BLOOD VESSEL EXTRACTION USING DEEP NEURAL NETWORK

Report submitted to the SASTRA Deemed to be University

As the requirement for the course

ECE300: MINI PROJECT

Submitted by

RAJAGIRI NARESH KUMAR

(Reg. No:125004170)

KALAGA V S K ADITYA

(Reg. No:125004390)

L VENKATESHWARLU

(Reg No.125004343)

SCHOOL OF ELECTRICAL & ELECTRONICS ENGINEERING

THANJAVUR 613 401

SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

Bonafide Certificate

This is to certify that the report titled **"BLOOD VESSEL EXTRACTION USING DEEP NEURAL NETWORK "** submitted as a requirement for the course**, ECE300: MINI PROJECT** for B.Tech. ELECTRONICS & COMMUNICATION ENGINEERING programme, is a bonafide record of the work done by **Mr. R.NARESH KUMAR (Reg. No:125004170), Mr. KALAGA V S K ADITYA (Reg. No:125004390), Mr. L.VENKATESHWARLU (Reg. No:125004343)** during the academic year 2023-24, in the School of ELECTRICAL & ELECTRONICS ENGINEERING, under my supervision.

Signature of project supervisor **:** 

Name With Affiliation **:** Dr. Narasimhan K, Sr. Asst Professor, SEEE

Date **:**

Project *Viva voce* held on _____

Examiner 1 **Examiner 2**

SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

SCHOOL OF ELECTRICAL & ELECTRONICS ENGINEERING

THANJAVUR – 613 401

Declaration

We declare that the report titled **"BLOOD VESSEL EXTRACTION USING DEEP NEURAL NETWORK "** submitted by us is an original work done by us under the guidance of **Dr. Narasimhan K, Sr. Asst Professor, SEEE, School of Electrical and Electronics Engineering, SASTRA Deemed to be University** during the sixth semester of the academic year 2023-24, in the **School of Electrical and Electronics Engineering.** The work is original and wherever I/We have used materials from other sources, I/We have given due credit and cited them in the text of the report. This report has not formed the basis for the award of any degree, diploma, associate-ship, fellowship or other similar title to any candidate of any university.

Signature of the candidates:

(R.Naresh Kumar) **(KALAGA V S K ADITYA)** **(L.Venkateshwarlu)**

Date:

Acknowledgements

We would like to thank our Honourable Chancellor **Prof. R. SETHURAMAN** for providing an opportunity and the necessary infrastructure for carrying out this project as a part of our curriculum.

First and foremost, we are thankful to God Almighty for giving us the potential and strength to work on this project fruitfully. We express our gratitude to our Honorable Vice-**Chancellor Dr. S. VAIDHYASUBRAMANIAM** and **Dr. S.SWAMINATHAN,** Dean of Planning & Development of SASTRA Deemed to be University for providing us with an encouraging platform for the course of study.

We also thank **Dr. R. CHANDRAMOULI**, Registrar, SASTRA Deemed to be University, for providing us the opportunity to work in the SEEE, SASTRA Deemed to be University for the course of the work.

We render our solicit thanks to **Dr. K. THENMOZHI**, Dean, SEEE, **Dr. A. KRISHNAMOORTHY**, Associate Dean Academics, SEEE, SASTRA Deemed to be University for introducing the Mini-project in our curriculum.

We owe a debt of earnest gratitude towards our guide **Dr. Narasimhan K**, Sr. Asst Professor, SEEE for his continued support and guidance throughout the course of our work. The intriguing questions raised by our panel members **Dr. Narasimhan K**, Sr. Asst. Professor, SEEE and **Dr. Lakshmi C**, Asst. Professor, SEEE helped us in shaping this project to its worth.

We also extend our sincere thanks to our parents for their continuous moral support and friends without whose thought-provoking suggestion and encouragement, along with our guide, this work would not have been what it is today

Abstract

The extraction of retinal blood vessel is more important. The retinal vascular problems can lead to blindness and even death due to poor vision of the patient. Understanding the characteristics of blood vessels can greatly assist in diagnosing conditions like hypertensive retinopathy and diabetic retinopathy, making it a valuable tool for ophthalmologists. Manual segmentation of blood vessels is time-consuming and requires a high level of skill. Therefore, automating this process with a computerized system saves significant time and allows more patients to receive timely consultations

In our project, we utilized the DRIVE dataset, which contains fundus images of the eye, capturing the internal eye structure. Fundus images are essential for diagnosing and monitoring various eye conditions. Our primary goal was to extract blood vessels from fundus images using deep neural architectures such as U-Net and Staircase-Net. U-Net employs an encoder-decoder architecture and has a total of 31,055,297 learnable parameters with accuracy of 94%, while Staircase-Net utilizes upsampling and downsampling techniques and has total of 1,727,970 learnable parameters with accuracy of 80%, though this model has slightly less accuracy. Still the higher performance of the other metrics makes it a good trade-off. The comparison of these models against traditional methods revealed that our CNN models outperformed the existing techniques significantly. By utilizing these advanced models for segmentation, we aim to enhance the efficiency and accuracy of diagnosing retinal vascular issues, ultimately improving patient care and outcomes. this automatic segmentation using deep learning helps in diagnosing process, allowing healthcare professionals to focus more on treatment and patient interaction.

Keywords: *Retinal vessel; segmentation; convolutional neural network; fundus image,U-Net,Staircase-Net.*

Abbreviations*:*

1. CNN*:* Convolutional Neural Network
2. CLAHE: Contrast Limited Adaptive Histogram Localisation
3. DRIVE: Digital Retinal Images for Vessel Segmentation
4. ReLU: Rectified Linear Unit
5. RGB: Red Green Blue
6. AUC: Area Under Curve
7. BCE: Binary Cross Entropy
8. IOU: Intersection Over Union

List of Figures

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

The retinal blood vessels play a crucial role in maintaining the health of the retina by supplying oxygen and nutrients. The central retinal artery, which enters the eye through the optic nerve, branches into upper and lower units, forming a complex network of capillaries resembling tree branches. Analyzing attributes such as vessel width, length, and tortuosity can assist in diagnosing various conditions, making this study valuable for ophthalmologists

Retinal vascular problems can lead to severe consequences, including blindness and even death in extreme cases due to impaired vision. Manual segmentation of retinal vessels, traditionally used for diagnosis, is laborious and time-consuming, requiring significant skill. Additionally, the low contrast of blood vessel regions makes accurate segmentation challenging.

Automated segmentation systems for early detection of retinal illnesses including hypertensive retinopathy and diabetic retinopathy are important. These systems save time, allowing more patients to consult doctors promptly, leading to early disease detection and intervention.

However, developing an effective automated system is complex due to factors like vessel shape variations, optic disc structures, and environmental conditions such as noise and contrast levels. Among supervised methods, deep neural network architectures, particularly convolutional neural networks (CNNs), have shown superior performance in vessel segmentation tasks.

Early approaches using Artificial Neural Networks (ANNs) have evolved into CNN-based models that achieve remarkable accuracy. These advancements in deep learning have significantly improved the accuracy and efficiency of automated retinal vessel segmentation systems, benefiting both patients and healthcare professionals.

In this work, segmenting retinal blood vessels is made faster by using image processing followed by a trained Convolutional Neural Network (CNN).

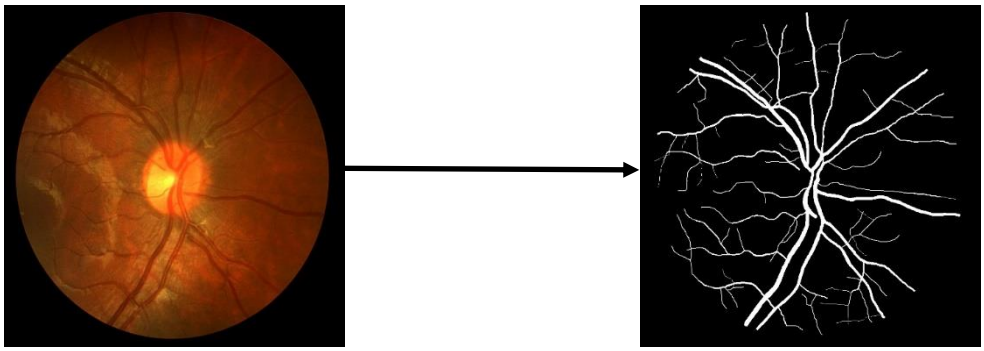


Fig. 1.1 – Sample of fundus image and segmented blood vessels

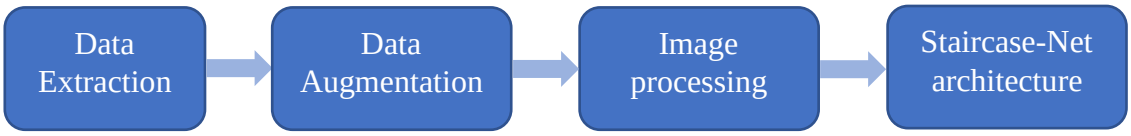
OBJECTIVE

The objective of the work is to study supervised Convolutional Neural Networks (CNN) called ‘U-Net’ and 'Staircase-Net' for automated retinal blood vessel segmentation. The model utilizes publicly available dataset (DRIVE) and employs a series of up-sampling and downsampling processes for feature extraction. The preprocessing steps involve image transformations, and the evaluation metrics include specificity, accuracy, sensitivity, and area under the curve.

Overall, our model contributes to the field of medical image analysis by providing a robust and reliable method for automated retinal blood vessel segmentation. This advancement has the potential to assist healthcare professionals in the early detection and diagnosis of various retinal diseases, ultimately improving patient outcomes and treatment strategies.

CHAPTER 2

METHODOLOGY



2.1 Data Extraction:

The dataset used in this project is DRIVE (Digital Retinal Images for Vessel Extraction) dataset. This dataset is a widely used dataset in the field of medical image analysis, particularly for tasks related to retinal vessel segmentation.

The images in the DRIVE dataset are captured from digital fundus camera, which is used to photograph the back of the eye(retina). It contains 40 high-resolution colored retinal images, is a cornerstone in diabetic retinopathy. These images, freely available in Kaggle.

Each picture is 564x584 pixels in size and has been JPEG compressed. The dataset's division into 20 training and 20 testing images, each accompanied by ground truth masks, these masks are created by experts who annotate the images to mark the pixels corresponding to blood vessels. This ensures comprehensive model assessment, making it an invaluable resource for advancing automated retinal image analysis. DRIVE has a standardized platform for developing and evaluating CNN-based segmentation algorithm

2.1.1 Challenges and Use Cases:

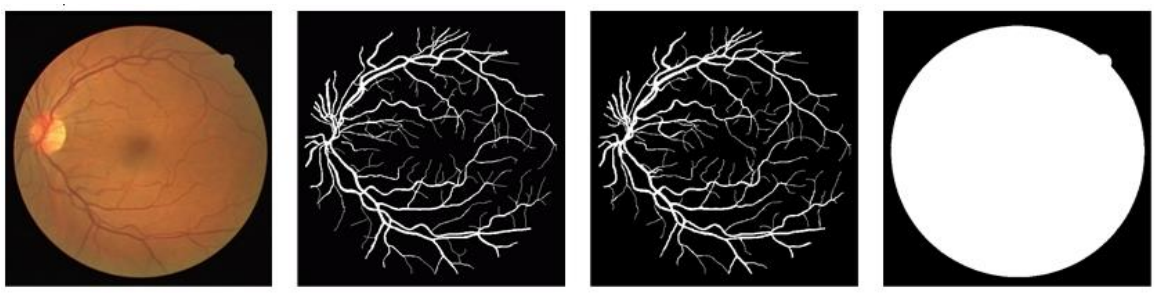
Retinal vessel segmentation is a challenging task due to the complex and fine structure of blood vessels, as well as variations in image quality and pathology. The DRIVE dataset is commonly used to develop and evaluate algorithms and deep learning models for automatic segmentation of retinal blood vessels.

2.1.2 Ethical Considerations:

When working with medical datasets like DRIVE, it's important to consider ethical guidelines regarding data usage, privacy, and patient consent. Researchers typically ensure that the data is used responsibly and in compliance with relevant regulations. By

using the DRIVE dataset responsibly and acknowledging its source, researchers contribute to advancements in medical imaging and healthcare by developing more accurate and efficient methods for retinal vessel segmentation and analysis.

	Testing	Training
Fundus Image	20	20
Mask	20	20



Fundus image Ground truth_1 Ground truth_2 Mask

Fig. 2.1.1 Drive training dataset

2.2 Data augmentation:

Data augmentation involves artificially expanding a dataset by applying various transformations to the existing data. By introducing these variations, data augmentation helps improve the generalization of machine learning models, by exposing them to a wider range of possible inputs without collecting additional labeled data.

As the dataset contains various size of images making as a unique size of 512X512.

Data augmentation techniques in this project are:

- **Horizontal Flip:** Horizontal flipping is the act of turning a picture such that the left side is now the right side and vice versa.
- **Vertical Flip:** data augmentation process involves flipping an image vertically, so that the top becomes the bottom and vice versa.
- **Elastic transform:** data augmentation process involves distorting an image using a combination of random elastic deformations and smoothing, to create more varied training examples for the model.
- **Grid Distortion:** Grid distortion data augmentation process involves dividing an

image into a grid and randomly displacing the grid points to create a distorted version of the original image for training the model.

- **Optical Distortion:** data augmentation process involves simulating lens distortion, such as barrel or pincushion distortion, to augment images for training the model.
- **Shift scale rotate:** Shift Scale Rotate data augmentation process involves randomly applying affine transformations including shifting, scaling, and rotating an image to create new training examples for the model.

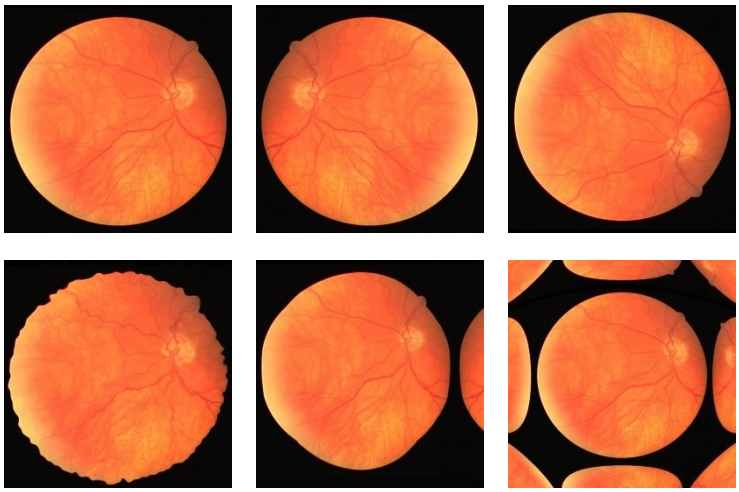


Fig.2.2.1 Data augmentation sample output

2.2.1 Data Augmentation Sample Output:

As a result, training data after augmentation is total with 140 funds images with their respective masks.

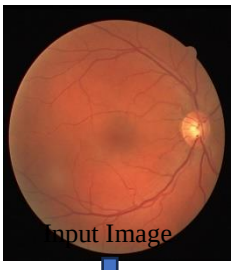
2.3 Image Processing:

Image preprocessing is essential for enhancing model performance by standardizing and optimizing input data. It includes tasks like normalization, resizing, and noise reduction, which ensure uniformity and quality across images, enabling more effective feature extraction and model training. Additionally, it also helps in improving generalisation of model.

2.3.1 Image processing techniques:

IMAGE PRE-PROCESSING

In image preprocessing fundus image of the retina is give as input and it undergoes following conversions to form pre-processed image.



Input Image

RGB to Lab Conversion

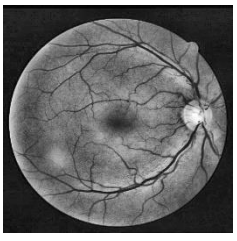
L Channel Extraction

CLAHE

Recombination and conversion to RGB

Green Channel Extraction

CLAHE



Pre-Processed Image

- **RGB TO LAB CONVERSION:**

  Converting RGB to LAB colour space in image preprocessing for blood vessel segmentation enhances model performance by separating colour information from luminance. LAB space separates lightness (L) from colour information (A and B), enabling better discrimination of blood vessels from background, especially in varying lighting conditions or image quality, ultimately improving segmentation accuracy

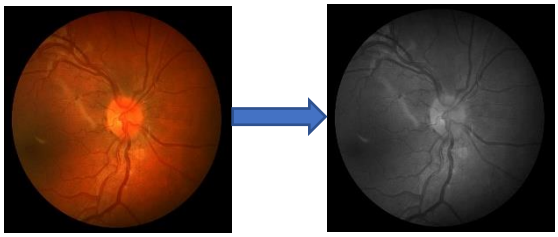


Fig. 2.3.1 sample image of RGB TO LAB

- **L Channel Extraction:**

  L channel for use in further image processing from the A and B components. L channel extraction involves isolating the luminance component from an image in LAB color space. This channel primarily represents the lightness or brightness information, providing crucial grayscale details essential for blood vessel segmentation. By focusing on luminance, L channel extraction enhances contrast and accentuates vessel structures, aiding in more accurate segmentation results.

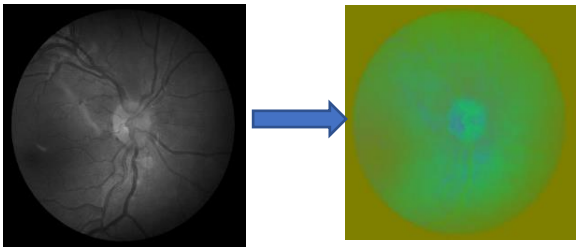


Fig. 2.3.2 sample image of L channel extraction

- **Recombination and Conversion back to RGB:**

Following CLAHE, the image is typically converted back to RGB color space for further analysis or visualization. This conversion ensures consistency in the color representation of the enhanced image while preserving the improvements made through preprocessing techniques

- **CLAHE (CONTRAST LIMITED ADAPTIVE HISTOGRAM LOCALISATION):**

The next step after L channel extraction in image preprocessing for blood vessel segmentation may involve applying Contrast Limited Adaptive Histogram Equalization (CLAHE). CLAHE enhances local contrast in the L channel, improving the visibility of blood vessels against varying background intensities and enhancing

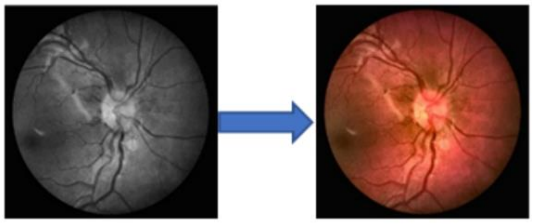


Fig.2.3.3 sample image of CLAHE

segmentation performance in regions with uneven illumination.

- **Green Channel Extraction:**
  After converting the image back to RGB, the next step involves extracting the green channel. In retinal imaging, the green channel often exhibits better contrast for blood vessels compared to other channels due to hemoglobin absorption characteristics. Isolating the green channel enhances vessel visibility and prepares the image for subsequent segmentation steps.

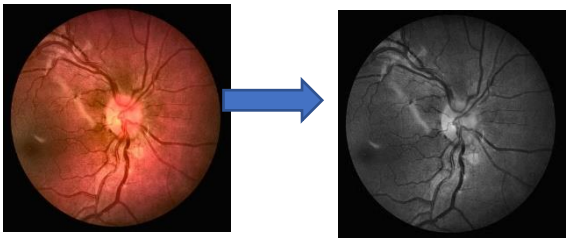


Fig.2.3.4 sample image of
green channel extraction

Applying the CLAHE algorithm will increase the contrast of the blood vessels alone.

- **CLAHE:**

Following green channel extraction, applying CLAHE (Contrast Limited Adaptive Histogram Equalization) further enhances contrast and improves the visibility of blood vessels in the green channel image. CLAHE adapts the histogram equalization process locally, enhancing local contrast while preventing over-amplification of noise, thereby facilitating more accurate blood vessel segmentation.

We Get A Final Pre-processed Image Here,

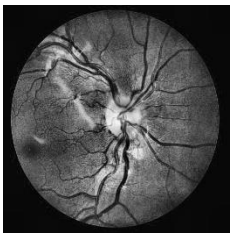


Fig. 2.3.5 Final pre-processed image

2.4 U-Net Architecture:

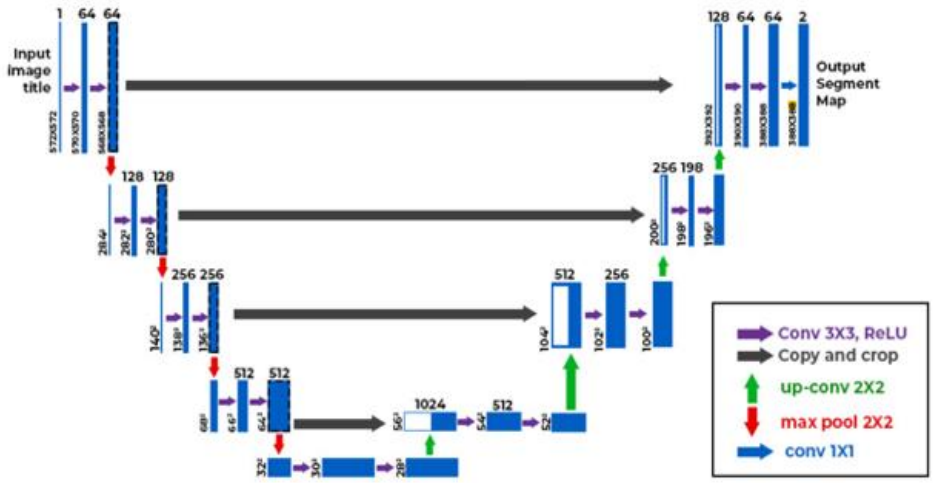


Fig.2.4.1 U-Net Architecture

The U-Net architecture is a convolutional neural network (CNN) commonly used for semantic segmentation tasks, including medical image segmentation such as blood

vessels. Its key feature is its U-shaped architecture, which consists of a contracting path (encoder) and an expansive path (decoder) with skip connections.

The contracting path contains encoding layers that captures contextual information and reduce the spatial resolution of the input

While the expansive path contains the decoder layers that decodes the encoded data and use the information from the contracting path

2.4.1 Contracting Path (Encoder):

- Consists of a series of convolutional and pooling layers.
- These layers progressively reduce spatial dimensions while increasing the number of feature channels, capturing high-level features.

2.4.2 Expansive Path (Decoder):

- Comprises up-sampling and convolutional layers.
- It gradually recovers the spatial dimensions while reducing the number of feature channels, reconstructing the segmented output.

2.4.3 Skip Connections:

- Connects corresponding layers between the encoder and decoder.
- These connections concatenate feature maps from the contracting path to the corresponding layers in the expansive path, enabling precise localization of features and alleviating information loss during downsampling.

Each pixel is scaled from 0 to 1.

In the U-Net architecture, commonly used activation functions include:

Rectified Linear Unit (ReLU):

ReLU is the most widely used activation function in deep learning, including U-Net. It introduces non-linearity by replacing negative values with zeros, helping the network learn complex relationships in the data.

Sigmoid:

Sigmoid activation function is often used in the output layer of U-Net for binary segmentation tasks. It squashes the output values between 0 and 1, making them interpretable as probabilities.

SoftMax:

SoftMax activation function is used in the output layer for multi-class segmentation tasks. It normalizes the output values across multiple classes, ensuring that the predicted probabilities sum up to 1 for each pixel.

These activation functions play a crucial role in introducing non-linearity, controlling the output range, and facilitating the learning process of the U-Net model for various segmentation tasks.

2.5 Staircase-Net architecture:

Staircase-Net is a variation of the U-Net CNN architecture. It's built with a series of blocks that alternate between expanding (unsampled) and contracting (down sampled) layers, and then combines their outputs together.

These blocks use filters whose sizes are determined by the input data, and they're part of convolutional layers that create feature maps by spotting patterns in pictures. Alongside convolution, pooling, and fully connected layers form the network structure, aiding in categorizing images. Activation layers add complexity by introducing non-linearities, helping the network learn intricate relationships within the data. Pooling layers then simplify the maps, reducing their size to speed up processing and prevent overfitting. Finally, the processed data flows through a feed-forward network for predictions and classification.

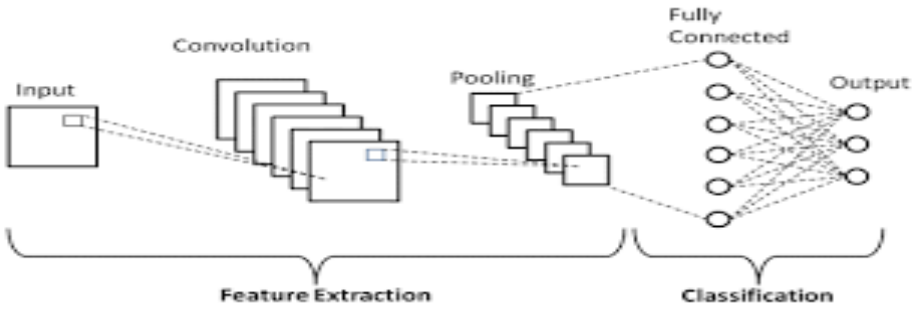


Fig.2.5.1 Basic CNN architecture

The Staircase-Net model is created by cascading two Fully Convolutional Neural Networks (FCNNs) that are comparable to each other.

The output of the first FCNN is concatenated with the input picture and then fed as input to the second FCNN. This model uses a series of convolution layers followed by up- and down-sampling in the FCNN layers.

Thin vessel features are extracted using the up-sampling layers, while thick vessel features are extracted using the down-sampling layers.

There are a total of 12 convolution blocks, containing 36 convolution layers with a filter size of 3x3 and same padding, ensuring the output shape matches the input shape. The activation function used is ReLU (Rectified Linear Unit), and the output layer of the FCNN uses a sigmoid activation function to produce a binary image with a filter size of 1x1.

To enhance the model's performance, batch normalization is applied. Up-sampling is achieved using Conv2D transpose with a specified number of filters, along with the preceding convolution block and padding. Down-sampling is performed using Maxpooling2D with the filters from previous layers.

The input shape features for the model are defined as (Batch size, size of the image, color channels), with a total of 17,24,258 trainable parameters. The input image size is set to 512x512, and the batch size can vary.

This architecture is implemented using TensorFlow and the Keras module, both of which are open-source tools commonly used in deep learning projects.

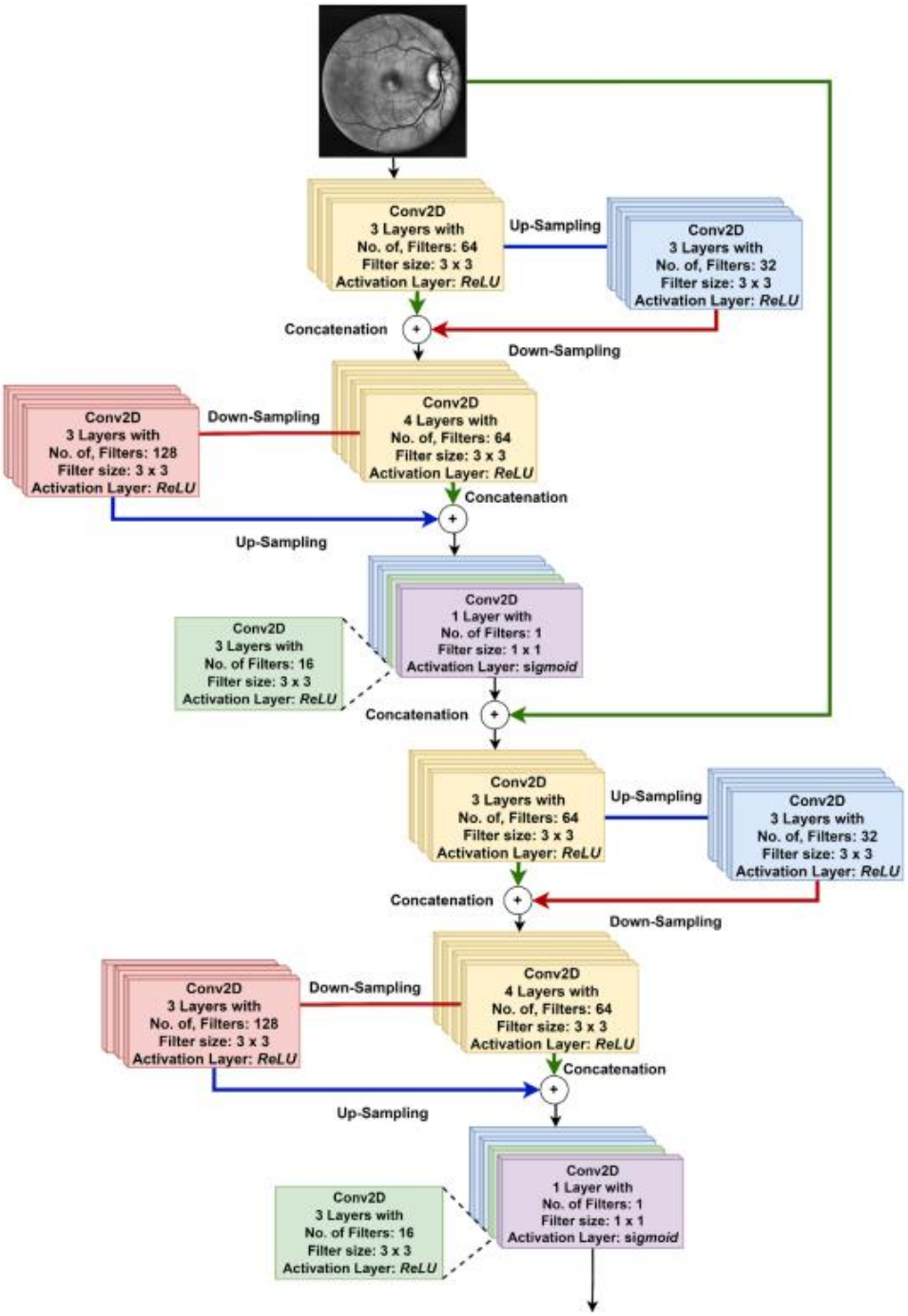


Fig 2.5.2 Staircase Architecture

CHAPTER 3

Metrics

The model's performance is evaluated using several metrics, including Binary Cross-Entropy (BCE) loss, specificity, sensitivity, area under the curve (AUC), accuracy, and validation loss.

Binary Cross-Entropy (BCE) loss is a common metric for binary classification tasks. It calculates the difference between the actual probability distribution of target labels and the predicted probability distribution.

The formula for BCE can be expressed as,

$$BCE = -(a * log(b) + (1 - a) * log(1 - b))$$

$$'a' - ground\ truth\ label\ (either\ 0\ or\ 1)\ \&\ 'b' - predicted\ probability.$$

Other metrics used for evaluation are sensitivity, specificity, Area under curve, accuracy, validation loss.

Sensitivity measures the model's ability to correctly identify positive samples, specifically the percentage of true positives (TP) among all real positives (TP + FN). The formula for sensitivity is :

$$sensitivity = TP/(TP + FN)$$

Specificity, on the other hand, assesses the model's accuracy in identifying negative samples. It is calculated using the formula TN / (TN + FP), where TN is true negatives

$$Specificity = TN / (TN + FP).$$

and FP is false positives The formula for specificity is :

The Area Under Curve (AUC) is used to evaluate how well the model distinguishes between positive and negative classes in binary classification. A higher AUC indicates better performance, with the AUC calculated based on the true positive rate (sensitivity) versus the false positive rate (1-specificity) across different classification thresholds.

Accuracy is a widely used metric that measures the proportion of correctly classified samples (true positives and true negatives) out of all samples evaluated.

$$The\ formula\ for\ calculating\ accuracy\ is:$$

$$Accuracy\ =\ (TP\ +\ TN)\ /\ (TP\ +\ TN\ +\ FP\ +\ FN)$$

$$where\ TP\ stands\ for\ true\ positives$$

$$TN\ for\ true\ negatives$$

$$FP\ for\ false\ positives$$

$$FN\ for\ false\ negatives.$$

A performance statistic called validation loss is used in machine learning to assess how well a model performs on a different validation dataset. The goal is to evaluate how effectively the model generalizes to fresh, untested data. The training loss—the discrepancy between the model's predictions and the actual labels on the training dataset—is minimized while the model is being trained. As a result, the validation dataset is used to test the model after each training epoch, and the validation loss is calculated by computing the error between the model's predictions and the validation data's real labels. This makes it possible to monitor the model's generalization abilities and decide when to halt training to prevent overfitting.

CHAPTER 4

OUTPUT REFFERENCES

4.1 Data Augmentation:

As a result of data Augmentation, train images and mask are increased to 140 samples.

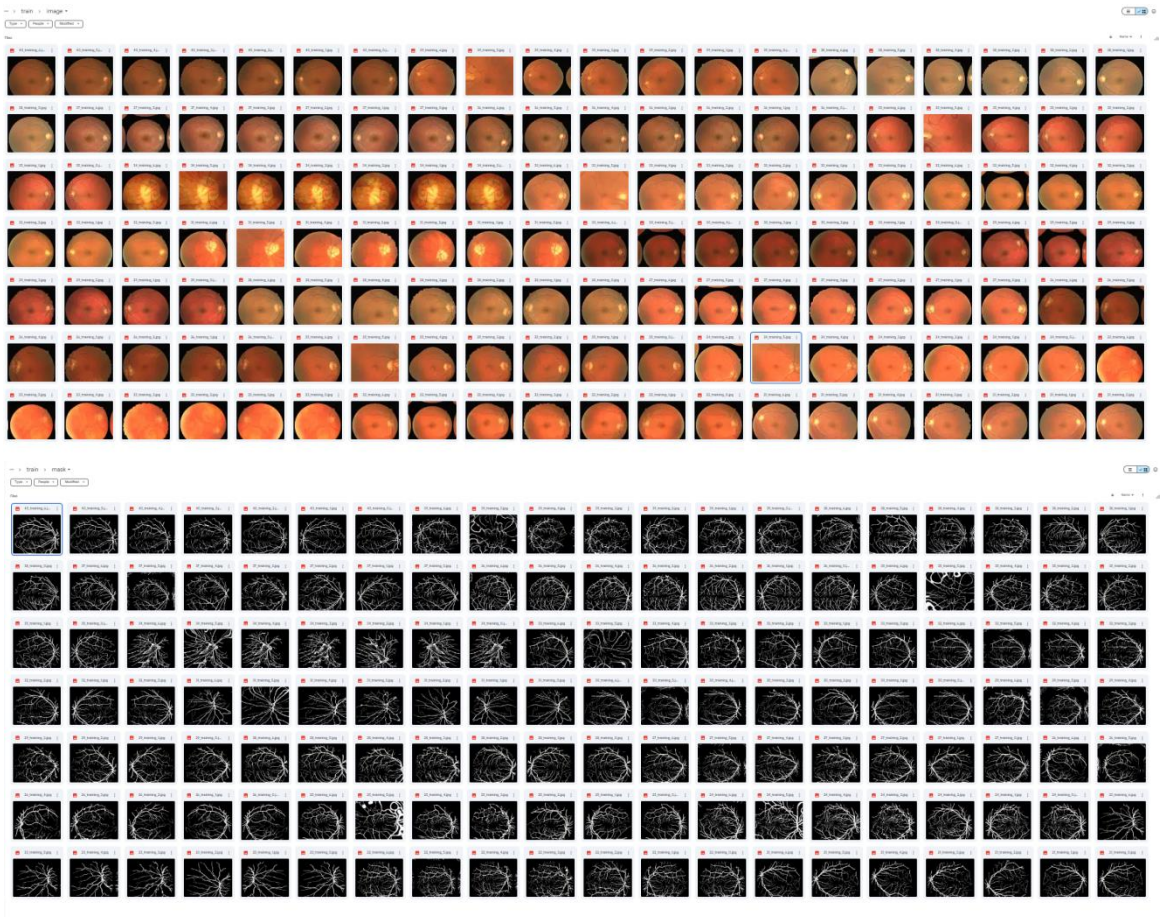


Fig 4.1 data augmentation of train Image and mask

4.2 Image preprocessing:

Image preprocessing is done to both augmented training and test images.

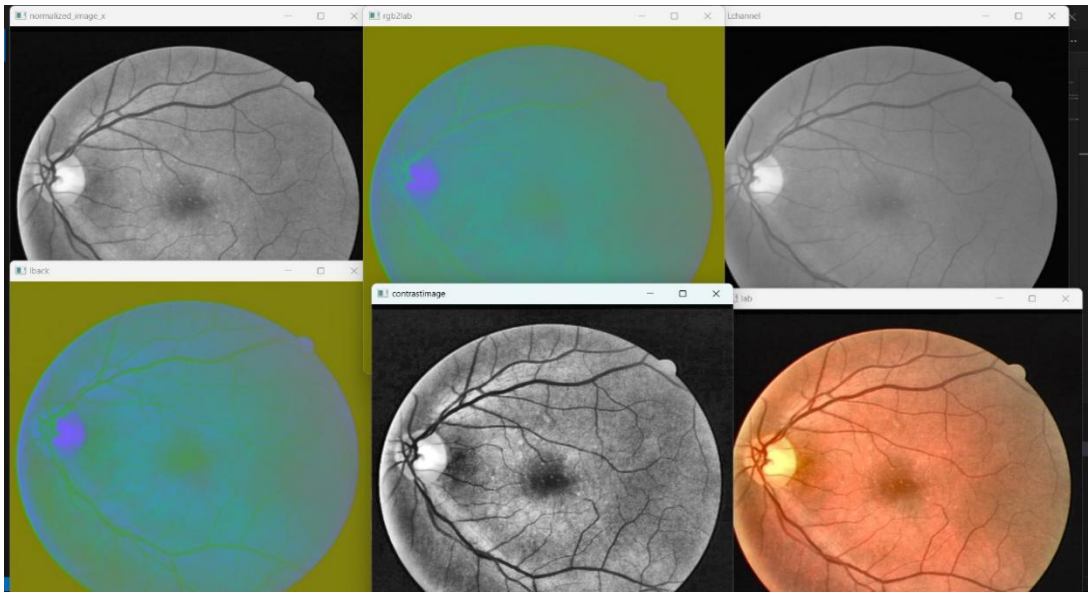


Fig 4.2.1 preprocessing of an single image

Fig 4.2.2 pre-processed images of train module

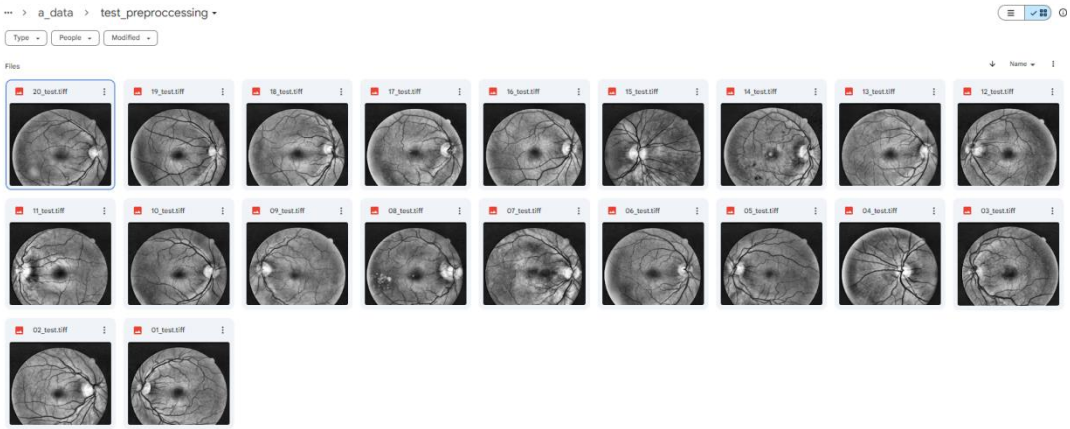


Fig 4.2.3 Pre-processed images of test module

4.3 U-Net Architecture:

The pre-processed image is sent into U-Net Architecture to extract blood vessel.

In U-Net the total number of params are 31,055,299 , including both Trainable and Non-trainable params. Here dice coefficient , IOU(Intersection over Union),Loss score of each epoch is found. Also Accuracy,F1,Jaccard and Precision score of each testing image is mentioned.

The total test accuracy is found to be 94%.

```
Accuracy: 0.94297
F1: 0.47791
Jaccard: 0.31461
Recall: 0.89684
Precision: 0.32837
```

```
Total params: 31,055,297
Trainable params: 31,043,521
Non-trainable params: 11,776
```

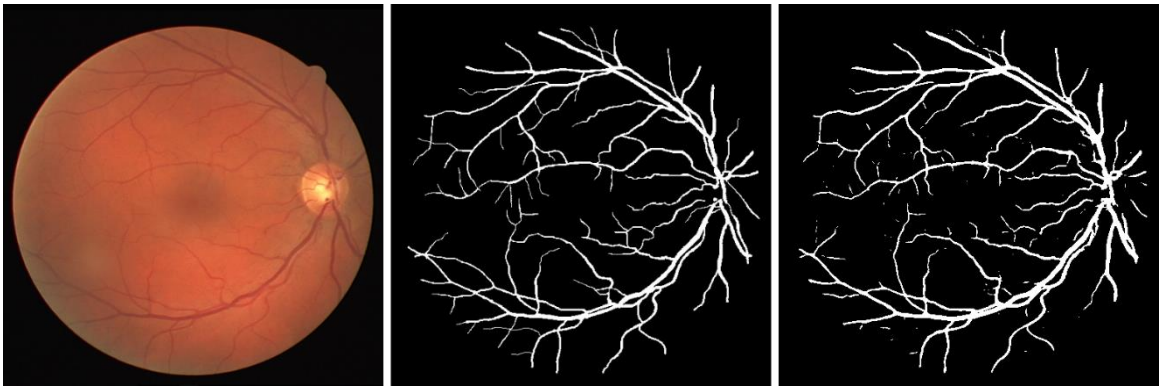
	A	B	C	D	E	F	G	H	I	J	K	L
1	epoch	dice_coef	iou	loss	lr	precision	recall	val_dice_coef	val_iou	val_loss	val_precision	val_recall
2	0	0.3780376315	0.238616541	0.6219624877	1.00E-04	0.5102250576	0.536582768	0.1379574835	0.07410833985	0.8620425463	0	0
3	1	0.554046154	0.3843589127	0.4459538758	1.00E-04	0.8312156796	0.4304736555	0.1060045734	0.05597634986	0.8939954042	0	0
4	2	0.6057375669	0.4353790283	0.3942624629	1.00E-04	0.8722323179	0.420853436	0.08210565895	0.04281457514	0.9178943634	0.04276244342	0.0006360654952
5	3	0.639667213	0.4710250199	0.3603328168	1.00E-04	0.8980386853	0.4152377546	0.06867617369	0.03557351604	0.9313238859	0.09515483677	0.0103362659
6	4	0.6658409238	0.4997493625	0.334159106	1.00E-04	0.914431572	0.4137726724	0.1813940257	0.1002230197	0.818606019	0.3359819055	0.06014819443
7	5	0.6878756881	0.5248548388	0.3121244013	1.00E-04	0.9271475673	0.4124450982	0.3860704899	0.2401761115	0.6139295101	0.5501759052	0.1664600372
8	6	0.7065890431	0.5468397141	0.2934109271	1.00E-04	0.9356087446	0.4119819105	0.4993616641	0.3334632218	0.5006383657	0.6583493352	0.2334594727
9	7	0.7238892913	0.5677111745	0.2761108279	1.00E-04	0.9444394708	0.4121139348	0.6412568092	0.4728165567	0.3587432504	0.9316070676	0.2633246481
10	8	0.7373198867	0.5843740106	0.2626800835	1.00E-04	0.9501190186	0.4115635157	0.6890238523	0.5260745287	0.3109761178	0.9503091574	0.303475976
11	9	0.7471348643	0.5967696309	0.2528650165	1.00E-04	0.9530514479	0.4107136726	0.7131353617	0.554445684	0.2868646383	0.9605694413	0.323666811
12	10	0.7568592429	0.6091985703	0.2431406528	1.00E-04	0.9570958614	0.4100022316	0.7173672915	0.5596551895	0.2826327682	0.9663696561	0.3124172688
13	11	0.7623607516	0.6163989902	0.2376393229	1.00E-04	0.9567441344	0.4100467861	0.7126136422	0.5540846586	0.2873863876	0.9764913917	0.2861623764
14	12	0.7704624534	0.6269891858	0.2295378298	1.00E-04	0.9612454772	0.4092773497	0.721249938	0.5644547343	0.278750062	0.9654180408	0.3061996698
15	13	0.7800082564	0.6397234201	0.2199918181	1.00E-04	0.9659203291	0.4096751213	0.7329343557	0.5787907839	0.2670656443	0.9580438137	0.3266369998
16	14	0.7862253785	0.6480676532	0.2137746066	1.00E-04	0.9684096575	0.4091753066	0.7395431399	0.5869640708	0.2604567707	0.9593272209	0.3367849588
17	15	0.7898028493	0.653013289	0.2101972252	1.00E-04	0.9689491391	0.4091564417	0.741561532	0.5895225406	0.2584383786	0.950805068	0.347387135
18	16	0.794219017	0.6590735912	0.2057807148	1.00E-04	0.9704807401	0.4092071056	0.7408982515	0.5887896419	0.2591017187	0.9632050395	0.3248896301
19	17	0.8009153605	0.6682875156	0.1990848184	1.00E-04	0.9735764265	0.4096126854	0.7315296531	0.5770048499	0.2684703469	0.9573252201	0.3208776712
20	18	0.8035151958	0.6719120741	0.1964848191	1.00E-04	0.9742278457	0.4093792737	0.7467265129	0.5960814953	0.2532734275	0.9566246271	0.3338769972
21	19	0.8080351949	0.6782528162	0.1919647157	1.00E-04	0.9760181308	0.4100270271	0.744756341	0.5936560631	0.2552436888	0.9639511108	0.3191407919
22	20	0.815181613	0.6883414388	0.1848183572	1.00E-04	0.9793774486	0.4101688266	0.7370634079	0.5840473771	0.2629365325	0.9667345881	0.3062570393
23	21	0.8174929023	0.6916572452	0.1825070977	1.00E-04	0.979824245	0.4106087387	0.7378460765	0.5850558281	0.2621539235	0.9679769278	0.3024964035
24	22	0.8177250624	0.6919531226	0.1822749376	1.00E-04	0.9792321324	0.4105207622	0.7419443727	0.5900133848	0.2580555677	0.9613358378	0.3164801598
25	23	0.8197066784	0.6948632002	0.1802934408	1.00E-04	0.9798471332	0.4102470577	0.7466204762	0.5959018469	0.2533795238	0.9459741712	0.3395248055
26	24	0.8292094469	0.7086026073	0.1707903147	1.00E-05	0.9845010042	0.4106274843	0.7570892572	0.6094423532	0.2429106981	0.964294374	0.3268859386
27	25	0.8355495334	0.7178662419	0.1644504666	1.00E-05	0.9878150821	0.4112090468	0.7560065985	0.6080735922	0.243993476	0.9664940238	0.3223534524
28	26	0.8386201859	0.7223979831	0.1613798887	1.00E-05	0.9893493056	0.4113298059	0.7553473711	0.6072335243	0.2446527034	0.9670934081	0.3205899298
29	27	0.8409429193	0.72584337	0.1590571553	1.00E-05	0.9904218316	0.4113613367	0.7549214363	0.6066899896	0.2450786084	0.9673460126	0.319850415
30	28	0.842851162	0.728685081	0.1571486443	1.00E-05	0.9912527204	0.4114063978	0.7545028329	0.6061556339	0.2454972565	0.9673911333	0.3190850317
31	29	0.8444845676	0.7311241031	0.1555153579	1.00E-05	0.9919210672	0.4113767743	0.7540963888	0.6056369543	0.2459035814	0.9674524665	0.3183608651
32	30	0.8454635739	0.73258847	0.1545363814	1.00E-06	0.9923727512	0.4106031656	0.7553402185	0.6072121659	0.2446596622	0.9661615491	0.3214530945
33	31	0.8458327055	0.733140409	0.1541671455	1.00E-06	0.9924094677	0.4113149941	0.755277276	0.6071310043	0.2447227538	0.966099143	0.3213011622
34	32	0.8460359573	0.7334448099	0.153963998	1.00E-06	0.9925054908	0.411308676	0.7552483082	0.607093215	0.2447517365	0.9660573602	0.32130602
35	33	0.8462266922	0.7337308526	0.1537732929	1.00E-06	0.9925845861	0.4113094211	0.755220294	0.6070568562	0.244779706	0.9660060406	0.3212631643
36	34	0.8464110494	0.7340073586	0.1535889804	1.00E-06	0.9926624298	0.4113019705	0.7551854849	0.607011795	0.2448145449	0.965993464	0.3212316632

Fig 4.3.1 Dice Coeff, iou and loss scores of each epoch

	Image	Acc	F1	Jaccard	Recall	Precision
0	01_test	0.9304275513	0.4400024564	0.2820533008	0.9388102725	0.2873355791
1	02_test	0.935710907	0.5159826531	0.3476931414	0.9350473613	0.356298588
2	03_test	0.9552574158	0.5236180496	0.3546629986	0.6898544521	0.4219414807
3	04_test	0.9441146851	0.5160864108	0.3477873742	0.9248253818	0.3579053466
4	05_test	0.9522323608	0.535327297	0.3654927793	0.8886288037	0.3830386066
5	06_test	0.9461479187	0.5110995671	0.3432731671	0.8820224719	0.3597932615
6	07_test	0.9448471069	0.4978117402	0.331391047	0.8967588537	0.344535795
7	08_test	0.9505004883	0.4982599954	0.3317884546	0.8404643882	0.3540888107
8	09_test	0.9540786743	0.4907352568	0.3251485593	0.8547008547	0.3441727985
9	10_test	0.9465904236	0.4798840967	0.3156891496	0.9273510409	0.3236944973
10	11_test	0.9378738403	0.459080643	0.297926456	0.9428376535	0.303406796
11	12_test	0.942150116	0.4752050386	0.3116517634	0.9084413866	0.321758283
12	13_test	0.939491272	0.4918631471	0.3261395981	0.9103521878	0.3369617697
13	14_test	0.9409141541	0.4564309528	0.2956984358	0.9413723219	0.3012461204
14	15_test	0.9440879822	0.4335021064	0.276733284	0.9248021108	0.2831036398
15	16_test	0.938709259	0.4695433986	0.3067995513	0.9009248701	0.3175120557
16	17_test	0.9475898743	0.4767889105	0.3130156508	0.8288097445	0.3346519833
17	18_test	0.9380073547	0.4410662081	0.2829281207	0.9117019764	0.2908991925
18	19_test	0.9316329956	0.4232477312	0.2684300759	0.9594397432	0.2715111478
19	20_test	0.9390640259	0.4226543299	0.2679528894	0.9295707472	0.2735054729

Get more storage...

Fig 4.3.2 Acc, F1, Jaccard and Precision score of images



Fundus Image Ground Truth Segmented Image

Fig 4.3.3 U-Net Output

Graphs of U-Net:

Graphs are plotted, according to their corresponding Scores.

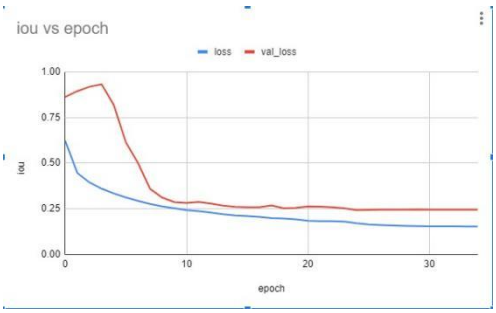


Fig 4.3.4 loss vs val_loss

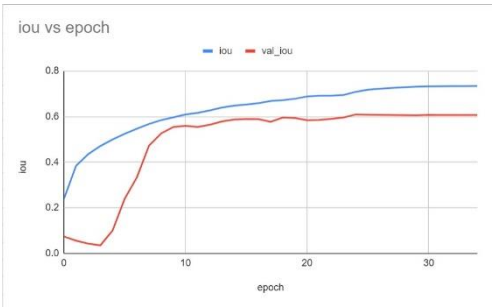


Fig 4.3.5 iou vs val_iou

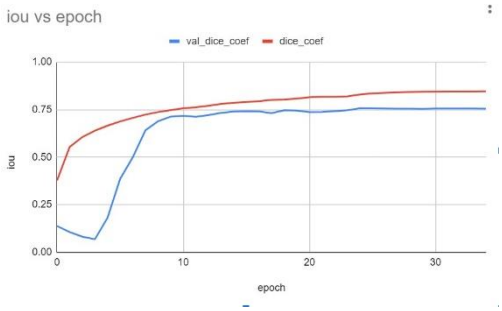


Fig 4.3.6 dice_coeff vs val_dice_coeff

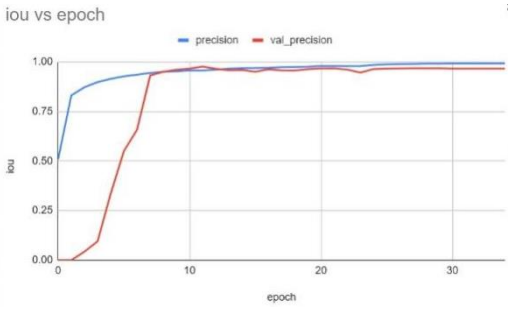


Fig 4.3.6 precision vs val_precision

5.4 Staircase Architecture:

The Preprocessed Image is sent into Staircase Architecture for extraction of blood vessel.

This model includes layer types, output shapes, and parameter counts. The breakdown of the key components:

Input Layer

- Input shape: (512, 512, 3) for RGB images.

Convolution Blocks

- Multiple convolution blocks with Conv2D layers followed by BatchNormalization and ReLU activation.
- Intermediate convolutional layers have 64, 32, 64, and 128 filters, respectively.
- Various Conv2DTranspose layers for upsampling.

Concatenation and Pooling Layers

- Concatenation layers combine features from different parts of the network.
- MaxPooling2D layers for downsampling.

Output Layer

- Final Conv2D layer with a sigmoid activation function for binary classification (1 channel output).

Total Parameters:

Total trainable parameters:	1,724,258 (6.58 MB)
Total non-trainable parameters:	3,712 (14.50 KB)
Grand total parameters	1,727,970 (6.59 MB)

This architecture is designed for blood vessel segmentation, utilizing a hierarchical approach with multiple convolutional blocks, upsampling, and downsampling layers to capture features at different scales and perform accurate segmentation

- In this staircase-net CNN there are total of 36 convolutional layers of filter size 3*3

with activation function as ReLu .

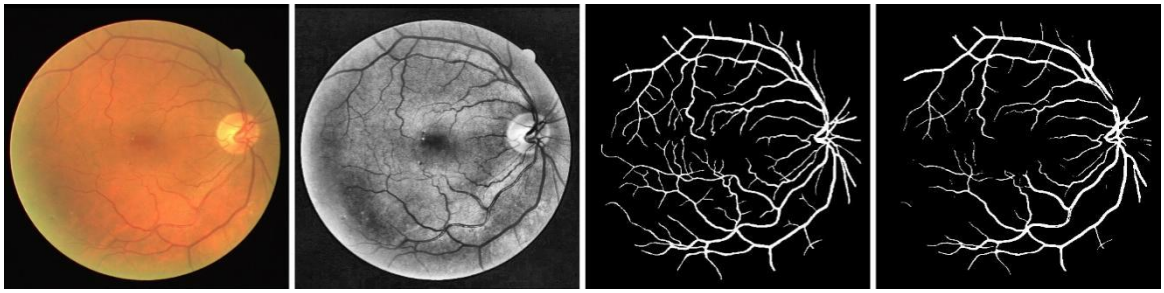
. After concatenation feature maps are followed by fully connected layers with filter size as 1*1 and activation function as sigmoid

Relu – rectified linear unit**:**

- Introduces nonlinearity between the layer.
- F(x) = max(0,x)
- For negative values the F(x) is zero.

Sigmoid activation function maps any value between 1 and 0.

- Can be interpreted as probability.
- F(x) = 1/(1+e^(-x))



Fundus Image Preprocessed Image Ground Truth Segmented Image

The model is runned upto 41st epoch, The Specificity , Sensitivity and AUC(Area Under Curve) score after each epoch is read.

Here is the example of last epoch with its corresponding scores;

epoch	accuracy	auc	binary_accuracy	binary_crossentropy	loss	lr	sensitivity	specificity	val_accuracy	val_auc	val_binary_accuracy	val_binary_crossentr	val_loss	val_sensitivity	val_specificity
0	0.7464396954	0.7349006534	0.7464396954	0.4919729531	0.4919729531	0.001	0.712747097	0.6835471272	0.7640046862	0.4685124325	0.7640046862	0.3658138385	0.3658138385	0	0.990137279
1	0.8027357459	0.7744178772	0.8027357459	0.315864265	0.315864265	0.001	0.6859166622	0.9616727829	0.7640046862	0.5313905478	0.7640046862	0.3007977605	0.3007977605	0	0.9967781305
2	0.8063395023	0.7934837937	0.8063395023	0.2267058939	0.2267058939	0.001	0.6636819575	0.9698274732	0.7640046862	0.616777122	0.7640046862	0.3139615059	0.3139615059	0	0.9982800484
3	0.8076337576	0.8078109622	0.8076337576	0.1769174188	0.1769174188	0.001	0.6556368472	0.973007977	0.7640046862	0.5571444631	0.7640046862	0.3513032794	0.3513032794	0	0.9991329312
4	0.8082566857	0.8211578727	0.8082566857	0.1510987431	0.1510987431	0.001	0.6528649926	0.9748131633	0.7648345828	0.74948138	0.7648345828	0.2909349799	0.2909349799	0.0174642764	0.9987300634
5	0.8087728024	0.8298338652	0.8087728024	0.1364006251	0.1364006251	0.001	0.6534455419	0.9757144451	0.7707479596	0.795635879	0.7707479596	0.2183518708	0.2183518708	0.1496060938	0.9973184468
6	0.8094085455	0.8382143378	0.8094085455	0.1270715445	0.1270715445	0.001	0.6567239165	0.9764336944	0.7794235349	0.836520195	0.7794235349	0.1564783752	0.1564783752	0.3598716523	0.9941766858
7	0.8096676469	0.8444613218	0.8096676469	0.1214652359	0.1214652359	0.001	0.6583240628	0.976805687	0.7825139761	0.8406580091	0.7825139761	0.136599645	0.136599645	0.450097084	0.9918715358
8	0.8098833561	0.8443084955	0.8098833561	0.1182847843	0.1182847843	0.001	0.6566167612	0.977093637	0.7845861316	0.8211408854	0.7845861316	0.1232584864	0.1232584864	0.5102260113	0.9898027182
9	0.8102757335	0.8481468658	0.8102757335	0.1150673479	0.1150673479	0.001	0.6608152986	0.9773783088	0.7862899899	0.830694437	0.7862899899	0.1125338674	0.1125338674	0.5639688373	0.9881539345
10	0.8106618524	0.8543915153	0.8106618524	0.1120778769	0.1120778769	0.001	0.6656461954	0.9776192904	0.7858805849	0.815408515	0.7858805849	0.1153248996	0.1153248996	0.5512894392	0.9885705113
11	0.8110601306	0.8516932726	0.8110601306	0.1095129102	0.1095129102	0.001	0.6683518291	0.9777824283	0.7865961194	0.8082592487	0.7865961194	0.1092739105	0.1092739105	0.5755111575	0.9876741171
12	0.8101327419	0.8461934071	0.8101327419	0.1130889803	0.1130889803	0.001	0.6591946483	0.9774592519	0.7850206494	0.8231292367	0.7850206494	0.1261091828	0.1261091828	0.610720396	0.9795807004
13	0.809623003	0.8425774574	0.809623003	0.1150526702	0.1150526702	0.001	0.6465359073	0.9776152372	0.7892335873	0.8028758764	0.7892335873	0.1060442105	0.1060442105	0.6633859673	0.9808594584
14	0.8106569648	0.8589704633	0.8106569648	0.1096656024	0.1096656024	0.001	0.6612765789	0.9778923988	0.7866994739	0.8267195563	0.7866994739	0.1065131798	0.1065131798	0.5929964019	0.9865727425
15	0.8110947609	0.8616310954	0.8110947609	0.1074248031	0.1074248031	0.001	0.667968154	0.977951467	0.7865005732	0.7552668724	0.7865005732	0.1101318747	0.1101318747	0.574170351	0.9875187874
16	0.8114362955	0.8624668717	0.8114362955	0.105498746	0.105498746	0.001	0.6712802052	0.9780868292	0.7871347666	0.8269556165	0.7871347666	0.1033751741	0.1033751741	0.6149947047	0.9852660894
17	0.8116565347	0.8634480238	0.8116565347	0.1040183455	0.1040183455	0.001	0.6748298407	0.9781195521	0.7871515155	0.8183488846	0.7871515155	0.1040078253	0.1040078253	0.6032655835	0.9858776331
18	0.8116731048	0.8592435718	0.8116731048	0.1040385589	0.1040385589	0.001	0.6752266288	0.9780381322	0.7858396769	0.8204022646	0.7858396769	0.1126141176	0.1126141176	0.5769056376	0.9862203598
19	0.8116803169	0.8586646318	0.8116803169	0.1041862369	0.1041862369	0.001	0.6730061769	0.9782201648	0.7876625061	0.797529282	0.7876625061	0.1011671573	0.1011671573	0.6357811689	0.9842004776
20	0.811932385	0.8583969474	0.811932385	0.1023657024	0.1023657024	0.001	0.6785393357	0.976100121	0.7877893448	0.8243326545	0.7877893448	0.1007850766	0.1007850766	0.6399296522	0.9842303395
21	0.812310636	0.8602010608	0.812310636	0.1007397994	0.1007397994	0.001	0.6823479533	0.9782986178	0.7879108191	0.8200159669	0.7879108191	0.1001410708	0.1001410708	0.6500096917	0.9832196236
22	0.8125941753	0.8593539	0.8125941753	0.0992327109	0.0992327109	0.001	0.68515414	0.9783405662	0.787733674	0.8016311526	0.787733674	0.1041203737	0.1041203737	0.6310747266	0.9850634336
23	0.8128752708	0.8593304753	0.8128752708	0.09797609597	0.09797609597	0.001	0.689065516	0.9783996344	0.7876199484	0.8090279698	0.7876199484	0.1027291864	0.1027291864	0.6369352083	0.9837821722
24	0.8128366647	0.8584564328	0.8128366647	0.09793130308	0.09793130308	0.001	0.6889000535	0.9783143401	0.7881835699	0.8163149357	0.7881835699	0.09846127033	0.09846127033	0.6647312045	0.9828317761
25	0.8126826882	0.8557316661	0.8126826882	0.09822124988	0.09822124988	0.001	0.6872684956	0.9783164859	0.7869547009	0.7919852138	0.7869547009	0.1052282006	0.1052282006	0.6699034572	0.9803847075
26	0.8102716804	0.8438839912	0.8102716804	0.1098740399	0.1098740399	0.001	0.6547892094	0.9780247211	0.6761656079	0.790346086	0.6761656079	0.3871145844	0.3871145844	0.7777204514	0.8192900419
27	0.8102300167	0.8420086503	0.8102300167	0.1108313426	0.1108313426	0.001	0.6618404984	0.9773436189	0.7862861753	0.796557188	0.7862861753	0.1090884581	0.1090884581	0.7200220227	0.9746311903
28	0.8118098378	0.8514847691	0.8118098378	0.102552563	0.102552563	0.001	0.6757036448	0.9783040881	0.788146019	0.8000448346	0.788146019	0.09858343005	0.09858343005	0.6655389071	0.9826682606
29	0.8126407266	0.8559697666	0.8126407266	0.09838434309	0.09838434309	0.001	0.6873606443	0.9783509374	0.7876052856	0.812086761	0.7876052856	0.1030544266	0.1030544266	0.6218503714	0.9859101176
30	0.8134373426	0.8570916057	0.8134373426	0.09448850155	0.09448850155	0.000100000005	0.6965128522	0.9784110188	0.7885103226	0.8360769153	0.7885103226	0.09419457614	0.09419457614	0.6743947268	0.9823468328
31	0.8137927651	0.8598909974	0.8137927651	0.09282676551	0.09282676551	0.000100000005	0.7029688954	0.9784903526	0.7885091901	0.8344308734	0.7885091901	0.0943693301	0.0943693301	0.6766447425	0.9821730852
32	0.8139324784	0.860260427	0.8139324784	0.0920234099	0.0920234099	0.000100000005	0.7050299644	0.9785088897	0.7885090113	0.8341879249	0.7885090113	0.09444023069	0.09444023069	0.6803894043	0.9818769693
33	0.8140476346	0.8607651591	0.8140476346	0.09129252285	0.09129252285	0.000100000005	0.706751883	0.9785020947	0.7884929776	0.8337620497	0.7884929776	0.0945831943	0.0945831943	0.6829187604	0.9816611409
34	0.8141497374	0.8612607989	0.8141497374	0.09059020132	0.09059020132	0.000100000005	0.7084051967	0.9784890413	0.7884734869	0.8332557982	0.7884734869	0.09476012737	0.09476012737	0.6852527261	0.9814693332
35	0.8142420053	0.8618611693	0.8142420053	0.08990007639	0.08990007639	0.000100000005	0.7100486159	0.9784745574	0.788449645	0.8329032063	0.788449645	0.09497297555	0.09497297555	0.6868451502	0.9813450575
36	0.8143232465	0.8628573418	0.8143232465	0.08926061541	0.08926061541	1.00E-05	0.7126708959	0.9783627391	0.7884025574	0.8344977498	0.7884025574	0.09502457082	0.09502457082	0.6891268492	0.9809386134
37	0.8143590689	0.8627634645	0.8143590689	0.08908353001	0.08908353001	1.00E-05	0.7117722631	0.9784908691	0.788374722	0.8351172805	0.788374722	0.09499411285	0.09499411285	0.6920412779	0.9806293249
38	0.8143730164	0.8627979159	0.8143730164	0.08897489309	0.08897489309	1.00E-05	0.7120248079	0.9784841537	0.7883583307	0.8353587985	0.7883583307	0.09499437362	0.09499437362	0.6941091418	0.9804368827

Fig 5.4.2 specificity, sensity and AUC value over each epoch

Epoch 40: val_loss did not improve from 0.09419
70/70 [==============================] - 99s 1s/step - loss: 0.0889 - accuracy: 0.8144 - binary_accuracy: 0.8144 - binary_crossentropy: 0.0889 - sensitivity: 0.7123 - specificity: 0.9785 - auc: 0.8628 - val_loss: 0.0950 - val_accuracy: 0.7884 - val_binary_accuracy: 0.7884 - val_binary_crossentropy: 0.0950 - val_sensitivity: 0.6953 - val_specificity: 0.9803 - val_auc: 0.8354 - lr: 1.0000e-05
Epoch 41/60
70/70 [==============================] - ETA: 0s - loss: 0.0888 - accuracy: 0.8144 - binary_accuracy: 0.8144 - binary_crossentropy: 0.0888 - sensitivity: 0.7126 - specificity: 0.9785 - auc: 0.8629

Epoch 41: val_loss did not improve from 0.09419

Epoch 41: ReduceLROnPlateau reducing learning rate to 1.0000000656873453e-06.
70/70 [==============================] - 99s 1s/step - loss: 0.0888 - accuracy: 0.8144 - binary_accuracy: 0.8144 - binary_crossentropy: 0.0888 - sensitivity: 0.7126 - specificity: 0.9785 - auc: 0.8629 - val_loss: 0.0950 - val_accuracy: 0.7883 - val_binary_accuracy: 0.7883 - val_binary_crossentropy: 0.0950 - val_sensitivity: 0.6958 - val_specificity: 0.9803 - val_auc: 0.8354 - lr: 1.0000e-05

This model got 94% testing accuracy , when trained model is applied to each testing image , their F1 , Jaccard , Recall and Precision score for each corresponding image is noted.

```
Accuracy: 0.94893
F1: 0.50441
Jaccard: 0.33779
Recall: 0.89300
Precision: 0.35295
```

	Image	Acc	F1	Jaccard	Recall	Precision
0	01_test	0.9365539551	0.4608052908	0.2993807658	0.9312106918	0.3061514603
1	02_test	0.9403152466	0.5352857313	0.3654540293	0.9379619028	0.3745064627
2	03_test	0.947177887	0.5299249754	0.3604747829	0.8352953767	0.3880574753
3	04_test	0.9509620667	0.5448751992	0.3744525547	0.9109743104	0.3886756238
4	05_test	0.955165863	0.546863554	0.3763332449	0.8737218184	0.397979798
5	06_test	0.9533843994	0.5392156863	0.3691275168	0.8546497729	0.3938525945
6	07_test	0.953289032	0.5289478746	0.3595711297	0.8603428857	0.3818595868
7	08_test	0.957572937	0.5317052632	0.3621243404	0.823636838	0.3925640388
8	09_test	0.9575881958	0.5134780326	0.3454224316	0.8645741232	0.3651811278
9	10_test	0.9539527893	0.5095282597	0.3418570416	0.9002153625	0.3553213193
10	11_test	0.9458694458	0.4885011895	0.3231899265	0.924420191	0.3319615912
11	12_test	0.9501953125	0.5092467298	0.3416036309	0.8962688542	0.3556652315
12	13_test	0.9455184937	0.5167817025	0.3484191797	0.9056089173	0.3615490224
13	14_test	0.9491539001	0.4898380985	0.3243613139	0.9263173133	0.3329517665
14	15_test	0.9521560669	0.4699070161	0.3071101044	0.9167216359	0.3159240737
15	16_test	0.9449577332	0.4942693912	0.3282588454	0.8933231978	0.3416513228
16	17_test	0.9519424438	0.5004758128	0.3337564123	0.8355620283	0.3572196751
17	18_test	0.9432067871	0.4626822578	0.300967227	0.9114176027	0.3100362757
18	19_test	0.9412651062	0.4586336627	0.2975500707	0.9515611322	0.3021262797
19	20_test	0.9483795166	0.4572437029	0.2963810316	0.906200318	0.3057611844

Fig 4.4.3 Accurecy, F1, Jaccard, Recall, Presion scores of corresponding testing images

Graphs of Staircase-Net:

This is graph plotted between training Loss(in blue colour) and Validation Loss(in orange

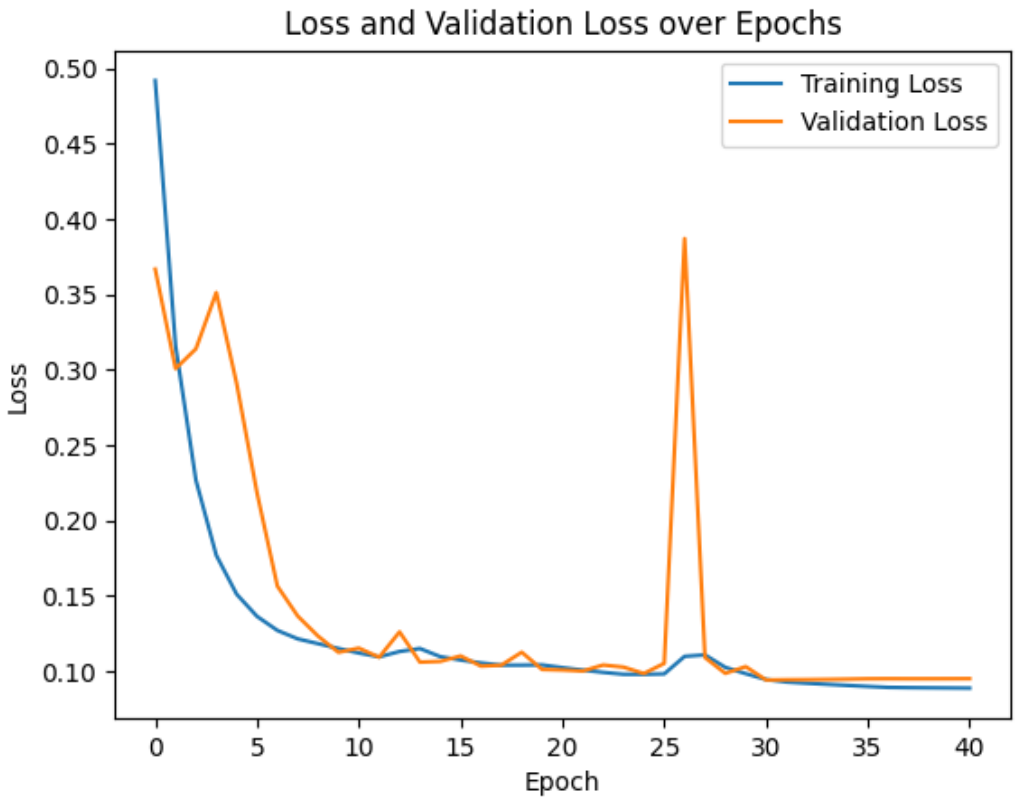


Fig 4.4.4 loss vs val_loss

colour) over Epochs.

4.5 Analysis

Architecture	F1	JACCARD	RECALL	PRECISION
U-Net	0.4779	0.3146	0.8968	0.3283
Staircase-Net	0.5044	0.3377	0.8929	0.3529

Chapter 5

Conclusion & Feature plans

Our paper introduces an original method for segmenting retinal blood vessels, focusing on scaling photographs to prevent image augmentation effects. Although our model experiences reduced sensitivity due to a high percentage of negative pixels in segmented images, the trade-off is justified by its strong performance across other metrics. Notably, the model's swift execution time, averaging 0.3632 seconds per picture, makes it suitable for clinical applications. Our network employs a cascaded design with up-sampling and down-sampling layers, prioritizing blood vessel localization and feature extraction. While sensitivity may be limited by negative pixels, we can enhance performance by augmenting data or emphasizing the balanced loss function for vessel features. Additionally, we explored the U-Net architecture for blood vessel segmentation and found it effective, enabling the diagnosis of hypertensive retinopathy, which can greatly benefit ophthalmologists in early detection and treatment planning.

Feature plans:

Segmented retinal images can be utilized for advanced preprocessing methods and subsequent classification of arteries and veins. This classification enables the calculation of the Arteriole to Venule Ratio (AVR), which is valuable for detecting various diseases.

The AVR values derived from this classification can be particularly informative for assessing the severity of hypertensive retinopathy.

Additionally, segmented images can aid in identifying instances of arteries and veins obstruction, and they can also be applied in the context of diabetic retinopathy analysis.

REFERENCES

" Staircase-Net: a deep learning-based architecture for retinal blood vessel segmentation by Varun. (2022) – https://link.springer.com/article/10.1007/s12046-022-01936-w

“Impact of Image Enhancement Technique on CNN Model for Retinal Blood Vessels Segmentation " by Ahmed j . afifi, Ali Shah, Soomoro, Gulsher Ali Baloch. (2019) - https://ieeexplore.ieee.org/abstract/document/8886493

" Encoder Enhanced Atrous (EEA) Unet architecture for Retinal Blood vessel segmentation " by Indumathi et al. (2016) - https://www.sciencedirect.com/science/article/pii/S1389041721000085

"Retinal Blood Vessel Segmentation Using Line Operators and Support Vector Classification" by E. Ricci et al. (2007) - https://ieeexplore.ieee.org/abstract/document/4336179

" Robust retinal blood vessel segmentation using convolutional neural network and support vector machine" by Bala subramanian et al. (2019) https://link.springer.com/article/10.1007/s12652-019-01559-w

" Fast and efficient retinal blood vessel segmentation method based on deep learning network " by Henda et al. (2018) - https://ieeexplore.ieee.org/document/8545136

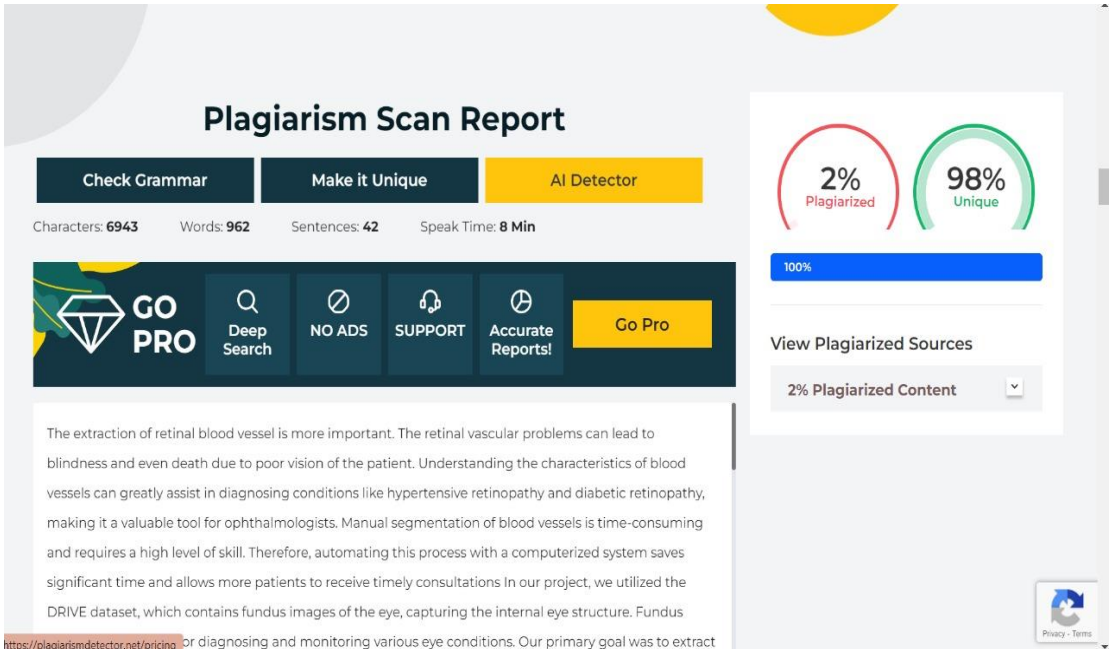
" Sine-Net: A fully convolutional deep learning architecture for retinal blood vessel segmentation" by Atli et al. (2021) – https://www.sciencedirect.com/science/article/pii/S221509862030330X

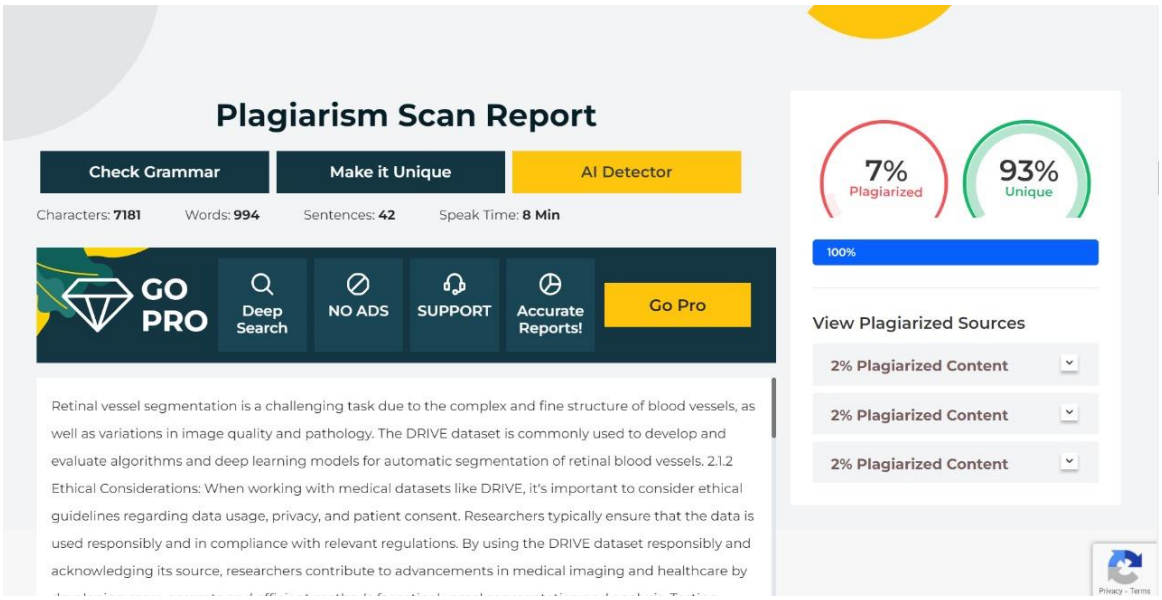
“Ridge-based vessel segmentation in color images of the retina” by Staal et al https://ieeexplore.ieee.org/abstract/document/1282003

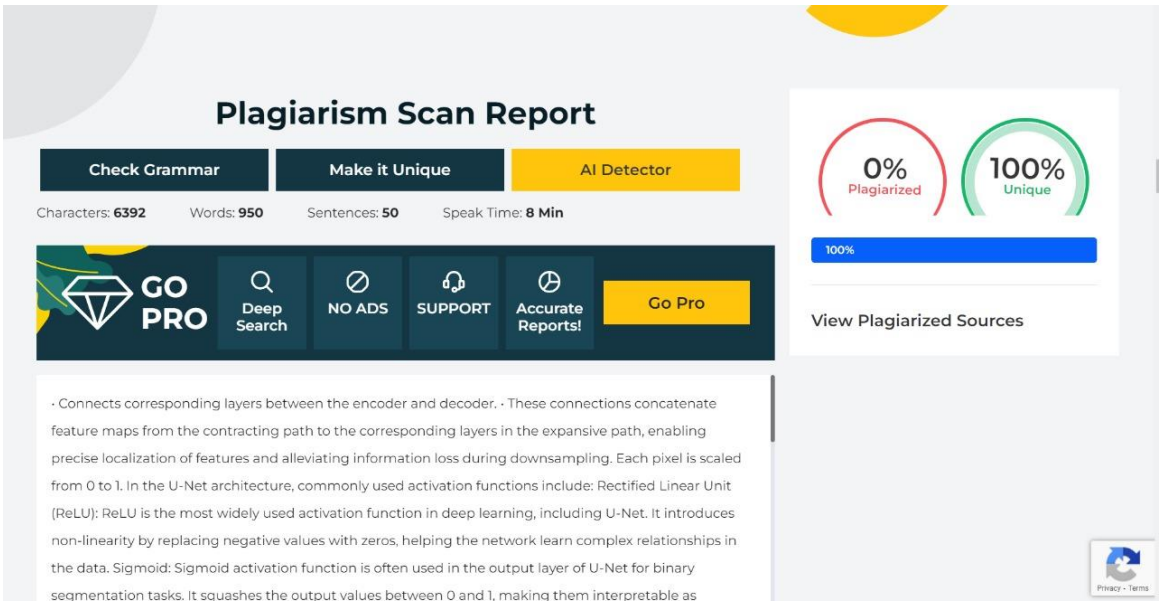
"Retinal Vessel Segmentation Using Deep Learning Networks with Maxo "Retinal Vessel Segmentation Using Deep Learning Networks with Max out and Residual Units" by Narasimhan et al. (2018) - UT and Residual Units" by Narasimhan et al. (2018) –

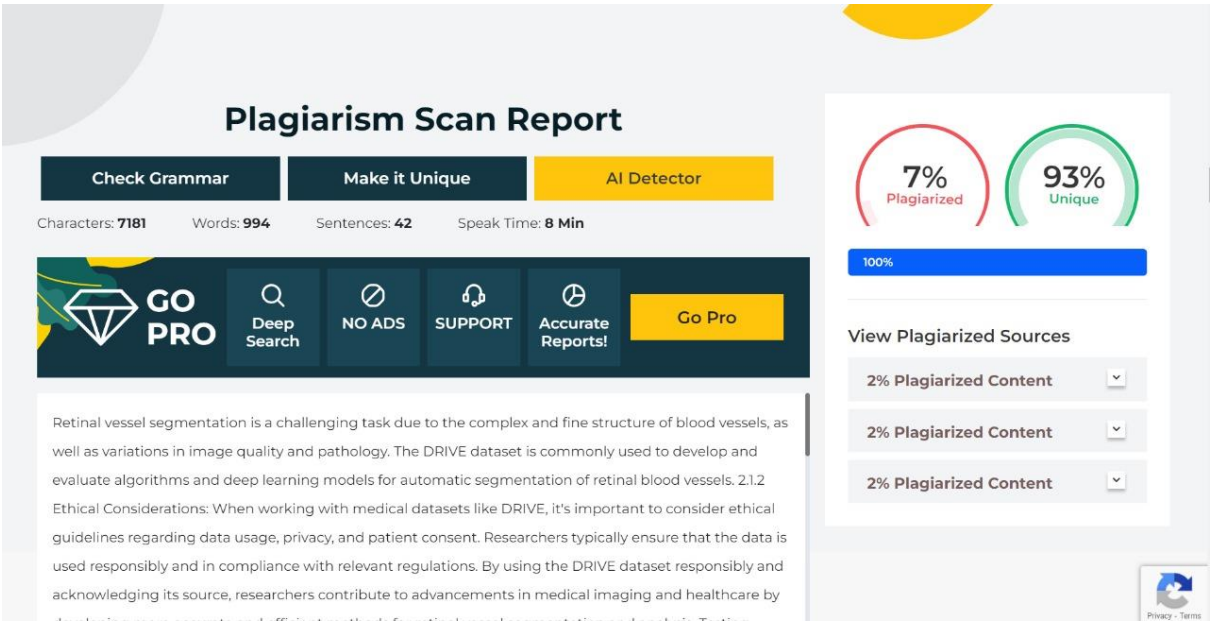
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6249017/

PLAGIARISM REPORT






Plagiarism Scan Report
Check Grammar
Make it Unique
AI Detector
Characters: 6392
Words: 950
Sentences: 50
Speak Time: 8 Min
GO PRO
Deep Search
NO ADS
SUPPORT
Accurate Reports!
Go Pro
· Connects corresponding layers between the encoder and decoder. · These connections concatenate feature maps from the contracting path to the corresponding layers in the expansive path, enabling precise localization of features and alleviating information loss during downsampling. Each pixel is scaled from 0 to 1. In the U-Net architecture, commonly used activation functions include: Rectified Linear Unit (ReLU): ReLU is the most widely used activation function in deep learning, including U-Net. It introduces non-linearity by replacing negative values with zeros, helping the network learn complex relationships in the data. Sigmoid: Sigmoid activation function is often used in the output layer of U-Net for binary segmentation tasks. It squashes the output values between 0 and 1, making them interpretable as
0%
Plagiarized
100%
Unique
100%
View Plagiarized Sources


Plagiarism Scan Report
Check Grammar
Make it Unique
AI Detector
Characters: 7181
Words: 994
Sentences: 42
Speak Time: 8 Min
GO PRO
Deep Search
NO ADS
SUPPORT
Accurate Reports!
Go Pro
Retinal vessel segmentation is a challenging task due to the complex and fine structure of blood vessels, as well as variations in image quality and pathology. The DRIVE dataset is commonly used to develop and evaluate algorithms and deep learning models for automatic segmentation of retinal blood vessels. 2.1.2 Ethical Considerations: When working with medical datasets like DRIVE, it's important to consider ethical guidelines regarding data usage, privacy, and patient consent. Researchers typically ensure that the data is used responsibly and in compliance with relevant regulations. By using the DRIVE dataset responsibly and acknowledging its source, researchers contribute to advancements in medical imaging and healthcare by
7%
Plagiarized
93%
Unique
100%
View Plagiarized Sources
2% Plagiarized Content
2% Plagiarized Content
2% Plagiarized Content