OPTIC NERVE LOCALIZATION IN THE RETINAL FUNDUS IMAGES OF EYE

A major project

Submitted in partial fulfilment of the requirements for

The award of the Degree Of

Master of Computer Application (MCA)

By

Name	**Enrollment no.**
Sanjukta Mukherjee	12022010010032
Saikat Santra	12022010010033
Abhrajit Kar	12022010010047

Under the guidance of

Mr. Supratim Ghosh

MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY WEST BENGAL

Utech

In Pursuit Of Knowledge And Excellence

DEPARTMENT OF COMPUTER APPLICATION

INSTITUTE OF ENGINEERING AND MANAGEMENT, 2024

OPTIC NERVE LOCALIZATION IN THE RETINAL FUNDUS IMAGES OF EYE

A major project

Submitted in partial fulfilment of the requirements for

The award of the Degree Of

Master of Computer Application (MCA)

By

Name	**Enrollment no.**
Sanjukta Mukherjee	12022010010032
Saikat Santra	12022010010033
Abhrajit Kar	12022010010047

Under the guidance of

Mr. Supratim Ghosh

MAULANA ABUL KALAM AZAD
UNIVERSITY OF TECHNOLOGY,
WEST BENGAL

DEPARTMENT OF COMPUTER APPLICATION

INSTITUTE OF ENGINEERING AND MANAGEMENT, 2024

DECLARATION CERTIFICATE

This is to certify that the work presented in the document entitled "Optic nerve localization in the retinal fundus images of eye" in partial fulfilment of the requirement for the award of degree of Master of Computer Application of Institute of Engineering & Management is an authentic work carried out under my supervision and guidance.

To the best of my knowledge the content of this document does not form a basis for the award of any previous Degree to anyone else.

Date:

Head of Dept.	**Mr. Supratim Ghosh**
Dept. of Computer Application	Dept. of Computer Application
Institute of Engineering and Management	Institute of Engineering and Management

CERTIFICATE OF APPROVAL

The forgoing thesis **“Optic nerve localization in the retinal fundus images of eye”** is hereby approved as a creditable study of research topic and has been presented in satisfactory manner to warrant its acceptance as prerequisite to the degree for which it has been submitted.

It is understood that by this approval, the undersigned do not necessarily endorse any conclusion drawn or opinion expressed therein but approve the work for the purpose for which it is submitted.

Viva-Voice held on…………………

(Internal Examiner) **(External Examiner)**

ACKNOWLEDGEMENT

We would like to express our special thanks of gratitude to our Guide, **Mr. Supratim Ghosh** who helped us a lot in this project, his valuable suggestions helped us to solve tough challenges and without his help this project could not have been completed in time. A special thanks to our Head of Department who gave us the golden opportunity to do this wonderful project on the topic **"Optic nerve localization in the retinal fundus images of eye"**, which helped us to gain a significant knowledge in the aforesaid subjects. Secondly, we would like to thank our friends who helped us a lot in finalizing this project within the given time frame.

Name	**Enrollment no.**
Saikat Santra	12022010010033
Abhrajit Kar	12022010010047
Sanjukta Mukherjee	12022010010032

ABSTRACT

The analysis of retinal images has been of great interest in the recent past. A lot of different techniques have been developed for image analysis including segmentation of the eye structures which proves to be a reliable solution for optical diagnosis. One such analysis task includes optic disc localization which is primarily beneficial for Ophthalmological and medical diagnosis. Although it is primarily beneficial for medical analysis it also can prove to be beneficial in non-medical fields such as computer vision, eye ball tracking, image processing etc.

The optic disc which is also known as the optic nerve head is a round section at the back of the eye. It is one of the most important parts of a human eye. The shape, size, position and other features of the optic disc can help diagnosing and monitoring various diseases of human eye including glaucoma, diabetic retinopathy and hypertensive retinopathy. The optic disc detection procedure can be automated using computer algorithms and different image processing techniques and machine learning algorithms.

To automate the process of medical diagnosis of eye diseases accurate optic disc detection is very essential. By accurately localizing the optic disc a lot of other features of the OD can be analysed. Features like shape, size, position can help in detecting abnormalities and tracking the changes in these features can help in early diagnosis of the diseases. This project aims towards building a U-Net model in which the optic disc is localized in the retinal fundus image of human eye. The work is divided into two parts: (a) The area of optic disc is detected and localized in ground-truth images and (b) a supervised learning algorithm is used to train a model which takes the greyscale images of the ground-truth annotations as input and returns a mask locating the area of binary disc.

TABLE OF ABBREVIATIONS

OD	Optic Disc
FOV	Field of view
FCNN	Fully convolutional neural network
FCN	Fully convolutional network
ReLU	Rectified Linear Unit
Adam	Adaptive Moment Estimation
BCE	Binary Cross Entropy
HRF	High resolution fundus
DRIVE	Digital Retinal Images for Vessel Extraction
ROI	Region of Interest
STARE	Structured Analysis of the retina
IoU	Intersection over Union
R-CNN	Region-based Convolutional Neural Network

TABLE OF CONTENTS

CHAPTER 1:

INTRODUCTION

The bright round section at the back of eye is known as the optic disc of the eye also known as the optic nerve head. At this position in the human eye the retina and optic nerve connect. The main artery and vein of the retina enters the eye through the OD. The retina is responsible for taking in the light that enters the eye and turn it into images that we see.

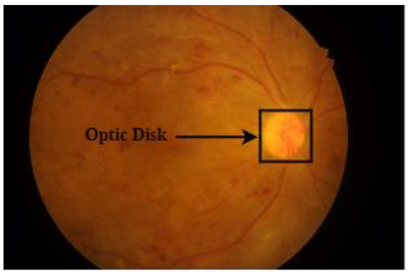

Fig 1.1: Fundus image of eye (IDRiD dataset)

The retina consists of two main parts: the macula and the peripheral retina. While macula is located at the centre of the eye it is responsible for seeing images directly at the front the peripheral retina which is responsible for peripheral vision makes up the rest of the retina.

The ROI or the OD is a bright raised area at the back of the eye is yellow – orange or pink in colour is located at the border of the macula and the peripheral retina. This part of the retina is also known as the blind spot due to the absence of rods and cones.

When light enters our eye it hits the back of the eye i.e., the retina. This light is taken by the photoreceptors and are turned into electrical signals. These signals through the retina reaches the optic disc and through the optic nerves it reaches our brain which enables us to see.

The small indentation at the centre of the optic disc where the optic nerve consisting of millions of fibres connects to the retina. The ground truth of the project is based on this criterion. To detect the optic disc, we locate the cluster of nerves which connect the optic disc to the retina. The binary image is pre-processed and converted into grayscale. The most intense region is considered as the target part.

Fig 1.2: Greyscale retinal fundus image with ground-truth annotation(DRIVE Dataset)

After finding the ground-truth a U-Net model is developed through which the corresponding greyscale images are passed and a mask locating the area of the optic disc is returned. A total of 9500 images and their corresponding masks are used for this project. 70% of these images were used for training and validation of the model and the rest 30% images were used for testing while developing the model. To find the real-life accuracy of the project different publicly available datasets are used and the predictions are compared with existing models.

CHAPTER 2:

OBJECTIVE

The OD is the entry point of the optic nerves which are responsible for carrying visual signals to the brain. The abnormalities in the shape, size, position and other features of the OD can indicate towards underlying retinal diseases. Early detection of these abnormalities in the features of the OD can help in early diagnosis of the diseases which can help in preventing partial vision loss or complete blindness.

Accurate OD localization is essential for automating the process of monitoring changes in the OD over time. Monitoring the changes in various features of the OD can provide valuable insights about the retinal diseases and the condition of the OD.

Our project aims at building a supervised learning model for automatic localization and segmentation of the optic disc in retinal fundus images. The main objective is to extract the optic disc from retinal fundus images accurately and efficiently, which is crucial for various medical applications such as diagnosing retinal diseases and monitoring ocular diseases. Therefore, the objectives of the project include:

- Detecting optic disc location in ground-truth annotations.
- Localizing the optic discs in retinal fundus images based on ground-truth.

In the project various image processing and deep learning techniques are implemented to achieve the objectives which efficiently localizes the optic disc in retinal fundus eye images.

CHAPTER 3:

LITERATURE REVIEW:

3.1 Background

OD localization is a very critical task in field of image processing and medical applications. The OD also known as the optic nerve head is important part of a human eye. Abnormalities in the OD indicate various retinal diseases and detection its position and features can help in early detection and diagnosis of these retinal diseases.

The process of OD localization can be automated using various machine learning algorithms and image processing techniques. Image processing techniques can help locating and returning the boundaries of the optic disc whereas the machine learning algorithms can help in accurately detecting the OD and automating the process.

For automating the localization process at first the detection of the OD needs to be accurate. Accurate detection can further help in feature extraction like size, shape, position etc which can help in detecting abnormalities in the OD and tracking them over time can help in early diagnosis.

3.2 OD Localization

“New optic disc localization approach in retinal images,” *IEEE Conference Publication IEEE Xplore*, Nov. 01, 2013. https://ieeexplore.ieee.org/document/6707418

This paper by Florin Rotaru et al., proposes an optic disc localization approach in retinal images. First the disc area is identified using a complex methodology.

Then from a histogram the pixels are classified into two classes "bright" and "dark" using a threshold. For every pixel in the image the texture measure and the pixel variance are computed. Lastly, the edges are extracted and a circular optic disc boundary is obtained by Hough transform.

"Optic Disc Localization using Interference Map and Localized Segmentation," *IEEE Conference Publication | IEEE Xplore*, Jul. 01, 2019. https://ieeexplore.ieee.org/document/8938221

The mentioned paper by Pratik Joshi et al., proposes method for OD localization and center prediction. Firstly, the green channels are extracted and a max inference map (IMP) is then generated highlighting the ROI. By bounding the ROI in a square region, a binary image specifically isolating the OD is obtained using threshold. The center of the region is marked as the OD center while the square region is marked as the OD.

"Automatic localization and segmentation of Optic Disc in retinal fundus images through image processing techniques," *IEEE Conference Publication | IEEE Xplore*, Apr. 01, 2014. https://ieeexplore.ieee.org/document/6996090

This paper by R.Geetha Ramani et al., proposes a framework for optic disc localization in retinal fundus images. A template is first created on the green channel fundus images based on the target image which serves as a reference for localization and segmentation. Cross Correlation is then performed between the source image and the template created by manually segmenting fundus images from gold standard database. The higher the correlation values the more the match The OD is marked finally by bounding it in a rectangle.

"Automatic localization of the optic disc based on iterative brightest pixels extraction," *IEEE Conference Publication | IEEE Xplore*, Jun. 01, 2014. https://ieeexplore.ieee.org/document/6845958

This paper by Chun Yuan Yu et al., proposes an OD localization method in three main steps: (a) preprocessing – where the noise is removed and gaussian filters are applied to enhance the optic disc region. (b) In this step the brightest pixels are iteratively selected and region is selected based on geometric figures. (c) The

OD is localized by locating the center points of selected region and identifying the OD.

“Automatic optic disc detection in digital fundus images using image processing techniques,” *IEEE Conference Publication | IEEE Xplore*, Feb. 01, 2014. https://ieeexplore.ieee.org/document/7034118

The paper "Automatic optic disc detection in digital fundus images using image processing techniques " by Snehal B.Akhade et al., uses Deep convolutional neural networks for image segmentation to detect Optic disc location. The paper uses the IDRiD (Indian Diabetic Retinopathy Image Dataset) for predicting the OD location. This paper also gives information about data augmentation , validation and prediction. The performance of the model is computed by comparing it with the existing labeled data and gives insights about future prospects of using CNN in medical image segmentation.

These papers represent a diverse range of approaches to optic disc detection and segmentation in fundus images including traditional image processing techniques, geometric models, evolutionary algorithms and deep learning methods. Our project demonstrates the process of developing a segmentation model using deep learning techniques. By leveraging the U-Net architecture and appropriate training strategies, accurate segmentation results are achieved.

CHAPTER 4:

DATASET PREPARATION

4.1 Mix-up Model

Dataset Mix-up is a data augmentation technique widely used in deep learning models. Mix-up dataset is used to improve generalization performance of the developed neural network models. Mix-up involves creating new data from existing data. For images the mix-up works by interpolating two original images to get a new one.

For the mix-up model multiple datasets were used namely Drive, Stare and HRF dataset. Images from these sources are interpolated, rotated and mirrored to get new medically sound image which can be used for training the model. Dataset Mix-up encourages the model to learn more robust and generalized decision boundaries by exposing it to a wider range of synthetic data points that lie along the interpolation paths between real data points.

4.2 Data augmentation

For the purpose of working on this project a large dataset was required to properly train the model and get accurate results. To increase the size of the initial dataset, data augmentation was performed on the initial dataset of fundus greyscale images and their corresponding binary images.

To perform data augmentation on the given dataset a set of steps are performed. The most important step is the detection of the optic disc in the binary image and then interpolating it. Two images are interpolated based on the positions of their regions of intertest. These images were then mirrored to artificially produce images for the training dataset.

Below is a flowchart depicting the dataset preparation process for this project.

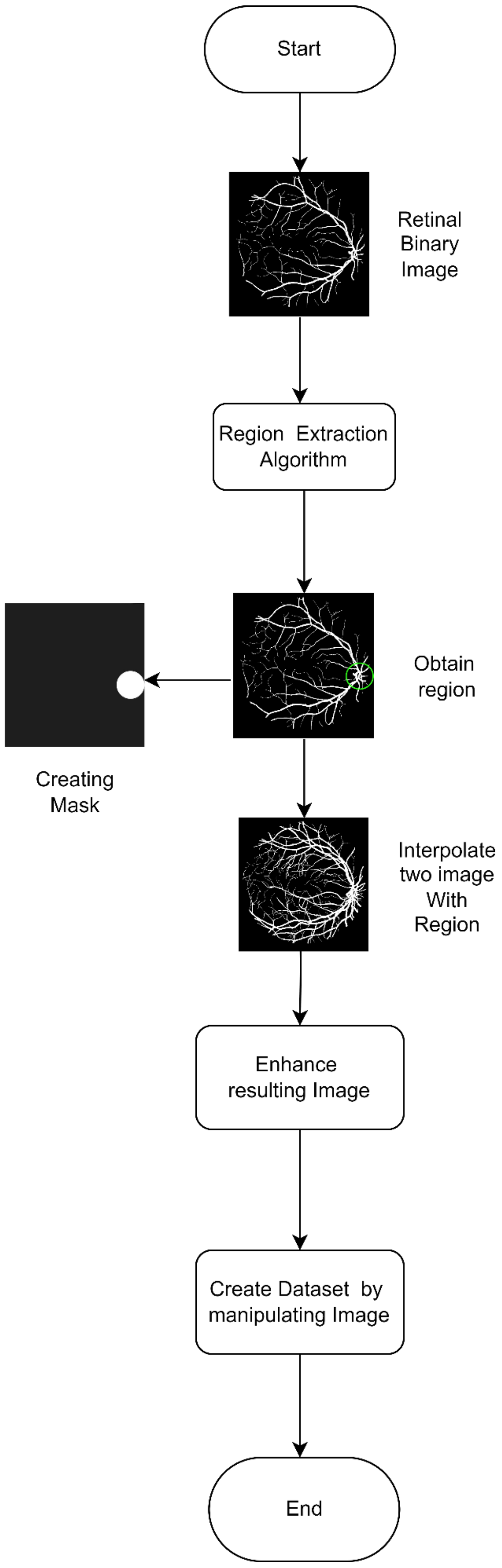
Start
Retinal Binary Image
Region Extraction Algorithm
Obtain region
Creating Mask
Interpolate two image With Region
Enhance resulting Image
Create Dataset by manipulating Image
End

Fig. 4.1: Flow chart representing steps of dataset preparation

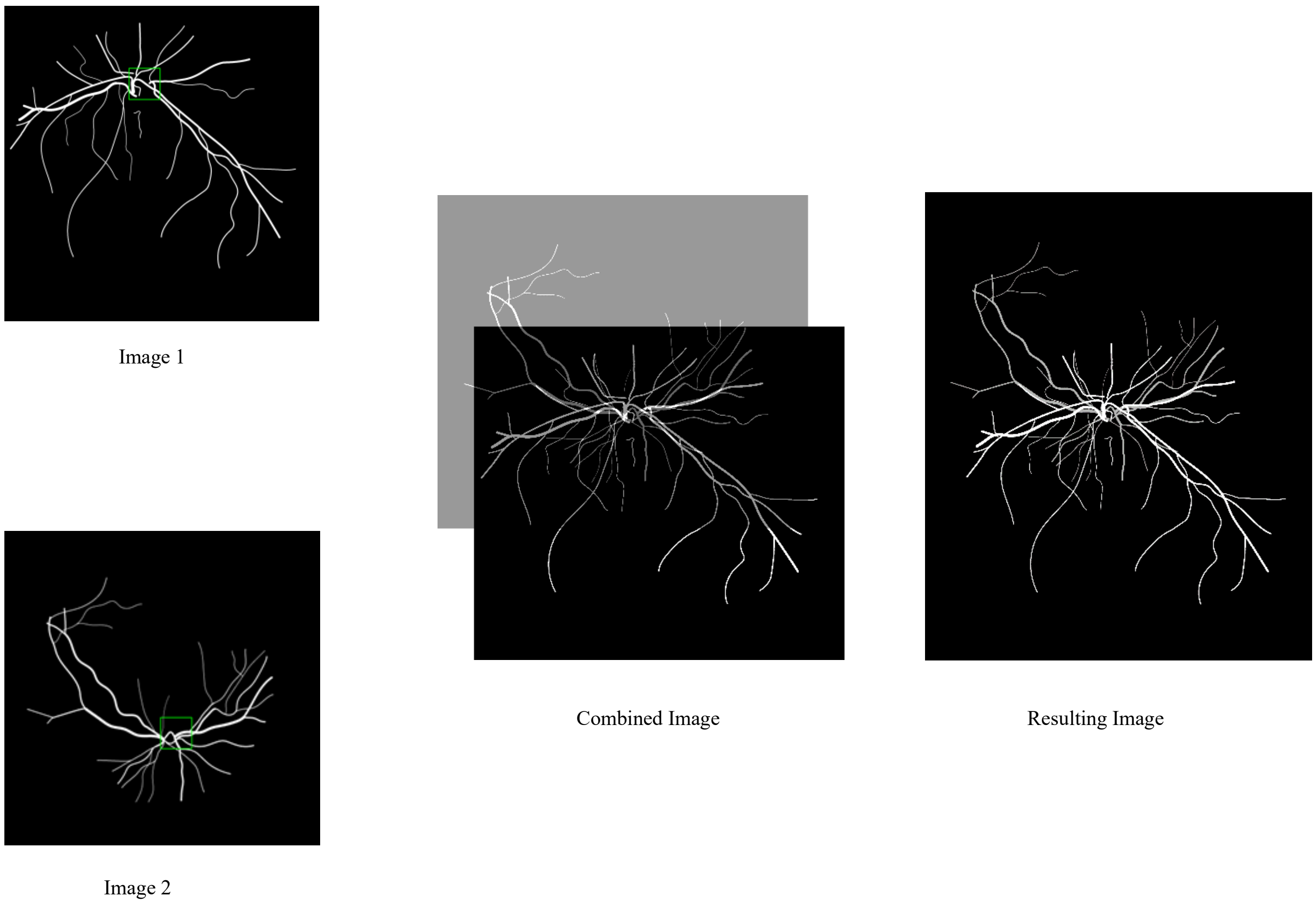

Fig. 4.2: Data augmentation by overlapping images

Above is the visual representation of how the images are actually interpolated and new images are formed. By carrying out the given set of steps a total of 7560 such images were produced which were usable to train the model.

From these 70% images were considered for training the model and remaining 30% was used to test the model.

The dataset hence produced can be downloaded from GitHub with the address: https://github.com/SaikatSantra9/binary-image-of-optic-disk/tree/main/Binary%20images

CHAPTER 5:

ROI DETECTION IN BINARY IMAGES

The ROI or OD detection in the binary images is one of the most important parts for training the model. The OD is detected and a masked binary image locating the optic disc is returned. This binary mask is treated as a ground-truth for locating the OD in the corresponding greyscale images of the binary mask.

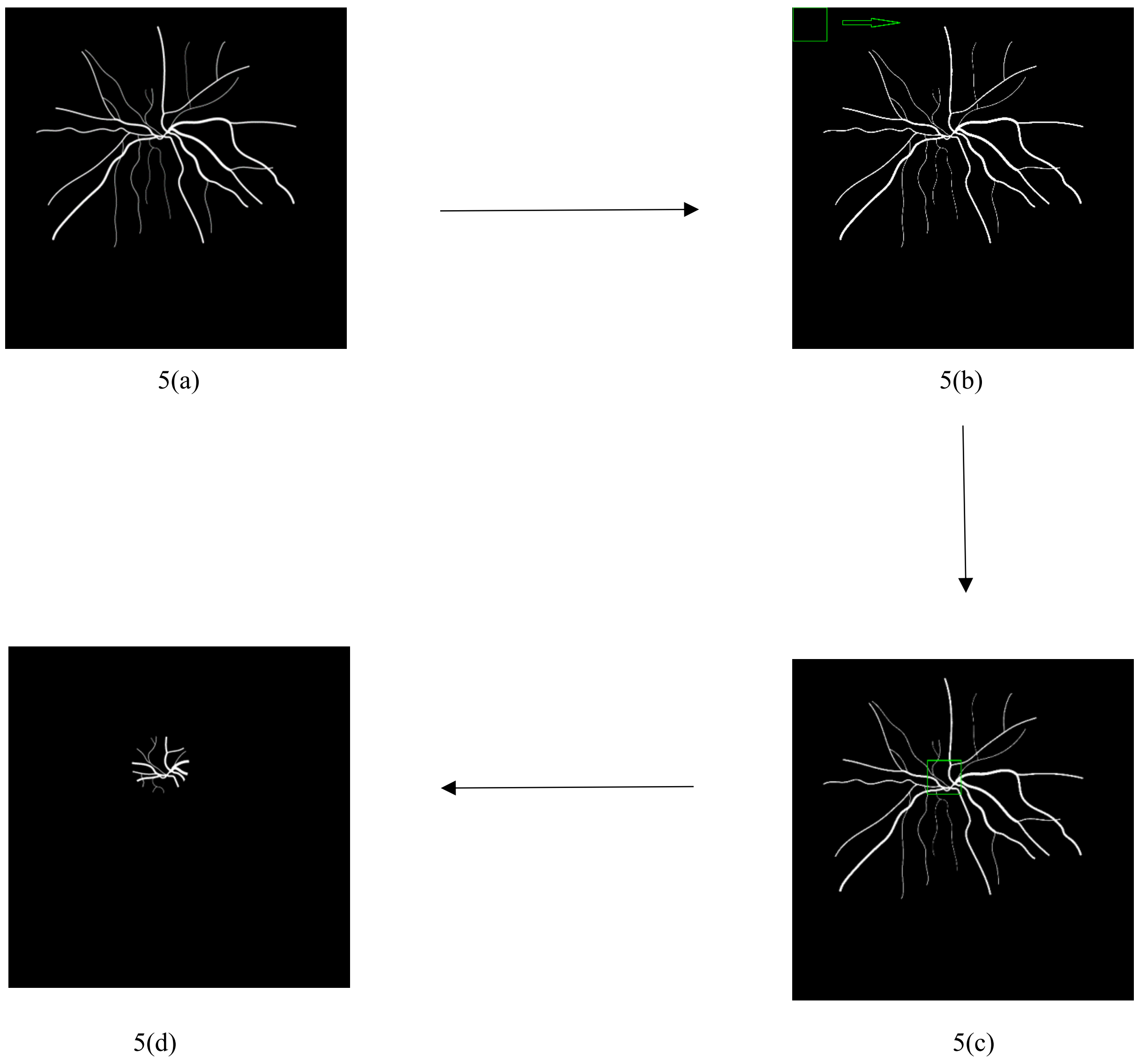

Fig 5.1: Steps performed in OD detection on ground-truth images.

The OD detection is divided into four steps which are explained as follows:

5.1 Preprocessing the Image:

As described in the figure 5(a) of the above figures, the binary image is loaded and is converted into grayscale. The image dimensions (height, width) are obtained and other variables needed for further processing are initialized. An appropriate window size is defined for further analysis.

5.2 Locating the ROI:

The intensity of each window is measured. Then this window is iterated throughout the entire image horizontally one row at a time as shown in figure 5(b) of the above images. The window with the maximum intensity is considered as the region of optic disc.

5.3 Highlighting the Optic Disc:

After we get the region of maximum intensity, we enclose it within a rectangle to highlight it from the rest of the image as shown in part 5(c) of the above images. The colour and thickness of the rectangle is adjusted and the most intense region of the rectangle is extracted.

5.4 Masking:

Now a circular mask is created as shown in part 5(d) of the above pictures. Its radius is defined based on the detected region's central point. This circular mask is then applied to the image.

Following the above four steps a total of 7560 binary masks are created. These binary masked images are used as the ground-truth for the corresponding greyscale images which are used for training and deploying the model. Each fundus image in the dataset is annotated with the ground-truth labels indicating the location and boundaries of the OD. The annotation serves as a reference label for training and evaluating the segmentation model.

CHAPTER 6:

METHODOLOGY

This project showcases a comprehensive approach to building and deploying deep learning model for image segmentation task. The project demonstrates a complete pipeline for training and testing a U-Net based image segmentation model using PyTorch. The methodology is schematically described in the flowchart below:

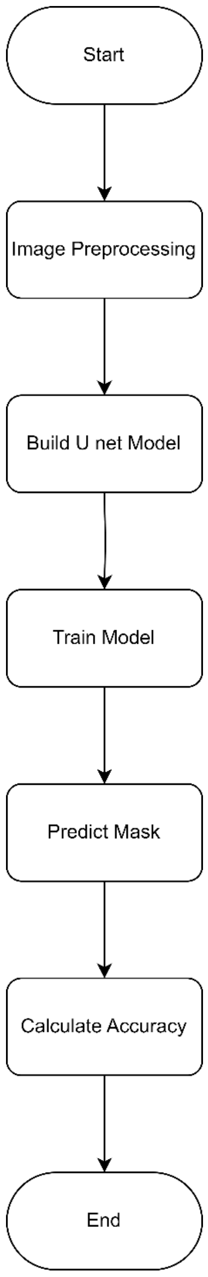

Fig. 6.1: Flow chart representing steps of building the model

Based on the above defined flowchart the project can mainly be divided into four different parts which are:

1. **Data Preparation:** Images and corresponding masks are resized and normalized.
2. **Model Architecture:** U-Net architecture is used along with ReLU activation function.
3. **Training the model:** The training script loads the dataset, initializes the U-Net model, defines the loss function (Dice Loss), and sets up the optimizer (Adam).
4. **Testing the Model:** For each test image, it generates a segmentation mask using the trained model.

6.1 Image Pre-Processing and Data Loading

Image Pre-Processing is a crucial step in any machine learning project. The project starts with loading the dataset which consists of the images and their corresponding masks. Here each pixel in the mask denotes the class of the object it belongs to.

The annotated fundus images are typically divided into training, validation and test sets. The training set is used to train the segmentation model, the validation set is used to tune the hyperparameters and monitor model performance during training and the test set is used to evaluate the final model performance. Once the dataset is prepared and split it is loaded into the memory using data loading utility provided by PyTorch.

After loading the data, the following steps are performed for pre-processing the image which enhances the quality of the fundus images which makes it suitable for segmentation tasks. The pre-processing steps include:

- **Image Resizing:** The images are resized to a standardized resolution of 512x512 pixels and converted to Pytorch tensors, to ensure uniformity and facilitate computational efficiency.

- **Normalization:** The pixel intensities are normalized to a common scale in the range [0,1], to ensure consistent input to the segmentation model.

- **Noise Reduction:** Filters are applied to reduce noise and artifacts in the images which improve segmentation accuracy.

- **Contrast Enhancement:** The image contrast is adjusted to enhance the visibility of the retinal structures including the optic disc for better segmentation results.

Meticulously preparing the dataset and ensuring its quality, consistency and suitability for the segmentation task lays a solid foundation for training accurate and robust optic disc segmentation model.

6.2 Model Architecture

The project implements a U-Net architecture for the image segmentation task. U-Net architecture was developed by Olaf Ronnerberger et al. for biomedical image segmentation in the year 2015 at the University of Freiburg, Germany. U-Net is one of the most common approaches for semantic segmentation. It is a FCNN designed to learn from small amount of training data. It is an improvement for the existing FCN for segmentation developed by Jonathan Long et al. in (2014).

U-Net is a U-shaped network architecture consisting an encoder path and a decoder path with skip connections that is connected by a bridge. The encoder (contraction path) downsamples the input image to extract features, while the decoder path upsamples the feature maps to generate the final segmentation mask. The U-Net model is composed of convolutional blocks, encoder blocks and decoder blocks.

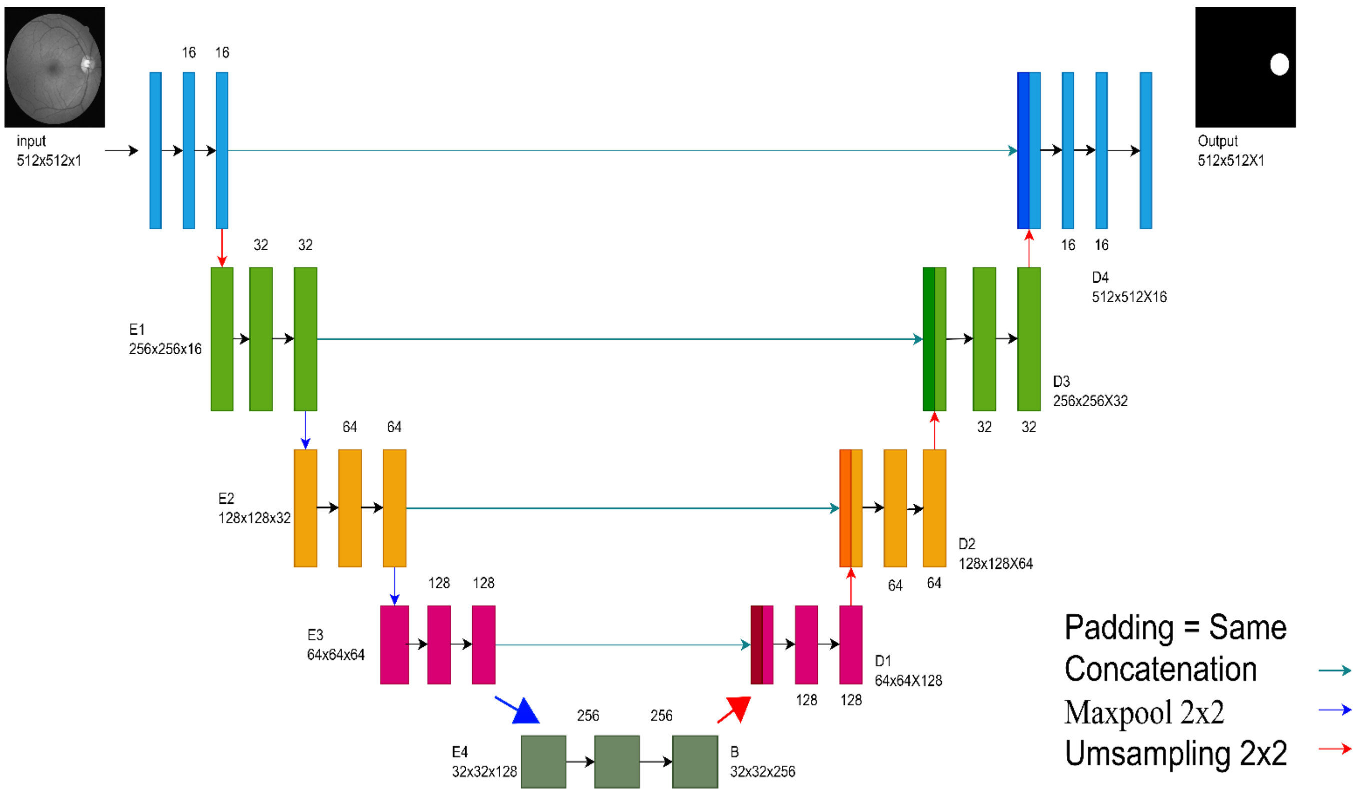

Fig. 6.2: Developed U-Net architecture

6.2.1 Encoder Path

One of the most important components of a U-Net architecture is the encoder or the contraction path. The encoder path is responsible for capturing and extracting features from the input image. Each layer in the decoder is designed to progressively reduce spatial dimensions of the input image and increase the number of feature channels. The encoder path works as a feature extractor and learns an abstract representation of the input image.

The encoder at different levels of abstraction captures features through a series of convolutional and max pooling operations using ReLU activation functions. ReLU introduces non-linearity for better generalization of training data. The model later recovers the spatial dimensions through the expansive path.

6.2.2 Max Pooling

In max pooling the input feature is divided into non-overlapping rectangular regions. The most prominent features from each region is selected and represented in the feature map. For a 2x2 max pooling layer the spatial dimensions of the feature maps get reduced by half. The maximum value of each region is represented in the feature map.

The spatial dimensions of the output feature map after the max pooling operations does not remain the same as the input images. Each pooling layer downsamples the input. Max pooling helps identifying the most prominent features and discards the less important features from the image and supressing the noise.

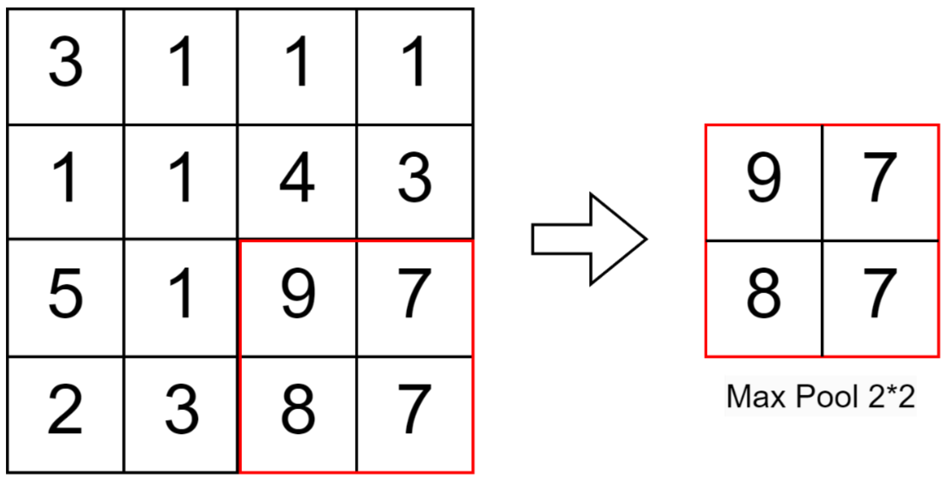

Fig 6.3: Internal operation of max-pooling layers

6.2.3 Kernel

The kernel or a convolutional filter is then applied to the input image through the convolutional operation. The kernel also helps in feature extraction from the input image being a learnable parameter. The kernel slides over the input image in the convolutional operations and computes element wise dot product for each pixel values.

Then all the dot product values are summed to produce a single value for the output feature map. In this project multiple kernels of different sizes are applied to the CNN and the CNN learns and gradually extract more abstract and complex features. Kernel helps identify the edges and structures from the input image. These features can further be identified and segmented.

The below given figure fig 6.4 describes the internal operations performed by a kernel and how output is computed for an input feature map and a kernel.

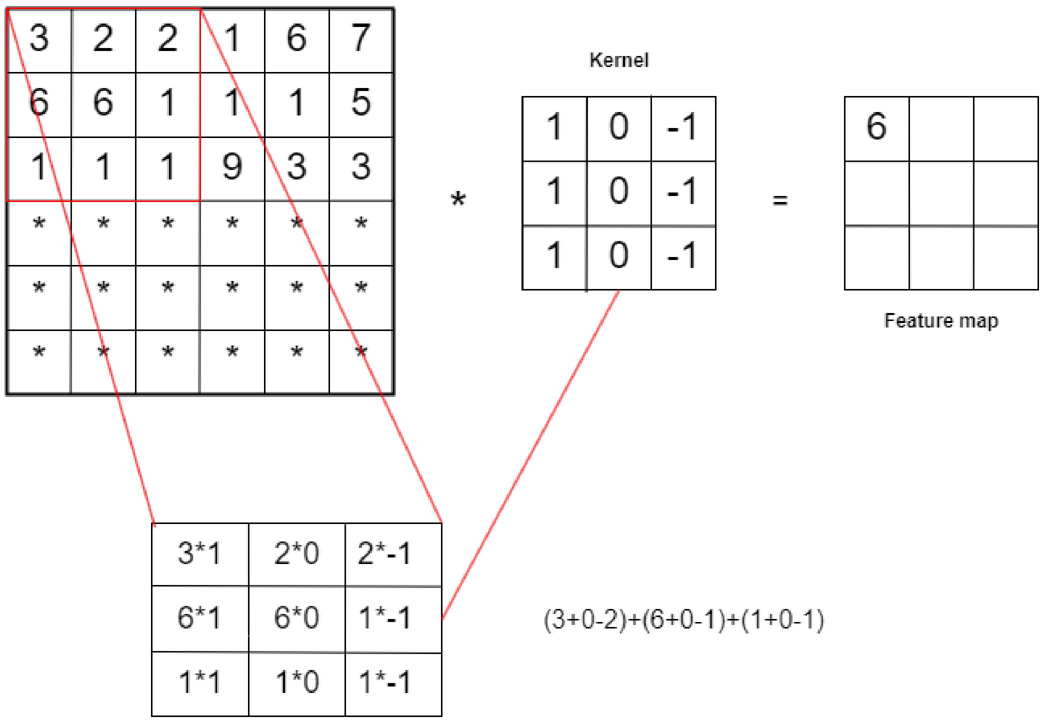

Fig 6.4: Internal operation of Kernel

6.2.4 Stride

The number of pixels by which the kernel shifts over the input image is known as the stride. The kernel starts iterating from the top left corner and starts sliding across the image by the size specified of the stride. For convolutional layers the stride is set to 1 which essentially means the kernel moves one pixel at a time.

When the kernel moves one pixel at a time it results it results in overlapping receptive fields. The overlapping helps capture fine-grained spatial information and preserves the spatial dimensions of the input feature map.

In transposed convolutional layers the kernel is set to 2 which doubles the size across each spatial dimensions of the input feature map. During upsampling process the size of stride is greater than that in convolutional layers to increase the resolution of the feature maps and for recovering the spatial dimensions lost during down-sampling.

Below given is an image describing the work of stride where a 6x6 input image with stride 2 results in a 2x2 output. The stride offers an additional avenue to reduce the number of parameters in image classification task.

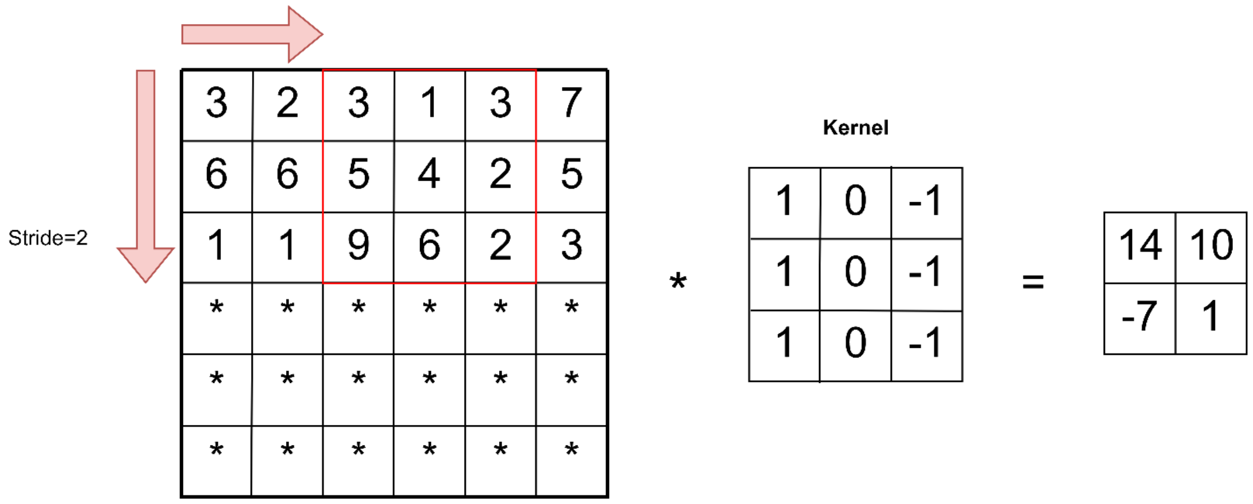

Fig 6.5: Internal operation of strides

6.2.5 Padding

Borders of zeroes are applied around the input image before applying convolutional layers. These borders are known as padding. Padding ensures the size of the output feature map is the same as that of the input feature map. Padding makes sure the spatial dimensions remain the same especially in cases where stride is greater than one.

As the kernel moves from top left corner of the input image, when padding is applied on the input image the kernel also convolves with the padding. This ensures spatial dimension equality in input image and output feature map.

When no padding is applied then the size of the output feature map will not be the same as that of the input feature map. This may lead to inaccuracy in feature extraction.

In this project padding of 1 pixel is added so the size of output feature map remains the same which facilitates dense connectivity throughout the network layers. The fig 6.6 shows the internal operations performed in the convolutional layers when padding is added.

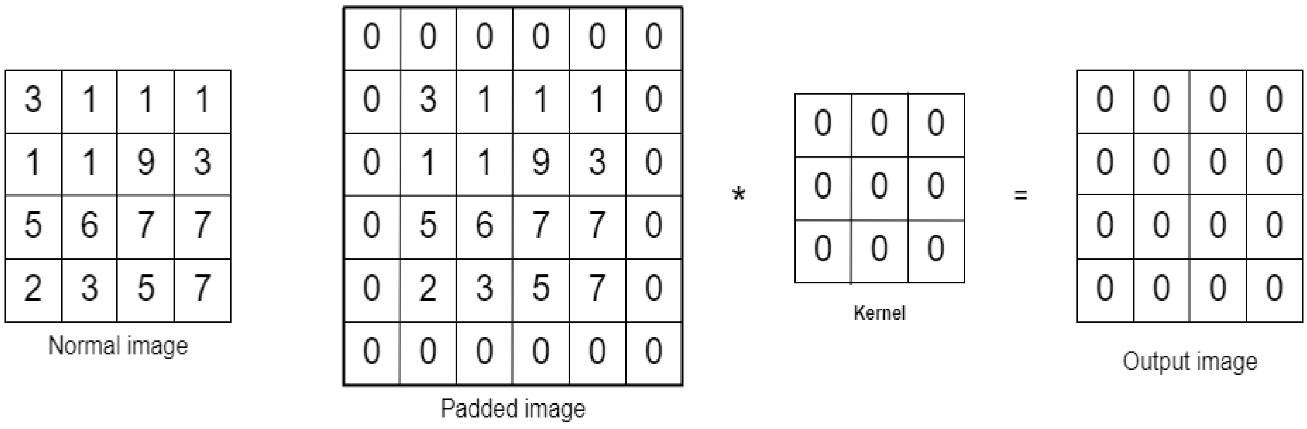

Fig 6.6: Internal operation of Padding

6.2.6 Skip Connections

There are connections which bypass one or more layers in the neural networks. These are called as the skip connections or the residual connections. The skip connections feed the input from any layer in the expansive path to the output of a corresponding layer in the expansive path. These facilitate better gradient flow and improve information propagation across the network.

The encoder blocks consist of convolutional layers followed by pooling layers. These blocks progressively downsample the spatial dimensions of the input image while extracting higher level features. Before downsampling the encoder blocks pass the feature maps through convolutional layers followed by activation functions, batch normalization and other non-linearities the feature maps of the encoder blocks are stored and passed as skip connections to the corresponding decoder blocks.

The decoder blocks are responsible for upsampling the feature maps back to the original input resolution while combining them with the skip connections to recover spatial details. Each decoder block consists of transposed convolutional layers (also known as deconvolution or upsampling layers) followed by convolutional layers, activation functions, and normalization layers.
The transposed convolutional layers in the decoder blocks increase the spatial dimensions of the feature maps. The skip connections from the encoder blocks are concatenated or added (depending on the implementation) with the upsampled feature maps in the decoder blocks.

This allows the decoder to leverage both high-level semantic information from the encoder and detailed spatial information from the earlier layers. By combining information from multiple resolution levels through skip connections, the decoder blocks refine the segmentation predictions and produce accurate segmentation masks.

The skip connections in the model architecture facilitate feature reuse and information fusion between encoder and decoder blocks, enabling the model to effectively capture both global context and fine-grained details for accurate optic disc segmentation.

6.2.7 Decoder block

The decoder block is a crucial component of the U-Net model, specifically designed for semantic segmentation tasks like this. The decoder block aims to upsample the low-resolution feature maps from the encoder pathway to match the original input resolution. It combines the upsampled features with skip connections from the corresponding encoder pathway to recover spatial details lost during downsampling.

The decoder block consists of transposed convolutional layers (also known as deconvolution or upsampling layers), followed by convolutional layers, activation functions, and normalization layers. The transposed convolutional layers increase the spatial dimensions of the feature maps, effectively performing upsampling.

Skip connections are connections from the encoder pathway to the decoder pathway which provide high-resolution feature maps from the encoder pathway directly to the corresponding decoder block. Skip connections facilitate the fusion of low-level and high-level features, aiding in precise localization and segmentation by combining detailed spatial information with semantic context.

In the decoder block, the upsampled feature maps from the transposed convolutional layers are concatenated with the skip connections. This fusion of features from different resolution levels allows the model to capture both local details and global context, enhancing the accuracy of segmentation predictions.

Lastly the output layer of the U-Net model is defined. It is a convolutional layer with a sigmoid activation function. It takes the features from the expansive path and produces the final segmentation output with the specified number of classes which in this case is 1. Class value is 1 as final output is a binary mask and hence sigmoid activation function is applied which squashes the output values in the range [0,1] and is used for binary classification problems.

6.2.8 Activation Function

An activation function is a mathematical operation applied to the output of each neuron in a neural network, typically following the weighted sum of inputs and biases. Activation functions introduce non-linearity to the network, allowing it to learn complex patterns and relationships in the data.

Some of the common activation functions include:

Sigmoid: Outputs values between 0 and 1, often used in binary classification problems.

Hyperbolic Tangent (tanh): Outputs values between -1 and 1, similar to the sigmoid function but centered at zero.

Rectified Linear Unit (ReLU): Outputs the input itself if it is positive; otherwise, it outputs zero. ReLU is widely used in deep learning models due to its simplicity and effectiveness.

Leaky ReLU: Similar to ReLU, but allows a small, non-zero gradient when the input is negative, helping to mitigate the "dying ReLU" problem.

Softmax: Used in multi-class classification problems to output probabilities across multiple classes, ensuring that the sum of the probabilities equals one.

Activation functions introduce non-linearity to the network, enabling it to approximate complex functions and relationships in the data. In this project the activation function used in various layers of the neural network architecture including the decoder block is ReLU. ReLU is one of the most commonly used activation function in CNNs due to its simplicity and effectiveness.

ReLU is a piecewise linear function defined as f(x) = max(0,x) f(x)=max(0,x)returns 0 if the input x is less than or equal to 0, and it returns the input x itself if it is greater than 0.

ReLU introduces sparsity in the network, as neurons that receive negative inputs output zero. This can lead to faster training and improved generalization. Unlike activation functions like sigmoid, ReLU does not suffer from the vanishing gradient problem, which can make the process of training the model difficult.

6.3 Training the model

The training phase in this project involves optimizing the parameters of the neural network. The training script loads the dataset, initializes the U-Net model, defines the loss function and sets up the optimizer. A suitable loss function (Dice loss) is chosen to quantify the difference between the predicted segmentation mask and the ground truth annotations. Next an optimization algorithm (Adam optimizer) is selected to minimize the chosen loss function. The optimizer adjusts the weight of the neural network parameters based on the gradients of the loss function with respect to other parameters.

The training loop runs for a specified number of epochs, iterating over the entire training dataset and updating the model parameters. After each epoch validation loss is computed to monitor model performance on unseen data and gradients are updated using back propagation.

The training loop iterates over the entire training dataset for a fixed number of epochs and for each epoch, the dataset is divided into batches of images and corresponding annotations. For each batch, the following steps are performed:

Forward Pass: The input images are fed into the neural network, and the model generates predictions for the optic disc localization or segmentation.

Loss Computation: The loss function is calculated based on the model's predictions and the ground truth annotations.

Backward Pass: Gradients of the loss function with respect to the model parameters are computed using backpropagation.

Parameter Update: The optimizer adjusts the model parameters (weights) based on the computed gradients to minimize the loss function.

Validation: Periodically during training (e.g., after each epoch), the model's performance is evaluated on a separate validation dataset.

Training might also be stopped early if the model's performance on the validation dataset does not improve or starts to degrade over several epochs, indicating overfitting. After training completes, the trained model parameters (weights) are saved to disk for future use in inference or deployment.

6.3.1 Optimizer

An optimizer is an algorithm or method used to adjust the parameters of a model during training in order to minimize a loss function. The goal of optimization is to find the set of model parameters that result in the best performance on the training data, typically measured by how well the model's predictions match the true labels.

There are various optimizers available, each with its own characteristics and update rules. Some common optimizers include:

- Stochastic Gradient Descent (SGD)
- Adam
- RMSProp
- Adagrad
- Momentum

In this project, the Adam optimizer is used for optimizing the parameters of the neural network during the training phase. The Adam optimizer is a popular choice for training deep neural networks due to its efficiency and effectiveness in handling sparse gradients and noisy data. Adam is known for its efficient convergence properties, making it suitable for training deep neural networks with large datasets.

Adam adapts the learning rates for each parameter by maintaining two moving averages of gradients and squared gradients. It computes the adaptive learning rates based on the first moment (mean) and the second moment (uncentered variance) of the gradients. Adam performs bias correction to account for the initialization bias of the moving averages. This helps in stabilizing the optimization process, especially during the early stages of training.

An optimizer plays a crucial role in training machine learning models by iteratively updating the model's parameters to minimize the loss function, ultimately leading to improved performance on the task at hand. By using Adam, the training process aims to minimize the chosen loss function efficiently and effectively, leading to improved performance of the model on the task of optic disc localization or segmentation in retinal images.

6.3.1.1 Working of Adam’s Algorithm

The core of Adam’s algorithm lies in its computation of adaptive learning rates for each parameter.

Initialization:

Adam initializes two vectors m and v both of same space as the parameter θ of the model. The vector m stores the moving average of the gradients and v keeps record of the moving average of squared gradients. The moving averages are key to Adam’s adaptive adjustments. Time step counter t is initialized to zero which keeps track of the number of iterations that the algorithm has completed.

The initial values are as follows:

- m0 = 0 (first vector)
- v0 = 0 (second vector)
- t = 0 (time step)

Compute Gradients:

For each iteration t, the Adam optimizer computes the gradient gt, which represents the derivative of the objective function with respect to the current model parameters θt. In essence, this gradient indicates the direction in which the objective function increases most rapidly at the current parameter values θt.

Update first vector (m):

After computing the gradient gt for each iteration t, Adam updates the first-moment vector m, which serves as a moving average of the gradients. This update blends the previous value of m with the new gradient gt, weighted by the hyperparameter β1 and its complement 1−β1. Essentially, this process integrates recent gradient observations while maintaining a memory of past gradients, resulting in a smoothed estimate of the gradient direction.

Update second vector(v):

Similarly, Adam updates the second-moment vector v, which estimates the variance of the gradients. This update involves blending the past squared gradients with the current squared gradient, weighted by the hyperparameters β2 and 1−β2. The second-moment vector provides insights into the unpredictability of the gradients.

Bias Correction:

To correct the bias in the moments m and v, which are initialized to zero, Adam adjusts the vectors by the respective decay rates β1t and β2t . This correction ensures that the moving averages are more representative, especially during the initial training stages.

Updates the parameters:

Finally, Adam updates the model parameters θ by incorporating the adaptive learning rates calculated in the previous steps. This update moves the parameters towards the minimum of the loss function, with the step size determined by the learning rate α.

6.3.2 Loss Function

A loss function, also known as a cost function or objective function, is a measure of how well a machine learning model's predictions match the true labels or targets in a training dataset. It quantifies the difference between the predicted output of the model and the ground truth labels, providing a single scalar value that represents the model's performance on a given task. The goal during training is to minimize this loss function, thereby improving the model's ability to make accurate predictions.

Key aspects of a loss function:

Quantifying Error: The loss function computes the discrepancy between the model's predictions and the true labels. It quantifies how far off the predictions are from the actual values.

Optimization Objective: Minimizing the loss function is the primary objective during model training. By adjusting the model parameters iteratively, the goal is to find the set of parameters that result in the smallest possible loss.

Differentiable: Loss functions are typically differentiable with respect to the model parameters, allowing for the use of gradient-based optimization algorithms, such as stochastic gradient descent (SGD), to update the parameters efficiently.

Task-specific: The choice of loss function depends on the nature of the machine learning task. Different tasks, such as classification, regression, or segmentation, require different loss functions tailored to the specific characteristics of the problem.

Interpretability: The value of the loss function provides interpretable feedback on the model's performance. A lower loss indicates better agreement between predictions and ground truth, while a higher loss suggests poorer performance.

In the project, Dice Loss function is used. The Dice Loss is a similarity coefficient-based loss function that measures the overlap between predicted and ground truth segmentation masks. It is particularly effective for tasks where class imbalance is prevalent, as in medical image segmentation. The goal is to minimize the Dice Loss during training, leading to more accurate and precise segmentation results.

6.3.2.1 Dice Loss Function

In the project, the Dice Loss function is used as the loss function. The Dice Loss is a similarity coefficient-based loss function commonly employed in semantic segmentation tasks, particularly in medical image analysis, where class imbalance is prevalent.

Dice Loss function and its characteristics:

Similarity Coefficient: The Dice coefficient, also known as the Sørensen–Dice coefficient, is a measure of the spatial overlap between two binary masks. It is defined as twice the intersection of the predicted and ground truth masks divided by the sum of their areas.

Soft Dice Loss: The Dice Loss function is derived from the Dice coefficient and is formulated to be differentiable, making it suitable for gradient-based optimization algorithms. It quantifies the dissimilarity between the predicted and ground truth segmentation masks.

Objective: The objective of the Dice Loss is to minimize the dissimilarity or maximize the similarity between the predicted and ground truth masks. This encourages the model to produce segmentation results that closely match the ground truth annotations.

Handling Class Imbalance: The Dice Loss is effective in handling class imbalance, which is common in medical image segmentation tasks. It mitigates the impact of class imbalance by focusing on the spatial overlap between the predicted and ground truth masks rather than relying solely on pixel-wise classification accuracy.

Differentiability: The Dice Loss function is differentiable with respect to the predicted segmentation mask, allowing for gradient-based optimization methods such as stochastic gradient descent (SGD) to minimize the loss efficiently during training.

Range: The Dice Loss is bounded between 0 and 1, where 0 indicates no spatial overlap between the predicted and ground truth masks (complete dissimilarity), and 1 indicates perfect spatial overlap (complete similarity).

6.3.2.2 Comparison between Dice loss and BCE loss

Dice Loss:

Advantages:

- Measures the spatial overlap between predicted and ground truth masks, making it well-suited for tasks like image segmentation.
- Robust to class imbalance, which is common in segmentation tasks.

- Provides a more intuitive measure of segmentation accuracy, as it directly evaluates the agreement between predicted and ground truth masks.

Disadvantages:
- Not as computationally efficient as BCE loss.
- May suffer from convergence issues in certain cases, especially when dealing with highly imbalanced datasets or when the predicted masks are noisy.

Binary Cross-Entropy Loss:

Advantages:
- Widely used and well-established loss function for binary classification tasks, including image segmentation.
- Computes the difference between pixel-wise probabilities and ground truth labels, providing a clear gradient signal for optimization.
- Generally, computationally efficient and easy to implement.

Disadvantages:
- Less intuitive for segmentation tasks compared to Dice Loss, as it focuses on pixel-wise classification rather than spatial overlap.
- Can be sensitive to class imbalance, leading to biased training especially when one class dominates the image.
- May not directly address the spatial consistency of segmentation masks, which is crucial for tasks with complex object shapes or overlapping regions.

Comparison:

For this project the dice loss function is used as finding the overlap between the predicted and the ground truth mask is crucial. It provides a direct measure of segmentation performance and is robust to class imbalance. By using the dice loss function more accurate results were achieved and proved to give more intuitive results for our project. The Dice loss quantifies the spatial overlap between the predicted and the ground-truth segmentation masks. It aims at maximizing the spatial similarity between the predicted and ground truth masks.

Dice Loss proves to be effective in handling class imbalance. It focuses on the spatial overlap between masks, making it less sensitive to class imbalances. This made dice loss more suitable for our segmentation task as class imbalance is common. Dice Loss is more robust to noise and class imbalance due to its focus on spatial overlap. It tends to produce more accurate and coherent segmentation results, particularly in medical image analysis tasks where precise delineation of structures is crucial. Dice Loss provides a measure of spatial overlap between masks, offering insights into the model's segmentation performance in terms of spatial coherence and overlap with ground truth annotations.

Even though BCE proves to be computationally efficient in many cases which makes it suitable for large scale segmentation tasks it didn’t quite give the desired results for our project. BCE Loss does not explicitly address class imbalance. In scenarios with imbalanced classes, the model may prioritize the majority class, leading to suboptimal performance for the minority class.
BCE Loss measures the discrepancy between the predicted pixel-wise probabilities and the binary ground truth labels. It encourages the predicted probabilities to match the ground truth labels closely. BCE Loss provides pixel-wise binary classification loss, making it straightforward to interpret the model's performance at the pixel level. However, it struggled with segmenting small or overlapping structures due to its pixel-wise nature.

6.3.3 Activation Function

An activation function is a mathematical operation applied to the output of each neuron in a neural network. It introduces non-linearity into the network, enabling it to learn complex patterns and relationships in the data. Activation functions help in transforming the input signal into an output signal, typically representing the neuron's level of activation or firing.

When a neuron in a neural network receives input data, it combines this input by multiplying each piece of data by a weight assigned to it. These weighted inputs are then summed up, and a bias term is added to the total sum. This process forms the initial output of the neuron. However, this output isn't directly sent out. Instead, it's passed through an activation function. The activation function serves a crucial role by adding nonlinearity to the neuron's output. This nonlinearity is vital because it enables the neural network to capture complex patterns and relationships in the data that wouldn't be possible with linear operations alone. By introducing nonlinearity, the activation function allows the neural network to learn and represent more sophisticated data patterns, making it capable of handling a wider range of tasks and data types effectively.

Without an activation function, a neural network would be restricted to modeling only linear relationships between its inputs and outputs. In essence, it would behave like a linear regression model, which can only capture linear patterns in the data. However, many real-world problems involve complex, nonlinear relationships between variables. Activation functions introduce nonlinearities into the neural network's computations, allowing it to learn and represent these intricate patterns in the data. Thus, the absence of an activation function would severely limit the neural network's ability to handle and make sense of complex data structures and relationships.

The activation function used in this project is the Rectified Linear Unit (ReLU). ReLU is one of the most widely used activation functions in deep learning due to its simplicity and effectiveness. Mathematically, ReLU is defined as:

$\mathrm{ReLU}(x)=\max(0,x)$

6.3.3.1 ReLU activation function

The ReLU activation function outputs the input value if it is positive, and zero otherwise. This results in a piecewise-linear function that introduces non-linearity to the neural network while preserving positive values. ReLU has several advantages, including computational efficiency, ease of implementation, and mitigation of the vanishing gradient problem.

In mathematical terms, the ReLU activation function can be represented as:

$$f(X)=\max(0, X)$$

If the input X is less than or equal to zero, the output of the function is zero.
If the input X is greater than zero, the output of the function is equal to the input X itself. In essence, ReLU acts as a threshold function, allowing positive values to

pass through unchanged while setting negative values to zero. This property makes ReLU particularly effective in deep neural networks as it helps mitigate the vanishing gradient problem and accelerates convergence during training.

The simplicity and effectiveness of ReLU have contributed to its widespread adoption in various neural network architectures, from simple feedforward networks to complex convolutional and recurrent networks. Indeed, while ReLU has proven to be a popular activation function, it can encounter a significant issue known as the Dying ReLU problem. This problem arises when the neuron's bias is set such that it consistently receives negative inputs during training.

6.3.3.2 Leaky ReLU activation function

Leaky ReLU (Rectified Linear Unit) is an alternative activation function commonly used in neural networks, particularly in deep learning models. It addresses the "dying ReLU" problem associated with the standard ReLU activation function.

In Leaky ReLU, instead of setting negative values to zero, as in the case of ReLU, negative values are multiplied by a small positive slope (typically a small constant, such as 0.01 or 0.1). This slight slope ensures that even when the input to a neuron is negative, the neuron still produces a small output, allowing information to flow through the network during both forward and backward propagation.

Mathematically, the Leaky ReLU function is defined as:

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha . x & \text{if } x <= 0 \end{cases}$$

where,
α is a small positive slope (usually a small constant, e.g., 0.01).

The main advantage of Leaky ReLU over traditional ReLU is that it helps to mitigate the "dying ReLU" problem by allowing gradients to flow through neurons with negative inputs, preventing them from becoming inactive during training. This can lead to improved training stability and performance, particularly in deeper neural networks where the dying ReLU problem is more likely to occur.

Additionally, Leaky ReLU retains the computational efficiency of ReLU and is easy to implement, making it a popular choice in many neural network architectures. However, it's worth noting that the choice of the slope parameter (α) may affect the performance of the network and often requires tuning through experimentation.

6.3.3.3 Comparison between ReLU and Leaky ReLU

ReLU (Rectified Linear Unit) performs better than Leaky ReLU in scenarios where the majority of the activations are positive. This is because ReLU sets all negative activations to zero, effectively removing them from the computation, which can lead to sparsity in the network. Sparsity can help in reducing computational complexity and improving the efficiency of training.

Additionally, ReLU is computationally more efficient than Leaky ReLU since it involves a simple thresholding operation without any additional parameters to

However, it's worth noting that ReLU can suffer from the "dying ReLU" problem, where neurons can become inactive (i.e., always output zero) during training, especially if they receive consistently negative inputs. Leaky ReLU addresses this issue by allowing a small, non-zero slope for negative inputs, ensuring that even inactive neurons can still contribute to the gradient flow during backpropagation.

The "dying ReLU" problem refers to a situation where neurons in a neural network using the ReLU activation function become inactive (i.e., always output zero) and cease to contribute to the learning process during training.

This problem typically occurs when a large gradient flows through a ReLU neuron, causing it to update its weights in such a way that it consistently receives negative inputs. Since ReLU sets negative inputs to zero, if a neuron consistently receives negative inputs, it will remain inactive (output zero) indefinitely. As a result, the neuron effectively "dies" and stops learning, leading to a phenomenon known as "dead neurons" or "dying ReLU."

The presence of dying ReLU neurons can hinder the learning process in neural networks by reducing the representational capacity of the model and impeding the flow of gradients during backpropagation. This can ultimately result in slower convergence during training and suboptimal performance of the network.

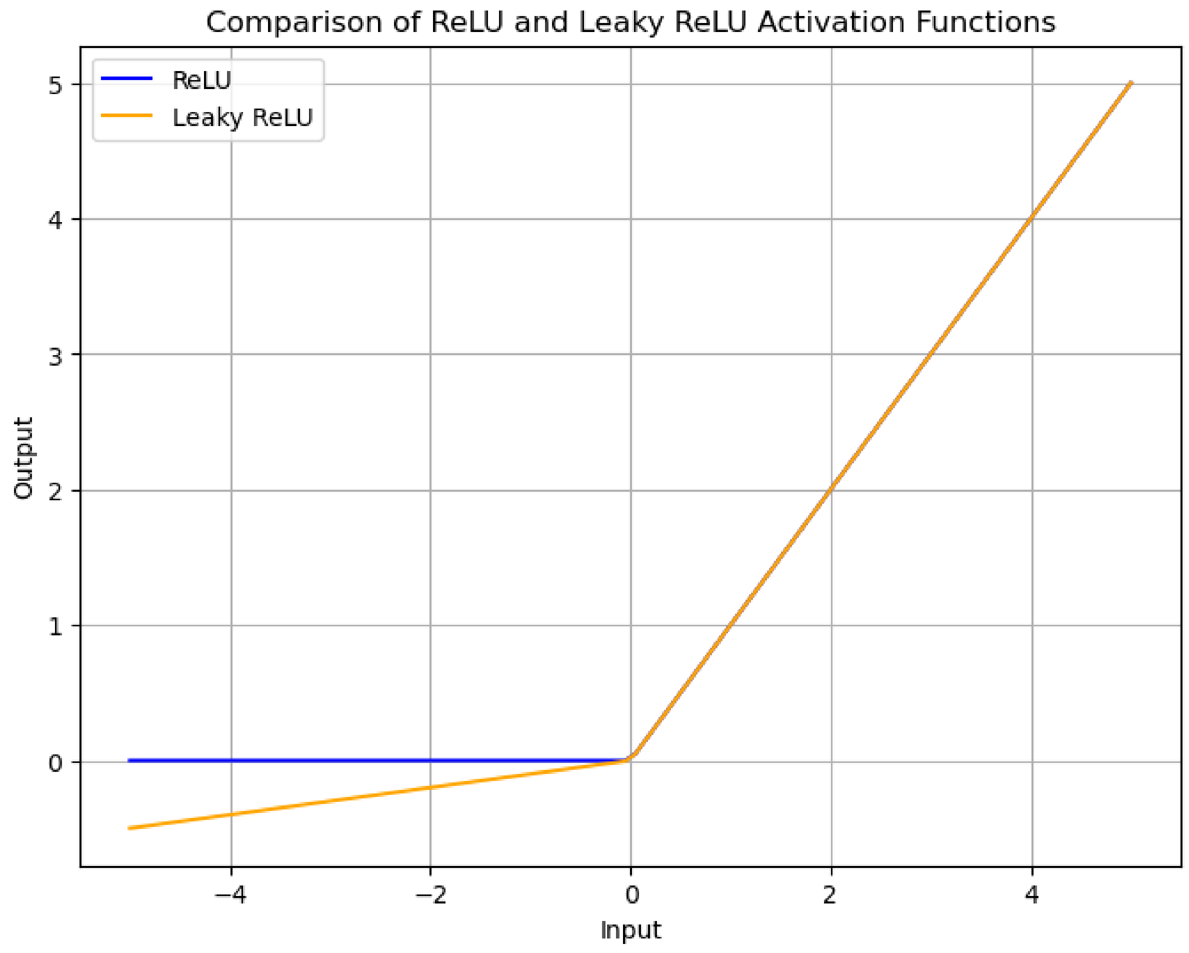

Fig 6.7: Graph depicting range of values for ReLU and leaky ReLU activation functions

The above graph describes the comparison between the range of values in the ReLU and Leaky ReLU activation function. For this project Leaky ReLU activation function is used as while using ReLU activation function it failed to provide output for some images but once leaky ReLU was used better results were achieved. Leaky ReLU's non-zero slope for negative inputs also helped prevent overfitting by introducing some regularization. By experimenting both activation functions and comparing their performance based on training loss and validation accuracy leaky ReLU was found to be a giving better result.

6.4 Testing the model

The testing phase is crucial for assessing the real-world performance of the trained model and ensuring that it can generalize well to unseen data. In the testing phase of the project, the trained neural network model is evaluated on a separate dataset that it has not seen during training.

This dataset is used to assess the performance and generalization ability of the model on unseen data. For testing the model's performance, three datasets were used and were compared individually. The first dataset is the DRIVE dataset, second dataset is the Stare dataset and lastly the HRF image dataset is used to check the real-world performance of the model.

The trained model is saved at the end of the training phase and is loaded during the testing phase. The testing script loads the dataset and iterates over the test dataset and generates segmentation mask using the trained model. It displays the input image and the predicted mask side by side for better visualization understanding. Some images detected multiple area as the output due to noise in test images hence contours are applied to the output images and the area with largest contour is displayed as the output.

Chapter 7:
RESULTS

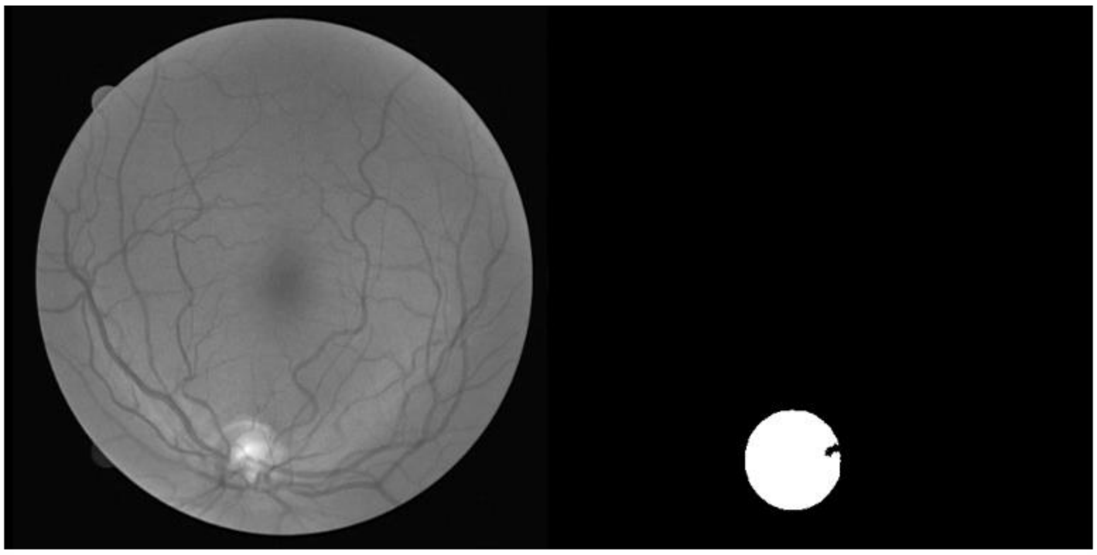

Fig 7.1: Detected optic disc mask 1

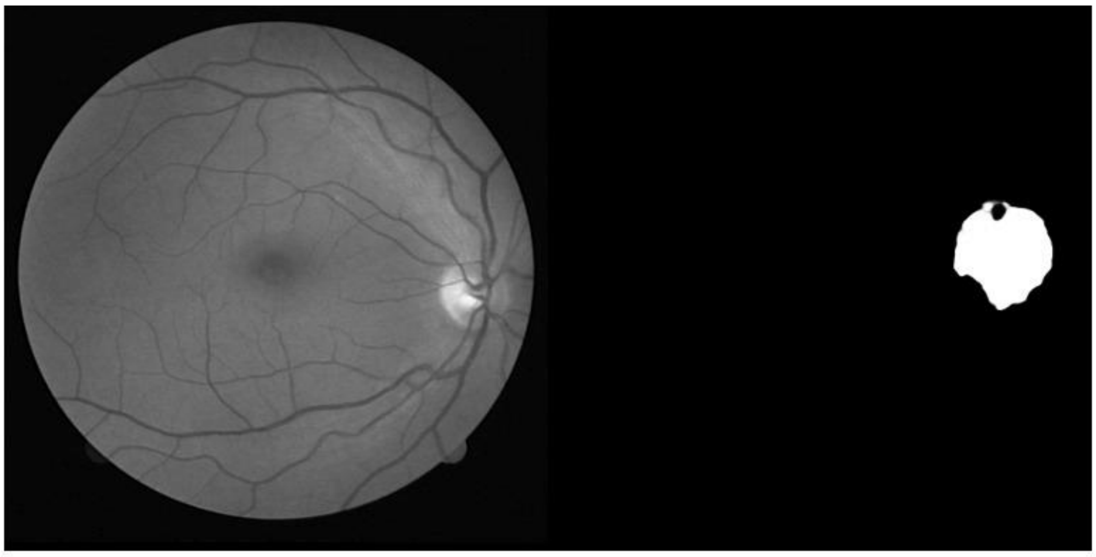

Fig 7.2: Detected optic disc mask 2

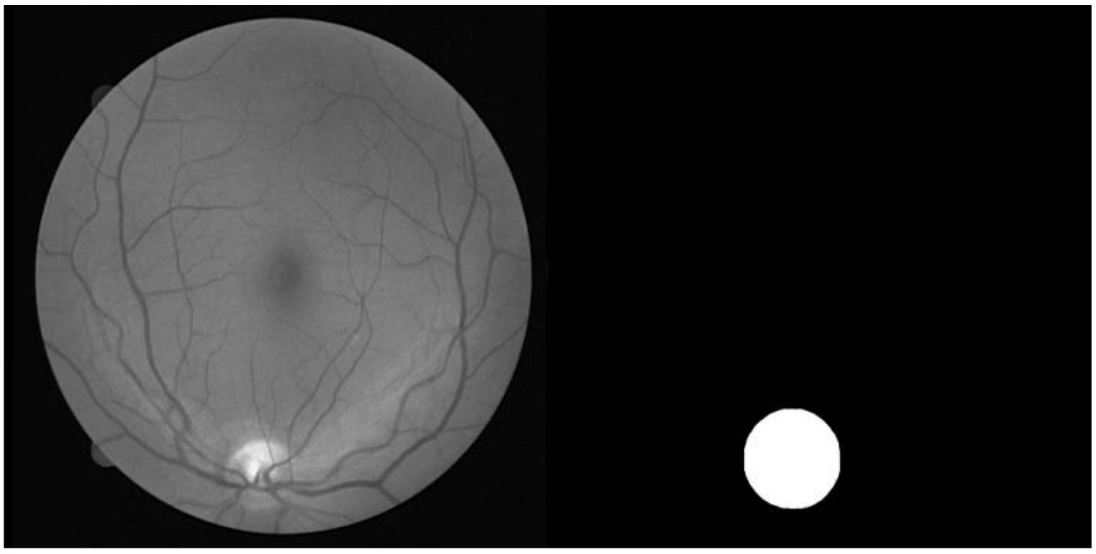

Fig 7.3: Detected optic disc mask 3

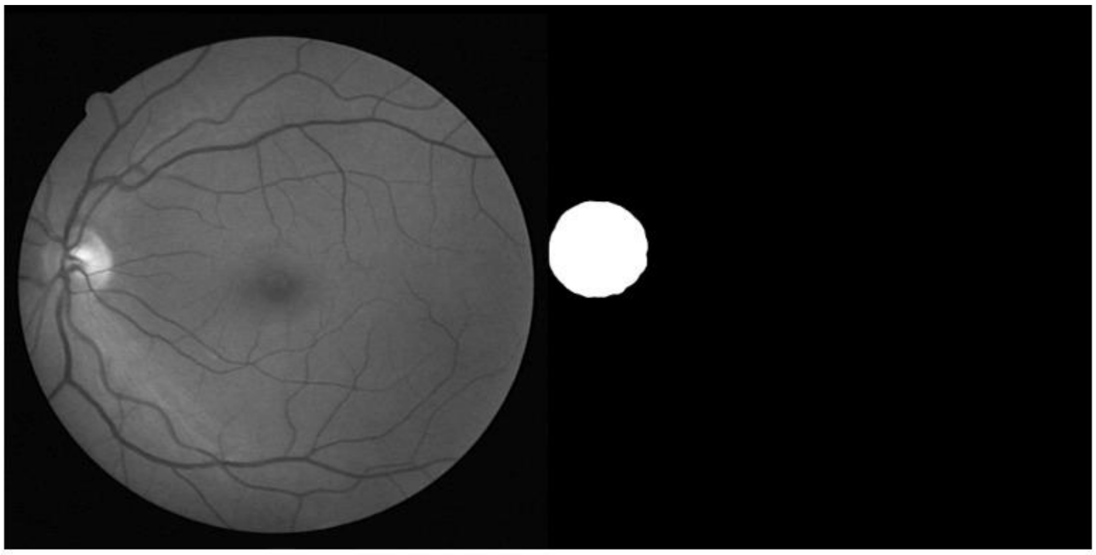

Fig 7.4: Detected optic disc mask 4

The above given pictures are the output returned by the developed U-Net model. The images were taken from DRIVE dataset and are displayed with their corresponding masks.

Image Name	Center X	Center Y	GT Center X	GT Center Y	Eucledian dis
01_test_grayscale.jpg	88	266	87	266	0
02_test_grayscale.jpg	462	279	462	279	0
03_test_grayscale.jpg	99	277	99	278	1
04_test_grayscale.jpg	357	278	357	278	0
05_test_grayscale.jpg	88	262	88	262	0
06_test_grayscale.jpg	471	275	472	275	1
07_test_grayscale.jpg	488	291	489	291	1
08_test_grayscale.jpg	473	271	479	276	7.81
09_test_grayscale.jpg	100	264	100	264	0
10_test_grayscale.jpg	465	291	465	291	0
11_test_grayscale.jpg	79	273	80	271	2.24
12_test_grayscale.jpg	88	257	88	256	1
13_test_grayscale.jpg	483	271	484	271	1
14_test_grayscale.jpg	478	275	478	275	0
15_test_grayscale.jpg	194	290	195	291	1.41
16_test_grayscale.jpg	475	262	475	263	1
17_test_grayscale.jpg	464	267	465	267	1
18_test_grayscale.jpg	470	251	478	262	13.6
19_test_grayscale.jpg	486	281	486	281	0
20_test_grayscale.jpg	476	292	476	291	1
21_training_grayscale.jpg	52	262	50	258	4.47
22_training_grayscale.jpg	465	272	470	274	5.39
23_training_grayscale.jpg	430	229	431	220	9.06
24_training_grayscale.jpg	471	320	468	318	3.61
25_training_grayscale.jpg	462	267	454	271	8.94
26_training_grayscale.jpg	52	261	83	249	33.24
27_training_grayscale.jpg	483	279	480	277	3.61
28_training_grayscale.jpg	484	265	481	270	5.83
29_training_grayscale.jpg	486	271	488	276	5.39
30_training_grayscale.jpg	479	263	484	291	28.44
31_training_grayscale.jpg	446	288	395	249	64.2
32_training_grayscale.jpg	486	291	489	278	13.34
33_training_grayscale.jpg	471	320	472	305	15.03
34_training_grayscale.jpg	288	450	319	291	161.99
35_training_grayscale.jpg	66	261	85	274	23.02
36_training_grayscale.jpg	469	260	473	276	16.49
37_training_grayscale.jpg	472	290	485	281	15.81
38_training_grayscale.jpg	481	268	492	272	11.7
39_training_grayscale.jpg	99	260	86	252	15.26
40_training_grayscale.jpg	484	276	488	270	7.21

Table 1: Difference between the centers of ground truth locations and the retinal image locations

The above table specifies the x and y co-ordinates of the center points of the predicted optic discs. The centroids of the predicted masks for the retinal fundus images and the masks for ground-truth annotations. The difference between the two centroids are calculated and can be seen giving accurate results in most cases.

7.1 ACCURACY

7.1.1 Accuracy using Dice loss function

The Dice loss function and Dice coefficient are commonly used evaluation metrics in medical image segmentation tasks. The Dice loss function is derived from the Dice coefficient and is used as a loss function during training of segmentation models. It is particularly useful for class-imbalanced problems, such as medical image segmentation, where the foreground (e.g., optic disc) may occupy only a small portion of the image compared to the background.

The Dice coefficient is a common evaluation metric used to quantify the similarity between two segmentation masks, typically the predicted and ground truth masks. Mathematically, the Dice coefficient is defined as

{2×true positive/ (2×true positive+ false positive+ false negative)}

In other words, the dice coefficient or the F1-score is calculated as 2 x intersection divided by total pixels in both images.

Dice = 2 x area of overlap / total area

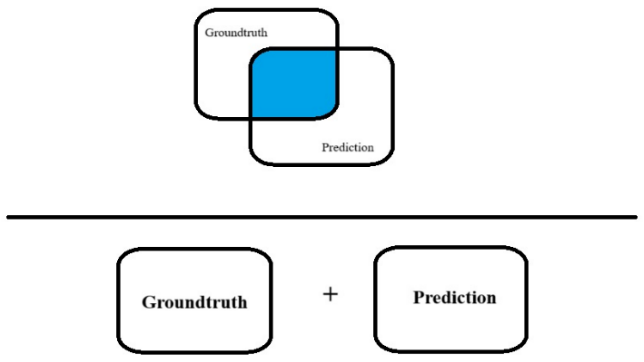

Fig 7.5: Pictorial description of area of overlap and total area.

The area of overlap and the total area is pictorially depicted in the above figure.

The Dice coefficient ranges from 0 to 1, with 1 indicating perfect overlap between the predicted and ground truth masks and 0 indicating no overlap. By calculating the dice coefficient an accuracy of about 88% was achieved.

7.1.2 Accuracy using Jaccard's index

The model calculates the Intersection Over Union (IoU) between the groundtruth mask and the predicted mask. The IoU is a common evaluation metric for segmentation tasks. It calculates the ratio of the intersection to the union of the true and predicted mask. The IoU value is a measure of how well the predicted mask aligns with the true mask.

$$IoU(A, B) = \frac{A \cap B}{A \cup B}$$

IoU is basically area of overlap divided by area of union. The IoU can be understood clearly with the help of the diagram below

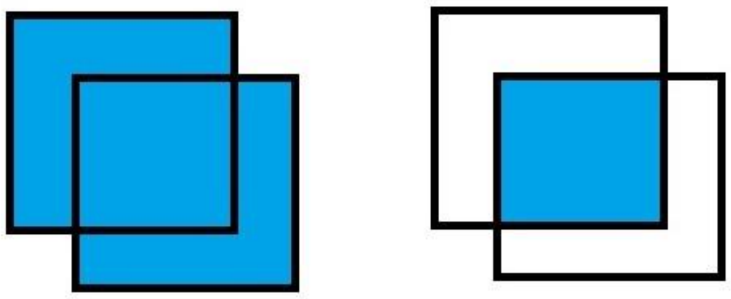

Fig. 7.6: Pictorial representation of Intersection over Union

In the numerator the area of overlap is calculated between the predicted mask and the ground truth mask. The area of union is the area bounded by both the predicted mask and the ground truth mask. Diving the area of overlap by area of union we get out required IoU.

Additionally, the percentage of the predicted mask in terms of the true mask in terms of the true mask is calculated and saved in a file to calculate average overlap. The Overall Jaccard Index (IoU) is found to be nearly 78 %.

7.2 COMPARATIVE ANALYSIS

7.2.1 Comparative analysis with HRF Dataset

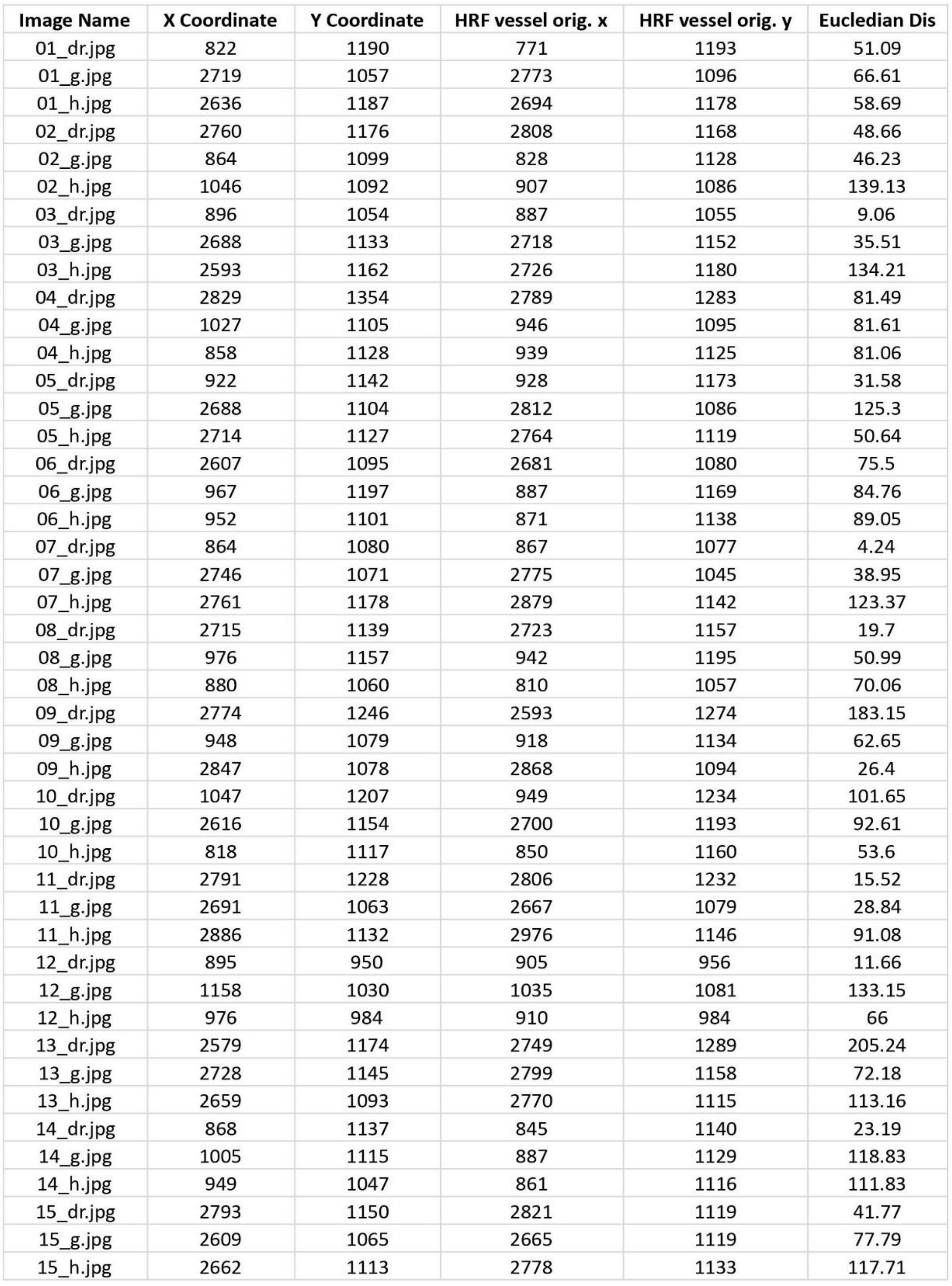

Image Name	X Coordinate	Y Coordinate	HRF vessel orig. x	HRF vessel orig. y	Eucledian Dis
01_dr.jpg	822	1190	771	1193	51.09
01_g.jpg	2719	1057	2773	1096	66.61
01_h.jpg	2636	1187	2694	1178	58.69
02_dr.jpg	2760	1176	2808	1168	48.66
02_g.jpg	864	1099	828	1128	46.23
02_h.jpg	1046	1092	907	1086	139.13
03_dr.jpg	896	1054	887	1055	9.06
03_g.jpg	2688	1133	2718	1152	35.51
03_h.jpg	2593	1162	2726	1180	134.21
04_dr.jpg	2829	1354	2789	1283	81.49
04_g.jpg	1027	1105	946	1095	81.61
04_h.jpg	858	1128	939	1125	81.06
05_dr.jpg	922	1142	928	1173	31.58
05_g.jpg	2688	1104	2812	1086	125.3
05_h.jpg	2714	1127	2764	1119	50.64
06_dr.jpg	2607	1095	2681	1080	75.5
06_g.jpg	967	1197	887	1169	84.76
06_h.jpg	952	1101	871	1138	89.05
07_dr.jpg	864	1080	867	1077	4.24
07_g.jpg	2746	1071	2775	1045	38.95
07_h.jpg	2761	1178	2879	1142	123.37
08_dr.jpg	2715	1139	2723	1157	19.7
08_g.jpg	976	1157	942	1195	50.99
08_h.jpg	880	1060	810	1057	70.06
09_dr.jpg	2774	1246	2593	1274	183.15
09_g.jpg	948	1079	918	1134	62.65
09_h.jpg	2847	1078	2868	1094	26.4
10_dr.jpg	1047	1207	949	1234	101.65
10_g.jpg	2616	1154	2700	1193	92.61
10_h.jpg	818	1117	850	1160	53.6
11_dr.jpg	2791	1228	2806	1232	15.52
11_g.jpg	2691	1063	2667	1079	28.84
11_h.jpg	2886	1132	2976	1146	91.08
12_dr.jpg	895	950	905	956	11.66
12_g.jpg	1158	1030	1035	1081	133.15
12_h.jpg	976	984	910	984	66
13_dr.jpg	2579	1174	2749	1289	205.24
13_g.jpg	2728	1145	2799	1158	72.18
13_h.jpg	2659	1093	2770	1115	113.16
14_dr.jpg	868	1137	845	1140	23.19
14_g.jpg	1005	1115	887	1129	118.83
14_h.jpg	949	1047	861	1116	111.83
15_dr.jpg	2793	1150	2821	1119	41.77
15_g.jpg	2609	1065	2665	1119	77.79
15_h.jpg	2662	1113	2778	1133	117.71

Table 2: Comparison with HRF Dataset

The above table depicts the comparison between the centroids of the predicted mask of our model and the ground-truth centroids provided with the dataset. The High-Resolution Fundus (HRF) dataset is a widely used collection of retinal fundus images that is frequently employed for research and development in the field of medical image analysis, particularly in ophthalmology.

The HRF is a public database currently consisting of 15 retinal images each from healthy patients, patients with diabetic retinopathy, and patients with glaucoma. For each image, binary gold standard vessel segmentation images are available, as well as masks delineating the field of view (FOV). These gold standard data were meticulously generated by experts in retinal image analysis and clinicians from collaborating ophthalmology clinics.

The above table describes the x and y positions of the centroids of the predicted masks by our model and compares it with the x and y positions of the centroids of the ground-truth annotations provided with the dataset. The results are almost accurate and gives better results than any other model for this comparison. Our work stands as a testament for the efficiency of our model which accurately localizes the optic disc in the retinal fundus images.

7.2.2 Prediction masks of HRF dataset

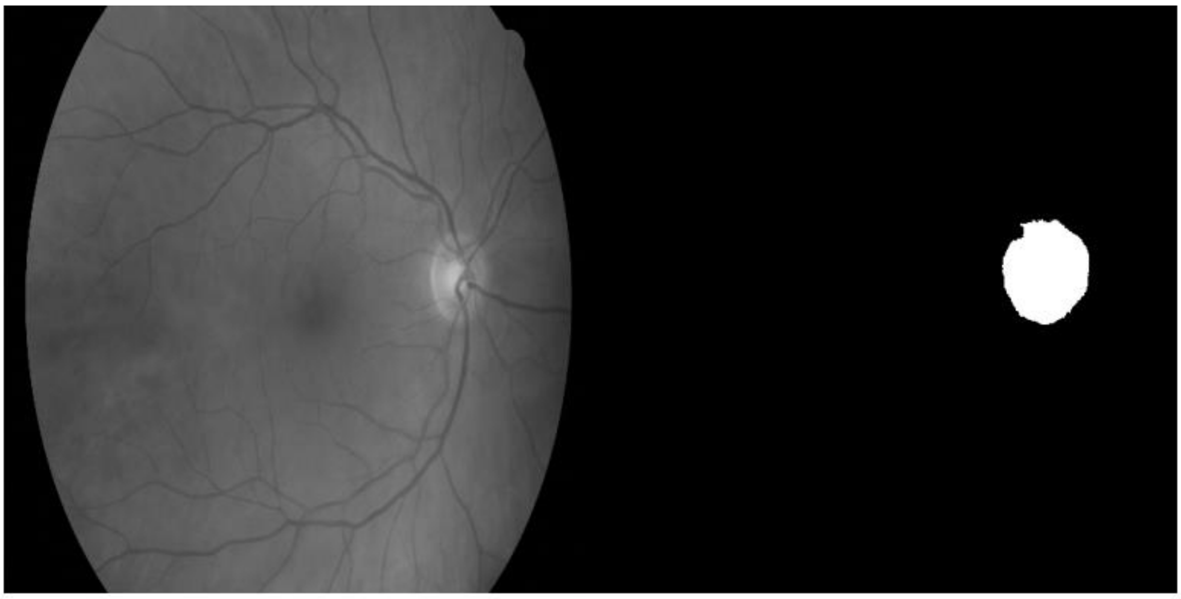

Fig 7.7: Detected optic disc mask 1

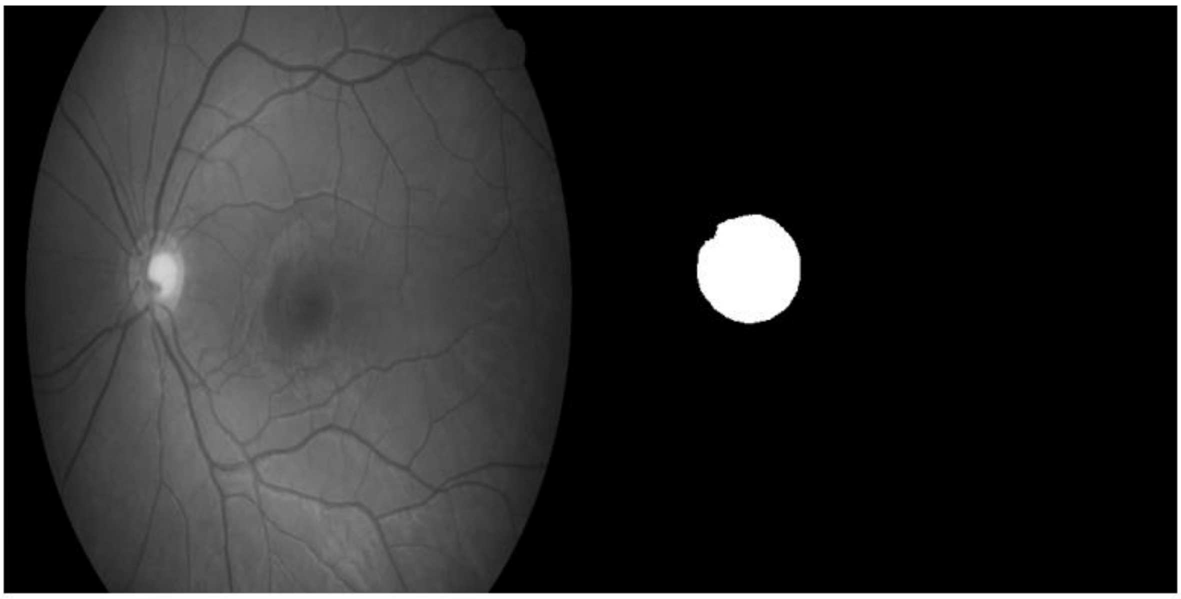

Fig 7.8: Detected optic disc mask 2

7.2.3 Comparative analysis with STARE dataset.

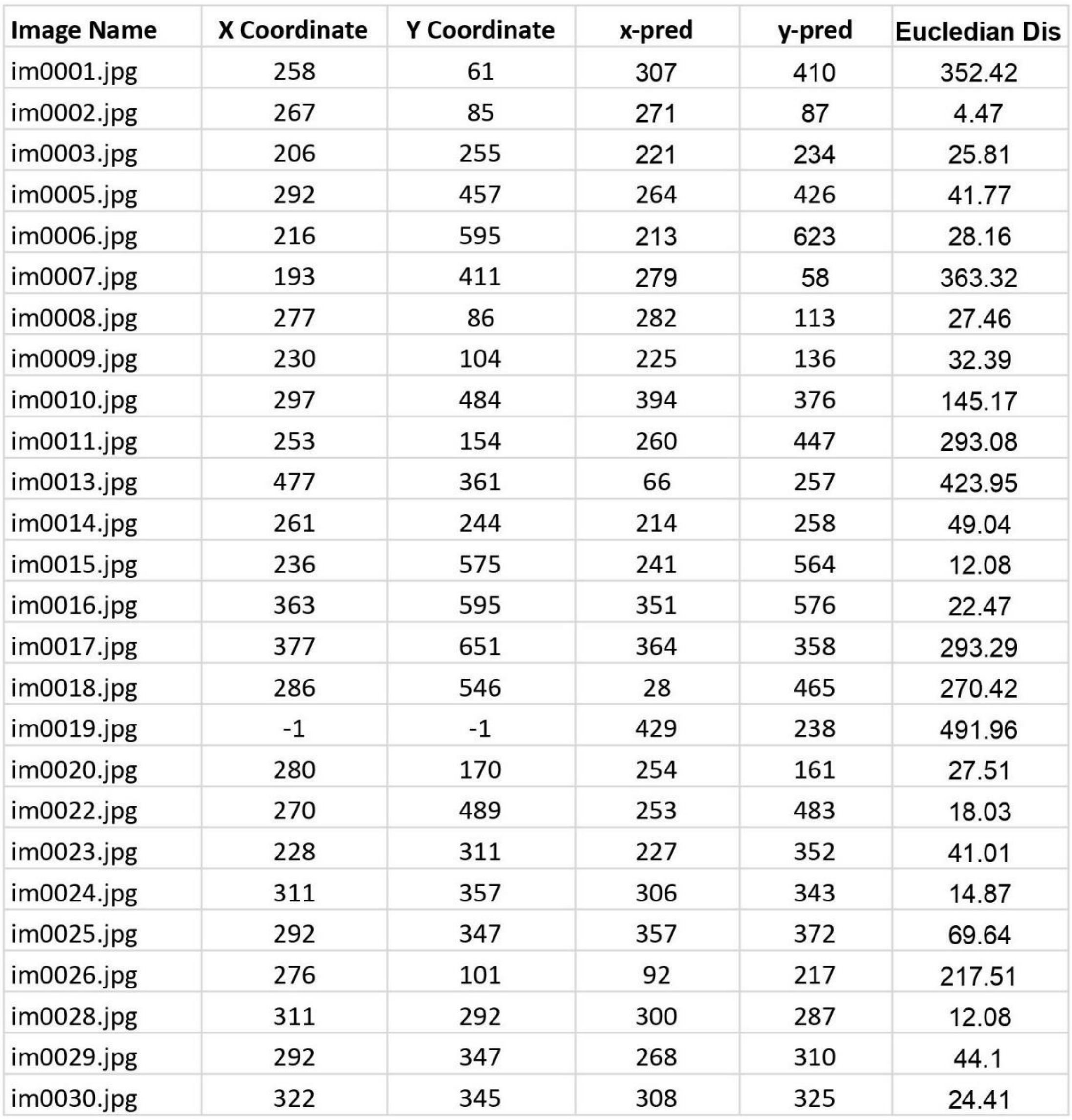

Image Name	X Coordinate	Y Coordinate	x-pred	y-pred	Eucledian Dis
im0001.jpg	258	61	307	410	352.42
im0002.jpg	267	85	271	87	4.47
im0003.jpg	206	255	221	234	25.81
im0005.jpg	292	457	264	426	41.77
im0006.jpg	216	595	213	623	28.16
im0007.jpg	193	411	279	58	363.32
im0008.jpg	277	86	282	113	27.46
im0009.jpg	230	104	225	136	32.39
im0010.jpg	297	484	394	376	145.17
im0011.jpg	253	154	260	447	293.08
im0013.jpg	477	361	66	257	423.95
im0014.jpg	261	244	214	258	49.04
im0015.jpg	236	575	241	564	12.08
im0016.jpg	363	595	351	576	22.47
im0017.jpg	377	651	364	358	293.29
im0018.jpg	286	546	28	465	270.42
im0019.jpg	-1	-1	429	238	491.96
im0020.jpg	280	170	254	161	27.51
im0022.jpg	270	489	253	483	18.03
im0023.jpg	228	311	227	352	41.01
im0024.jpg	311	357	306	343	14.87
im0025.jpg	292	347	357	372	69.64
im0026.jpg	276	101	92	217	217.51
im0028.jpg	311	292	300	287	12.08
im0029.jpg	292	347	268	310	44.1
im0030.jpg	322	345	308	325	24.41

Table 3: Comparison with STARE Dataset

The above table depicts the comparison between the centroids of the predicted mask of our model and the ground-truth centroids provided with the dataset. The STARE dataset is a widely used collection of retinal fundus images that is frequently employed for research and development in the field of medical image analysis, particularly in ophthalmology.

The STARE is a public database currently consisting of 400 retinal images. For each image, binary gold standard vessel segmentation images are available, as well as masks delineating the field of view (FOV). These gold standard data were meticulously generated by experts in retinal image analysis and clinicians from collaborating ophthalmology clinics.

The above table describes the x and y positions of the centroids of the predicted masks by our model and compares it with the x and y positions of the centroids of the ground-truth annotations provided with the dataset. The results are almost accurate and gives better results than any other model for this comparison. Our work stands as a testament for the efficiency of our model which accurately localizes the optic disc in the retinal fundus images.

7.2.4 Predicted masks of STARE dataset

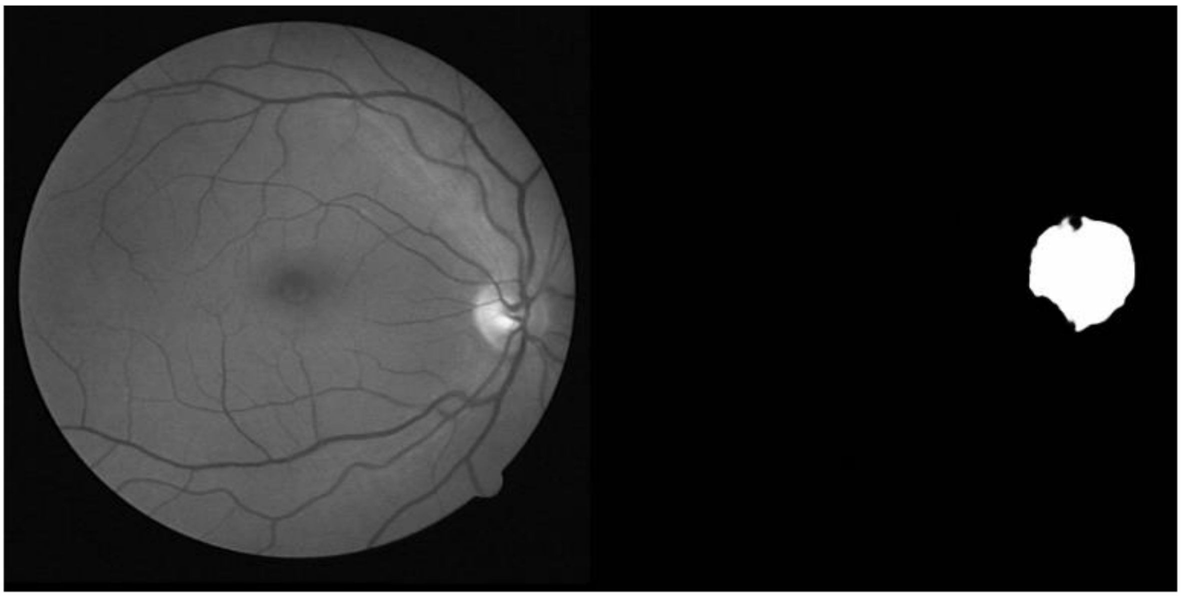

Fig 7.9: Detected optic disc mask 1

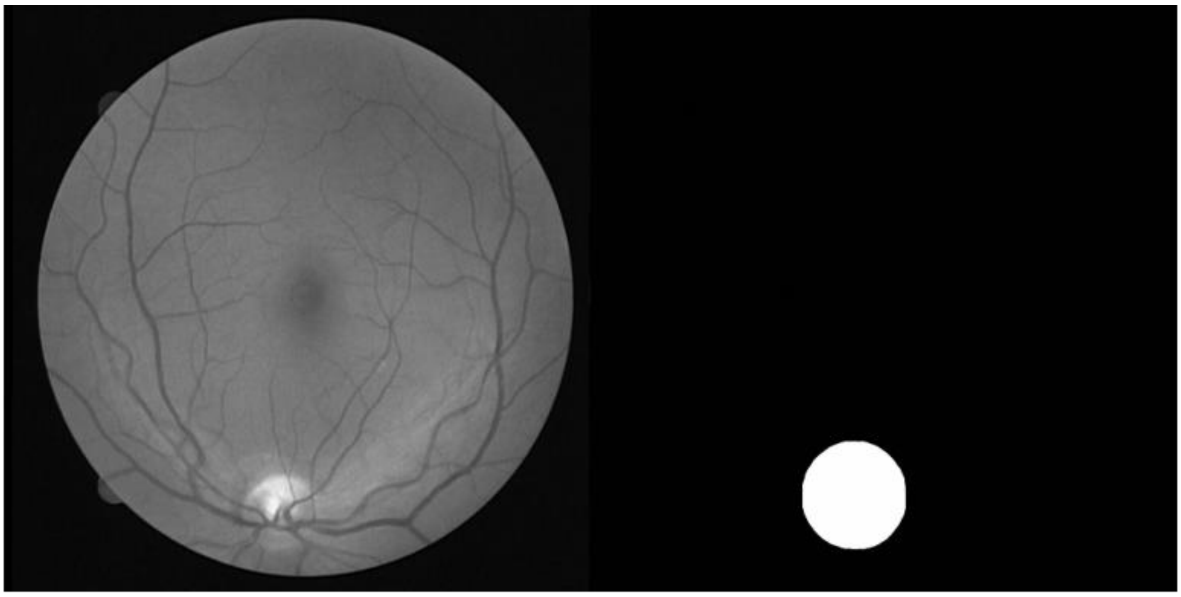

Fig 7.10: Detected optic disc mask 2

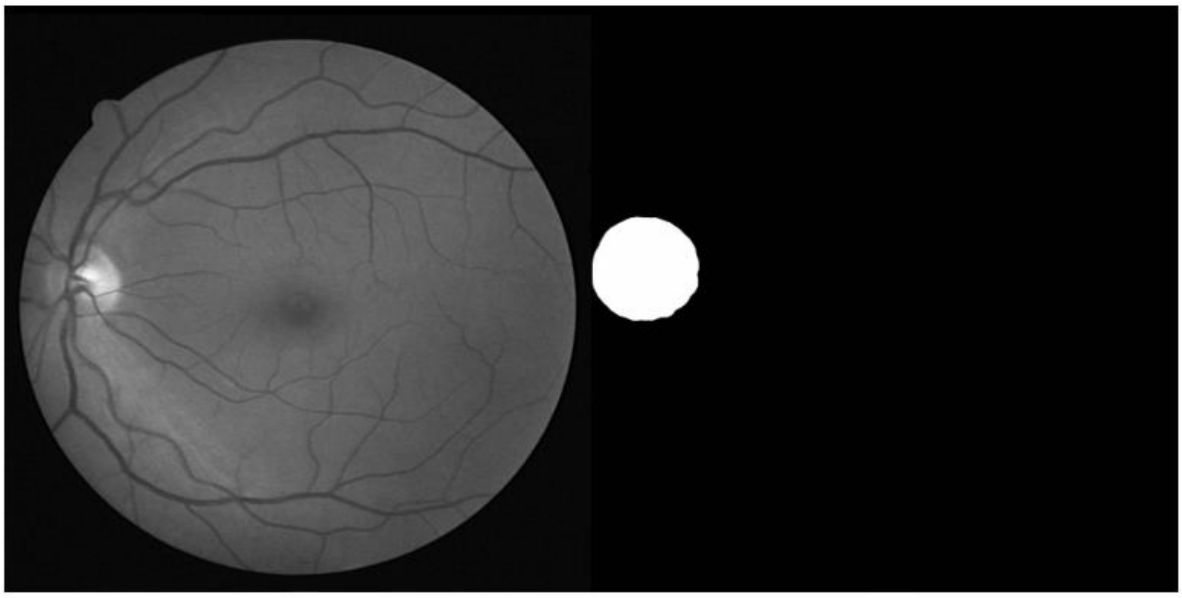

Fig 7.11: Detected optic disc mask 3

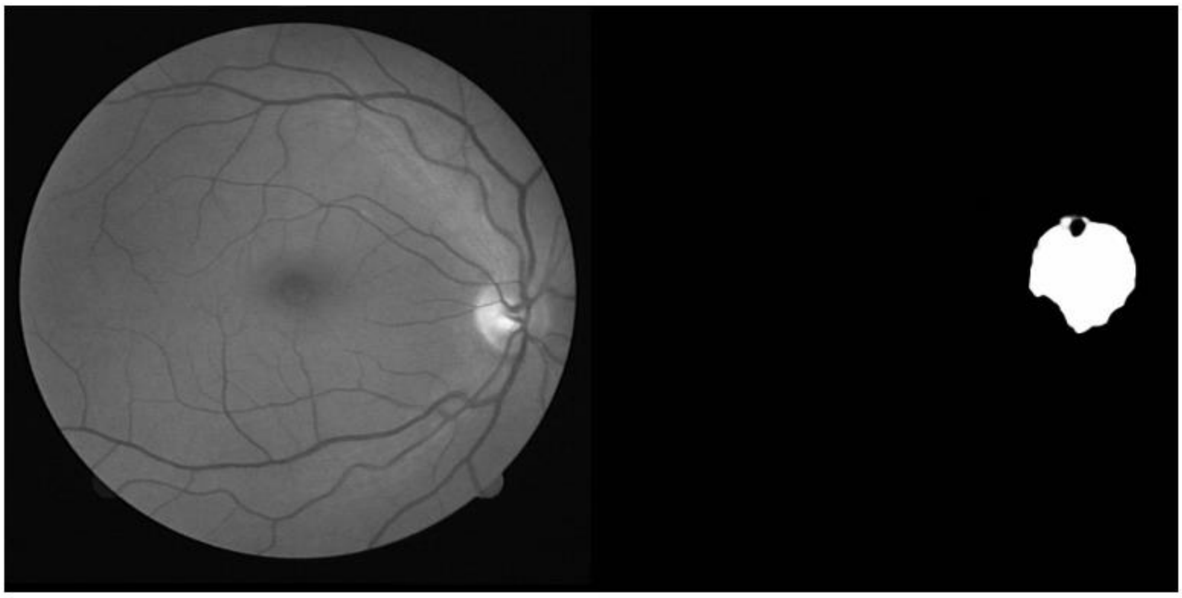

Fig 7.12: Detected optic disc mask 4

CHAPTER 8:

CONCLUSION

The developed system can be seen fulfilling both the objectives of the project. The proposed methodology efficiently detects the optic disc in retinal fundus images. The locations of the optic disc in the ground-truth images are detected based on the principle of raster scan. The window with maximum intensity was chosen as the location. Next a supervised learning model was developed which takes green channel fundus images as input and returns a mask localizing the area of optic disc in the fundus images.

Finally, contours were added to the resulting masks and the contour with largest area was displayed as output. Through the utilization of deep convolutional networks and image processing techniques, the model demonstrates promising results in accurately localizing and segmenting the optic disc. By leveraging data augmentation, training, and validation strategies, the model achieves robust performance across various epochs.

Furthermore, the comparison with existing labelled data validates the efficacy of the approach. The difference between the centroids of the predicted masks and the ground-truth masks for different datasets stand as a testament of real-life implementation of the model. It also suggests how accurate the results of the model are and how efficiently the model works. The model returns an average IoU of about 78% and provides better results than previous works on this field.

CHAPTER 9: FUTURE SCOPE

The existing model can be modified more by using different algorithms for localizing the optic disc to increase the accuracy as well as the neural network needs to be modified and worked on to decrease the computation time and achieve even better results. This will improve the overall performance of the developed system. Continuous refining and optimizing the algorithms and techniques used in optic disc segmentation is required which could lead to better accuracy and efficiency in detecting optic discs from retinal images.

Exploring advanced deep learning architectures and techniques, such as attention mechanisms, adversarial training, or self-supervised learning, could also potentially improve the performance and generalization capabilities of the system. The scope of the project and also be expanded to include segmentation of other anatomical structures or abnormalities in retinal images, such as blood vessels, lesions, or macula, could provide a more comprehensive diagnostic solution.

The next step of this project is to work on video datasets in which we will try to track the position of the optic disc continuously in each frame and return its position and other features. Integrating the developed optic disc detection system into clinical workflows and systems or wearables could also be beneficial.

Integration of optic disc detection into wearable devices would involve incorporation of technology that can continuously monitor and analyze the optic disc's health in real-time. This will offer a proactive approach to eye health monitoring, allowing for early detection of potential issues and timely intervention and will prove to be beneficial for patients.

REFERENCES

- O. Ronneberger, P. Fischer, and T. Brox, "U-NET: Convolutional Networks for Biomedical Image Segmentation," in *Lecture Notes in Computer Science*, 2015, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28.

- "Automatic localization of the optic disc based on iterative brightest pixels extraction," *IEEE Conference Publication | IEEE Xplore*, Jun. 01, 2014. https://ieeexplore.ieee.org/document/6845958

- "Automatic localization and segmentation of Optic Disc in retinal fundus images through image processing techniques," *IEEE Conference Publication | IEEE Xplore*, Apr. 01, 2014. https://ieeexplore.ieee.org/document/6996090

- "Optic Disc Localization using Interference Map and Localized Segmentation," *IEEE Conference Publication | IEEE Xplore*, Jul. 01, 2019. https://ieeexplore.ieee.org/document/8938221

- "New optic disc localization approach in retinal images," *IEEE Conference Publication | IEEE Xplore*, Nov. 01, 2013. https://ieeexplore.ieee.org/document/6707418

- "Automatic optic disc detection in digital fundus images using image processing techniques," *IEEE Conference Publication | IEEE Xplore*, Feb. 01, 2014. https://ieeexplore.ieee.org/document/7034118

- "Deep Convolutional Networks for Image Segmentation: Application to Optic Disc detection," *IEEE Conference Publication | IEEE Xplore*, Mar. 01, 2020. https://ieeexplore.ieee.org/document/9113204

- G. Y. Kim, S. H. Lee, and S. M. Kim, “Automated segmentation and quantitative analysis of optic disc and fovea in fundus images,” *Multimedia Tools and Applications*, Apr. 2021, doi: 10.1007/s11042-021-10815-1.

- B. J. Bhatkalkar, S. V. Nayak, S. V. Shenoy and R. V. Arjunan, "FundusPosNet: A Deep Learning Driven Heatmap Regression Model for the Joint Localization of Optic Disc and Fovea Centers in Color Fundus Images," in *IEEE Access*, vol. 9, pp. 159071-159080, 2021, doi: 10.1109/ACCESS.2021.3127280.

- J. Dietter *et al.*, “Optic disc detection in the presence of strong technical artifacts,” *Biomedical Signal Processing and Control*, vol. 53, p. 101535, Aug. 2019, doi: 10.1016/j.bspc.2019.04.012.

- M. W. Khan, “RVD: a Handheld Device-Based FunDUs video dataset for retinal vessel segmentation,” *arXiv.org*, Jul. 13, 2023. https://arxiv.org/abs/2307.06577

- B. Jin, P. Li, P. Wang, L. Shi, and J. Zhao, “Optic disc segmentation using Attention-Based U-Net and the improved Cross-Entropy convolutional neural network,” *Entropy*, vol. 22, no. 8, p. 844, Jul. 2020, doi: 10.3390/e22080844.

- N. Modi and J. Singh, “Role of eye tracking in human computer interaction,” *ECS Transactions*, vol. 107, no. 1, pp. 8211–8218, Apr. 2022, doi: 10.1149/10701.8211ecst.

- M. Maynard-Reid, “U-Net image segmentation in Keras - PyImageSearch,” *PyImageSearch*, Mar. 23, 2023. https://pyimagesearch.com/2022/02/21/u-net-image-segmentation-in-keras/

- A. A. Awan, “A complete guide to data augmentation,” Nov. 23, 2022. https://www.datacamp.com/tutorial/complete-guide-data-augmentation