A Project Report
on

Retina Gray Scale Image Perception using Deep Learning

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

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

19WH1A0562	**Arisham Dhanuhya**
19WH1A05A3	**Penthala Harshini Reddy**
19WH1A05A7	**Nagaram Anjali**

under the esteemed guidance of

Ms. M. Shanmuga Sundari
Assistant Professor

Department of Computer Science and Engineering
BVRIT HYDERABAD
College of Engineering for Women
(NBA Accredited – EEE, ECE, CSE and IT)
(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)
Bachupally, Hyderabad – 500090

June, 2023

DECLARATION

We hereby declare that the work presented in this project entitled **"Retina Gray Scale Image Perception using Deep Learning"** submitted towards completion of Project Work in IV year of B.Tech., CSE at 'BVRIT HYDERABAD College of Engineering For Women', Hyderabad is an authentic record of our original work carried out under the guidance of Ms. M. Shanmuga Sundari, Assistant Professor, Department of CSE.

A. Dhanuhya
(19WH1A0562)

P. Harshini Reddy
(19WH1A05A3)

N. Anjali
(19WH1A05A7)

BVRIT HYDERABAD
College of Engineering for Women
(NBA Accredited – EEE, ECE, CSE and IT)
(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)
Bachupally, Hyderabad – 500090

Department of Computer Science and Engineering

Certificate

This is to certify that the Project Work report on "**Retina Gray Scale Image Perception Using Deep Learning**" is a bonafide work carried out by Arisham Dhanuhya (19WH1A0562), Penthala Harshini Reddy (19WH1A05A3), Nagaram Anjali (19WH1A05A7) in the partial fulfillment for the award of B.Tech. degree in **Computer Science and Engineering, BVRIT HYDERABAD College of Engineering for Women, Bachupally, Hyderabad**, affiliated to Jawaharlal Nehru Technological University Hyderabad, Hyderabad under my guidance and supervision.

The results embodied in the project work have not been submitted to any other University or Institute for the award of any degree or diploma.

Head of the Department
Dr. E. Venkateswara Reddy
Professor and HoD,
Department of CSE

Guide
Ms. M. Shanmuga Sundari
Assistant Professor

External Examiner

Acknowledgements

We would like to express our sincere thanks to **Dr. K V N Sunitha, Principal**, **BVRIT HYDERABAD College of Engineering for Women**, for providing the working facilities in the college.

Our sincere thanks and gratitude to our **Dr. E. Venkateswara Reddy, Professor & HOD**, Department of CSE, **BVRIT HYDERABAD College of Engineering for Women** for all the timely support and valuable suggestions during the period of our project.

We are extremely thankful and indebted to our internal guide, **Ms. M. Shanmuga Sundari, Assistant Professor**, Department of CSE**, BVRIT HYDERABAD College of Engineering for Women** for her constant guidance, encouragement and moral support throughout the project.

Finally, we would also like to thank our Project Coordinator, all the faculty and staff of **CSE** Department who helped us directly or indirectly, parents and friends for their cooperation in completing the project work.

Ms. A. Dhanuhya

(19WH1A0567)

Ms. P. Harshini Reddy

(19WH1A05A3)

Ms. N. Anjali

(19WH1A05A7)

Contents

S.No.	Topic	Page No.
	Abstract	i
	List of Figures	ii
	List of Tables	iii
1.	INTRODUCTION	1
	1.1 Objectives	1
	1.2 Methodology	1
	1.2.1 Dataset	1
	1.2.2 The proposed CNN model	2
	1.3 Organization of Project	4
2.	LITERATURE REVIEW	5
3.	THEORETICAL ANALYSIS OF PROPOSED PROJECT	15
	3.1 Requirements Gathering	15
	3.1.1 Software requirements	15
	3.1.2 Hardware requirements	15
	3.2 Technologies Description	15
4.	DESIGN	20
	4.1 Introduction	20
	4.2 Architecture Diagram	21
	4.3 Algorithm	
	4.3.1 U-Net	22
	4.3.2 SegNet	25
5.	IMPLEMENTATION	29
	5.1 Coding	29
	5.2 Input	29
	5.3 Evaluation Metrics	29
	5.4 Results	33
6.	Conclusion and Future Scope	37
7.	References	38
8.	Appendix	41

ABSTRACT

Millions of people in the world are affected by ocular fundus diseases such as diabetic retinopathy, age-related macular degeneration, glaucoma, retinal detachment, and fundus tumors, without accurate diagnoses and timely appropriate treatment, these fundus diseases can lead to irreversible blurred vision or even blindness. Manual vessel segmentation is always difficult in a retinal image because of the varying thickness and width of retinal blood vessels. Retinal blood vessel segmentation has become mandatory for any type of retinal disease.

Segmentation of retinal blood vessels is considered an effective technique for diagnosing these diseases. We are aiming to solve the problems of serious segmentation errors of eyes by using network architectures, like semantic pixel-wise segmentation (SegNet) and U-Net. The accurate segmentation of retinal blood vessels in the fundus is of great practical significance to help doctors diagnose fundus diseases.

List of Figures

S.No	Description	Page No.
1	Dataset Files Architecture	2
2	Sample Images	2
3	Human Eye Image	20
4	Architecture Design	21
5	Architecture of U-Net	22
6	Max Pooling	23
7	Architecture of SegNet	26
8	Training Images	29
9	ROC Curve of SegNet	33
10	ROC Curve of SegNet	34
11	Outputs of SegNet Model	36
12	Outputs of U-Net Model	36

List of Tables

S.No	Description	Page No.
1	IOU Scores	34
2	Precision Score	35

1. INTRODUCTION

As a vital part of the eye, the retina plays a significant role in the human body. Humans rely on their eyes for 80% of the ways they receive information from the outside world. The quality of human eyesight is significantly influenced by the retina. Clinical medicine currently relies on ophthalmological tools and medical imaging technology to diagnose and treat individuals with retinal injury or detached blindness.

The retina produced by the current imaging technology has significant flaws, which makes it difficult for medical treatment and performing attentive observation and correct judgment. Therefore, some CNN Algorithms can be used. These algorithms will help the doctors by providing a grayscale image of the retina image in which has a clear vision of optic nerves, and blood vessels of the eye which helps in better understanding of the eye for the doctors. Then from the image doctors can treat the patients with better analysis.

In Order to early detection of Eye Disease, Convolutional Neural Network (CNN) approach is proposed using the images (retinal) as input to it. CNNs were applied individually to solve certain problems. CNN is a method of deep learning which has a magnificent record of advantages for image analysis and interpretation. CNN is considered one of the important methods to fix an automatic analysis of retinal images.

1.1. Objective: The task of identifying blood vessels and nerves in the retina images is time consuming when followed traditional methods. The objective of this project is identification of nerves and blood vessels accurately that will help doctors in detecting the diseases in early stage and lead to save lives of people. The features are extracted and identified using various fully connected deep learning techniques. Using a publically available dataset of retinal images, we would train a convolutional neural network model to identify nerves and blood vessels.

1.2. Methodology:

1.2.1. Dataset: Proper dataset is required for the training and the testing phase. The dataset for the experiment is downloaded from the Kaggle which is DRIVE(Digital Retinal Images for Vessel Extraction) contains 40 different retina images. The images are split into 70% training set, 30% testing set. Each having the

images classified into – 1st manual, images, mask as you can see in figure 1. The images are in TIF format.

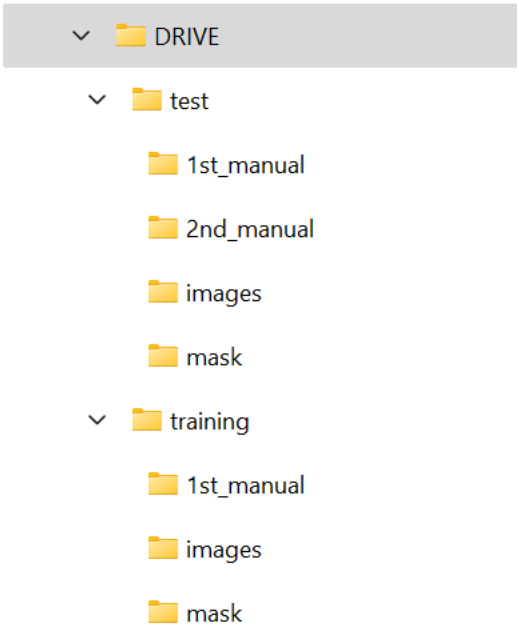

Fig 1: Dataset Files Architecture

Sample Images in figure 2 are as follows:

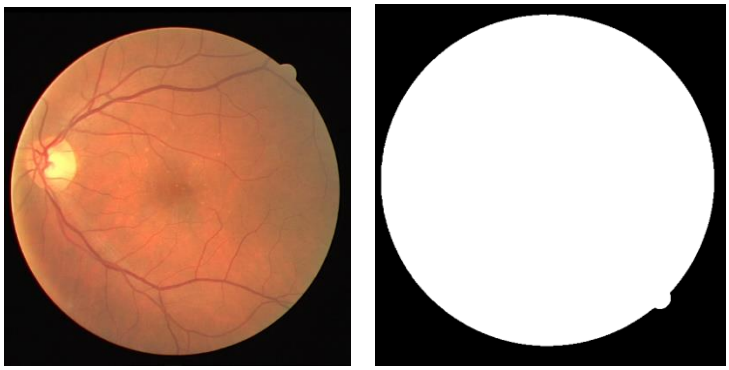

Fig 2: Sample Images

1.2.2 The Proposed CNN Model:

In this proposed system, Convolutional Neural Networks (CNNs) are chosen for the image segmentation of retina images due to their inherent advantages in image analysis tasks. Convolutional Neural Networks excel in capturing and learning intricate visual features present in images, making them particularly suitable for analyzing retinal images that contain complex structural patterns. The hierarchical nature of Convolutional Neural

Networks allows them to automatically extract and represent these features at different levels, enabling effective segmentation.

U-Net is a widely utilized Convolutional Neural Network (CNN) architecture that excels in image segmentation tasks. Its unique design combines a contracting path, which captures contextual information, and an expansive path, which enables precise localization. U-Net's contracting path consists of convolutional and pooling layers that progressively reduce the spatial dimensions while increasing the number of feature channels. This allows the network to learn high-level semantic features. On the other hand, the expansive path uses up-convolutional layers to recover the spatial resolution, creating a segmentation map that matches the input image's dimensions. U-Net's skip connections between corresponding layers in the contracting and expansive paths facilitate the transfer of detailed information, aiding in accurate segmentation of objects.

Similarly, SegNet is another notable CNN architecture specifically designed for semantic image segmentation. It employs an encoder-decoder structure to capture both global and local information. The encoder network consists of convolutional and pooling layers that progressively reduce spatial dimensions while extracting relevant features. These features are then fed into the decoder network, which employs upsampling layers to recover the original resolution. However, instead of using up-convolutional layers like U-Net, SegNet employs pooling indices from the corresponding encoder layers to perform pixel-wise upsampling. This approach helps retain important spatial information for precise segmentation. Furthermore, SegNet includes skip connections similar to U-Net, enabling the transfer of fine-grained details between encoder and decoder layers.

By leveraging the strengths of Convolutional Neural Networks, specifically U-Net and SegNet are powerful architectures for image segmentation tasks. The proposed system aims to achieve precise segmentation of retina images.They share the characteristic of utilizing an encoder-decoder structure to extract and combine features at multiple scales, enabling accurate segmentation of objects or regions of interest in images.The established success of these architectures in image classification tasks ensures the system's ability to effectively analyze and segment retina images based on their visual characteristics.

1.3. Organization of Project

The technique which is developed is taking input as a retina image and extracts the features and filters for the presence of blood vessels and nerves, these features extracted are segmented based on models' trainings and experiences. And the image is finally segmented as clear visible nerves of retina.

2. LITERATURE REVIEW

Title: Light-Field Imaging Reconstruction Using Deep Learning Enabling Intelligent Autonomous Transportation System

Authors: Davi Ribeiro Militani, Demóstenes Zegarra Rodríguez, Juan Casavílca Silva, Lunchakorn Wuttisittikulkij , Muhammad Saadi , Renata Lopes Rosa, and Sattam Al Otaibi

Summary: Plenoptic cameras, also referred to as light-field (LF) cameras, enable the recording of the 4D LF distribution of target scenes. Commercial plenoptic cameras, intelligent transportation systems, and other sectors and applications have all been investigated using THE light-field (LF) imaging. Because surface flaws and parallax problems must be taken into account when building accurate maps for the Intelligent Autonomous Transport System (IATS), light-field (LF) camera images are essential. To produce high-quality photos, a learning-based architecture that directly imitates the LF distribution is provided. The performance of the suggested framework is assessed using a variety of image quality assessment techniques, including PSNR, SSIM, IWSSIM, FSIM, GFM, MDFM, and HDR-VDP. Deep learning algorithms like CNN, GAN , SRCNN. In this work, LF image reconstruction framework that efficiently learns the picture distribution, reconstructs high-quality and densely sampled LF images, and extracts the spatioangular information of a scene. The suggested model additionally improves the spatial and angular resolutions by simulating the local LF distribution.

Title: Title: Harnessing Multi-View Perspective of Light Fields for Low-Light Imaging

Authors: Kranthi Kumar Rachavarapu, Kaushik Mitra and Mohit Lamba

Summary: The main goal of this effort was to improve LF recorded under dim lighting. Light Field (LF) offers special benefits like depth estimation and post-capture focussing, but low light situations severely restrict these capabilities. For Low-Light Light Field (L3F) restoration, a deep neural network called L3Fnet that not only improves the visual quality of each LF view but also maintains the epipolar geometry across views. L3F-wild dataset, which contains LF captured late at night with virtually zero lux values, to further examine the effectiveness of low-light restoration techniques. LF Processing Algorithm is

used . completing trials on the L3F-20, L3F-50, and L3F-100 datasets, the effectiveness of L3Fnet on LFs at various levels of low light was demonstrated. Results on the L3F-wild dataset demonstrated that L3Fnet could now automatically adjust to various light levels as a result of this preprocessing module.

Title: Raindrop Removal With Light Field Image Using Image Inpainting
Authors: Hang Su , Tao Yang , Nathan Crombez , Yassine Ruichek, Tomas Krajnik , Xiaofei Chang and Zhi Yan

Summary: In this study, an image inpainting technique for removing raindrops from light field images. First thing is to identify raindrop regions using the depth map created from the light field image, and these regions are then expressed as a binary mask. By shifting the focus to the far distance, the original raindrops image is made better. precision of the visual localisation percentages for the original sub-aperture photographs, improved images, and images with raindrops removed are 80%, 82%, and 86%, respectively.To assess the suggested picture restoration technique, image quality analysis is carried out. For the perception tasks in adverse weather, it's crucial to take advantage of both the rich ray information in light field images and the most recent advancements in deep learning-based image processing. Additional applications of the restored images include object detection and visual. Making raindrop removal using light field image an end-to-end is intriguing paradigm going forward. Additionally, merging the light field dataset with the EU long-term dataset for long-term vehicle autonomy is a focus of future work.

Title: A Double-Deep Spatio-Angular Learning Framework for Light Field-Based Face Recognition
Authors: Alireza Sepas-Moghaddam, Fernando Pereira, Kamal Nasrollahi , Mohammad A. Haque, Paulo Lobato Correia and Thomas B. Moeslund

Summary: In this paper, a framework for light field-based face identification using double-deep spatio-angular learning is proposed. Using two deep networks in succession, intra-view/spatial and inter-view/angular data The long short-term memory (LSTM) recurrent network, whose inputs are VGG-Face descriptions, generated using a VGG-16 convolutional neural network, is a component of the proposed double-deep learning system (CNN). The LSTM network then evaluates a series of VGG-Face spatial

descriptions. Utilizing the IST-EURECOM light field face database, a wide range of experiments tackling complex recognition problems have been carried out. Although each SA image's location within the multi-view array and the order in which the SA images are to be scanned are known, there is still some additional information regarding the inter-view angular information/dependencies, such as parallax, that could be further utilised during the learning process to improve the recognition accuracy and/or convergence speed. Future studies will examine how to further use the additional angle information by extending the LSTM for spatio-angular visual recognition challenges.

Title: Deep Light Field Super-Resolution Using Frequency Domain Analysis and Semantic Prior

Authors: Gangyi Jiang, Mei Yu, Yeyao Chen , Yo-Sung Ho and Zhidi Jiang

Summary: We have been widely concerned by the limited size of the imaging and angular resolutions. To this end, this paper proposes a new. enhance the spatial and angular resolutions of LFI. a frequency restoration process with new cascaded 2D and 3D convolutional neural networks. The designed network to enhance its representation ability and domain transformation. The field of LF research's "hot spot" has been LFSR.

Generally speaking, LFSR can be classified into two categories: angular SR, which synthesizes fresh views to improve angular resolution, and spatial SR, which improves the spatial resolution of LF images. Firstly, bidirectional optical flows and spatial interpolation are utilized to reconstruct the intermediate. the spatial and angular information frequency components. In addition, semantic priors containing the network to further improve its expression ability. The recovered results are transformed by inverse DCT to reconstruct the high-quality LFI. Unfortunately, effectively determining disparity from low-resolution (LR) LFI is difficult and expensive. CNN was employed to handle LF data because of its widespread performance in image restoration tasks and its ease of usage. Since LFI has a 4D structure, adopting 4D convolution for modelling makes sense, but it produces a large network architecture and necessitates a large amount of training data. In order to create a 2D CNN for modelling, the spatial-angular properties of LFI were examined. This led to the development of bidirectional recurrent CNN, spatial-angular interactive network, and spatial-angular separable convolution Demonstrates that the proposed method can effectively generate LFI with high spatial-angular resolution. performance in both subjective and objective aspects.

Title: CapsField: Light Field-Based Face and Expression Recognition in the Wild Using Capsule Routing
Authors: Ali Etemad, Alireza Sepas-Moghaddam, Fernando Pereira and Paulo Lobato Correia

Summary: Such as biometrics and affective computing are recognition in the wild in this regard, this research suggests that learning hierarchical links between capsules, created from each LF image, be preserved and made public. A detailed performance assessment research employing the new to perform personal identity, and expression recognition advances after the first automatic facial recognition. Despite recent developments in face and expression analysis, some conditions still cannot be accurately assessed. Systems for analysis cameras with lenslet light fields(LF), have recently come into prominence as they are able to simultaneously capture the intensity of light rays coming from multiple directions in space. The visual scene is "seen" by an LF camera. A multi-view is formed by the collection of 2D SA pictures that have been rendered face image analysis. The performance of the recognition solutions presents the biggest hurdle in this situation. This work takes the recognition paradigm into account. It suggested a solution by CapsField: CapsField is a brand-new deep learning approach that is suggested for face and expression identification in the real world. CapsField blends convolutional and capsule networks to take advantage of both the spatial and angular data present in LF images. Recently, a few solutions based on capsule networks have been put forth in the context of facial image analysis, including age and gender classification and expression identification, demonstrating how well-suited capsule networks are to handle variations in face pose. CapsField takes things a step further by using dynamic routing between capsules for the first time to harness the angle variances between the various views acquired in LF photos. Additionally, CapsField ignores the spurious dimensions and gives higher weights to the more significant features to strengthen its robustness.

Title: Light Field Image Super-Resolution via Mutual Attention Guidance
Authors: Yao Lu and Zijian Wang

Summary: Progress in light field image super-resolution has been significantly accelerated by deep learning-based techniques. However, the majority of them disregard

lining up various light field sub-aperture elements.image prior to aggregation, producing less than ideal super-resolution outcomes. For sub-aperture feature aggregation, we want to provide an effective feature alignment technique. In order to do this, we create a technique for mutual attention for sub-aperture feature alignment and suggest a mutual attention guidance block (MAG).With the help of the centre attention guidance module (CAG) and the surrounding attention guidance module, MAG achieves the mutual attention mechanism between the centre feature and surrounding feature (SAG). To create bidirectional center-view, SAG aligns the refined surrounding-view feature with the original surrounding-view feature after CAG aligns the center-view feature with the surrounding-view feature.

Title: AIFNet: All-in-Focus Image Restoration Network Using a Light Field-Based Dataset

Authors: Bin Chen, Jizhou Li, Lingyan Ruan , and Miu-Ling Lam

Summary: The performance of the image such as object recognition and image segmentation degrades when images are blur. Image converges out of image plane. This region is called circle of confusion (COC). However, defocus blur is undesirable for most computer vision and image processing tasks. Competitive results have been obtained in addressing single image motion deblurring by using DOF and LFDOF. However, the performance of existing deblurring techniques is still unsatisfactory.

So in order to restore the defocussed image in this paper they have implemented a convolutional neural network(CNN) architecture AIFNet for removing spatially varying defocus blur from a single defocused image for multiple image datasets they have used recognition techniques and Light Field Synthetic aperture. AIFNet consists of three modules: defocus map estimation, deblurring, and domain adaptation. The effects and performance of various network components are extensively evaluated. We also compare our methods with existing solutions using several publicly available image datasets. We leverage the photographic features of light fields. superior deblurring results. We fully analyze the contributions of. our method against existing techniques using different testing datasets. our AIFNet presents state-of-the-art performance. illustrate the benefits of all-in-focus image restoration.

Title: Quality Prediction on Deep Generative Images
Authors: Alan C. Bovik, Dae Yeol Lee, HyunsukKo , and Seunghyun Cho

Summary: Deep neural networks have been used for several tasks, including the creation of images. For applications like image reduction, generative adversarial networks (GANs) in particular are capable of creating incredibly lifelike images. Similar to normal compression, monitoring and managing the encoding process requires the ability to automatically evaluate the perceptual quality of generated images. On GAN-generated content, traditional image quality algorithms are ineffectual, particularly in textured regions and at high compressions. Here, we provide a new image quality predictor for generative images that is "naturalness"-based. Utilizing a multi-stage parallel boosting technique based on structural similarity features and assessments of statistical similarity, we have developed a novel GAN picture quality predictor. According to our experimental findings, our suggested GAN IQA model provideshigher-qualityy predictions on both generative image datasets and conventional image quality datasets. The SSQP model, which we suggested, was developed utilising two groupings of attributes that stand for structural and statistical similarity. In order to understand the nonlinear relationship between the proposed characteristics and the subjective scores, we additionally used a multi-stage parallel boosting method. We developed a database of generated images with high quality made up of GAN generative images, and we performed an empirical analysis on it. The experimental results show that SSQP performs better than existing FR models on the new database by a wide margin. On three conventional image quality datasets, it also achieved similar prediction accuracies as current DNN-based IQA models.

Title: An Off-Axis Flight Vision Display System Design Using Machine Learning
Authors: Jianlin Zhao, Shan Mao, Zhenbo Ren

Summary: An advanced optoelectronic tool known as the AFLIGHT simulator can recreate the procedures involved in takeoff and landing as well as the surrounding environment on the ground.for pilots. This study investigates a deep neural network for an off-axis flight visual display system design, focusing primarily on a free-form surface for take-off and landing pilot training. The ZPL-macro programming in the reverse-engineered ZEMAX software is used to propose the surface type and its related

expression, together with optical requirements for a flight vision display system. The network has great accuracy in predicting the free-form surface shape from the original structures and parameters. Finally, we explore these data for neural network learning and training. This approach and neural network may provide follow-up research for complex and high-quality optical system design based on artificial intelligence methodology, greatly cut design time, increase design accuracy, and lessen the dependence of optical engineers on experience.

Title: A Deep Learning for Unsupervised Anomaly Localization in Industrial Images
Authors: Chandranath Adak, Shaohua Yan , Xian Tao , Xin Zhang and Xinyi Gong

Summary: As we know, deep learning-based visual inspection has been highly successful with the help of supervised learning methods. However, in real industrial scenarios, the scarcity of defect samples, the cost of annotation, and the lack of knowledge of defects may render supervised-based methods ineffective. In recent years, unsupervised anomaly localization algorithms have become more widely used in industrial inspection tasks. The researchers in this field by comprehensively surveying recent achievements in unsupervised anomaly localization in industrial images using deep learning. The survey reviews covering different aspects of anomaly localization, mainly covering various concepts, challenges, taxonomies, benchmark datasets, and quantitative performance comparisons of the methods reviewed. In reviewing the achievements to date, this paper provides detailed predictions and analysis of several future research directions. This review provides detailed technical information for researchers interested in industrial anomaly localization and who wish to apply it to the localization of anomalies in other fields.

Title: The Recognition Framework of Deep Kernel Learning for Enclosed Remote Sensing Objects
Authors: DAZHENG FENG, JIE CHEN , LONG SUN , and MENGDAO XING
Summary: Remote sensing image target recognition is used in various fields, such as ships, tanks, airplanes, and vehicles, which are closed targets. The features of these targets include target outlines that are obvious and target discriminant features that are significantly different from the surrounding environment, and the targets are characterized as small and dense. Therefore, the recognition of these types of targets is a popular topic.

We proposed a recognition framework consisting of a remote sensing image target recognition method based on deep saliency kernel learning analysis, which uses a target region extraction method based on the visual saliency mechanism and implements a nonlinear deep kernel learning saliency feature analysis method to realize target extraction and recognition. Experimental results show that a 95.9% recognition rate is achieved for SAR remote sensing target recognition on the public MSTAR data set, a 96% recognition rate on the UC Merced Land Use data set, and an 85% recognition rate on a self-built visible light remote sensing image data set. The recognition framework can be used for video recognition.

Title: Deep Light Field Spatial Super-Resolution Using Heterogenous Imaging
Authors: Gangyi Jiang, Haiyong Xu, Mei Yu, Yeyao Chen, Yo-Sung Ho

Summary: Light field (LF) imaging expands traditional imaging techniques by simultaneously capturing the intensity and direction information of light rays, and promotes many visual applications. However, owing to the inherent trade-off between the spatial and angular dimensions, LF images acquired by LF cameras usually suffer from low spatial resolution. Many current approaches increase the spatial resolution by exploring the four-dimensional (4D) structure of the LF images, but they have difficulties in recovering fine textures at a large upscaling factor. To address this challenge, this paper proposes a new deep learning-based LF spatial super-resolution method using heterogeneous imaging (LFSSR-HI). The designed heterogeneous imaging system uses an extra high-resolution (HR) traditional camera to capture the abundant spatial information in addition to the LF camera imaging, where the auxiliary information from the HR camera is utilized to super-resolve the LF image. Specifically, an LF feature alignment module is constructed to learn the correspondence between the 4D LF image and the 2D HR image to realize information alignment. Subsequently, a multi-level spatial-angular feature enhancement module is designed to gradually embed the aligned HR information into the rough LF features. Finally, the enhanced LF features are reconstructed into a super-resolved LF image using a simple feature decoder. To improve the flexibility of the proposed method, a pyramid reconstruction strategy is leveraged to generate multi-scale super-resolution results in one forward inference. The experimental results show that the proposed LFSSR-HI method achieves significant advantages over the state-of-the-art

methods in both qualitative and quantitative comparisons. Furthermore, the proposed method preserves more accurate angular consistency.

Title: Unrolling Graph Total Variation for Light Field Image Denoising

Authors: Gene Cheung, Huy Vu , Kazuya Kodama , RinoYoshida , Takayuki Hamamoto

Summary: A light field (LF) image is composed of multiple sub-aperture images (SAIs) from slightly offset viewpoints. To denoise a noise-corrupted LF image, leveraging recent development in deep algorithm unfolding, we pursue a hybrid graph-model-based / data-driven approach. Specifically, we first connect each pixel in a target patch of an SAI to neighbouring pixels within the patch, and to pixels in co-located "similar" patches in adjacent SAIs. Given graph connectivity, we formulate a maximum a posteriori (MAP) problem using graph total variation (GTV) as signal prior. We then unroll the iterations of a corresponding optimization algorithm into a sequence of neural layers. In each unrolled layer, we learn relevant features per pixel from data using a convolutional neural net (CNN) in a supervised manner, so that edge weights can be computed as functions of feature distances. Each neural layer can be interpreted as a graph low-pass filter for a 4D LF image patch. Experiments show that our proposal outperformed two model-based and two deep-learning-based implementations in numerical and visual comparisons.

Title: Camera -Based Light Emitter Localization Using Correlation of Optical Pilot Sequences

Authors: Ashwin Ashok, Kristin J. Dana , Macro Grusteser , MD Rashed Rahman , Narayan B. Madayam , Shubham Jain and T. V. Sethuraman

Summary: Visual identification of objects using cameras requires precise detection, localization, and recognition of the objects in the field-of-view. The visual identification problem is very challenging when the objects look identical and features between distinct objects are indistinguishable, even with state-of-the-art computer vision techniques. The problem becomes significantly more challenging when the objects themselves do not carry rich geometric and photometric features, for example, in visual identification and tracking of light emitting diodes (LED) for visible light communication (VLC) applications. In this paper, we present a camera based visual identification solution where objects or regions of interest are tagged with an actively transmitting LED. Motivated by the concept of pilot symbols, typically used for synchronization and channel estimation in radio

communication systems, the LED actively transmits unique pilot symbols which are detected by the camera across a series of image frames using our proposed spatio-temporal correlation-based algorithm. We setup the visual identity as a problem of localization of the LED on the camera image, which involves identifying and the unique ID corresponding to the LED. In this paper, we present the algorithm and trace-based evaluation of the identification accuracy under real-world conditions including indoor, outdoor, static, and mobile scenarios.

3. THEORETICAL ANALYSIS OF PROPOSED PROJECT

3.1. Requirements Gathering

3.1.1. Software Requirements

- Operating System: Windows
- Programming Language: Python
- Packages: Numpy, Matplotlib, Random, SimpleITK, Tensorflow
- Software: Jupyter Notebook

3.1.2. Hardware Requirements

- Processor: Intel Core i5 or above
- RAM: 8GB

3.2. Technological Description

Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors.

This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them

have later been patched and updated by people with no Python background - without breaking.

Tensorflow

TensorFlow is an extensively used open-source software library that offers robust dataflow and differentiable programming capabilities. It is widely adopted for diverse technological applications, particularly in the field of machine learning, specifically neural networks. With its flexible design, TensorFlow caters to the needs of both research and production environments, making it a reliable choice for implementing cutting-edge technologies. Originally developed by the Google Brain team for internal purposes, TensorFlow was later released under the Apache 2.0 open-source license. This licensing choice expanded its accessibility, allowing a wider community to leverage its functionalities and contribute to its development.

For deep learning tasks, TensorFlow provides a comprehensive API called tensorflow.keras, which simplifies the process of building and training neural networks. This API encompasses a range of modules and functions tailored to meet the demands of constructing efficient and scalable deep learning models. The layers module offers a diverse set of layer types, including convolutional, dense, and pooling layers, enabling users to design complex network architectures. The Model module within tensorflow.keras empowers developers to create customized models by specifying the input and output layers, facilitating the construction of neural network topologies tailored to specific technological requirements. This flexibility allows for the incorporation of various architectures and configurations to achieve optimal performance.

TensorFlow also provides specialized modules for data preprocessing and augmentation, streamlining the preparation of input data for training. The ImageDataGenerator module efficiently loads and preprocesses image data, while the preprocess_input function ensures compatibility of input images with specific neural network architectures, optimizing the training process. To enhance training efficiency and effectiveness, TensorFlow offers a range of callback functions. These functions, such as ModelCheckpoint and EarlyStopping, enable automatic model saving during training and early termination based on specified conditions, respectively. This proactive approach ensures model integrity and enhances training workflow.

Furthermore, TensorFlow facilitates the utilization of pre-trained models, enabling transfer learning for improved efficiency and performance. The load_model function allows for the integration of pre-trained weights and architectures, serving as a solid foundation for training on new datasets or solving related tasks. This transfer learning capability saves valuable time and computational resources. Overall, TensorFlow, with its versatile technological capabilities and the streamlined workflow provided by the tensorflow.keras API, empowers developers and researchers to implement state-of-the-art machine learning technologies with ease. Its robust features make it a preferred choice for various technological applications, driving advancements in the field of deep learning.

Numpy

Numpy is a versatile package for array processing that offers a range of functionalities. It revolves around a high-performance multidimensional array object, providing efficient storage and manipulation of data. Numpy is widely used in scientific computing with Python, serving as a fundamental tool in this domain. Its key features include a powerful N-dimensional array object, advanced functions for broadcasting, support for integrating C/C++ and Fortran code, and capabilities for linear algebra, Fourier transforms, and random number generation. While Numpy is primarily known for its scientific applications, it can also serve as an effective container for generic data. This flexibility allows Numpy to seamlessly and rapidly integrate with various databases, enabling efficient data handling and processing.

Matplotlib

Matplotlib is a powerful 2D plotting library for Python that offers high-quality figures suitable for publication purposes. It supports a wide range of output formats and can be utilized in various environments, including Python scripts, IPython shells, Jupyter Notebooks, web application servers, and graphical user interface toolkits. Matplotlib aims to provide an intuitive and straightforward experience for simple plotting tasks while also offering advanced capabilities for complex visualizations.

With just a few lines of code, you can generate a diverse range of plots, histograms, power spectra, bar charts, error charts, scatter plots, and more. The pyplot module, especially when combined with IPython, provides a user-friendly interface similar to MATLAB. For

advanced users, Matplotlib offers extensive customization options, allowing precise control over line styles, font properties, axes properties, and more through an object-oriented interface or MATLAB-like functions. To explore the capabilities of Matplotlib, you can refer to the sample plots and thumbnail gallery provided by the library.

SimpleITK

SimpleITK is a comprehensive library that provides a simple and efficient interface to the Insight Segmentation and Registration Toolkit (ITK) for image analysis and processing tasks. It is specifically designed to handle medical imaging data, offering a wide range of functionalities for tasks such as image registration, segmentation, filtering, and visualization. SimpleITK provides an abstraction layer over ITK, making it easier for users to work with medical images and leverage the power of ITK algorithms.

One of the key features of SimpleITK is its support for a diverse set of image formats commonly used in the medical imaging field, including DICOM, NIfTI, Analyze, and many others. This makes it convenient for reading, writing, and manipulating medical images in various formats.

SimpleITK offers a simple and intuitive programming interface, making it accessible to both novice and experienced developers. It provides a consistent API across different programming languages, including Python, allowing users to seamlessly switch between languages while working with medical image data.

SimpleITK includes a wide range of image processing algorithms and filters that can be applied to medical images. These include basic operations such as image smoothing, resampling, and thresholding, as well as advanced techniques like image registration, segmentation using region-growing or level set methods, and feature extraction.

Additionally, SimpleITK integrates with popular visualization libraries such as Matplotlib and VTK (Visualization Toolkit), enabling users to visualize and interact with medical images in a convenient and customizable manner.

Overall, SimpleITK provides a powerful and user-friendly toolkit for medical image analysis and processing tasks, allowing researchers, clinicians, and developers to

efficiently work with medical image data and apply advanced algorithms for a variety of applications in healthcare and medical research.

Jupyter Notebook

The Jupyter Notebook is a web-based application that allows users to create and share documents containing live code, equations, visualizations, and text. It is an open-source project maintained by Project Jupyter. Originally derived from the IPython project, the Jupyter Notebook was created as a separate project to provide a flexible and interactive computing environment. The name "Jupyter" is a combination of the core supported programming languages it initially supported: Julia, Python, and R. While it ships with the IPython kernel, which enables programming in Python, Jupyter supports a wide range of programming languages through various kernels. Currently, there are more than 100 kernels available, allowing users to write code in different languages within the Jupyter Notebook environment. The Jupyter Notebook has gained popularity due to its versatility and its ability to seamlessly combine code, visualizations, and explanatory text, making it a powerful tool for data analysis, research, and interactive programming.

4. DESIGN

4.1. Introduction:

The ability to see is provided by the eyes. They receive light from the environment and transmit visual data to the human brain. The eyes have a 200-degree field of view, which includes the front and sides of oneself (peripheral vision). To see images, movement, and depth, many parts of the eyes must operate together. The eyes are capable of seeing millions of distinct colors.

Retina: The innermost layer of light-sensitive tissue in the eyes of most vertebrates and some mollusks is called the retina as in figure 1. From the eye's optics, the retina receives a two-dimensional image of the world, examines it, and transmits nerve impulses conveying the sorted image to the visual brain via the optic nerve to form vision. In many aspects, the retina acts in a way similar to the film or picture sensor of a camera. As a vital part of the eye, the retina plays a significant role in the human body. Humans rely on their eyes for 80% of the ways they receive information from the outside world. The quality of human eyesight is significantly influenced by the retina as you can see in figure 3. Clinical medicine currently relies on ophthalmological tools and medical imaging technology to diagnose and treat individuals with retinal injury or detached blindness.

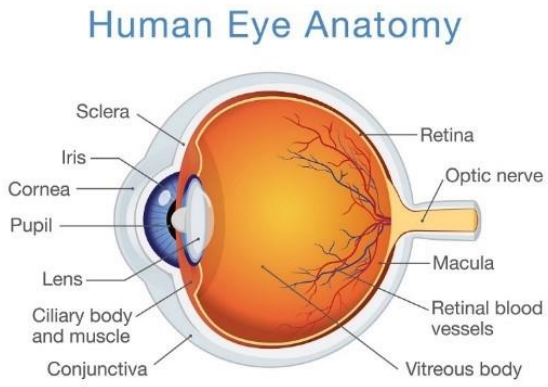

Fig 3: Human Eye Anatomy

Retina Blood Vessels. The outer retina, which is primarily made up of photoreceptors, and the retinal pigment epithelium are supplied by the choriocapillaris and the central retinal artery, respectively. The middle retinal artery passes along the lower border of the optic nerve sheath before leaving the optic nerve and accessing the eye. In the interior retina, the artery splits into three capillary layers. The retinal vessels carry blood to the intraretinal neurons [1]. The choriocapillaris, which is located beneath the retinal pigment epithelium, is the only source of oxygen diffusion for the avascular photoreceptor layer.

Pathological conditions [2] and noise in the retinal picture's present certain difficulties in the extraction of the blood vessels. Additionally, it has been noted that variation between blood vessels and the retinal framework in retinal imaging is minimal, which makes it challenging to undertake the careful observation and sound judgment while administering medical therapy. As a result, several deep learning approaches are applicable.

These techniques will help the doctors by providing a grayscale image of the retina image which has a clear vision of optic nerves, and blood vessels of the eye which helps in better understanding of the eye for the doctors[3]. Then from the image doctors can treat the patients with better analysis.

4.2. Architecture Diagram

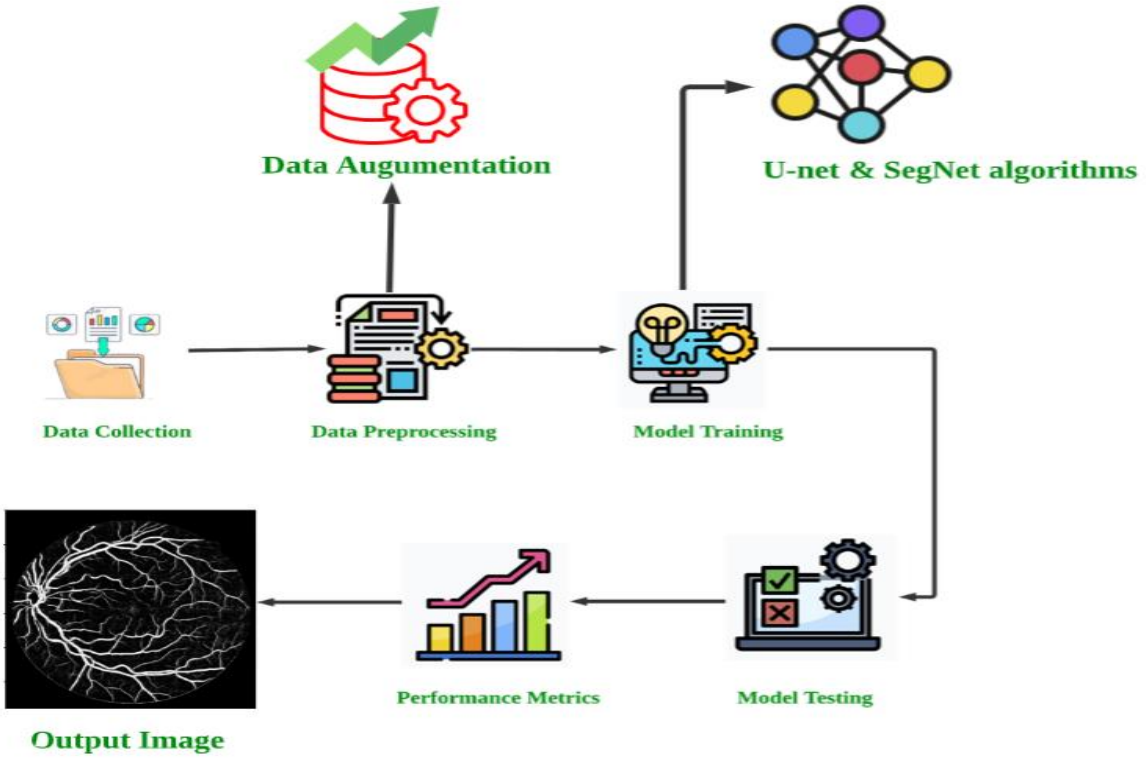

Fig 4: Proposed Systems Architecture Design

The proposed model's architecture as in figure 4 can be broken down into six phases or six steps. Firstly, Data is gathered from Kaggle [19] which consists of 40 images and has a size of 29.98 MB and then data augmentation is done. Data augmentation is done to increase the dataset in order to boost the model's performance since the more variations there are, the more experiences the model may get. SegNet and U-Net are utilized to prepare and train the prototype. Then the model is used to test before being made available

to the public. Calculating the metrics to determine the degree to which the model findings may be trusted for accuracy.

4.3. Algorithm

4.3.1. U-Net: The acronym "U-Net" stands for "U-shaped Network". The U-Net algorithm is a convolutional neural network (CNN) architecture that has gained significant popularity and has become a standard approach for accurate and detailed segmentation of objects or regions of interest in medical images. It was introduced in the paper titled "U-Net: Convolutional Networks for Biomedical Image Segmentation" by Olaf Ronneberger, Philipp Fischer, and Thomas Brox in 2015.

U-Net is specifically designed to address the challenges of medical image segmentation, where precise localization and extraction of anatomical structures or abnormalities are crucial for diagnosis and treatment planning. The algorithm's architecture combines an encoder-decoder structure with skip connections, enabling the network to capture both high-level context and fine-grained details.

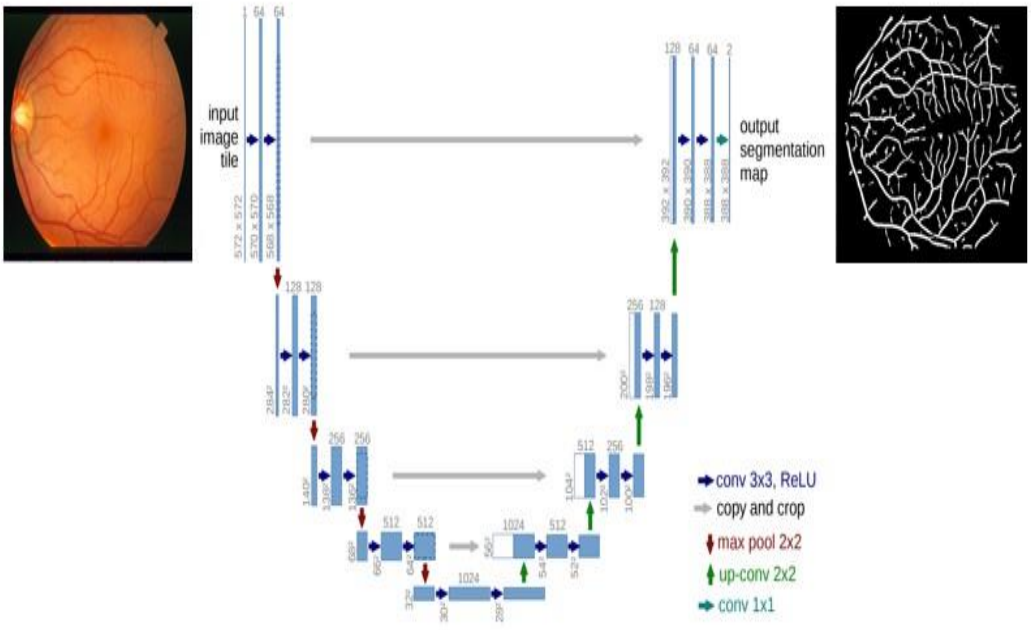

Fig 5: Architecture of U-Net

The U-Net architecture consists of two main parts: the contracting path (encoder) and the expansive path (decoder) as you can see in figure 5. The contracting path captures

contextual information and learns high-level semantic features, while the expansive path recovers spatial resolution and generates accurate segmentation maps.

The contracting path starts with a series of convolutional layers followed by a rectified linear unit (ReLU) activation function, which introduces non-linearity and allows the network to model complex relationships. These convolutional layers extract features from the input image, progressively reducing the spatial dimensions while increasing the number of feature channels.

Convolution layer: The convolutional layer is considered an essential block of the CNN. Convolutional layers are good for feature extraction from images as they deal with the spatial redundancy by weight sharing. It contains a set of filters (or kernels), parameters of which are to be learned throughout the training. Each filter convolves with the image and creates an activation map. The output of the convolution operation is computed by convolving an input (I) with a number of filters.

Max pooling: Pooling is a down sampling operation that reduces the dimensionality of the feature map as you can see in figure 6. Max Pooling is a convolution process where the Kernel extracts the maximum value of the area it convolves. It calculates the maximum value for each patch of the feature map.

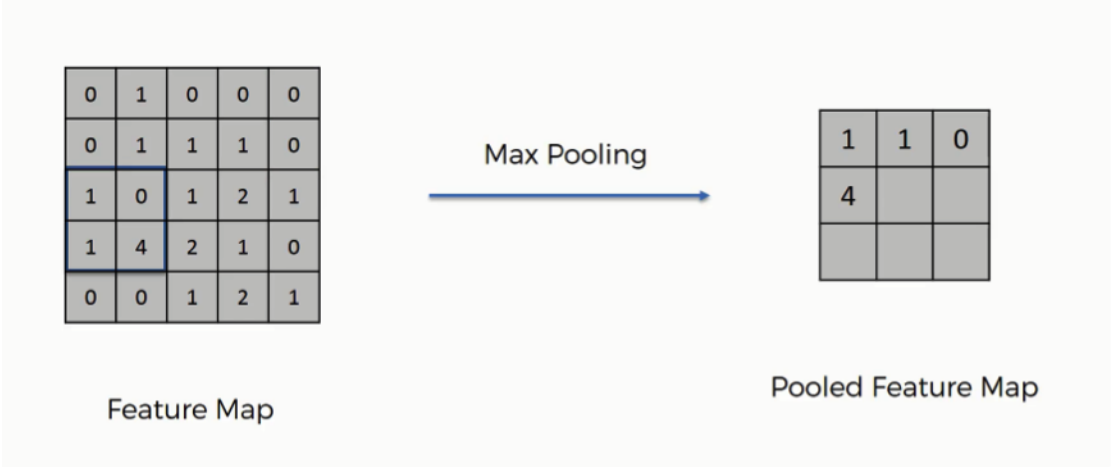

Fig 6: Max Pooling

ReLU: The rectified linear activation function or ReLU for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero.

In each step of the contracting path, max-pooling layers are applied to downsample the feature maps, reducing their spatial dimensions while retaining the most salient information. The downsampling process allows the network to capture larger contextual information and learn high-level features. The downsampling is typically performed using 2x2 max-pooling with a stride of 2.

The expansive path aims to recover the spatial resolution and generate segmentation maps that match the dimensions of the original input image. It consists of up-convolutional layers (also known as transpose convolutions or deconvolutions) that perform upsampling operations to restore the spatial dimensions. These up-convolutional layers are followed by concatenation with the corresponding feature maps from the contracting path (skip connections).

The skip connections are a key component of the U-Net architecture. They allow the network to transfer fine-grained details and spatial information from the contracting path to the expansive path. By fusing features from different scales and levels of abstraction, U-Net can effectively localize objects and capture intricate details in the segmented regions. The skip connections help to mitigate the problem of information loss during the downsampling process, improving the accuracy of segmentation.

In the U-Net architecture, the skip connections are implemented by concatenating the feature maps from the contracting path with the corresponding upsampled feature maps in the expansive path. This concatenation operation helps in preserving spatial information and providing the network with access to both local and global context during the upsampling process.

At the end of the expansive path, a pixel-wise softmax activation function is applied to the output layer. This activation function converts the network's output into a probability map, where each pixel represents the probability of belonging to a particular class (segmentation label). The softmax function ensures that the probabilities sum up to one for each pixel, allowing U-Net to provide fine-grained segmentation masks with class probabilities at the pixel level.

The U-Net algorithm is widely used in medical image segmentation tasks due to its accuracy and ability to capture fine details. It has been applied to various medical imaging modalities, such as magnetic resonance imaging (MRI), computed tomography (CT), and microscopy images. U-Net has shown remarkable performance in segmenting organs, tumors, lesions, cells, and other structures of interest in medical images.

One of the advantages of U-Net is its ability to learn from limited labeled data. It leverages the concept of data augmentation, where the input images are artificially modified through transformations like rotations, flips, and scaling. This augmentation increases the diversity of the training data and helps the network generalize better to unseen data.

Moreover, U-Net can benefit from transfer learning by using pre-trained weights from a larger dataset or a related task. This approach enables the network to leverage learned features and accelerate the training process, especially when the available medical datasets are small.

In summary, the U-Net algorithm is a powerful CNN architecture specifically designed for accurate and detailed medical image segmentation. Its encoder-decoder structure, combined with skip connections, allows the network to capture both contextual information and fine-grained details. U-Net has demonstrated remarkable performance in various medical segmentation tasks, making it a valuable tool for medical professionals and researchers in the field of image analysis and interpretation.

4.3.2. SegNet: The acronym "SegNet" stands for "Semantic Segmentation Network". The SegNet algorithm is a convolutional neural network (CNN) architecture specifically designed for semantic segmentation of images. It was introduced in the paper titled "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation" by Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla in 2015.

SegNet aims to address the task of pixel-level semantic segmentation, where the goal is to assign a semantic class label to each pixel in an input image. This fine-grained segmentation is crucial for various computer vision applications, such as autonomous driving, medical imaging, and scene understanding.

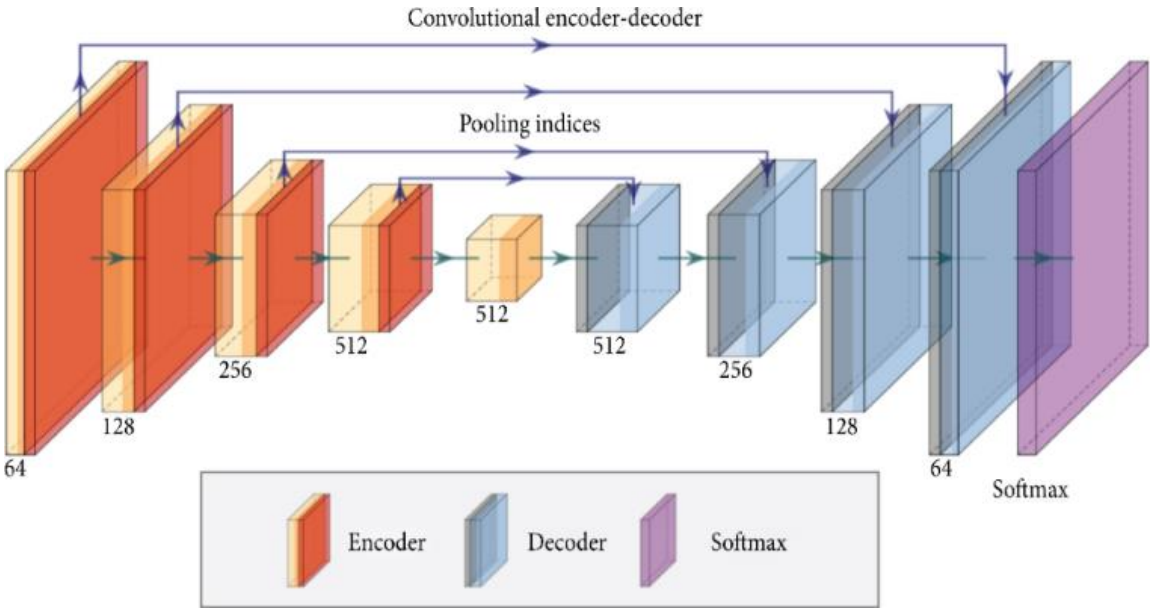

Fig 7: SegNet Architecture

The architecture of SegNet as you can see in figure 7 consists of an encoder-decoder structure with skip connections. The encoder part captures the contextual information and learns high-level features, while the decoder part recovers spatial resolution and generates pixel-wise segmentation maps.

The encoder of SegNet comprises a series of convolutional and pooling layers. These layers progressively reduce the spatial dimensions of the feature maps while increasing the number of feature channels. The convolutional layers extract local features, while the max-pooling layers perform down-sampling operations to capture global context and reduce the computational burden.

During the encoding process, SegNet saves the indices of the max-pooling operations for each pixel in the pooling layers. These indices are later used in the decoder part to upsample the feature maps efficiently, ensuring precise localization of objects.

The decoder part of SegNet consists of upsampling layers followed by convolutional layers. The upsampling layers use the saved indices from the corresponding encoder pooling layers to perform precise upsampling and recover the spatial resolution. This process helps in maintaining the fine-grained details of the segmented regions.

The skip connections in SegNet play a vital role in transferring both local and global contextual information from the encoder to the decoder. These connections enable the network to combine low-level and high-level features, capturing both fine details and high-level semantics. By fusing features from multiple levels of abstraction, SegNet can effectively localize objects and improve the accuracy of segmentation.

After the decoding process, The pixel-wise softmax function is an essential component in the SegNet algorithm for image segmentation. It is used to convert the network's output into a probability map where each pixel represents the probability of belonging to a particular class. Here's a detailed discussion about the pixel-wise softmax function in SegNet:

In SegNet, the pixel-wise softmax function is applied after the decoding process of the network. The decoder generates feature maps that contain information about different classes present in the input image. However, these feature maps need to be transformed into a pixel-level segmentation map that assigns a semantic class label to each pixel.

The softmax function is a popular activation function used in classification tasks. It takes a vector of real numbers and outputs a probability distribution over multiple classes. In the case of SegNet, the softmax function is applied pixel-wise, meaning it is independently applied to each pixel of the feature maps.

The pixel-wise softmax function takes the output feature maps from the decoder and computes the softmax activation for each pixel independently. It calculates the exponentiated value of each pixel's activation and normalizes these values by dividing by the sum of the exponentiated activations across all classes. This normalization ensures that the probabilities sum up to one for each pixel.

The result of the pixel-wise softmax function is a probability map where each pixel contains a probability value between 0 and 1, indicating the likelihood of belonging to a particular class. The pixel with the highest probability is often assigned as the predicted class for that pixel.

By applying the pixel-wise softmax function, SegNet converts the learned feature representations into meaningful pixel-level predictions. These predictions can then be used for further analysis, such as object detection, instance segmentation, or further post-processing steps.

The pixel-wise softmax function is crucial in SegNet as it allows the network to output per-pixel probabilities, enabling fine-grained segmentation. It ensures that the segmentation results are consistent and coherent across the image, providing detailed class assignments for each pixel.

One of the advantages of SegNet is its memory efficiency. By using pooling indices during upsampling, SegNet avoids the need for learning and storing additional parameters. This makes it computationally efficient and allows it to be deployed on resource-constrained devices.

SegNet has been successfully applied to various computer vision tasks, including road segmentation in autonomous driving, medical image segmentation, and scene understanding. Its ability to accurately capture detailed segmentation maps with limited computational resources has made it a popular choice in these domains.

In summary, the SegNet algorithm is a CNN architecture designed for pixel-level semantic segmentation. Its encoder-decoder structure, along with skip connections and pooling indices, enables the network to capture both local and global context, recover spatial resolution, and generate accurate segmentation maps. SegNet's memory efficiency and impressive segmentation performance make it a valuable tool for various computer vision applications.

5. IMPLEMENTATION

5.1. Coding

GitHub link: https://github.com/harshini1307/Retina-Gray-Scale-Image-Perception-using-Deep-Learning

5.2. Input

The training dataset is obtained from Kaggle names as DRIVE(Digital Retinal Images for Vessel Extraction) which has a total of 40 images,those are divided as training and testing datasets. Each having sub folders like manual,mask and ground images of retinal images. The training set consists of 20 images, similary test set consists of 20 images. These images are multiplied using data augmentation steps and generated 1,90,000 patches for training and 3200 patches for testing. These are given to the model for training. A sample of the images used for training are in figure 8:

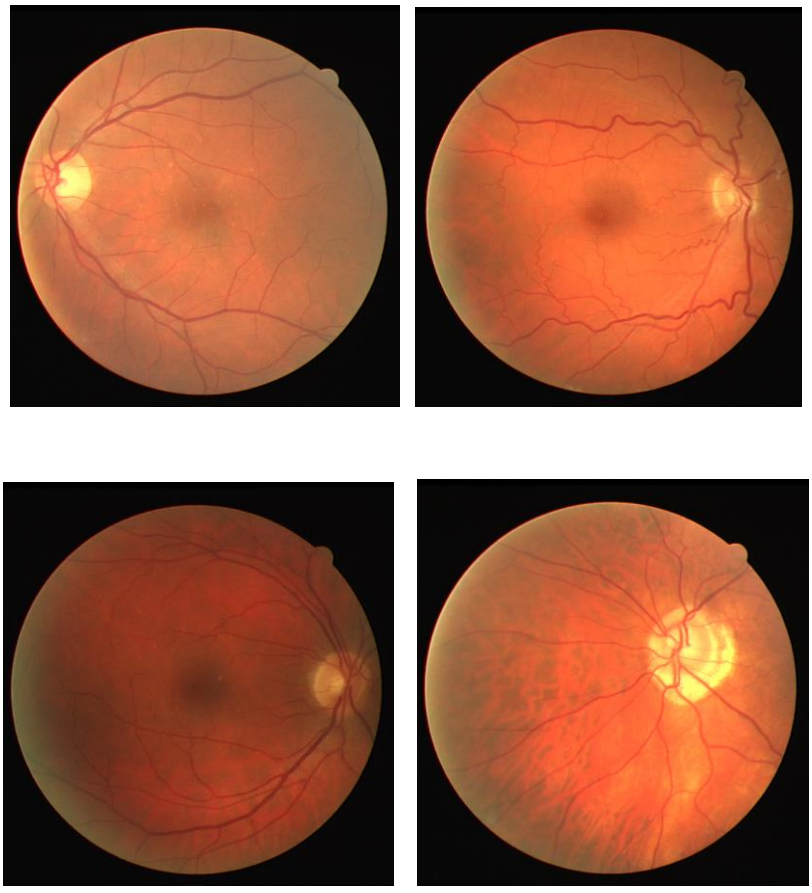

Fig 8: Training Images

5.3. Evaluation Metrics

True Positive (TP): True positives represent the cases where the model correctly identifies the positive class or the target class. In image segmentation, a true positive refers to a pixel or region that is correctly classified as belonging to the segmented object or the positive class.

False Positive (FP): False positives occur when the model incorrectly identifies the negative class as the positive class. In image segmentation, a false positive refers to a pixel or region that is mistakenly classified as belonging to the segmented object or the positive class, even though it belongs to the background or negative class.

False Negative (FN): False negatives occur when the model incorrectly identifies the positive class as the negative class. In image segmentation, a false negative refers to a pixel or region that belongs to the segmented object or the positive class but is mistakenly classified as part of the background or negative class.

True Negative (TN): True negatives represent the cases where the model correctly identifies the negative class or the background class. In image segmentation, a true negative refers to a pixel or region that is correctly classified as belonging to the background or negative class.

These terms are important for evaluating the performance of image segmentation models. True positives and true negatives indicate correct predictions, while false positives and false negatives represent classification errors. By analyzing these quantities, we can assess the accuracy, precision, recall, and other performance metrics of the model, helping us understand its strengths and weaknesses in segmenting objects of interest.

ROC Curve (Receiver Operating Characteristic Curve):

The ROC curve evaluates the performance of an image segmentation model by plotting the true positive rate (TPR) against the false positive rate (FPR) at different classification thresholds.

1. True Positive Rate (TPR), also known as Sensitivity or Recall:

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

where TP represents the number of true positive predictions (correctly predicted positive instances) and FN represents the number of false negative predictions (positive instances incorrectly predicted as negative).

2. False Positive Rate (FPR): $FPR = FP / (FP + TN)$

where FP represents the number of false positive predictions (negative instances incorrectly predicted as positive) and TN represents the number of true negative predictions (correctly predicted negative instances).

In image segmentation, the ROC curve can be obtained by treating the segmented regions as the positive class and the background as the negative class.

To construct the ROC curve, the model's output (such as probability maps or binary masks) is thresholded to generate segmentation predictions. Different threshold values are applied to obtain a range of TPR and FPR values, which are then plotted on the ROC curve.

The ROC curve in image segmentation helps to understand the trade-off between correctly identifying the segmented regions (TPR) and incorrectly classifying background regions as positive (FPR). The area under the ROC curve (AUC-ROC) provides a single value that summarizes the model's overall performance, with higher AUC-ROC indicating better segmentation accuracy.

IOU (Intersection over Union):

IOU also known as Jaccard Index, is another commonly used evaluation metric in image segmentation tasks. It measures the overlap between the predicted segmentation mask and the ground truth mask.

IOU is calculated by dividing the area of intersection between the predicted and ground truth masks by the area of their union. The formula for IOU is as follows:

IOU = Intersection Area / Union Area

To calculate IOU, the predicted segmentation mask and the ground truth mask are typically binarized, where each pixel is assigned either a value of 0 or 1, indicating background or foreground, respectively.

Here's a step-by-step explanation of the calculation:

1. Intersection Area: The intersection area is the number of pixels where both the predicted and ground truth masks have a value of 1 (foreground). It represents the common region correctly identified by the model and present in the ground truth.

2. Union Area: The union area is the total number of pixels where either the predicted or the ground truth mask has a value of 1 (foreground). It represents the combined region covered by both the predicted and ground truth masks.
3. IOU Calculation: The intersection area is divided by the union area to obtain the IOU score. IOU ranges from 0 to 1, where 0 indicates no overlap between the predicted and ground truth masks, and 1 indicates a perfect match.

The IOU score provides a measure of the spatial agreement between the predicted and ground truth masks. It quantifies how well the model captures the object boundaries and the overall shape of the segmented region. A higher IOU score indicates a higher degree of overlap and better segmentation accuracy.

IOU is particularly useful when evaluating image segmentation tasks where objects may have irregular shapes or varying sizes. It offers a robust evaluation metric that considers both false positives and false negatives. IOU is often used in conjunction with other metrics like precision, recall, F1 score, and pixel accuracy to comprehensively assess the performance of image segmentation models.

In summary, IOU (Intersection over Union) measures the overlap between the predicted and ground truth masks in image segmentation tasks. It is calculated by dividing the intersection area by the union area. IOU provides a quantitative assessment of segmentation accuracy and is commonly used as an evaluation metric in deep learning applications.

Precision:

Precision in image segmentation represents the accuracy of the model in correctly identifying pixels belonging to the segmented regions. It is calculated as the ratio of true positives to the sum of true positives and false positives. A high precision value indicates that the model is accurate in predicting positive instances, minimizing false positives.

$$\text{Precision} = TP / (TP + FP)$$

where TP represents the number of true positive predictions and FP represents the number of false positive predictions.

In the context of image segmentation, precision helps evaluate the model's ability to delineate the segmented regions accurately. It is particularly important when false positives can have significant consequences, such as in medical imaging, where misclassifying healthy tissue as a tumor can lead to incorrect diagnoses.Precision, along

with recall, provides a more detailed understanding of the model's performance in image segmentation tasks.

In summary, the ROC curve, F1 score, and precision are valuable evaluation metrics in image segmentation tasks in deep learning. The ROC curve allows for analyzing the trade-off between true positive and false positive rates, providing an overall performance assessment. The F1 score combines precision and recall, offering a balanced evaluation of segmentation accuracy. Precision specifically measures the accuracy of the model in identifying positive instances, which is crucial in applications where false positives need to be minimized. These metrics help assess and compare the performance of image segmentation models, assisting in the development and refinement of accurate and robust segmentation algorithms.

5.4. Results

Given the model an image and the model (SegNet and U-Net) should be used to detect and segment, the given retina images into clear and highly visible images.

On the images produced through data augmentation, the models are trained and tested. The expected blood vessels and nerves are segmented by the models. SegNet performs better than U-Net in comparison.

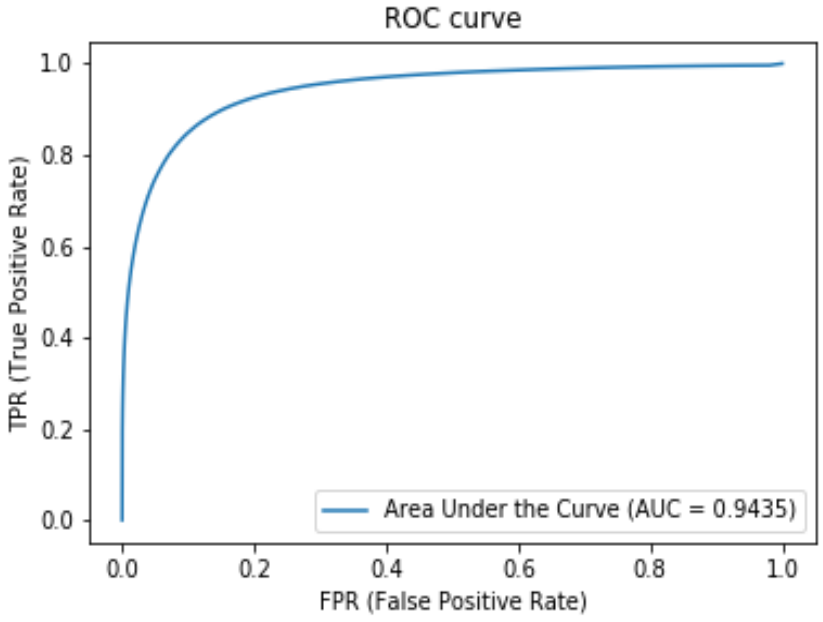

Fig 9: ROC Curve of SegNet Model

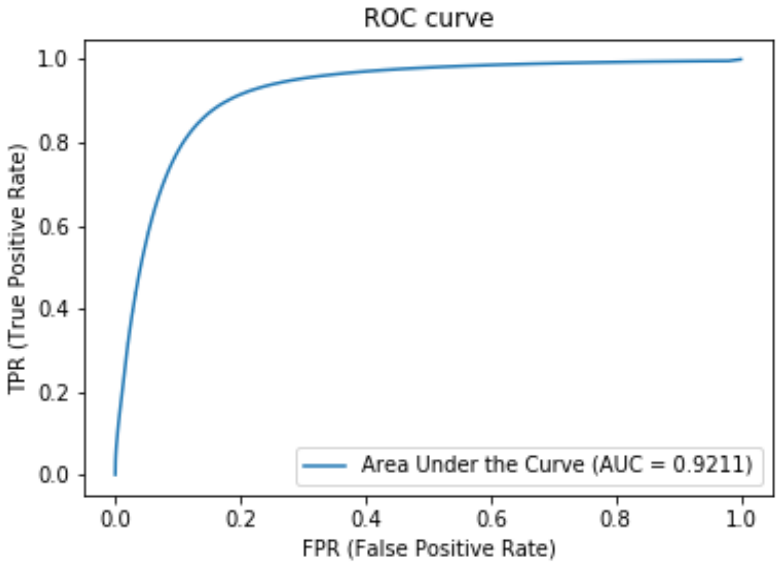

Fig 10 : ROC Curve of U-Net Model

The accuracy of the models (SegNet and U-Net) is compared with the ROC Curves in figure 9 and 10 respectively. The results of these models can be used accurately to segment retinal images, this would help radiologists as well as surgeons to treat vision related diseases early stage. Segmenting the nerves and blood vessels in the retina makes it easy to identify retinal detachments and diabetic retinopathies. To make up for the lack of retina pictures, data augmentation is used, and the generated images, models are trained and tested using the produced images. The nodules are segmented with the aid of the models U-Net and SegNet. These computer aided systems help doctors in early diagnosis of vision diseases leading to early-stage detection and faster recovery.

Table 1. Table of IOU Scores

Test Image Number	IOU Score (U-Net)	IOU Score (SegNet)
1	0.6790903017545704	0.6365927112445451
2	0.6447718292723801	0.6625801305109736
3	0.5740892206907846	0.5920574438436353
4	0.573264365078184	0.5889569787132367
5	0.5356950021002534	0.5792545229379039
6	0.5866107515780031	0.5331600010208813
7	0.41097745335553093	0.5118223953348539
8	0.45761387781450225	0.5015580963077432

9	0.564992488617618	0.5037232217856151
10	0.5834934630175601	0.5886502702604742
11	0.5097429216396013	0.5959501459394441
12	0.6413425020881613	0.6195627540330504
13	0.6145038189090989	0.5593122595599315
14	0.557326304823828	0.5655937225196098
15	0.20302959146119232	0.3869145836899708
16	0.63697112378596	0.6069155299031435
17	0.56949084415939	0.5252586515305362
18	0.6168109863522012	0.6036148123995249
19	0.5851502398920876	0.656654112921717
20	0.605185110813416	0.600061691781596
mean	0.5349122393430385	0.5034154148158003

Table 2. Table of Precision Scores

Test Image Number	Precision Score (U-net)	Precision Score (SegNet)
1	0.816282397571103	0.8592352741878994
2	0.7426445966206926	0.8312727388767162
3	0.645363362810145	0.7270375365716748
4	0.6883451482781158	0.7865678076778487
5	0.6247314890072808	0.7847476837218063
6	0.8731062550076593	0.912119612662637
7	0.45637951068927063	0.67057942542617
8	0.5650198818645806	0.7194679508342702
9	0.9095755159100104	0.9277139992135912
10	0.7099924417123255	0.8071893721280601
11	0.5678992643565113	0.7609701433798587
12	0.8023131148025351	0.8519416054179478
13	0.877060026832778	0.9122038444839401
14	0.6193890767278064	0.6801368586574706
15	0.20607206920728283	0.45766525609972564
16	0.8410281733607208	0.8857992723928138
17	0.8461054511164505	0.8827048069391623

18	0.7726906714724407	0.82136910827925
19	0.6381175703421407	0.7901325779136708
20	0.792014676638404	0.8329231754435648
mean	0.6184388954781285	0.7193163023025715

Output Images generated by both models (SegNet and U-Net):

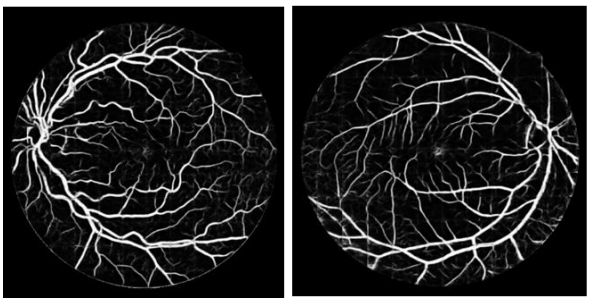

Fig 11: Outputs of SegNet Model

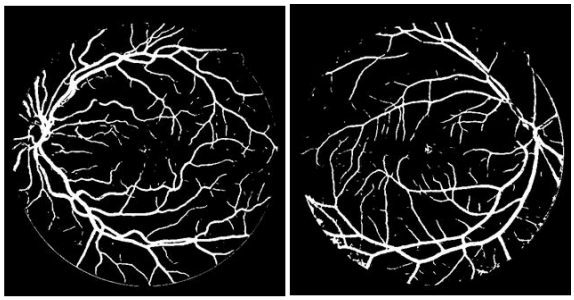

Fig 12: Outputs of U-Net Model

In the above figures 11 and 12 we can see that the first two images are the outputs of SegNet and the second two images are the output of the same image for U-Net. As we can observe the images in terms of visibility SegNet has more clear detection of optic nerves than U-Net.

SegNet has performed better compared to U-Net. The shortcomings of the detection and segmentation are overcome by these models, which improve the accuracy of properly detecting the nerves and blood vessels.

6. CONCLUSION AND FUTURE SCOPE

It has been noted that the variations between blood vessels and the retinal framework in retinal imaging is minimal, which makes it challenging to undertake careful observation and sound judgment while administering medical therapy. As a result, several deep learning approaches are used. For that purpose, the retinal images are taken from digital retinal images for the vessel extraction dataset which undergoes the data preprocessing and augmentation steps. Then the images are subjected to techniques of segmentation for extraction of blood vessels from images with help of SegNet and U-net algorithm which does pixel-wise classification of images. Later the extracted blood vessels are compared for better accuracy among both algorithms by using the ROC curve. The SegNet algorithm works better in comparison with the U-net algorithm i.e., SegNet gives around 94% of accuracy. Future work can be carried out on huge datasets and predict the diseases based on the segmented images.

REFERENCES

[1] Lalitha R, Aditi Padyana, “Detection of Diabetic Retinopathy in Retinal Fundus Image Using Yolo-Rf Model”, Ieee, 26-28 November 2021.

[2] Muhammad Zubair Khan, Yugyung Lee, “Retinal Image Analysis To Detect Neovascularization Using Deep Segmentation”, Ieee,15 July 2021.

[3] Jongwoo Kim, Loc Tran, “Retinal Disease Classification From Oct Images Using Deep Learning Algorithms”, Ieee, 18 October 2021.

[4] Davi Ribeiro Militani, Demóstenes Zegarra Rodríguez, Juan Casavílca Silva, Lunchakorn Wuttisittikulkij, Muhammad Saadi, Renata Lopes Rosa, and Sattam Al Otaibi, “Light-Field Imaging Reconstruction Using Deep Learning Enabling Intelligent Autonomous Transportation System”, IEEE Transactions On Intelligent Transportation Systems, Vol. 23, No. 2, February 2022.

[5] Kranthi Kumar Rachavarapu, Kaushik Mitra and Mohit Lamba, “Harnessing Multi-View Perspective of Light Fields for Low-Light Imaging”, IEEE Transactions On Image Processing, Vol. 30, 2021.

[6] Hang Su, Tao Yang, Nathan Crombez, Yassine Ruichek, Tomas Krajnik, Xiaofei Chang and Zhi Yan, “Raindrop Removal with Light Field Image Using Image Inpainting”, date of current version April 6, 2020.

[7] Alireza Sepas-Moghaddam, Fernando Pereira, Kamal Nasrollahi, Mohammad A. Haque, Paulo Lobato Correia, and Thomas B. Moeslund “CapsField: Light Field-Based Face and Expression Recognition in the Wild Using Capsule Routing”, IEEE Transactions On Image Processing, Vol. 30, 2021

[8] Gangyi Jiang, Mei Yu, Yeyao Chen, Yo-Sung Ho, and Zhidi Jiang, “Deep Light Field Super-Resolution Using Frequency Domain Analysis and Semantic Prior”, IEEE Transactions On Multimedia, Vol. 24, 2022.

[9] Gangyi Jiang, Mei Yu, Yeyao Chen, Yo-Sung Ho, and Zhidi Jiang, “Deep Light Field Super-Resolution Using Frequency Domain Analysis and Semantic Prior”, IEEE Transactions On Multimedia, Vol. 24, 2022.

[10] Yao Lu and Zijian Wang, “Light Field Image Super-Resolution via Mutual Attention Guidance”, date of publication September 14, 2021.

[11] Bin Chen, Jizhou Li, Lingyan Ruan, and Miu-Ling Lam “AIFNet: All-in-Focus Image Restoration Network Using a Light Field-Based Dataset”, IEEE Transactions On Computational Imaging, Vol. 7, 2021.

[12] Alan C. Bovik, Dae Yeol Lee, HyunsukKo, and Seunghyun Cho, “Quality Prediction on Deep Generative Images” IEEE Transactions On Image Processing, Vol. 29, 2020.

[13] Jianlin Zhao, Shan Mao, Zhenbo Ren, “An Off-Axis Flight Vision Display System Design Using Machine Learning” IEEE Photonics Journal, Vol. 14, April 2022.

[14] Chandra Nath Adak, Shaohua Yan, Xian Tao, Xin Zhang and Xinyi Gong “Deep Learning for Unsupervised Anomaly Localization in Industrial Images: A Survey”, IEEE Transactions On, Vol. 71, 2022.

[15] Dazheng Feng, Jie Chen, Long Sun, and Mengdao Xing “The Recognition Framework of Deep Kernel Learning for Enclosed Remote Sensing Objects”, date of current version July 13, 2021.

[16] Gangyi Jiang, Haiyong Xu, Mei Yu, Yeyao Chen, Yo-Sung Ho, “Deep Light Field Spatial Super-Resolution Using Heterogeneous Imaging”,2022.

[17] Gene Cheung, Huy Vu, Kazuya Kodama, RinoYoshida, Takayuki Hamamoto “Unrolling Graph Total Variation For Light Field Image Denoising”,2022.

[18] Ashwin Ashok, Kristin J. Dana, Macro Grusteser, MD Rashed Rahman, Narayan B. Madayam, Shubham Jain and T. V. Sethuraman, “Camera-Based Light Emitter

Localization Using Correlation of Optical Pilot Sequences “, date of current version March 9, 2022.

[19] https://www.kaggle.com/datasets/andrewmvd/drive-digital-retinal-images-for-vessel-extraction.

8. APPENDIX

SegNet Model :

```
def get_segnet(input_img, n_filters=16, dropout=0.5, batchnorm=True):

    c1 = conv2d_block(input_img, n_filters=n_filters*1, kernel_size=3,
batchnorm=batchnorm)
    p1 = MaxPooling2D((2,2)) (c1)
    p1 = Dropout(dropout*0.5)(p1)

    c2 = conv2d_block(p1, n_filters=n_filters*2, kernel_size=3, batchnorm=batchnorm)
    p2 = MaxPooling2D((2,2)) (c2)
    p2 = Dropout(dropout)(p2)

    u8 = Conv2DTranspose(n_filters*2, (3, 3), strides=(2,2), padding='same') (p2)
    u8 = Dropout(dropout)(u8)
    c8 = conv2d_block(u8, n_filters=n_filters*2, kernel_size=3, batchnorm=batchnorm)

    u9 = Conv2DTranspose(n_filters*1, (3,3), strides=(2,2), padding='same') (c8)
    u9 = Dropout(dropout)(u9)
    c9 = conv2d_block(u9, n_filters=n_filters*1, kernel_size=3, batchnorm=batchnorm)

    outputs = Conv2D(1, (1, 1), activation='sigmoid') (c9)
    model = Model(inputs=[input_img], outputs=[outputs])
    return model
```

U-Net Model:

```
def get_unet(input_img, n_filters=16, dropout=0.5, batchnorm=True):
    # contracting path
    c1 = conv2d_block(input_img, n_filters=n_filters*1, kernel_size=3,
batchnorm=batchnorm)
    p1 = MaxPooling2D((2,2)) (c1)
    p1 = Dropout(dropout*0.5)(p1)

    c2 = conv2d_block(p1, n_filters=n_filters*2, kernel_size=3, batchnorm=batchnorm)
    p2 = MaxPooling2D((2,2)) (c2)
    p2 = Dropout(dropout)(p2)

    c3 = conv2d_block(p2, n_filters=n_filters*4, kernel_size=3, batchnorm=batchnorm)
    p3 = MaxPooling2D((2, 2)) (c3)
    p3 = Dropout(dropout)(p3)

    c4 = conv2d_block(p3, n_filters=n_filters*8, kernel_size=3, batchnorm=batchnorm)
    p4 = MaxPooling2D(pool_size=(2, 2)) (c4)
    p4 = Dropout(dropout)(p4)

    c5 = conv2d_block(p2, n_filters=n_filters*4, kernel_size=3, batchnorm=batchnorm)

    # expansive path
```

```
u6 = Conv2DTranspose(n_filters*8, (3, 3), strides=(2, 2), padding='same') (c5)
u6 = concatenate([u6, c4])
u6 = Dropout(dropout)(u6)
c6 = conv2d_block(u6, n_filters=n_filters*8, kernel_size=3, batchnorm=batchnorm)

u7 = Conv2DTranspose(n_filters*4, (3, 3), strides=(2, 2), padding='same') (c6)
u7 = concatenate([u7, c3])
u7 = Dropout(dropout)(u7)
c7 = conv2d_block(u7, n_filters=n_filters*4, kernel_size=3, batchnorm=batchnorm)

u8 = Conv2DTranspose(n_filters*2, (3, 3), strides=(2,2), padding='same') (c5)
u8 = concatenate([u8, c2])
u8 = Dropout(dropout)(u8)
c8 = conv2d_block(u8, n_filters=n_filters*2, kernel_size=3, batchnorm=batchnorm)

u9 = Conv2DTranspose(n_filters*1, (3,3), strides=(2,2), padding='same') (c8)
u9 = concatenate([u9, c1], axis=3)
u9 = Dropout(dropout)(u9)
c9 = conv2d_block(u9, n_filters=n_filters*1, kernel_size=3, batchnorm=batchnorm)

outputs = Conv2D(1, (1, 1), activation='sigmoid') (c9)
model = Model(inputs=[input_img], outputs=[outputs])
return model
```