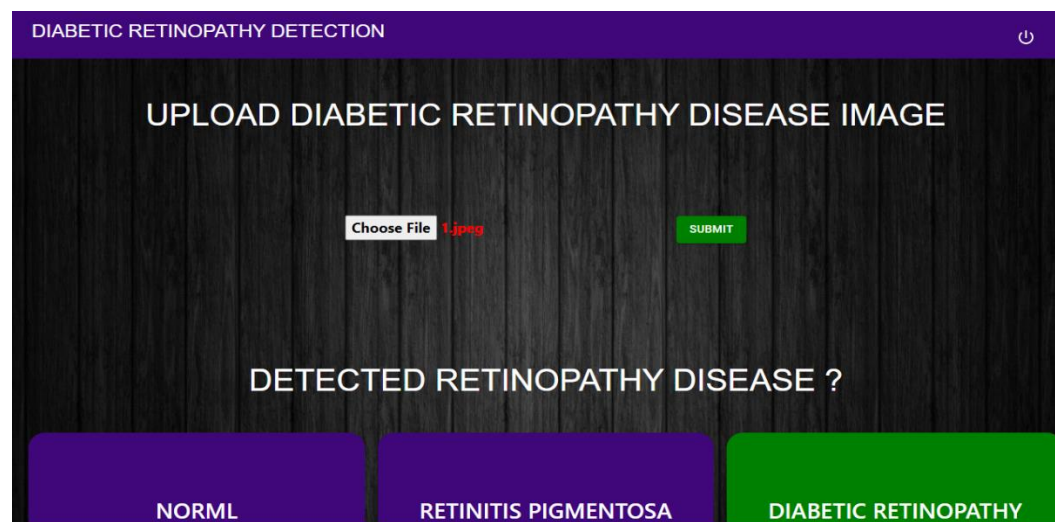


Figure 5.11 Diabetic retinopathy diagnosis

If the uploaded image is that of an eye scan of a retinitis pigmentosa is mentioned along with it. The below image shows that the uploaded image is that of a retinitis pigmentosa.

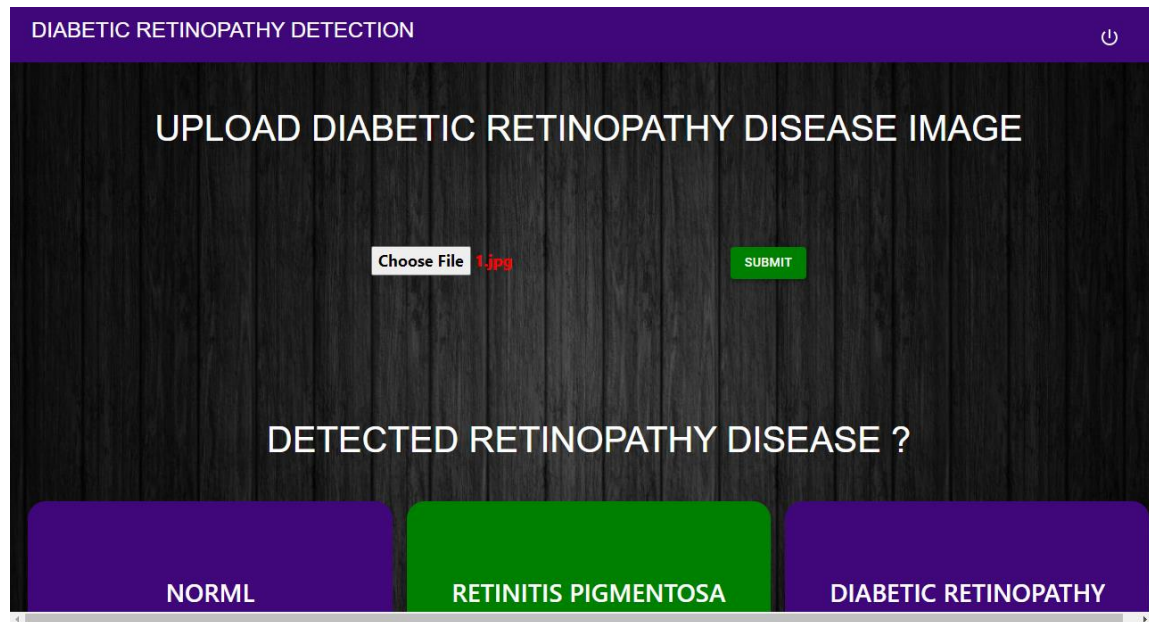


Figure 5.12 Retinitis pigmentosa diagnosis

5.2 DISCUSSION

Thus, from the above results and discussion, it is clear that we have efficiently made a project for performing diagnosis for retinitis pigmentosa and diabetic retinopathy. Thus, we have successfully implemented the scope of the project. Provides a technological solution to effectively facilitating better identification of the occurrence of diabetic retinopathy as well as retinitis pigmentosa. Facilitates the doctors to provide proper treatment to the patient. The accuracy of detection is very high.

CHAPTER 6

SUMMARY AND CONCLUSION

6.1 SUMMARY

In this project, we are going to determine occurrence of diabetic retinopathy and retinitis pigmentosa. . We train that datasets by extracting the features using an effective deep learning algorithm called the Resnet algorithm. Thus, this method provides an effective and cheap method to determine the presence of diabetic retinopathy or retinitis pigmentosa.

6.2 CONCLUSION

The project has been successfully implemented to perform diagnosis for diabetic retinopathy and retinitis pigmentosa and provide prior measures to avoid the disease. This also help in providing efficient treatment in a most cheap way and eventually reduce the time required for finding retinal diseases. In the current state it is done manually which consumes more time and also involves human error rate. So, by this project reduces the time required for manual classification and eliminates the human error rate by this project.

REFERENCE

1. Bogunovic .H et al., "RETOUCH: The Retinal OCT Fluid Detection and Segmentation Benchmark and Challenge", in IEEE Transactions on Medical Imaging, 2019, vol. 38, no. 8, pp. 1858-1874.
2. Ekatpure .S and R. Jain, "Red Lesion Detection in Digital Fundus Image Affected by Diabetic Retinopathy", Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 2018, pp.1-4.
3. Fu .H et al., "Segmentation and Quantification for Angle-Closure Glaucoma Assessment in Anterior Segment OCT," in IEEE Transactions on Medical Imaging, 2017, vol. 36, no. 9, pp. 1930-1938.
4. Gorczynska .L, J. V. Migacz, R. S. Jonnal, R. J. Zawadzki and J. S. Werner, "Influence of speed and resolution on OCT angiography and Doppler OCT imaging in human retinal and choroidal capillary systems", IEEE Photonics Conference (IPC), Waikoloa, HI, USA, 2016, pp. 140-141.
5. Han .X, Y. Sun and Y. Chen, "Residual Component Estimating CNN for Image Super-Resolution", IEEE Fifth International Conference on Multimedia Big Data (BigMM), Singapore, 2019, pp. 443-447.
6. Herliana .A, T. Arifin, S. Susanti and A. B. Hikmah, "Feature Selection of Diabetic Retinopathy Disease Using Particle Swarm Optimization and Neural Network", 6th International Conference on Cyber and IT Service Management (CITSM), Parapat, Indonesia, 2018, pp. 1-4.
7. Hui .L and S. Yu-jie, "Research on face recognition algorithm based on improved convolution neural network", 13th IEEE Conference on Industrial Electronics and Applications (ICIEA), Wuhan, China, 2018, pp. 2802-2805.
8. Kannoja .S.P and G. Jaiswal, "Ensemble of Hybrid CNN-ELM Model for Image Classification", 5th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 2018, pp. 538-541.
9. Kepp .T, J. Ehrhardt, M. P. Heinrich, G. Hüttmann and H. Handels, "Topology-Preserving Shape-Based Regression Of Retinal Layers In Oct Image Data Using Convolutional Neural Networks", IEEE 16th International Symposium on Biomedical Imaging , Venice, Italy, 2019, pp. 1437-1440.
10. Lee .P, C. Chan, S. Huang, A. Chen and H. H. Chen, "Blood vessel extraction from OCT data by short-time RPCA," IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 2016, pp. 394-398.
11. Liu .J et al., "Saliency-based image processing for retinal prosthesis", International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), Shanghai, China, 2019, pp. 297-297.
12. Matveyev .A.L et al., "Multi-factor modeling of OCT-scan formation in the presence of scatterer motions", International Conference Laser Optics (ICLO), St. Petersburg, Russia, 2020, pp. 1-1.
13. Najeeb .S, N. Sharmile, M. S. Khan, I. Sahin, M. T. Islam and M. I. Hassan Bhuiyan, "Classification of Retinal Diseases from OCT scans using

- Convolutional Neural Networks", 10th International Conference on Electrical and Computer Engineering (ICECE), Dhaka, Bangladesh, 2018, pp. 465-468.
14. Nath .S .S, B. K. Anoop and P. Sankar, "Classification of Outer Retinal Layers Based on KNN-Classfier", International Conference on Emerging Trends and Innovations In Engineering And Technological Research (ICETIETR), Ernakulam, 2018, pp. 1-4.
 15. Naz .S, T. Hassan, M. U. Akram and S. A. Khan, "A practical approach to OCT based classification of Diabetic Macular Edema", International Conference on Signals and Systems (ICSigSys), Sanur, 2017, pp. 217-220.
 16. Novosel .J, S. Yzer, K. A. Vermeer and L. J. van Vliet, "Segmentation of Locally Varying Numbers of Outer Retinal Layers by a Model Selection Approach", in IEEE Transactions on Medical Imaging, vol. 36, no. 6, pp. 1306-1315, June 2017.
 17. Perdomo .O, S. Otálora, F. A. González, F. Meriaudeau and H. Müller, "OCT-NET: A convolutional network for automatic classification of normal and diabetic macular edema using sd-oct volumes" ,IEEE 15th International Symposium on Biomedical Imaging, Washington, DC, USA, 2018, pp. 1423-1426.
 18. Qiao .L, Y. Zhu and H. Zhou, "Diabetic Retinopathy Detection Using Prognosis of Microaneurysm and Early Diagnosis System for Non-Proliferative Diabetic Retinopathy Based on Deep Learning Algorithms", in IEEE Access, vol. 8, pp. 104292-104302, 2020.
 19. Qummar .S et al., "A Deep Learning Ensemble Approach for Diabetic Retinopathy Detection", in IEEE Access, vol. 7, pp. 150530-150539, 2019.
 20. Soltan .A et al., "Wearable Glasses for Retinal Pigmentiosa Based on Optogenetics", IEEE International Symposium on Circuits and Systems (ISCAS), Sapporo, Japan, 2019, pp. 1-5.
 21. Soltan .A, Y. Liu, N. Armstrong, M. Akhter, B. Corbett and P. Degenaar, "Comparison between Different Optical Systems for Optogenetics based Head Mounted Device for Retina Pigmentosa", 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Berlin, Germany, 2019, pp. 382-385.
 22. Tong .W, M. Stamp, M. Hejazi, D. Garrett, S. Prawer and M. R. Ibbotson, "The Effects of Phase Durations on the Spatial Responses of Retinal Ganglion Cells to Epi- and Sub-Retinal Electrical Stimulation", 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Berlin, Germany, 2019, pp. 1795-1800.
 23. Vahadane .A, A. Joshi, K. Madan and T. R. Dastidar, "Detection of diabetic macular edema in optical coherence tomography scans using patch based deep learning", IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), Washington, DC, 2018, pp. 1427-1430.
 24. Vasilyeva .T, V. Kadyshev, A. Marakhonov and R. Zinchenko, "DNA Diagnosis in Case Series of Hereditary Retinal Dystrophy", Cognitive Sciences, Genomics and Bioinformatics (CSGB), Novosibirsk, Russia, 2020, pp. 101-105.

25. Vidyasari .R, I. Sovani, T. L. R. Mengko and H. Zakaria, "Vessel enhancement algorithm in digital retinal fundus microaneurysms filter for nonproliferative diabetic retinopathy classification", 2nd International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering, Bandung, Indonesia, 2011, pp. 278-281.
26. Xi .R, M. Hou, M. Fu, H. Qu and D. Liu, "Deep Dilated Convolution on Multimodality Time Series for Human Activity Recognition", International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 2018, pp. 1-8.
27. Zang .P et al., "DcardNet: Diabetic Retinopathy Classification at Multiple Levels Based on Structural and Angiographic Optical Coherence Tomography", in IEEE Transactions on Biomedical Engineering, 2018.

APPENDIX

USAGE

python3 food.py --dataset dataset/images --model cavity.model

import the necessary packages

from sklearn.preprocessing import LabelBinarizer

from sklearn.model_selection import train_test_split

from sklearn.metrics import classification_report

from pyimagesearch.preprocessing import ImageToArrayPreprocessor

from pyimagesearch.preprocessing import AspectAwarePreprocessor

from pyimagesearch.datasets import SimpleDatasetLoader

from pyimagesearch.nn.conv import FCHeadNet

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.optimizers import RMSprop

from tensorflow.keras.optimizers import SGD

from tensorflow.keras.optimizers import Adam

from tensorflow.keras.applications import VGG16

from tensorflow.keras.layers import Input

from tensorflow.keras.models import Model

from imutils import paths

import numpy as np

import argparse

```

import os

# construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-d", "--dataset", required=True,
                help="path to input dataset")
ap.add_argument("-m", "--model", required=True,
                help="path to output model")
args = vars(ap.parse_args())

# construct the image generator for data augmentation
aug = ImageDataGenerator(rotation_range=30, width_shift_range=0.1,
                          height_shift_range=0.1, shear_range=0.2, zoom_range=0.2,
                          horizontal_flip=True, fill_mode="nearest")

# grab the list of images that we'll be describing, then extract
# the class label names from the image paths
print("[INFO] loading images...")
imagePaths = list(paths.list_images(args["dataset"]))
classNames = [pt.split(os.path.sep)[-2] for pt in imagePaths]
classNames = [str(x) for x in np.unique(classNames)]

# initialize the image preprocessors

```

```

aap = AspectAwarePreprocessor(224, 224)

iap = ImageToArrayPreprocessor()

# load the dataset from disk then scale the raw pixel intensities to
# the range [0, 1]

sdl = SimpleDatasetLoader(preprocessors=[aap, iap])

(data, labels) = sdl.load(imagePaths, verbose=500)

data = data.astype("float") / 255.0

# partition the data into training and testing splits using 75% of
# the data for training and the remaining 25% for testing

(trainX, testX, trainY, testY) = train_test_split(data, labels,
                                                    test_size=0.25, random_state=42)

# convert the labels from integers to vectors

trainY = LabelBinarizer().fit_transform(trainY)

testY = LabelBinarizer().fit_transform(testY)

# load the VGG16 network, ensuring the head FC layer sets are left
# off

baseModel = VGG16(weights="imagenet", include_top=False,
                    input_tensor=Input(shape=(224, 224, 3)))

```

```

# initialize the new head of the network, a set of FC layers

# followed by a softmax classifier

headModel = FCHeadNet.build(baseModel, len(classNames), 256)


# place the head FC model on top of the base model -- this will
# become the actual model we will train

model = Model(inputs=baseModel.input, outputs=headModel)


# loop over all layers in the base model and freeze them so they
# will *not* be updated during the training process

for layer in baseModel.layers:

    layer.trainable = False


# compile our model (this needs to be done after our setting our
# layers to being non-trainable

print("[INFO] compiling model...")

opt = Adam(lr=0.001)

model.compile(loss="categorical_crossentropy", optimizer=opt,
              metrics=["accuracy"])


# train the head of the network for a few epochs (all other
# layers are frozen) -- this will allow the new FC layers to
# start to become initialized with actual "learned" values

```

```

# versus pure random

print("[INFO] training head...")

model.fit_generator(aug.flow(trainX, trainY, batch_size=32),
                    validation_data=(testX, testY), epochs=15,
                    steps_per_epoch=len(trainX) // 32, verbose=1)

# evaluate the network after initialization

print("[INFO] evaluating after initialization...")

predictions = model.predict(testX, batch_size=32)

print(classification_report(testY.argmax(axis=1),
                            predictions.argmax(axis=1), target_names=classNames))

# now that the head FC layers have been trained/initialized, lets
# unfreeze the final set of CONV layers and make them trainable
for layer in baseModel.layers[15:]:
    layer.trainable = True

# for the changes to the model to take affect we need to recompile
# the model, this time using SGD with a *very* small learning rate
print("[INFO] re-compiling model...")

opt = SGD(lr=0.0001)

model.compile(loss="categorical_crossentropy", optimizer=opt,
              metrics=["accuracy"])

```

```

# train the model again, this time fine-tuning *both* the final set
# of CONV layers along with our set of FC layers

print("[INFO] fine-tuning model...")

model.fit_generator(aug.flow(trainX, trainY, batch_size=32),
                    validation_data=(testX, testY), epochs=10,
                    steps_per_epoch=len(trainX) // 32, verbose=1)

# evaluate the network on the fine-tuned model

print("[INFO] evaluating after fine-tuning...")

predictions = model.predict(testX, batch_size=32)

print(classification_report(testY.argmax(axis=1),
                             predictions.argmax(axis=1), target_names=classNames))

# save the model to disk

print("[INFO] serializing model...")

model.save(args["model"])

# USAGE

# python3 recognize.py --image 1.jpg --conf config/config.json

# import the necessary packages

from pyimagesearch.utils import Conf

```

```

#from pyimagesearch.notifications import TwilioNotifier

from imutils.video import VideoStream

from imutils import paths

from threading import Thread

from tensorflow.keras.preprocessing.image import img_to_array

from tensorflow.keras.models import load_model

from datetime import datetime

from datetime import date

#import RPi.GPIO as GPIO

import numpy as np

import argparse

import imutils

import pickle

import time

import cv2

import os


font = cv2.FONT_HERSHEY_SIMPLEX


# construct the argument parser and parse the arguments

ap = argparse.ArgumentParser()

ap.add_argument("-i", "--image", required=True,

```

```

        help="path to the input image")

ap.add_argument("-c", "--conf", required=True,
                help="path to the input configuration file")

args = vars(ap.parse_args())

# load the configuration file and initialize the Twilio notifier
conf = Conf(args["conf"])
#tn = TwilioNotifier(conf)

# grab the paths to gesture icon images and then initialize the icons
# dictionary where the key is the gesture name (derived from the image
# path) and the key is the actual icon image
print("[INFO] loading icons...")
imagePaths = paths.list_images(conf["assets_path"])
icons = {}

# loop over the image paths
for imagePath in imagePaths:
    # extract the gesture name (label) the icon represents from the
    # filename, load the icon, and then update the icons dictionary
    label = imagePath.split(os.path.sep)[-1].split(".")[0]
    icon = cv2.imread(imagePath)

```

```

icons[label] = icon

# grab the top-left and and bottom-right (x, y)-coordinates for the
# gesture capture area
TOP_LEFT = tuple(conf["top_left"])
BOT_RIGHT = tuple(conf["bot_right"])

# load the trained gesture recognizer model and the label binarizer
print("[INFO] loading model...")
model = load_model(str(conf["model_path"]))
#lb = pickle.loads(open(str(conf["lb_path"]), "rb").read())

# start the video stream thread
print("[INFO] starting video stream thread...")
#vs = VideoStream(src=0).start()
#vs = VideoStream(usePiCamera=True).start()
#time.sleep(2.0)

#frame = load_img(args["image"])
#cv2

# loop over frames from the video stream
while True:

    # grab the frame from the threaded video file stream and grab the

```

```

# current timestamp

imagePaths = list(paths.list_images("dataset/images"))

classNames = [pt.split(os.path.sep)[-2] for pt in imagePaths]

classNames = [str(x) for x in np.unique(classNames)]

frame = cv2.imread(args["image"])

timestamp = datetime.now()


# resize the frame and then flip it horizontally

frame = imutils.resize(frame, width=400)

frame = cv2.flip(frame, 1)


# clone the original frame and then draw the gesture capture area

#clone = frame.copy()

#cv2.rectangle(clone, TOP_LEFT, BOT_RIGHT, (0, 0, 255), 2)


#roi = frame[TOP_LEFT[1]:BOT_RIGHT[1], TOP_LEFT[0]:BOT_RIGHT[0]]

# roi = cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY)

# roi = cv2.threshold(roi, 75, 255, cv2.THRESH_BINARY)[1]

#visROI = roi.copy()


# now that we have the hand region we need to resize it to be

# the same dimensions as what our model was trained on, scale

```

```

        # it to the range [0, 1], and prepare it for classification

roi = cv2.resize(frame, (224, 224))

roi = roi.astype("float") / 255.0

roi = img_to_array(roi)

roi = np.expand_dims(roi, axis=0)


        # classify the input image

proba = model.predict(roi)[0]

label = classNames[proba.argmax()]


# if label == "images" or "plates":


#     print("No Food")


# elif label == "chapathi" :


print(label)

if label == "idly":

    cv2.putText(frame,'Each Idly gives 33 calories',(0,20), font, 0.5, (0,0,0), 1,
cv2.LINE_AA)

elif label == "dosa":

    cv2.putText(frame,'Each dosa gives 133 calories',(0,20), font, 0.5, (0,0,0),
1, cv2.LINE_AA)

```

```

elif label == "rice":

    cv2.putText(frame,'Each bowl of rice gives 136 calories',(0,20), font, 0.5,
(0,0,0), 1, cv2.LINE_AA)

elif label == "chapathi":

    cv2.putText(frame,'Each chapati gives 104 calories',(0,20), font, 0.5,
(0,0,0), 1, cv2.LINE_AA)


# check to see if the label from our model matches the label


cv2.imshow("ROI", frame)

# cv2.imshow("Security Feed", clone)

# cv2.imshow("Passcode", canvas)

key = cv2.waitKey(1) & 0xFF


# if the `q` key was pressed, break from the loop

if key == ord("q"):

    reset_lights()

    break


# do a bit of cleanup

```

`cv2.destroyAllWindows()`

`vs.stop()`