Detection and Mosaicing through Deep Learning Models for Low-Quality Retinal Images

Master in Electric and Electronic Engineering

Tales Veríssimo Souza Correia

Leiria, March of 2023

Detection and Mosaicing through Deep Learning Models for Low-Quality Retinal Images

Master in Electric and Electronic Engineering

Tales Veríssimo Souza Correia

Dissertation under the supervision of Professor Paulo Jorge Simões Coelho (Polytechnic of Leiria), and Professor António Manuel Trigueiros da Silva Cunha (UTAD)

Leiria, March of 2023

Originality and Copyright

This dissertation/project report is original, made only for this purpose, and all authors whose studies and publications were used to complete it are duly acknowledged.

Acknowledgments

This work exists solely thanks to the people who believed in me.

I would like to express my deepest gratitude to Doctor Paulo Jorge Simões Coelho and Doctor António Manuel Trigueiros da Silva Cunha for their unwavering support, wisdom, and patience in guiding me to develop this work.

I want to thank my mother, grandmother, and father, who played an important role in supporting me from overseas, as did my entire family. I could not have done it without them. I love you all.

I cannot express how grateful I am to Alexandra and Alda, who welcomed me as a family member and supported me throughout this journey, making me feel at home. They hold a very special place in my heart.

I also want to thank my colleagues and friends Marcella, Thierry, Tiago, and Vitor for their unwavering support in difficult moments and for making everything seem light and easy at times.

Abstract

Glaucoma is a severe eye disease that is asymptomatic in the initial stages and can lead to blindness, due to its degenerative characteristic. There isn't any available cure for it, and it is the second most common cause of blindness in the world. Most of the people affected by it only discovers the disease when it is already too late.

Regular visits to the ophthalmologist are the best way to prevent or contain it, with a precise diagnosis performed with professional equipment. From another perspective, for some individuals or populations, this task can be difficult to accomplish, due to several restrictions, such as low incoming resources, geographical adversities, and travelling restrictions (distance, lack of means of transportation, etc.). Also, logistically, due to its dimensions, relocating the professional equipment can be expensive, thus becoming not viable to bring them to remote areas.

In the market, low-cost products like the D-Eye lens offer an alternative to meet this need. The D-Eye lens can be attached to a smartphone to capture fundus images, but it presents a major drawback in terms of lower-quality imaging when compared to professional equipment.

This work presents and evaluates methods for eye reading with D-Eye recordings. This involves exposing the retina in two steps: object detection and summarization via object mosaicing. Deep learning methods, such as the YOLO family architecture, were used for retina registration as an object detector. The summarization methods presented and inferred in this work mosaiced the best retina images together to produce a more detailed resultant image.

After selecting the best workflow from these methods, a final inference was performed and visually evaluated, the results were not rich enough to serve as a pre-screening medical assessment, determining that improvements in the actual algorithm and technology are needed to retrieve better imaging.

Keywords: Glaucoma, Deep Learning, Mosaicing.

Contents

List of Figures

List of Tables

1. Introduction

Human eyes play a key role in daily life, allowing humans to see and perceive things all around their reach. Alongside the other senses, humans can feel and comprehend all kinds of phenomena in the world. Eyesight shares a great deal in most basic daily activities (such as reading or writing for example), or in other cases, activities that demand more focus (perceiving art, aiming, etc.).

Being such an essential part of the human body, maintaining the eye in good health is extremely important, and the best way to keep track of its health is to have regular visits to the ophthalmologist. One of the most common eye checkups performed is the retinal fundus capture, which consists of the exposition of the back of the eye, where the macula and fovea are two of the principal areas where happens actual formation of an image from the light incidence [1].

With the recent advances in video summarization and machine learning, with a simple video recording taken from a smartphone using an amplifying lens, it is possible to convert it to a single image with the desired features to point out what could be an indication of glaucoma. The summarization process of condensing or extracting the key information and content from a video, representing the video's essential aspects in a concise and manageable form. It aims to provide a shorter version or overview of the original video, enabling users to quickly grasp the main points, events, or highlights without having to watch the entire video. This process doesn't intend to substitute the diagnosis process in a professional fundus machine with specialist assistance. However, it allows a quick summarization based on the captured video to perform a pre-screening that may alert the individuals to seek professional assistance.

1.1. Objectives

This work intends to develop a methodology capable of performing a retina summarization from a single video recording, expanding the video frames into a whole new image with detailed information that cannot be perceived during a recording. This unique image will serve as a pre-screening for a medical professional to evaluate the status of the retina and assess the diagnosis of glaucoma incidence. The steps to do so are:

- Detect retina with available methods for a video recording [2];
- Apply summarization techniques that use retinal images obtained from the recording and retrieve a full image of the exposed retina.

1.2. Dissertation Structure

This dissertation is divided into seven chapters. Chapter one provides an overview of the work, including the area of intervention, motivations, and problems to be addressed. It also briefly presents the main goals and contributions of the dissertation.

Chapter two delves into the biological background, specifically eye anatomy, glaucoma disease, and retinal imaging in general and briefly.

The third chapter covers technical concepts and the current state-of-the-art, including an overview of machine learning, object detection, and summarization techniques.

Chapter four describes the methodology used to obtain the results presented in the dissertation. It includes the pipeline for preparing data, inputting them to a detection network method, and feeding the final summarization method with only images with the object of interest.

Chapter five presents the results of the several explored subjects. This chapter analyzes the results from the proposed method and the object detection networks and provides visual results of summarization methods.

The sixth chapter six discusses the achievements of the dissertation and outlines future directions for further improvements and developments.

The last chapter depicts the publications produced because of this dissertation.

2. Initial Concepts and Background

2.1. The human eye

Human eyes take an important role in daily life, making it possible for humans to see and perceive things all around their reach. Alongside the other senses, humans can feel and comprehend all kinds of phenomena in the world. Eyesight shares a great deal in most basic daily activities, such as reading and watching television, or in some other cases, activities that demand more focus, such as perceiving art, aiming, and some athletic-related sports, such as boxing.

To better understand how the human eye works, this chapter intends to show its functionality alongside its anatomy and how glaucoma affects a person's eyesight. Figure 1 presents a simplified schematic of the eye and its anatomy. The first part of the eye that receives the reflected light is the cornea, which focuses the light through the pupil and regulates the amount of light that will reach the retina fundus.

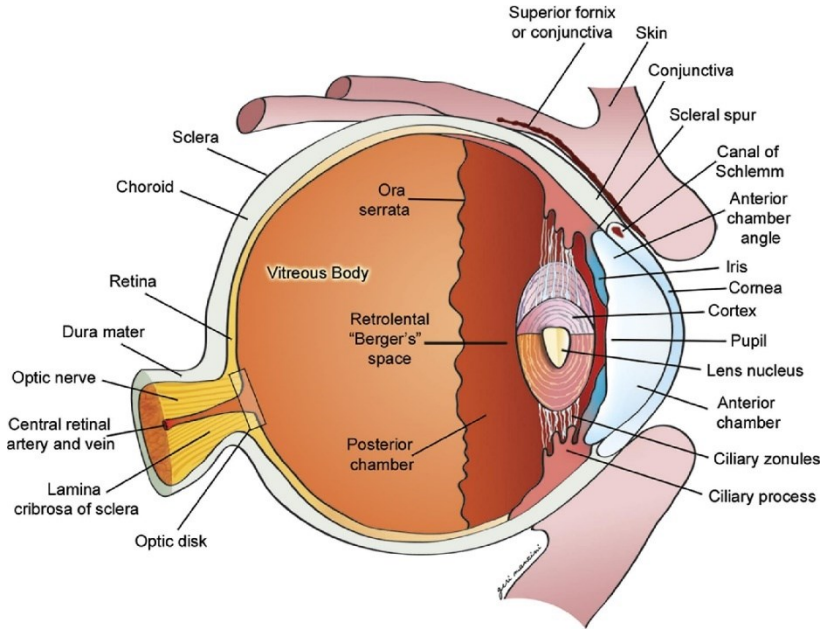


Figure 1: Detailed Human Eye – obtained from [3].

To create an image, the eye receives the light reflection and sends the information to the brain through the optic nerve to convert it to an actual image [4]. The retina converts light into electrical impulses, but it doesn't have the same sensibility in all its extension. As a

result, only a small area is responsible for transmitting the information about where the eye is focusing. This area is called the macula.

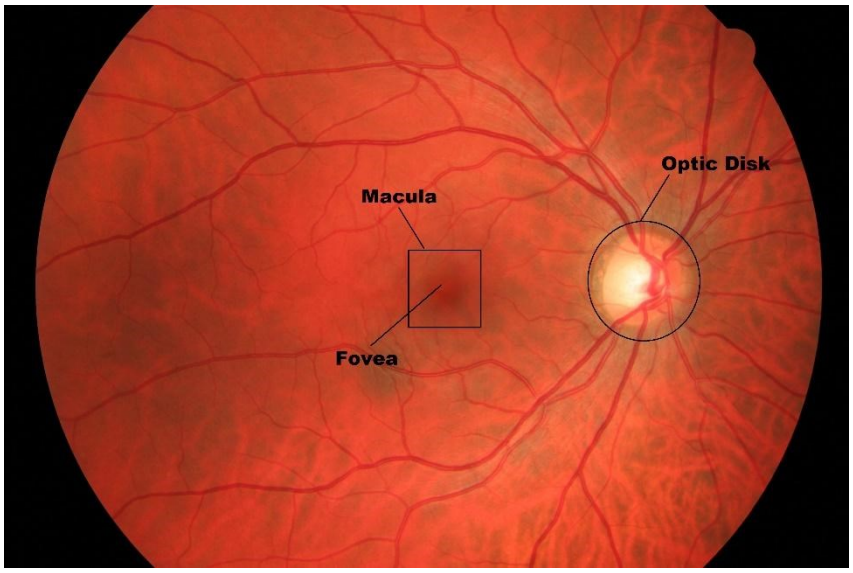


Figure 2: Fundus image with Macula, Fovea, and Optic Disk highlighted – obtained from [5].

The Figure 2 represents a photography from a public dataset that shows a fundus image with a clear perception of the main areas of the retina. The larger area the receives light reflections creates the peripherical vision, a blurry image not fully perceived by the optic nerve. The macula and fovea are two essential structures in the retina of the eye responsible for our central vision and the ability to see fine details. The macula has a center that contains a great density of light receptors responsible for converting electrical impulses to the optic nerve. This area is known as fovea, where the higher concentration of receptor cones are located, making it extremely important to maintain it healthy to guarantee no significant impairment [4]. Both the macula and fovea can be affected by several eye conditions, including macular degeneration [6], diabetic retinopathy [7], and glaucoma [8]. These conditions can cause vision loss or distortion and may require medical treatment to prevent further damage to the retina.

2.2. Glaucoma Disease

While vital for most activities and quintessential sense for humans, the human eye must be treated similarly to any other organ in the body. Having precautions and constant care is essential to maintain it working properly, like preventing damage from direct sunlight

absorption. However, some other issues may not be simple as protecting the eyes from the sun.

This study focuses on glaucoma as a disease that can damage the optical disc in a painless way, is not noticeable within its early stages, and is usually only detected with visual testing of the eye fundus [9]. Some factors that can intensify the degeneration caused by the disease are related to genetic disposition, myopia, diabetes, and hypertension. It is the second leading blindness cause in the world. The most common variation of glaucoma is the open-angle, responsible for around three-quarters of the blindness from glaucoma-related [9]. Figure 3 shows the difference between normal vision and open-angle glaucoma affected.

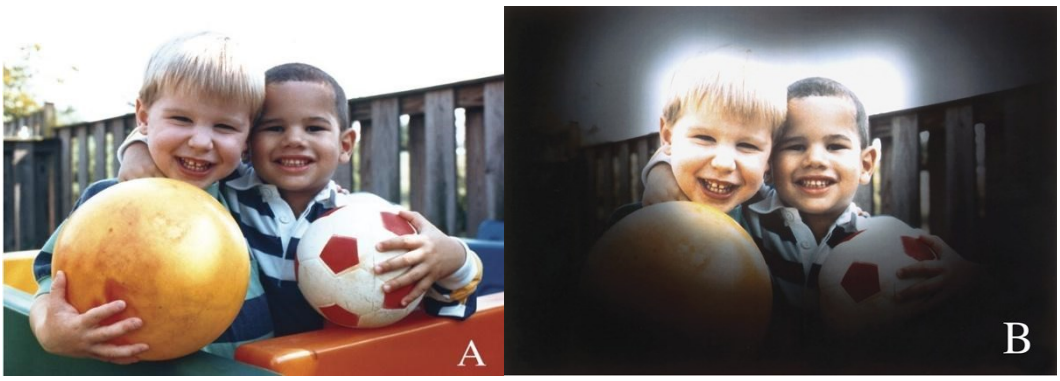


Figure 3: Comparison of a healthy eyesight (A) and glaucoma-affected one (B). – obtained from [10].

As a non-curable disease, it is vital to seek medical assistance as soon as any signal of vision impairment is present. It can be prevented or delayed in early treatment. As of today, there is a campaign running by the World Health Organization called The World Report on Vision that seeks to create awareness and increase investment in making efforts to mitigate global blindness, the main goal is to set feasible challenges to make eye care become more knowledgeable for those who suffers more from the disease, usually on the low-income countries and people with difficult access to medical care [9].

2.3. Visual Image Analysis

Although some processes rely on precise equipment and can make an accurate medical analysis of the fundus of the retina, in this study it is intended to use data from the alternative methods, such as the newly devices that can be coupled on the mobile phones and enhance the cameras to be able to record the fundus of the retina. Some of them are iExaminer, D-Eye, Peek Retina, or iNview [11].

The D-Eye lens, illustrated in Figure 4, was used to gather data to achieve the objective of this study, being low-cost and portable, it is an ideal device to perform fundus capture, with the backlash of losing precision on the record and losing detailed information depending on the resolution definitions of the used camera [2]. Although, new cellphone devices may be more suitable for this task than the one in Figure 4.

Figure 4: D-Eye device coupled with a cellphone – obtained from [2].

In today state of the art, it is already possible to extract the area of interest, that is, the fundus retinal image, due to an implementation realized in a previous work, and with that data it is intended to enhance the previous algorithm with an implementation using an image stitching technic to perform a glaucoma image detection based on machine learning, specifically deep learning emphasis using pictures from public and private databases, that will be later explained in next topic in this work.

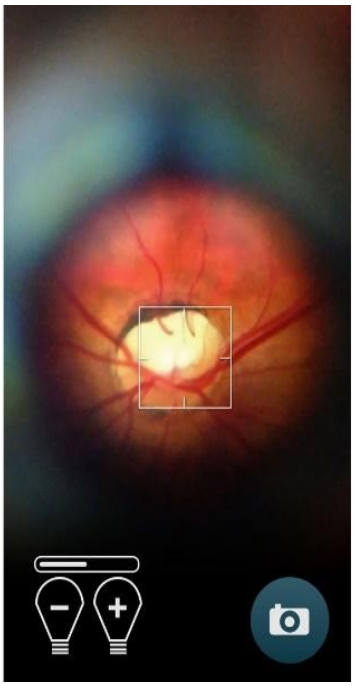


Figure 5: D-Eye capture application – obtained from [12].

Figure 5 shows an image from the retinal fundus after a record capture. Although, compared with a professional image capture with clinical equipment, it is notorious the difference between them. This is one of the points of applying a technic capable of stitching the frames of a recorded video to obtain the full spectrum of the retina, trying to mitigate quality and information loss.

3. State of the Art and Fundamentals

The advent of Video Summarization came from the need to gather important content from a significant amount of data, extracting the desired information from previously set keyframes that match the objective. With that in mind, selecting the keyframes and the features allied with the method to retrieve this information is crucial. Over the past years, many methods and techniques have been introduced, and some of them will be shared later in this document, many of them being derived from machine learning, more specifically, deep learning.

This chapter will present some topics relevant to this work's objectives, focusing on matters that might help with the objective. The reviewed studies with the selected inclusion criteria were searched in Science Direct and IEEE Xplore databases. The following research terms were used to research this systematic review: “video summarization”, “glaucoma detection” and “object detection”. There was a total of 42 reviewed studies, and after more criterion selection, 18 studies were selected in the final analysis. The research was performed from 10 April 2022 to 15 January 2023.

As presented in Figure 6, were identified 42 studies from the selected sources, without duplicated papers. Two additional records were added to the results, gathered from different queries to the databases. After analyzing each research article’s metadata, namely the title, abstract, and keywords, 6 studies were excluded from the analysis because they were medically specific and did not directly relate to evaluating the video summarization or glaucoma detection. The full text of the remaining 38 articles was assessed considering the inclusion criteria, and consequently, 20 articles were excluded. Finally, the remaining 18 papers were examined and included in qualitative and quantitative syntheses.

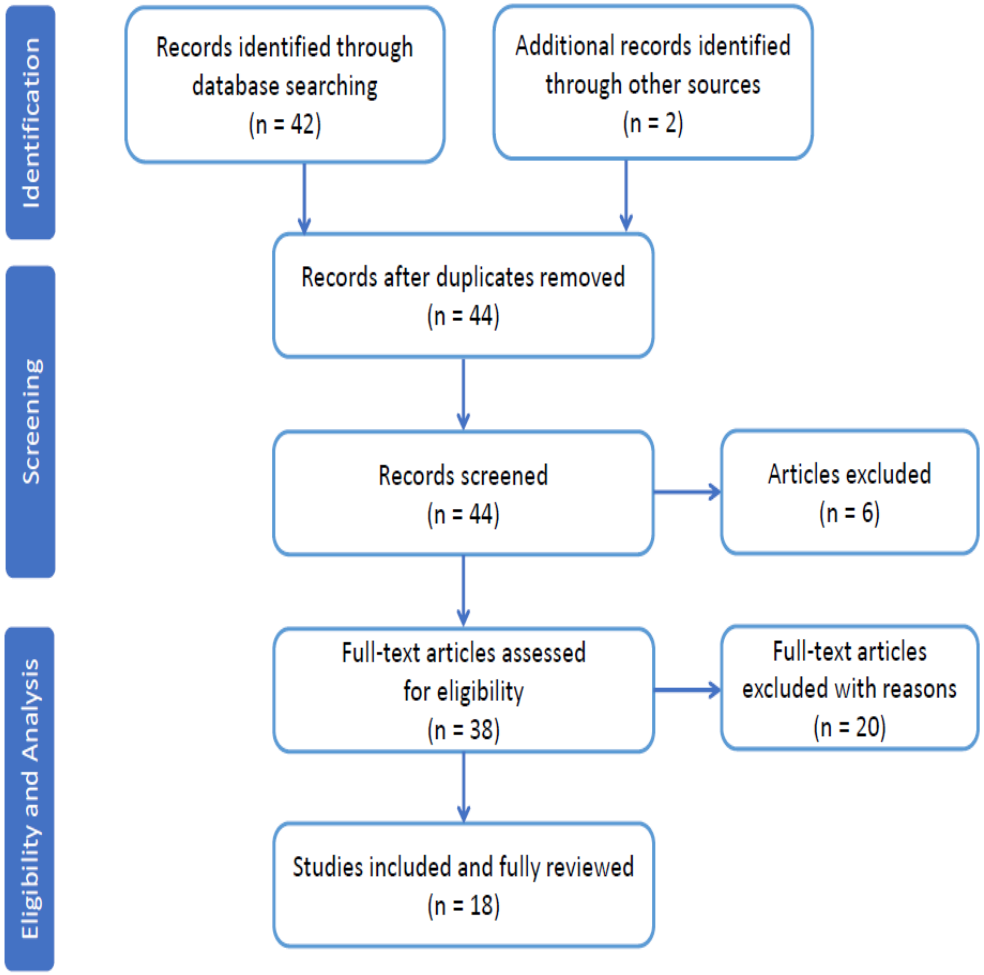


Figure 6: Flow diagram of the selection of the papers.

3.1. Machine learning fundamentals

In classical programming, the common way to build a functional algorithm is through programming that respects an order of a set of rules. In this case, the output is an expected result. With the machine learning approach, the flow of programming changes in the sense of how the algorithm works, inputting data and results (labeled data) to make the computer perform its comprehension, forming its own rules. It is then possible to change the input data and produce assets for those rules to test new data [13], as presented in Figure 7.

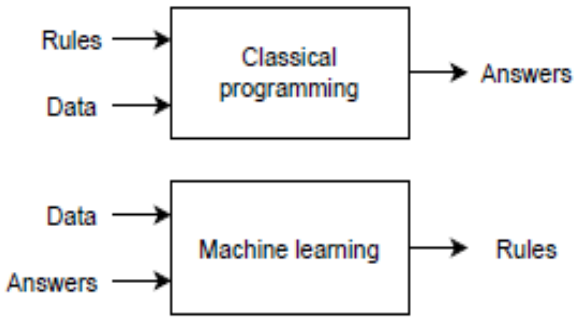


Figure 7: Classical x Machine Learning programming comparison – obtained from [13].

It is important to state that with the actual computational power and data available, some more complex technics can be implemented, like Neural Networks (NNs) and deep learning [14].

Neural Networks consist of methods of function activation, and the name derivates from how cerebral neurons work, making a chain reaction from synapses around the brain when an input triggers the initial neuron [14]. In machine learning, NNs are composed of input, output, and hidden layers, with each layer being connected to the subsequential one. Each one has a weight associated with it, which is essential to estimate the output and finally obtain a loss score between the prediction and the true target to adjust the weights of the layers based on this loss. This discrepancy, referred to as the distance score, can be calculated by comparing the predictions made by the neural network with the true targets. Once the distance score is obtained, the weights are then adjusted using an optimization algorithm such as Gradient Descent, which uses a backpropagation algorithm based on the loss score. In essence, learning entails determining the appropriate values for the layer weights [13]. Figure 8 briefly depicts the explanation.

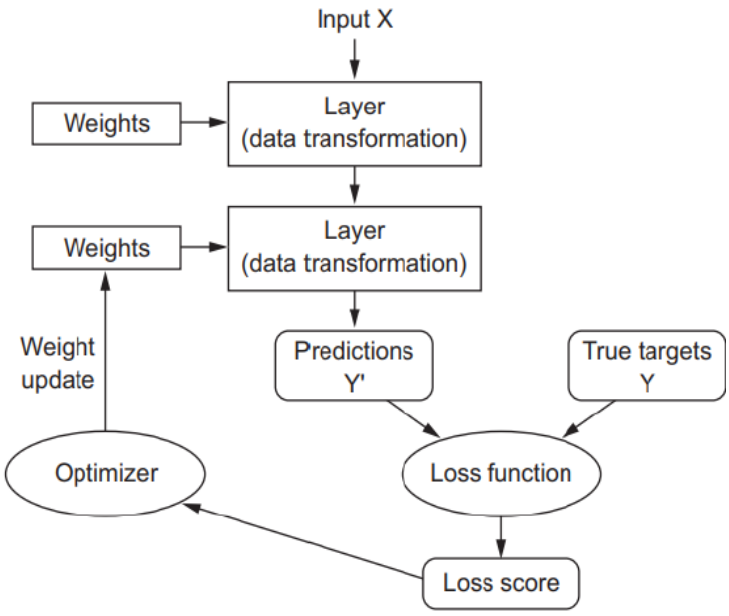


Figure 8 Neural Network workflow – obtained from [13].

The deep learning concept would be a neural network but with lots of layers, creating a sensation of depth from the input to the output. Due to this architecture, it is possible to use a pre-trained model in a small image dataset, if it was previously trained on a large dataset (this process is named as Fine-tuning). It is possible, for example, to use a trained model to perceive small objects in a video and then reuse it on another dataset to identify some key features from the last model [13].

Some examples of deep learning methods are Convolutional Neural Networks (CNNs), very useful for gesture recognition [15], speech recognition [16], and, as will be presented in this document, video summarization [17]–[32]. Some of the reviewed works use advanced deep learning methods [33]–[35], and apply 3D-CNN techniques, which are ideal when treating high volumetric data.

In the convolutional layer, a filter is moved across the input image to identify patterns and key features, such as horizontal or vertical lines, for object classification. The network initially focuses on small details and features in the first convolutional layers, which are then amplified for more complex objects in subsequent layers. Essentially, the deeper the network, the greater its capability to detect complex objects [14].

Typically, an activation function is included after each convolutional layer. Without this layer, the network would only be capable of performing linear classification, since convolutional layers involve multiplication operations. Due to their linearity, the

combination of multiple linear functions results in another linear function. Activation functions add non-linear capabilities to the network [17].

3.2. Video Summarization

With the new machine learning methods and the huge amount of data gathered from these past years, some works show promising results. This section presents the current state of the art from the most recent studies of Video Summarization.

Some of the known technics for video summarization rely on a set of tasks that the machine needs to perform to summarize relevant parts from video frames, considering the key features that have been selected to be present in the final set of image [31]. Figure 9 shows the basic structure of how a video summarization algorithm works.

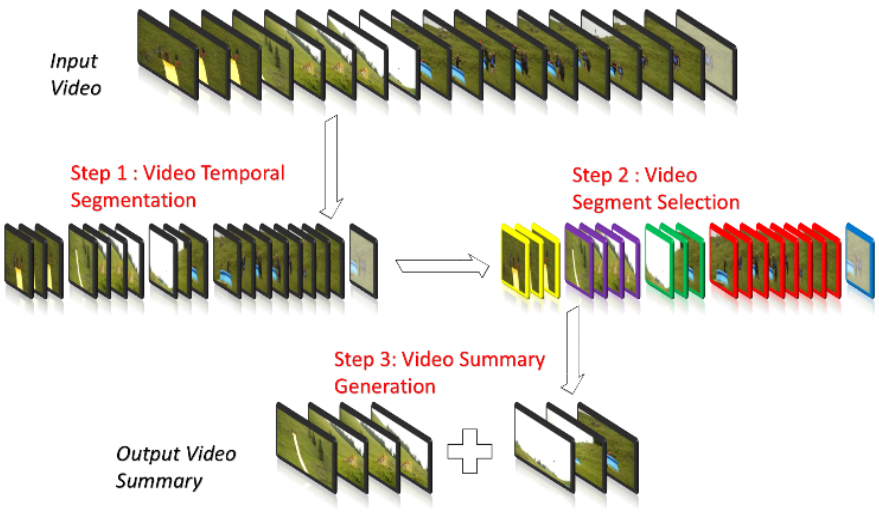


Figure 9: Video summarization algorithm structure – obtained from [36].

The input video is usually segmented frame by frame. Then, based on what features the algorithm was trained to summarize, it will select the best shots or frames as output, depending on the chosen method or criteria.

Long Short-Term Memory (LSTM) is a modern Recurrent Neural Network (RNN) useful to capture temporal dependencies between frames. Still, it has the issue of only being capable of handling short video segments within a range from 30 to 80 frames. To overcome this, the method proposed by Lin et al. [15] employs a three-dimension Convolutional Neural

Network (CNN) to extract features allied with a Deep Hierarchical LSTM Network with an attention mechanism for Video Summarization (DHAVS) framework. This method was applied in SumMe [31], and TVSum [32] datasets and compared it with recent results from other works with similar approaches. The F-Score obtained from DHAVS in SunMe was 45.2% and in the TVSum was around 60%.

In contrast to what was presented in [15], Zhao et al. [16] claims that RNNbased methods neglect global dependencies and multi-hop relationships between frames. To overcome that situation, a Hierarchical Multimodal Transformer (HMT) method is proposed to summarize lengthy videos by hierarchically separating the layers of dependency between shots, thus reducing memory and computational consumption. The metrics were also like the previously mentioned work and the datasets, where this method achieved an F-Score of 44.1% on SunMe and 60.1% on TVSum.

An attention mechanism is proposed to work with a dual-path attentive network by Liang et al. [33] to overcome the systematical stiffness of the recurrent neural networks. It was stated that their method improves the processing time and reduces the computational power. At the same time, it is possible to train the model in parallel, thus being scalable in more extensive datasets. The results for F-Score from training and testing in SumMe and TVSum, with 51.7% and 61.5%, respectively, were higher than what was presented in [15] and [16].

Feng et al. [19] proposed a video summarization technique that uses two different feature extraction that converts frame-level features into shot-level features based on CNN, named Video Summarization with netVLAD and CapsNet (VCVS). Their method improved computational and hardware work while using a feature fusion algorithm with capsule neural (CapsNet) networks to enhance the video features. The F-score presented is 49.5% on SumMe and 61.22% on TVSum.

Some video summarization methods [20]–[22], can only extract the content of static images from those videos. Huang et al.[23] comes with a method to do both video and motion summarization, relying on transitions effects detection (TED) for automatic shots segmentation, using CapsNet, and a self-attention method to summarize the results. The scores for this method were 46.6% on SumMe and 58% on TVSum.

A multiscale hierarchical attention approach is proposed by [24] for pattern recognition using intra-block and inter-block attention methods, exploring short and long-range temporal

representations of a video. This method was developed because the attention mechanism is easier to implement than RNN. The achieved results are 51.1% on SumMe and 61.0% on TVSum.

Chai et al.[25] propose a graph-based structural analysis in a three-step method that can detect the differences in continuous frames and establish the correct video summarization. For the tests, they used VSUMM and Youtube datasets [26], in which, compared to similar analyzed works, they achieved an F-score of 67.5% and 56.7%, respectively.

Another interesting approach, presented by Hussain et al.[34]. A survey on multi-view video summarization (MVS) claims that this technique is not addressed regularly as other mainstream summarization methods. Gathering the video records from simultaneous cameras and with different angles, the paper reviews the recent and most significant works that englobe MVS.

A self-attention binary neural tree (SABTNet) method is proposed by Fu et al.[35] to perform video summarization, subdividing the video and then extracting it to shot-level features, altogether with a self-attention imbued. This work is the first to introduce such an approach. Similarly, to the previously presented, the method was tested on SumMe and TVSum datasets, with F-scores of 50.7% and 61.0%, respectively.

The work from Harakannanavar et al.[17] an approach based on ResNet-18, a CNN with eighteen layers, was used with kernel temporal segmentation (KTS) for the videos to create a temporally consistent summary. This method was benchmarked with the usual datasets, SumMe and TVSum, obtaining 45.06% and 56.13% on F-scores, respectively.

An interesting method is proposed in [18], a CNN with a Global Diverse Attention (SUM-GDA) mechanism. It implies that the GDA provides relations within pair-frames and those pairs with all others in the video, stating that it overcomes the long-range issue from RNN models. They performed tests with supervised, unsupervised, and semi-supervised scenarios, with the usual datasets with the addition of VTW dataset [37]. As expected, the F-scores obtained from the tests were higher in the supervised training, in which was obtained 52.8% on SumMe, 61% on TVSum, and 47.9% on VTW.

Table 1 presents a resumed overview of the previously mentioned methods.

Table 1: State-of-the-art scores comparison

Method	F-Score (%)		Remarkables
	SumMe	*TVSum*	
3D-CNN with DHAVS [15]	45.2	60	Employs LSTM for long videos
HMT [16]	44.1	60.1	Reduces computational consumption by separating dependency between shots
Dual-Path Attentive Network [33]	51.7	61.5	It improves computational consumption, improves process time, and model can be trained in parallel
VCVS with CapsNet [19]	49.5	61.22	Improves computational and hardware work
TED with CapsNet [23]	46.6	58	Summarization can be done in video and motion, not only static images
Multiscale Hierarchical Attention [24]	51.1	61	Captures short and long-range dependencies, also can perform motion detection
SABTNet [35]	50.7	61	Shot-level segmentation and feature extraction
ResNet-18 with KTS [17]	45.06	56.13	Temporal consistent summarization
SUM-GDA [18]	52.8	61	Provides pair-frames relations within all video

It is relevant to imply that all those works are the most recent in terms of video summarization, making them a starting point as testing approaches to glaucoma detection, as for the next reviewed papers, they are more oriented to methods that have direct impact on this matter.

3.3. Glaucoma Detection

A multimodal model to automatically detect glaucoma was proposed by [38] to combine deep neural networks focused on macular optical coherence tomography (OCT) and color fundus photography (CFP). Their dataset consisted of the UK Biobank dataset [39] with 1193 healthy and 1283 healthy and glaucomatous frames, respectively. The OCT-developed model was based on Densenet with MRSA initialization. For the CFP model, transfer learning with the Inception Resnet V4 model, pre-trained on ImageNet data was used. Then a gradient-boosted decision tree was introduced with XGBoost to create four separate baseline models (BM1 to BM4), enhancing specific features they wanted to highlight. After

testing the model, the authors stated that mixing demographic and clinical features boosted the accuracy of diagnosis, obtaining around 97% of precision in correct results.

Trying to solve the issues of overfitting and extensive sets of data for training, Nayak et al.[40] proposed a method with a feature extraction called evolutionary convolutional network (ECNet) to perform automated glaucoma detection. They also applied an evolutionary algorithm named real-coded genetic algorithm (RCGA), which maximizes the inter-class distance and minimizes the intra-class variability to optimize the weight of the layers. Then it is applied a set of classifiers, such as K-nearest neighbor (KNN), extreme learning machine (ELM), backpropagation neural network (BNN), support vector machine (SVM), and kernel ELM (K-ELM), to enhance the model. They used a dataset from Kasturba Medical College, Manipal, India, using a Zeiss FF 450 fundus camera, containing 1426 retinal fundus images, 589 healthy, and 837 with glaucoma.

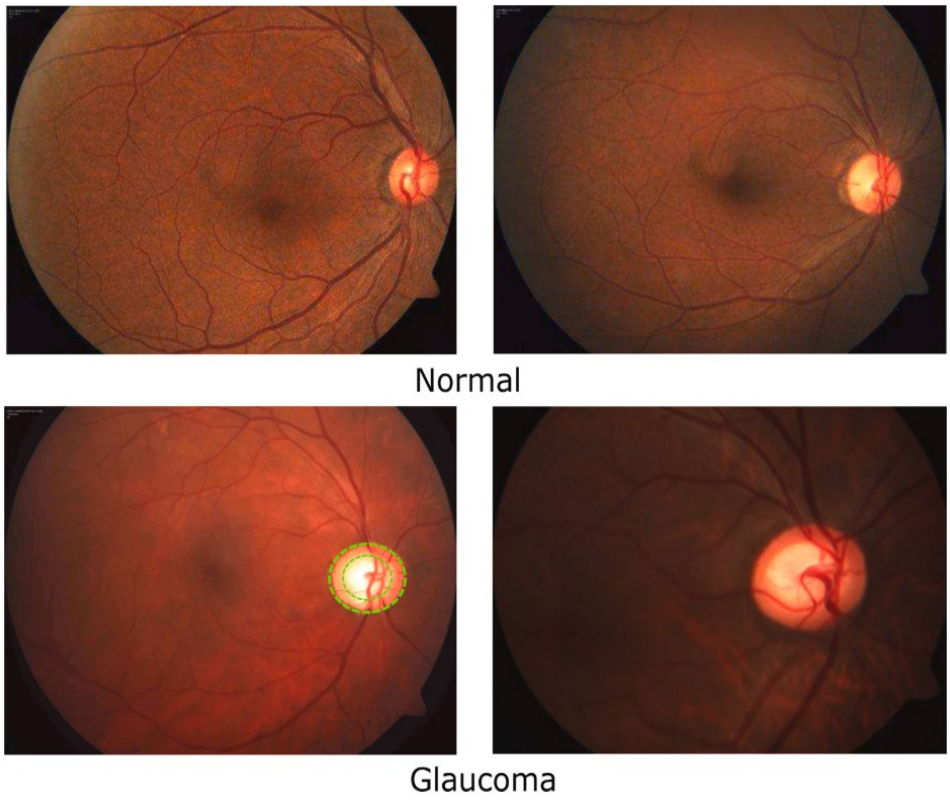


Figure 10: Optic disc and optic cup comparison – obtained from [2].

In Figure 10, the green circle's segmentation shows the CDR discrepancy in size between those pictures. As for the results, the classifier that scored the best was SVM, with 97,2% of obtaining the correct diagnosis.

Li et al.[41] proposed an attention-based CNN for glaucoma detection (AG-CNN), using large-scale attention-based glaucoma (LAG) database. These large-scale fundus images have 5824 positive and negative glaucoma images obtained from Beijing Tongren Hospital. When this work was proposed, no other work incorporated human attention in medical recognition. These attention maps were obtained through a simulated eye-tracking experiment and incorporated into the LAG dataset. The method had an F-Score of 95.1%.

An artificial intelligence technique method presented by Venugopal et al.[42] relies on Phase Quantized Polar Transformed Cellular Automaton (PQPT-CA) for training on fundus images for glaucoma detection in early stages, using the ACRIMA database [43], with 705 fundus images within glaucoma and normal ones. This approach was chosen because of the recent results in image processing, slightly changing the existing architecture of the automaton to fit the proposed method, they could use it to extract the features boosting the accuracy by around 24%, being 21% faster, and reducing the false positive results in 54%.

As for Zulfira et al.[44], they proposed a method that uses the classical parameter cup-to-disc ratio (CDR) allied with peripapillary atrophy (PPA) to enhance the precision of classification. They use an active contour snake (ACS) to segment the desired areas to calculate the CDR and Otsu's segmentation and threshold technique to acquire the PPA, and then the features are extracted with a grey-level co-occurrence matrix (GLCM). Dynamic ensemble selection (DES) is applied to classify glaucoma to make the final discrimination. The model was evaluated with three different databases where the ground truth was provided by ophthalmologists. Applying this method to RIM-ONE dataset [45] it was obtained an accuracy score of 96%.

Proposing the usage of 3D spectral-domain OCT, claiming that are potential information in these scans to help in glaucoma detection, Garcia et al.[46] brings a new perspective by presenting a method that uses the spatial dependencies of the features extracted from a B-scan of an OCT. Their database was composed of 176 healthy and 144 ill eyes. The method employed included a slide-level feature extractor and a volume-based predictive model. They also used an LSTM network to combine the recurrent dependencies that will be further mixed into the latent space to provide a holistic feature vector generated by the proposed method of the sequential-weighting module (SWM). The best results were achieved using RAGNet-VGG16 architecture with an accuracy of 92%.

Gupta et al.[47] comes with a robust network to detect glaucoma in retinal fundus images based on CDR. They used two main modules, CLAHE, to improve the retinal images and the second to find the CDR after the image segmentation based on EfficientNet and U-net. They performed the tests in DRISHTI-GS and RIM-ONE datasets. The result for this method using the Dice coefficient for similarity was 96%, and the pixel accuracy for the optic disc and cup was 96.54% and 96.89%, respectively.

Table 2 presents a resumed overview of the previously mentioned methods.

Table 2: State-of-the-art scores comparison

Method	Dataset / N. of Images	Accuracy
OCT & CFP & Systemic & Ocular Model [38]	UK Biobank / 2476	0.97
RCGA with SVM [40]	Kasturba Medical College / 1426	0.972
Full AG-CNN [41]	LAG / 5824	0.951
DES-MI [44]	RIM-ONE / 250	0.96
RAGNet-VGG16with SWM [46]	Private Dataset / 905	0.92
CLAHE with EfficientNet + U-net [47]	RIM-ONE / 766	0.966

3.4. Object detection

During this study, a tool that has proven to be important in object detection, specifically YOLO (You Only Look Once) [48]. In a similar work, this tool had already been used, and it showed to be capable of performing retina identification with a reasonable margin of precision [49]. Now with new versions of this detector, the objective is to compare them and find which is more efficient for the proposed goal.

YOLO is an object detection algorithm that uses a convolutional neural network (CNN) to detect objects in images and videos [48]. The algorithm is designed for real-time object detection and is known for its high speed and accuracy. Figure 11 illustrates how it works by dividing an image into a grid of cells, and each cell is responsible for predicting the bounding boxes of objects within its region. For each cell, the algorithm predicts the coordinates of the bounding box, the object class, and the confidence score of the prediction. The center coordinates represent the bounding box coordinates (x, y) and the width and

height of the box (w, h). The confidence score represents the probability that the prediction is correct and is used to filter out false positive detections.

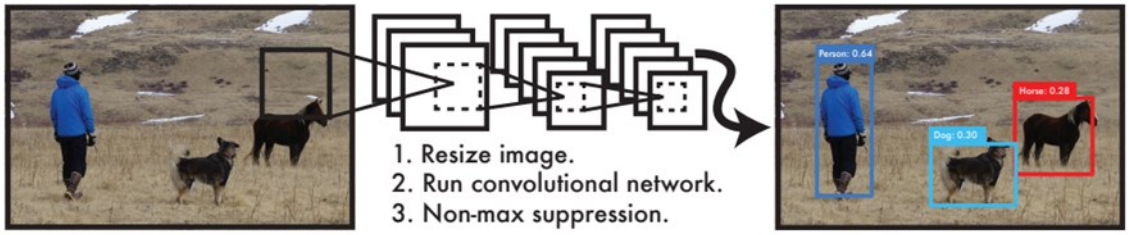


Figure 11: Yolo detection schematic – obtained from [48].

The YOLO algorithm is optimal when trained on a large dataset of images, and it learns to predict the object class, bounding box coordinates, and confidence scores for each box. In the inference step, the algorithm applies a non-maximum suppression (NMS) algorithm to filter out overlapping bounding boxes and keep only the most confident detections. The NMS algorithm compares the confidence scores of the bounding boxes and removes the boxes with lower scores if they overlap with boxes with higher scores.

The system divides an image into a grid with SxS size; if a recognizable object is within one of the grid cells, that cell identifies that object. The model predicts B bounding boxes and corresponding confidence scores for each grid cell. The confidence scores indicate the model's confidence in the existence of an object within the box and the predicted box's accuracy. This confidence is formally defined as the product of the probability of an object being present and the intersection over union (IOU) between the predicted box and the ground truth. The confidence score should be set to zero if there is no object in the cell. On the other hand, if an object is present, the confidence score should equal the IoU between the predicted box and the ground truth.

Also, for each grid cell, the model also predicts C conditional class probabilities, which represent a specific class's probability given an object's presence. These probabilities are only calculated for cells that contain an object. It's worth noting that only one set of class probabilities is predicted per grid cell, regardless of the number of boxes B. During the testing phase, the class probabilities and the confidence scores for each box are multiplied together described in the Equation (1 :

$$\Pr(Class_i|Object) * \Pr(Object) * IoU\ \frac{truth}{pred} = \Pr(Class_i) * IoU\ \frac{truth}{pred} \quad (1)$$

This way, it gives the confidence scores for a specific class for each box. It fits both the probability and the precision of the predicted box for that object. Figure 12 shows how the model runs as described before.

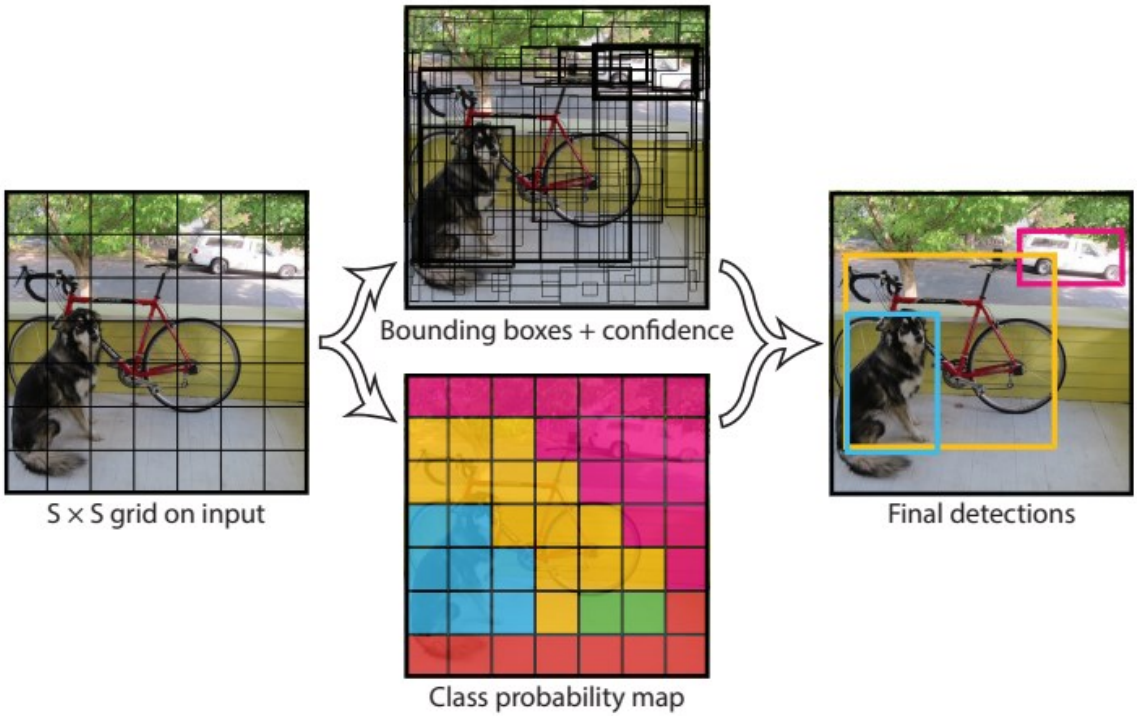


Figure 12: Model of object prediction based on grid segmentation – obtained from [48].

There are different versions of YOLO, such as YOLOv4 [50], YOLOv5 [51], and more recently, YOLOv8 [52]. Each version improves the model's performance by introducing new architectures and techniques, such as anchor-free approaches, new data augmentation techniques, and efficient feature extractions.

3.5. Summary from the state-of-the-art

After a brief introduction of state-of-the-art and some fundamentals, it is noticeable that in the past years, new methods and algorithms have been implemented to bring solutions to recurring problems. More than ever, machine learning is taking a massive part in those approaches. When talking about specific objectives, like glaucoma detection, it is known that the algorithm must adapt to extract the correct and desired key features.

Some of the works showed that RNNs and LSTM are not the best methods to treat long video summarization due to limitations on data length, being more useful on short-length videos, within the 30-80 frames range, like speech recognition videos. CNNs have presented some of the best results within long-length video summarization, thus being useful in some areas like medical, face recognition, security, and summarization of large data in general. Most of the reviewed articles were dependent on CNN methods due to their excellent performance. Still, it is important to note that being a powerful tool will also require equivalent computational power.

Overall, the key to achieving satisfactory results in video summarization depends on a good comprehension of the features needed to be summarized and choosing the method or combinations of methods that best suit the desired outcome. Also, selecting an ideal classifier can help achieve better results. In glaucoma detection, a great field of study can still be developed.

It is also important to state that the reviewed papers used public or private databases from high-quality images or videos. One of the main ideas of this work is the pursuit of the best method that can provide a reliable summary with low computational consumption due to the usage of smartphones with lower-quality image acquisition. This work aims for early detection with a fast and trustworthy algorithm. Instead of the conventional methods proposed in past works, a new algorithm capable of using a low-quality smartphone video of fundus recording and converting the resulting video to a single image with relevant features to finally bring a significant diagnostic, and of course, with a professional medical validation for those results.

As for object detection, the YOLOs proved to be a powerful resource that brings efficiency with low processing, indicating that it should deliver a good result in low-quality imaging. Of course, only one will be selected to perform a full test on the result after a brief comparison between the YOLO models.

Based on the information gathered, the plan will conduct a detection using a combination of the previously mentioned methods, aided by a private dataset obtained from the D-Eye device and the ACRIMA dataset for training and testing. Finally, a summary of the results will be performed.

4. Methodology

This chapter will present the methodology implemented to achieve the main objectives of this dissertation: retina detection through object detection methods and retina mosaicing of from the previously detected retinas, where the object detection and mosaicing methods are put together to perform a retina reading within a video input. Figure 13 presents how the workflow is intended to run.

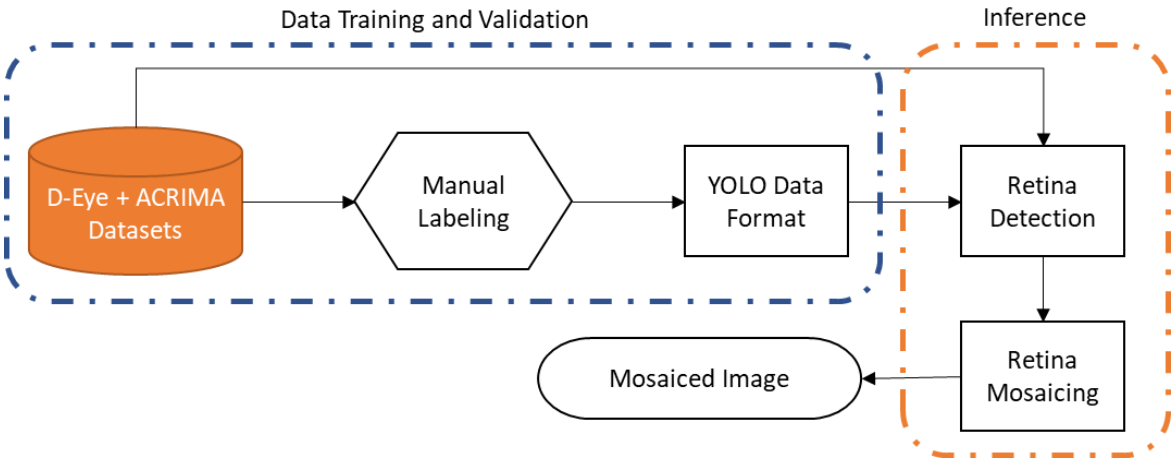


Figure 13: Methodology workflow.

The first step is to prepare the dataset, the D-Eye and ACRIMA datasets, for training and validation. To perform that, it is required to normalize the data information to the selected object detection data format, in this case, YOLO format, which will be explained in detail further in this chapter. Since there are many versions and evolutions of the YOLO algorithm [50], [51], [53]–[55], a brief evaluation will be performed to compare the one that fits best with the objective. Then a final step will perform the and mosaicing of the selected images previously filtered by the YOLO method and evaluate the result based on a known metric for detection and visualization inference for the mosaicing method.

4.1. Evaluation

A commonly used metric is applied to assess how well an object detection algorithm is performing - the Intersection over Union (IoU) [56]. This metric uses sets A and B, where A represents the pixels proposed by the algorithm as belonging to an object, and B represents the true object pixels. The precision is calculated using the following Equation (2):

$$IoU(\boldsymbol{A},\boldsymbol{B}) = \frac{\boldsymbol{A} \cap \boldsymbol{B}}{\boldsymbol{A} \cup \boldsymbol{B}}; \quad IoU(\boldsymbol{A},\boldsymbol{B}) \in [0,1] \tag{2}$$

For this metric, it is commonly known that a threshold bigger than 0.5 means there was a success in object detection. Likewise, above this threshold means that it was a failure. Although IoU is a useful metric to measure object localization accuracy, it may not always be the most appropriate metric, especially in scenarios where multiple objects must be detected. In such cases, average precision provides a more comprehensive evaluation of the detection performance, considering both precision and recall. Given a specific number of classes, for each class of objects **c** in the configuration data, it is possible to calculate the average precision (AP) [57], following this Equation (3):

$$AP(c) = \frac{TP(c)}{TP(c) + FP(c)} \tag{3}$$

Where TP(c) is the number of true positives and FP(c) is the number of false positives for a single class, where true positives are the cases where the model correctly predicts a positive example as positive. In contrast, false positives occur when the model incorrectly predicts a negative example as positive. An average precision close to 1 means the best result possible, and likewise, a value close to 0 means poor efficiency.

The mean average precision is a popular metric between object detection methods, therefore, it will be used in this work as the main metric for evaluating the YOLO's performance. Hence, the mean average precision (mAP) for *n* number of classes is represented by [57] in Equation (4):

$$mAP = \frac{1}{n_{class}} \sum_{i=1}^{n_{class}} AP_i \tag{4}$$

Where the n_{class} is the number of classes, and AP_i is the average precision for the i-th class. To use mAP for evaluating object detection algorithms, we first calculate the AP for each class and then take the average over all classes. Typically, we use a pre-defined threshold for intersection over union (IoU) between the predicted and ground truth bounding boxes to determine true positives and false positives. In practice, the mAP metric is often used to compare different object detection models, where a higher mAP value indicates better performance. It can also be used to track the performance of a single model over time or during training.

Another important metric that will help evaluate the methods and models is the Recall, similar to AP but with only one change; it shows the precision on sensitivity for false negatives (FN). In the context of machine learning and statistics, it measures a model's ability to identify all positive examples in a dataset correctly. It is also known as sensitivity or hit rate. It is described as the following Equation (5):

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

Altogether with the AP, they form the F1 score [58], or the F1 curve, which measures the model's accuracy that considers both the precision and recall of the model's predictions. The F1 score is the harmonic mean of precision and Recall and ranges from 0 to 1, with higher values indicating better performance. The following Equation (6) describes it:

$$F1\ Score = \frac{AP * Recall}{AP + Recall} \tag{6}$$

The comprehension in the F1 score is to determine a metric for two ratios in a balanced way; the higher both are, the higher the F1 score is. Since the objective is to achieve the precision of a single class “retina” and the absence of it, the F1 score fits to evaluate this work. To summarize the concept from the evaluation methods, Figure 14 presents the hierarchy of the metrics from the input data following the final F1 score.

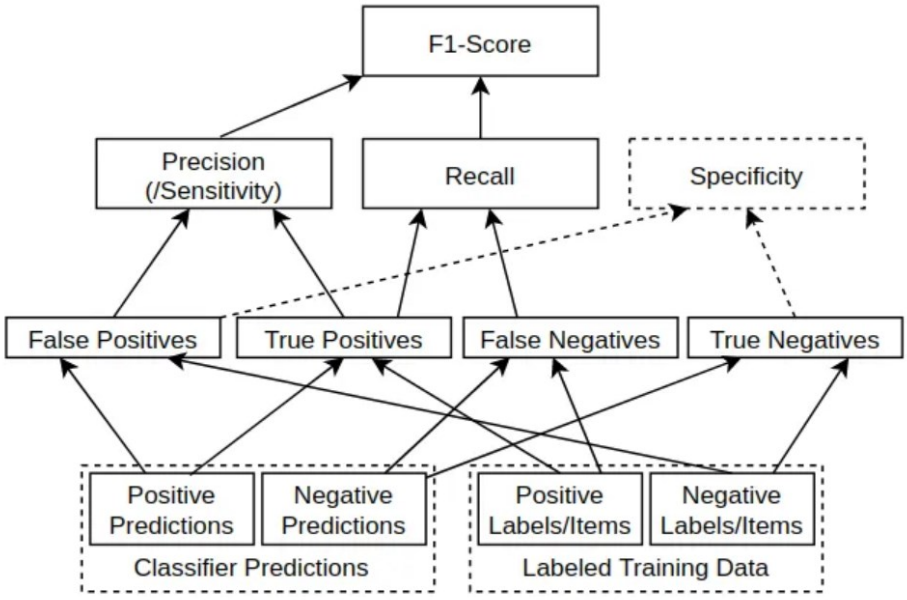


Figure 14: Hierarchy of the Metrics – obtained from [58].

4.2. Datasets

This dissertation used two retinal datasets, the D-Eye dataset, which is private, and the publicly available ACRIMA dataset [43].

The D-Eye dataset consists of 48 low-resolution video captures of the close up eye with exposed retina, each with a resolution of 1080x1920 pixels. 125 images were extracted from a single video in this dataset for training and validation purposes. The ACRIMA dataset comprises 705 images with a resolution of 577x577 pixels, including 396 glaucomatous and 309 normal images. 258 images were randomly selected from this dataset to be trained, along with the 125 images from the D-Eye dataset. Table 3 presents how this selection was divided.

Table 3: Dataset division for training and validation

	Resolution (pixels)	Train	Validation
D-Eye	1080x1920	100 images	25 images
ACRIMA [43]	577x577	201 images	57 images
Total		301 images	79 images

With the final images selected, it must be prepared to start training in the object detection section, the YOLO labeling format, which will be explained in the next section.

4.3. YOLO Network

In this section, will be presented a selection of YOLO network models that are going to be evaluated in terms of precision and performance. The best YOLO model will be chosen to infer the video inputs for the following steps in this work.

To perform this is important to note that the several architectures of the YOLOs are similar for various available versions. Therefore, for the training and validation, a custom dataset annotation must be implemented and be able to fit all investigated versions.

The dataset annotation must be divided into image and label folders, and inside them must have two other folders separating the train data and validation data. For each image file, a label *.txt* file must exist with the corresponding filename, containing the class of the

object to be detected and the bounding box area with the 4 points of interest. The annotation must follow this example format in Table 4:

Table 4: Example of YOLO label annotation format.

Class ID	X	Y	W	H
0	0.296198	0.502646	0.282187	0.172840

Where:

- Class ID is the label index of the object;
- X is the ratio between the center *x* coordinate of the bounding box and the width of the image;
- Y is the ratio between the center *y* coordinate of the bounding box and the height of the image;
- W is the ratio between the width of the bounding box and the width of the used image;
- H is the ratio between the bounding box's height and the used image's height.

Another file is required to perform the training in the YOLO framework, namely the configuration file in *.yaml* format. This file contains the required information needed to run the training and validation. Figure 15 shows how it should be for custom data.

```
1
2   path: ../train_data # dataset root dir
3   train: ../train_data/images/train  # train images (relative to 'path')
4   val: ../train_data/images/val  # val images (relative to 'path')
5   test:  # test images (optional)
6
7   # Classes
8   nc: 1  # number of classes
9   names: ['retina']  # class names
```

Figure 15: Custom data *yaml* file.

With the dataset prepared and the configurations done, it was time to perform training and validation within the object detection methods YOLOv5, YOLOv6, YOLOv7, and YOLOv8. These were chosen based on a previous research contribution of this investigation group [49], and now with these new versions are expected to have better results.

For each version of YOLO, were performed six different tests, having as variable parameters the learning rate and epochs. Other parameters were kept due to the time limit operation of the machine used, which was a virtual machine from *Google Colab* [59]. The momentum parameter was set to 0.937 for all tests, the image size normalized to 640 pixels, and the batch size selected was 32 due to speed and processing limitations. All tests were performed on a Tesla T4 with 12 GB RAM, using Python version 3.8.10, PyTorch 1.13.1, and Cuda version 116, provided by *Google Colab*.

4.3.1. YOLOv5 Training

Starting with the YOLOv5 [55], initially released in May 2020, developed by Ultralitcs, they claim this version is a huge improvement from what YOLOv3/YOLOv4 were capable of, switching from the framework Darknet to PyTorch, and being able to train with a huge amount of data and converging faster in terms of learning.

The YOLOv5 architecture consists of three key elements: the backbone, the neck, and the output. The input component is responsible for data pre-processing, which includes techniques such as mosaic data augmentation and adaptive image filling. To ensure adaptability to various datasets, YOLOv5 also features an adaptive anchor frame calculation system at the input stage, allowing the model to automatically adjust the initial anchor frame size for different datasets [60].

The backbone component of YOLOv5 is a convolutional neural network designed to aggregate and create image features at varying levels of granularity. It primarily employs the cross-stage partial network (CSP) [50] and spatial pyramid pooling (SPP) [61] techniques to extract feature maps of varying sizes from the input image through multiple convolution and pooling operations. The Bottleneck CSP architecture is utilized to minimize computational demands and enhance inference speed, while the SPP structure enables feature extraction at different scales, generating three-scale feature maps that help to boost detection accuracy. Figure 16 shows how are the structures from these architectures.

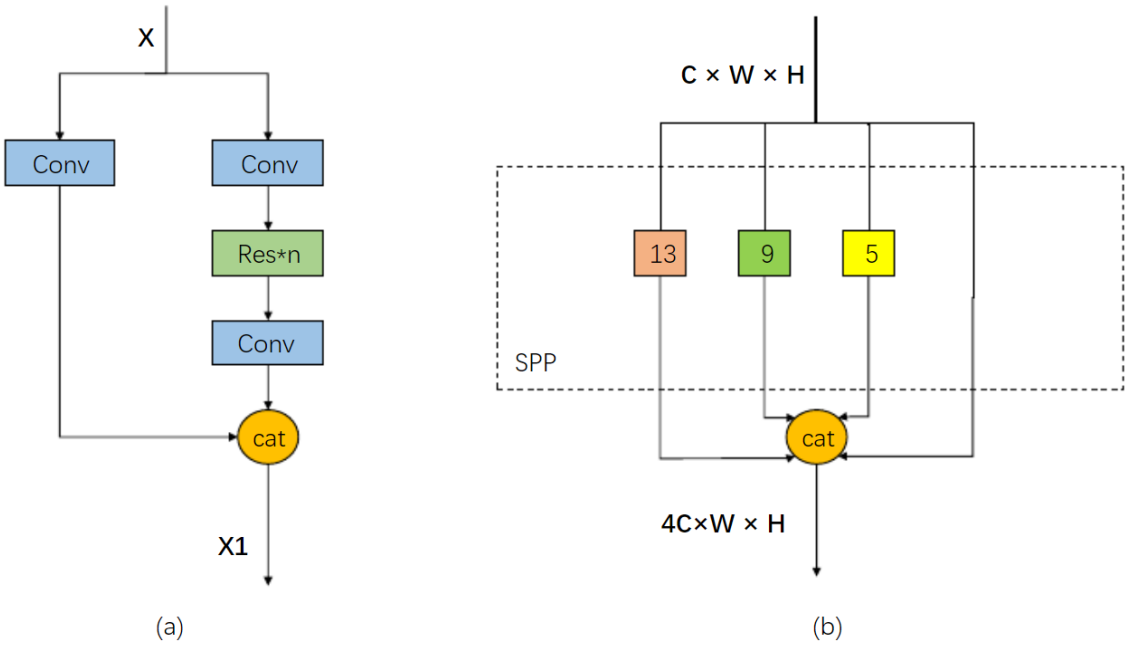


Figure 16: Structure from CSP(a) and SPP(b) – obtained from [62].

The neck component is a series of layers that integrate and merge image features and transmit them to the prediction stage. This component employs both the feature pyramid network (FPN) [63] and pyramid attention network (PAN) [64] structures. The FPN structure brings strong semantic features from higher-level feature maps down to lower-level maps. In contrast, the PAN structure carries strong localization features from lower-level maps up to higher-level maps. These two structures complement each other to enhance the features extracted from different network layers during the backbone fusion process, improving detection capabilities. Finally, the head output component is used to predict targets of varying sizes on the feature maps, representing the final detection stage.

Table 5 shows how the conducted training with the selected variable parameters worked for YOLOv5 for retina detection.

Table 5: YOLOv5 training accuracy on retina detection.

	Batch	Learning Rate	Epochs	mAP(0.5) (%)	mAP(0.5:0.95) (%)
Train 1	32	0.001	10	6.37	2.29
Train 2	32	0.001	50	98.48	70.58
Train 3	32	0.001	100	98.51	80.67
Train 4	32	0.01	10	96.29	68.82
Train 5	32	0.01	50	98.67	91.2
Train 6	**32**	**0.01**	**100**	**99.13**	**92.2**

As shown in Table 5, the mAP (0.5), which describes the average precision for a threshold bigger than 0.5, is achieved with high confidence for all training sets except for the first test, which has a lower learning rate and a lower epoch value. For the mAP (0.5:0.95), which is defined as a metric for different thresholds ranging from 0.5 to 0.95 in scales of 0.05, the higher scores are presented in both trainings with the higher learning rate tested and up to at least 50 epochs. The increment in the latter test, named Train 6, was only 1% against Train 5, showing that 100 epochs is sufficient to the proposed work. The mAP is the main evaluation metric in the training phase (the higher the score, the higher the probability for the trained model to detect the retina in a new set of samples).

For the inference test, it was selected a random video from the D-Eye private dataset, and the results matched well the training data results, as depicted in Figure 17. The test data input had a total of 439 consecutive frames of retina being detected.

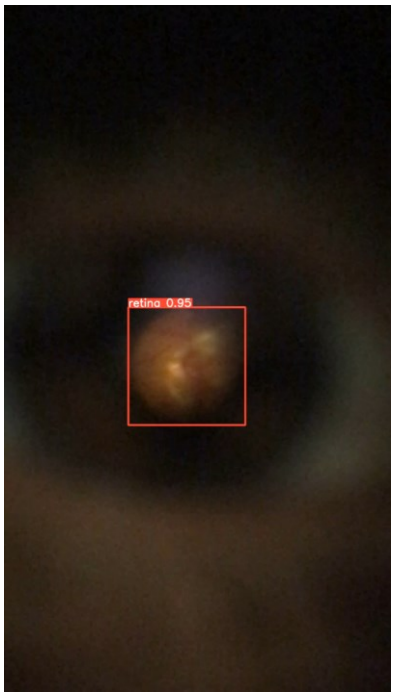


Figure 17: Example of retina detection, with 95% mAP, with Train 6 model.

The YOLOv5 also provides the F1 confidence curve, which illustrates the precision and Recall trade-off in a binary classification task as the confidence threshold for predicting class labels varies. In addition, it provides a visual representation, as shown in Figure 18, of the effect of changing the confidence threshold on balance between precision and recall and helps identify the optimal point to achieve a desired balance between the two.

The F1 curve from Figure 18 shows that the model from Train 6, when near a perfect confidence score, it shows a great number of false positives or false negatives, which cause the F1 score to diminish. Thus, it shows that the model is not confident enough to precisely affirm that an object is a retina or not above 90%, which makes sense compared to the Table 5 results.

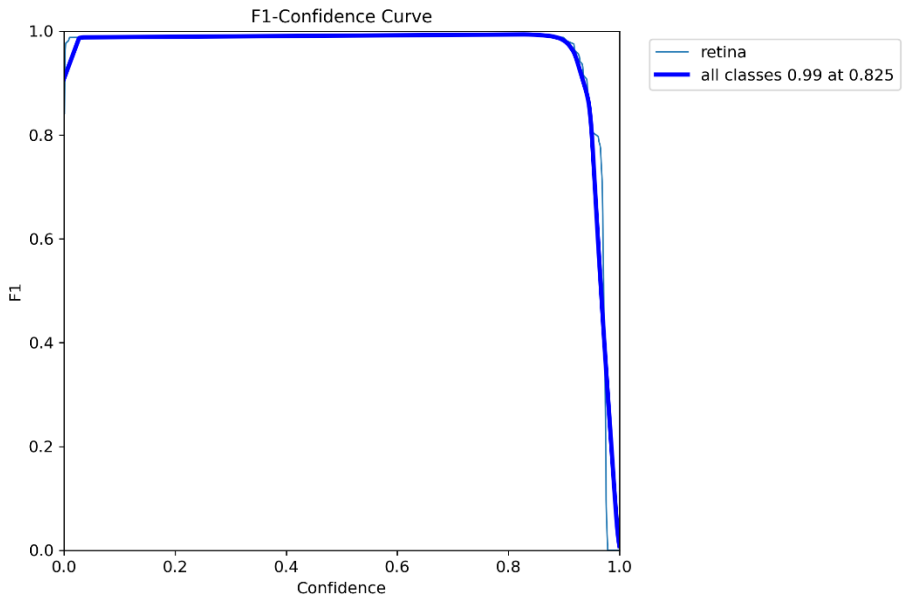


Figure 18: F1 Confidence curve from Train 6.

The graph in Figure 18 shows 2 curves representing the same object. The difference is that the YOLO method uses different equations to describe a single class, in this case, the only class is the presence of retina. By default, the model retrieves the curves to describe all classes. In our custom dataset, the "all classes" is quite similar to the retina class and mean to represent the same result.

4.3.2. **YOLOv6 Training**

As a successor to YOLOv5, the YOLOv6 [53] is a single-stage object detector developed by Meituan Inc. and actually came after YOLOv7 by a short time after its release. It is important to notice that these versions are not official releases as the original author stopped updating within the launch of YOLOv3 because of personal beliefs. The authors of the YOLOv6 model decided to develop a tool that helps solve problems in industrial applications. They introduced improvements from the previous model, upgrading the network design and adding some features that potentially will help overcome the previous versions in terms of speed and computational power. Figure 19 shows the architecture from the YOLOv6.

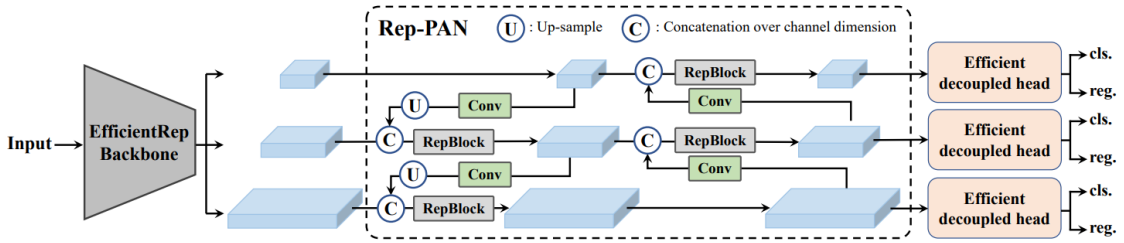


Figure 19: YOLOv6 Architecture – obtained from [53].

They chose the RepVGG [66] for the backbone as this architecture is more powerful and as fast as the mainstream architectures for small networks. As for large networks, they use a more efficient version of CSP, the CSPStackRep block. The neck is like the previous YOLO versions, using the PAN topology, but they performed an enhancement with RepBlocks to have a Rep-PAN architecture. In the end, the output is composed of a decoupled head structure with a simpler design. These changes were made for a better computational response.

YOLOv6 uses a more streamlined anchor-free detection approach. In contrast to anchor-based detectors, which require cluster analysis before training to identify the optimal anchor set, the latter can increase the complexity of the detector and lead to additional delays in edge applications where numerous detection results need to be transferred between hardware steps. Nevertheless, anchor-free detection is popular due to its robust generalization ability and simpler decoding logic. In addition, a recent analysis revealed that the speed of the anchor-free detector has a significant enhancement compared to the extra delay associated with the complexity of the anchor-based detector.

To enhance the detection accuracy by obtaining better positive samples, YOLOv6 incorporates the SimOTA [65] algorithm for the dynamic allocation of such samples. However, in contrast to the YOLOv5 shape-matching-based label assignment strategy and cross-grid matching technique, which increase the number of positive samples and enable fast network convergence, this approach is static and doesn't adapt to the evolving network training process.

Also similar to YOLOv5 architecture, YOLOv6 utilizes the SIoU [66] bounding box regression loss function to facilitate the network's learning process. Typically, training the target detection network involves the specification of at least two loss functions, namely

classification loss and bounding box regression loss. The choice of loss function significantly affects the detection accuracy and training efficiency.

Despite this method being very promising, it did not show satisfactory results in retinal detection training. The model developed by Meituan Inc. also did not provide full results from the training, validation, or testing, lacking some key information that the previously tested model provided. Figure 20 shows that the model struggled to run an inference test with the provided dataset.

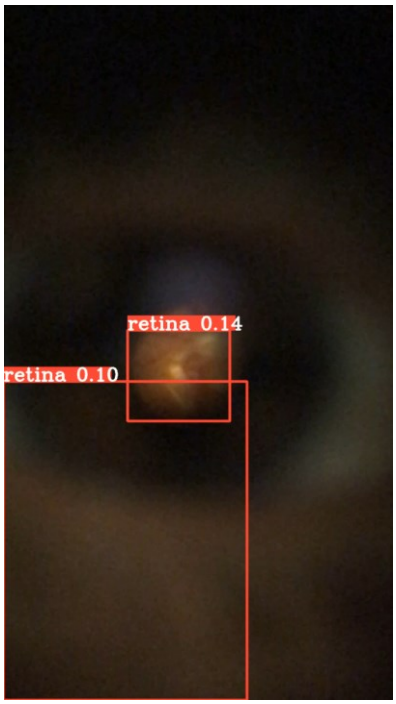


Figure 20: Example of failure on retina detection with Train 6 of YOLOv6.

Nonetheless, as the same tests were performed in all models, the F1 confidence curve from the best training result of YOLOv6 shows that the model architecture did not fit the retina dataset. Therefore, a more complete dataset may be required for the model to deliver better results than the ones shown in Figure 21.

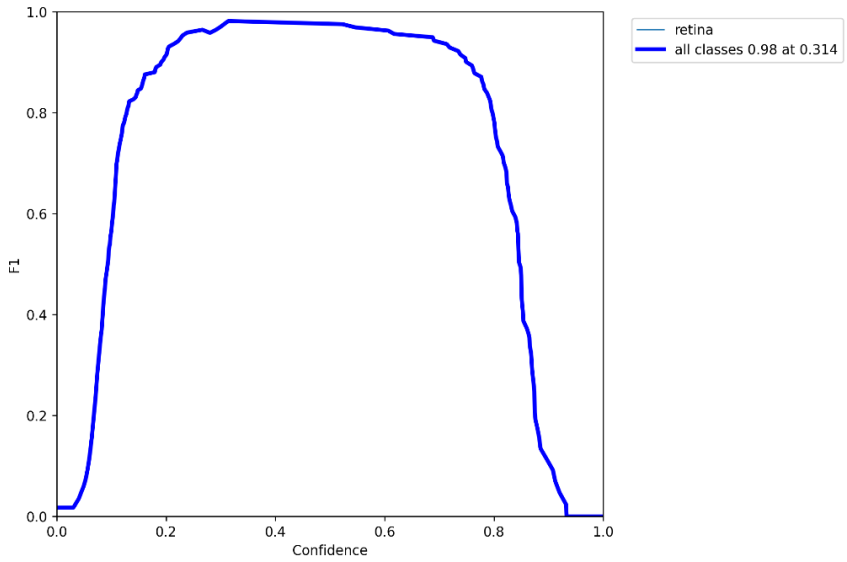


Figure 21: F1 confidence curve for the best training in YOLOv6

4.3.3. YOLOv7 Training

YOLOv7 was released in July 2022 [54] by the same authors of YOLOv4, and authors claim that this version surpasses all known object detectors in accuracy and speed, with much less computational power, and can be trained in small datasets without any pre-trained weights. Figure 22 presents a comprehension of the YOLOv7 in a set of images trained in the COCO dataset [67].

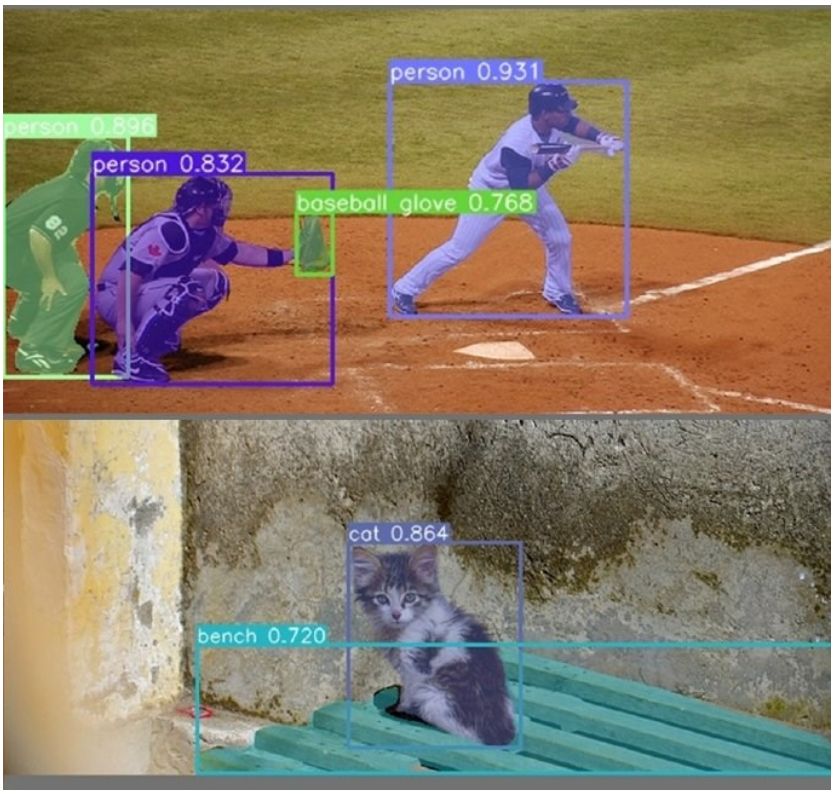


Figure 22: YOLOv7 instance segmentation with trained COCO dataset [54].

The architecture is also similar to older versions of YOLO, with some improvements, starting with the backbone block, Extended ELAN, or E-ELAN, an improved version of the ELAN (Efficient Layer Aggregation Network) [68] main architecture. ELAN is a neural network architecture that uses a unique module known as the "layer aggregation unit" to combine feature maps generated by various convolutional layers. The layer aggregation unit utilizes an attention mechanism to assign weights to each feature map, which are then combined to form a single feature map that can be utilized for further processing. The main idea of the extended version is to be able to improve the learning ability without making any changes in the gradient path.

It also introduced a model scaling for concatenation-based models, whose purpose is to adjust the model's attributes and re-scale it to meet the needs of different inference speeds. Generally, scaling a model involves adjusting hyperparameters such as the number of layers or neurons in each branch, activation functions, learning rate, and regularization strength. The objective is to achieve a balance between the model's capacity to learn complex patterns in the data and its efficiency in terms of speed and resource utilization while still attaining desirable performance on the specific task at hand. That’s the difference of this architecture from the most notorious approaches with concatenation-based architectures, where different

scaling factors can't be analyzed one by one, trying to scaling-up model depth will make a notable difference between the input and output of this specific layer, making it less efficient, for example. Figure 23 represents how model scaling works for concatenation-based models.

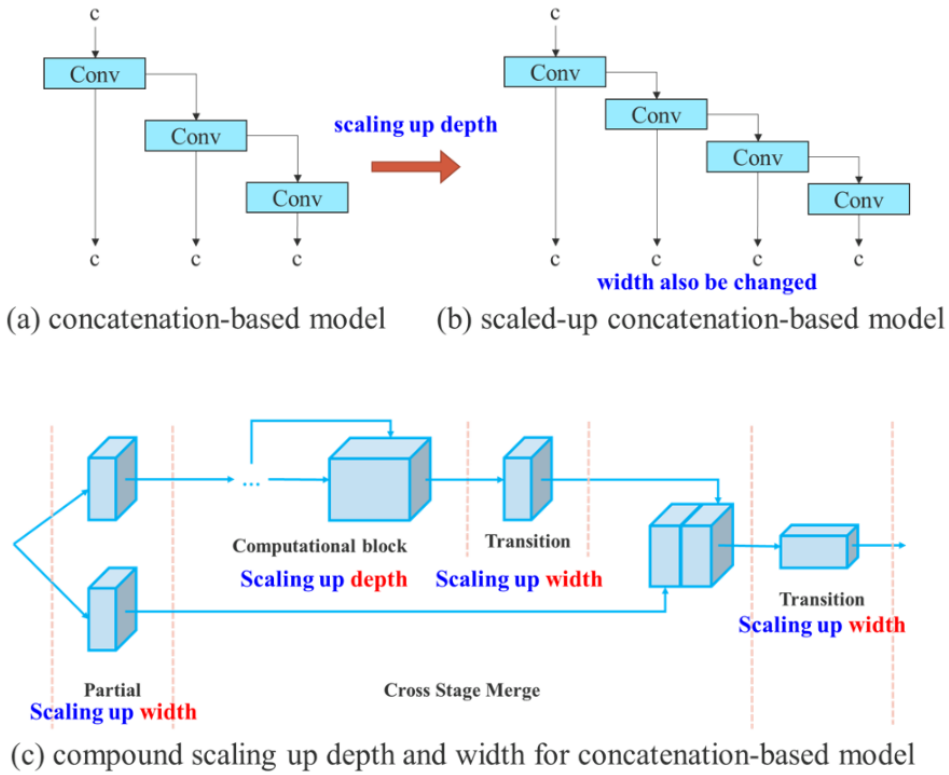


Figure 23: Model scaling differences in (a) to (b) and the proposed method on (c) – obtained from [54].

Another characteristic of the YOLOv7 architecture is the implementation of a planned re-parametrized convolution. The authors performed a series of tests to evaluate the performance of RepConv [69] and some other architectures, and as a result, they found out that the identity connection in RepConv destroys the residual in ResNet [70] and the concatenation in DenseNet [71], providing a diversity of gradients for feature maps. Thus, they opt for RepConvN, which does not have an identity connection, to develop the architecture for planned re-parameterized convolution. Then, replacing a convolutional layer with residual or concatenation with re-parameterized convolution should not include an identity connection.

Based on Deep Supervision [72], a commonly used technique in training deep neural networks, YOLOv7 employs multiple heads instead of being limited to one. The head responsible for generating the final output is called the lead head, while the auxiliary head is used for training purposes in the intermediate layers. To improve the training of the deep

network, a label assigner mechanism has been introduced that employs the network prediction results and ground truth to assign soft labels. Unlike traditional label assignment methods that generate hard labels based on given rules from the ground truth, reliable soft labels are generated using calculation and optimization methods that consider the quality and distribution of prediction output in conjunction with the ground truth.

After this brief explanation of the architecture of YOLOv7, Table 6 shows the training tests performed for mean average precision.

Table 6: YOLOv7 training accuracy on retina detection.

	Batch	Learning Rate	Epochs	mAP(0.5) (%)	mAP(0.5:0.95) (%)
Train 1	32	0.001	10	0.65	0.092
Train 2	32	0.001	50	3.98	0.88
Train 3	32	0.001	100	52.22	10.38
Train 4	32	0.01	10	1.85	0.24
Train 5	32	0.01	50	27.59	10.84
Train 6	**32**	**0.01**	**100**	**95.58**	**56.74**

As it is possible to observe in Table 6, the YOLOv7 model has struggled to deliver high accuracy for short epoch values and a lower learning rate. The best result was in the last training set with a 56.74 at mAP(0.5:0.95). The poor results compared to YOLOv5 may be the same reason for the low score on YOLOv6, which may be because of the low samples of dataset input.

Examining further, an interesting phenomenon occurs in Train 3. After hitting the high score for mean average precision in the 64^{th} epoch, the score dropped under 0,2 at mAP(0.5), as depicted in Figure 24. In retrospect, Figure 25 presents the scores from the Train 6 set, which differs from Train 3 only in the learning rate, but in its final epochs, it was possible to see that the precision was still following an improvement tendency. This may have occurred because the model with a lower training rate possibly started to converge due to high variance in the training process of this model.

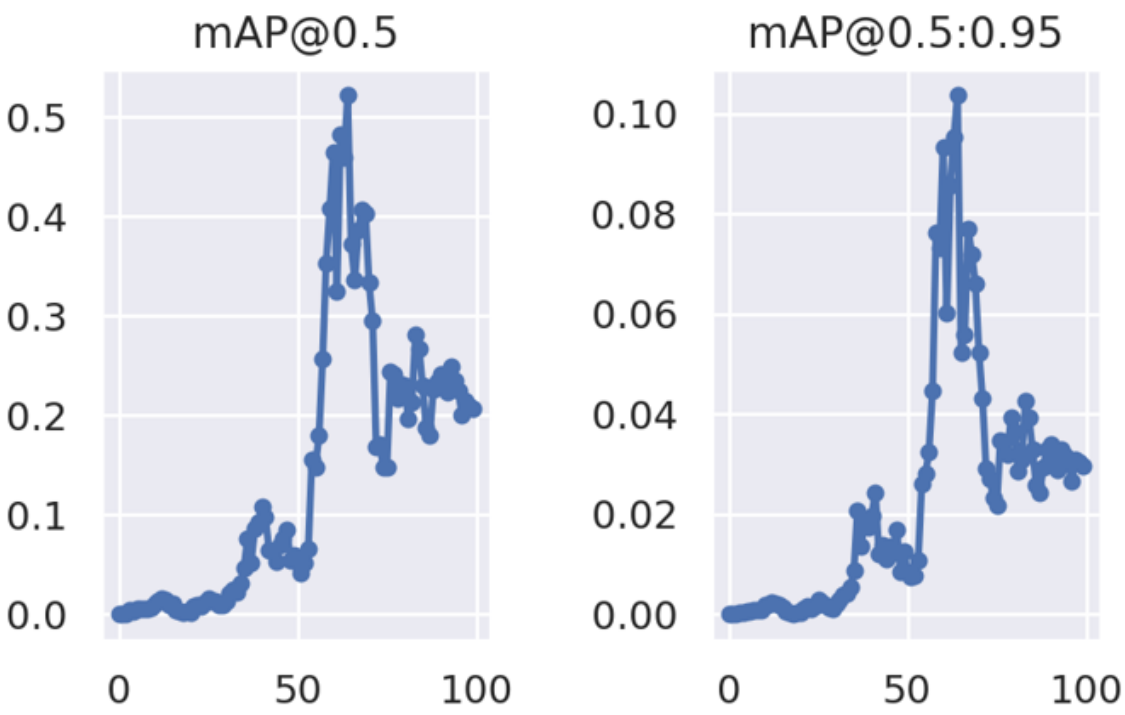


Figure 24: Mean average precision for Train 3 on YOLOv7.

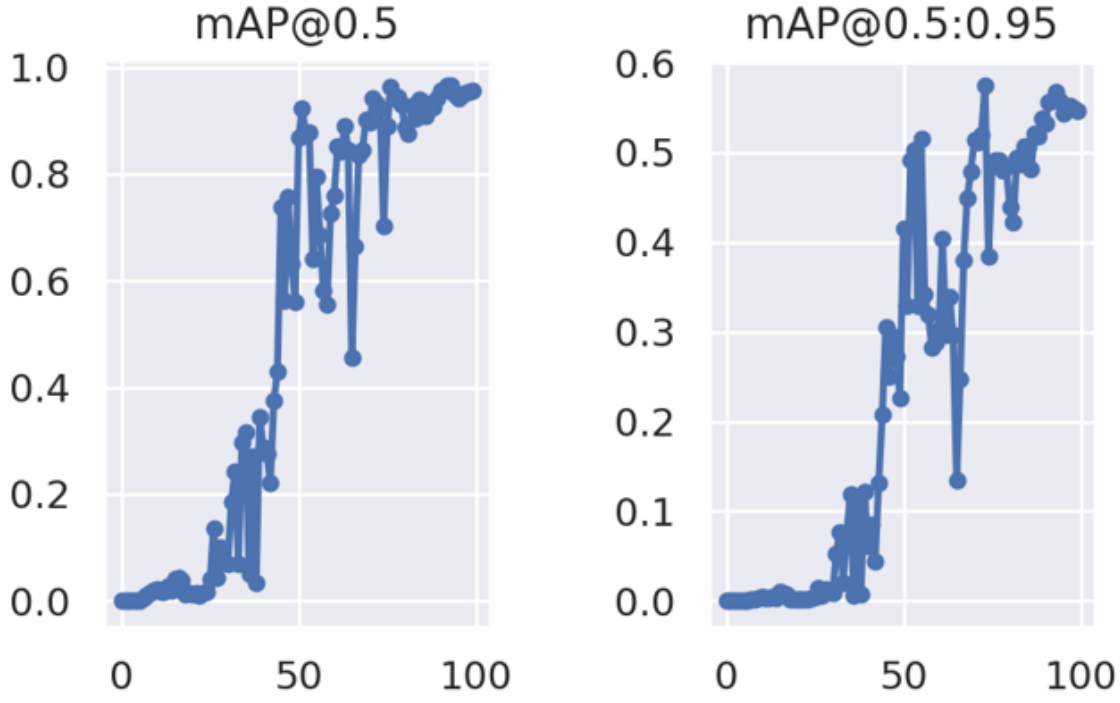


Figure 25: Mean average precision for Train 6 on YOLOv7.

4.3.4. **YOLOv8 Training**

The final object detection model to be trained is the YOLOv8 which is technically the real successor to YOLOv5. They were both built by Ultralitics, with the newer version released in January 2023, of course, more powerful and faster for both detection and segmentation tasks than its predecessor.

YOLOv8 offers substantial developer-convenience features. Unlike other models where tasks are separated into multiple Python files that need to be executed, YOLOv8 provides a Command-Line Interface (CLI) that simplifies the process of training a model, with a Python package included that provides a more user-friendly coding experience compared to previous versions of the model. As of today, no paper was released for this version, but there is an extent content on its repository on GitHub[1]. Figure 26 shows how is the architecture of this model.

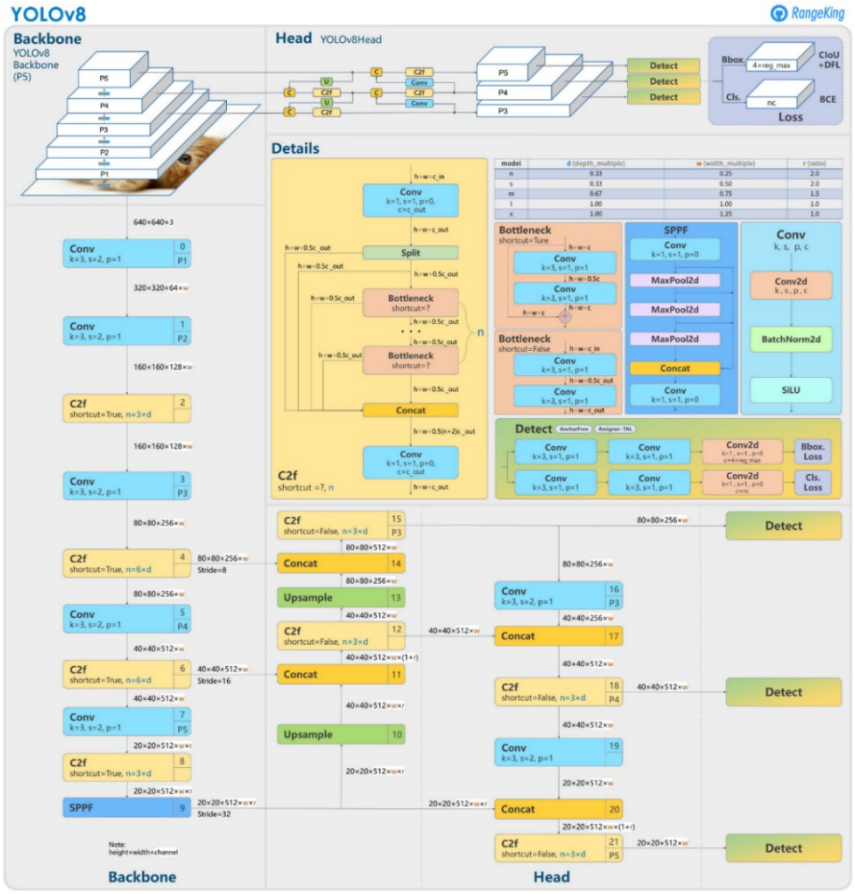


Figure 26: YOLOv8 model architecture – obtained from [55].

[1] https://github.com/ultralytics/ultralytics

A recent publication from Roboflow [52] website, a cloud-based platform that supplies tools for annotation, image management, and augmented computer vision datasets, presented an accuracy test named RF100, used to evaluate the model in a custom dataset. It used a 100-sample dataset from their repository, subdivided into seven categories, and compared the YOLOv8 results with YOLOv5 and YOLOv7, running each model for 100 epochs measuring the mAP(0.5). Figure 27 presents a plot bar graphic with the scores from these categories.

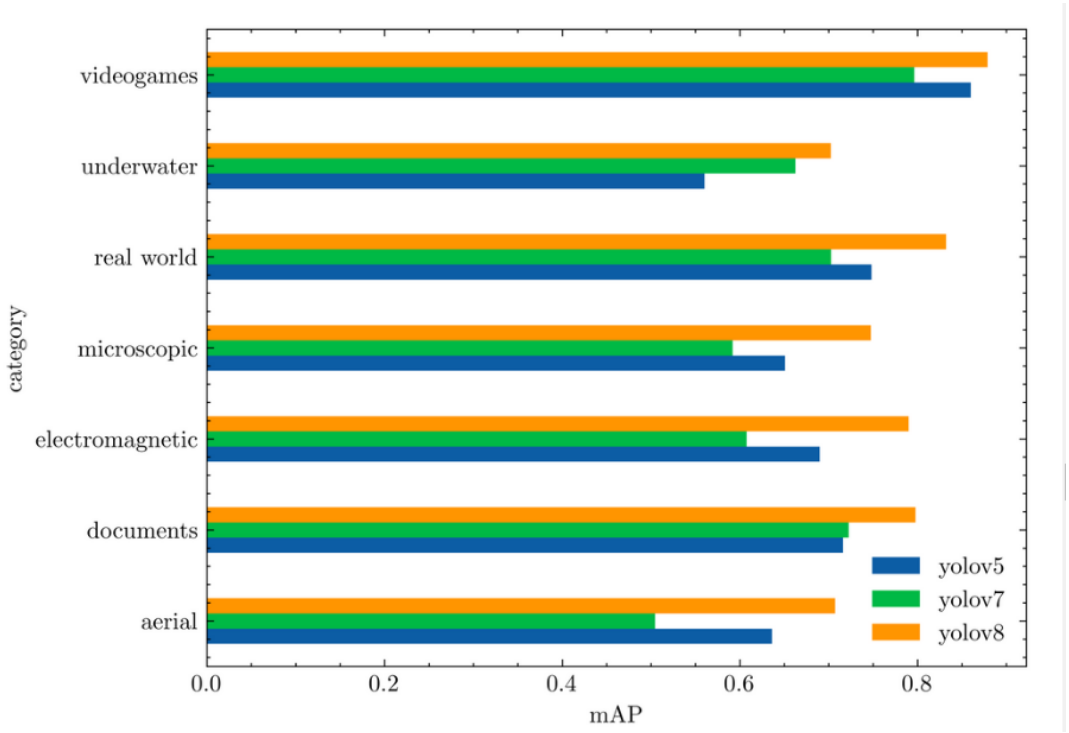


Figure 27: MAP (0.5) comparison between the YOLO model for each category [52].

In these terms, it is possible to see that YOLOv8 has superior mAP within all the categories, and even YOLOv5 has achieved high scores against YOLOv7 in five categories. Following the same training steps from the previous YOLO models, Table 7 presents the results for the training and validation for the mean average precision for YOLOv8.

Table 7: YOLOv8 training accuracy on retina detection.

	Batch	Learning Rate	Epochs	mAP(0.5) (%)	mAP(0.5:0.95) (%)
Train 1	32	0.001	10	98.36	85.78
Train 2	32	0.001	50	98.5	90.23
Train 3	32	0.001	100	98.5	91.15
Train 4	32	0.01	10	98.5	91.04
Train 5	32	0.01	50	98.52	90.05
Train 6	**32**	**0.01**	**100**	**98.89**	**92.26**

In this case, YOLOv8 performed far better than any other model, even in Train 1, which has a short number of training epochs, with high average precision in this training above the best result delivered in YOLOv7 model, but not being able to surpass the best result from YOLOv5. Oddly enough, this model is more powerful than others even with lower epochs for training, and even on a lower learning rate, indicating that YOLOv8 may have the good architecture, achieving high precision scores on small datasets retina images.

4.4. **Mosaic methods**

For the next step in this work, some tools will be presented and evaluated to perform the mosaicing of videos obtained from recordings made by D-Eye. Then, after collecting raw retina images, those tools will process these to extract as much information as possible, enhancing the details of the various retina frames available and delivering a single image as the result. The challenge will be to use some of these methods to obtain an adequate result to perform glaucoma detection, either by medical analysis or by a glaucoma automatic detection method.

4.4.1. Deep Image Stitching

To perform stitching of images from arbitrary views using deep learning, a method was proposed [73] to estimate a global homography between two images and then coarsely stitch those images based on this homography, and finally, a revision network to refine the stitched image. Figure 28 represents the pipeline of this proposed method.

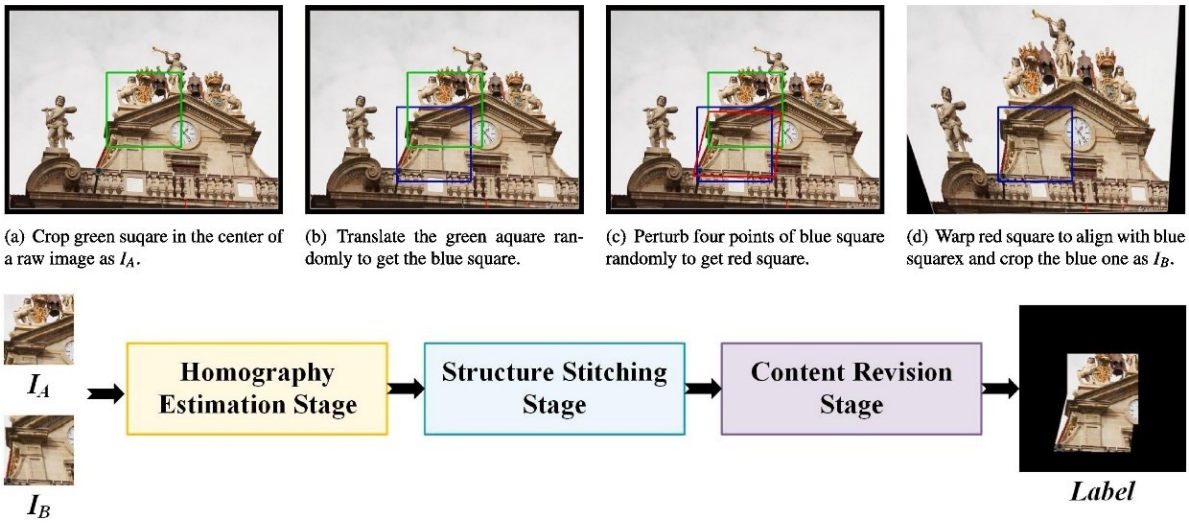


Figure 28: Pipeline from Deep Image Stitching – obtained from [73].

For the homography estimation phase, they prepared a pre-trained homography estimation network to free the CNNs from costly tasks. The steps for homography calculation involve feature extraction, feature matching, and homography solving by a direct linear transformation (DLT) [74] algorithm. The global homography estimation provides alignment information that can be used in the next stage of stitching the structure. Applying the homography can change the alignment of image pairs with a large parallax to image pairs with a small parallax. This process simplifies the problem of stitching images with a large parallax into stitching images with a small parallax, which makes the image stitching task with CNNs much easier.

The stitching phase was divided into two parts, structure stitching, and content revision. For the structure stitching it was used an improved Spatial Transformer Layer (STL) [75], proposing a Structure Stitching Layer (SLL), which will obtain the structure information about the stitched images. In addition, SLL will serve as a translator that can be used in the content revision network.

The content revision is implemented to eliminate the residues from the overlapping regions while maintaining low distortion in the other areas. To achieve this, an encoder-decoder network was applied to the stitching result, whereas the first module extract features resulting in 8 convolutional layers, with the number of filters per layer set to 64, 64, 128, 128, 256, 512, and 512. To reduce the computational work, a 2×2 max-pooling layer was applied, reducing the dimensions of the feature maps after the 2nd, 4th, and 6th convolutional layers. The decoder works similarly to the encoder, reorganizing the feature basis into a more

complex feature representation while matching the desired stitching result. Figure 29 presents the architecture structure for the stitching and content revision phases.

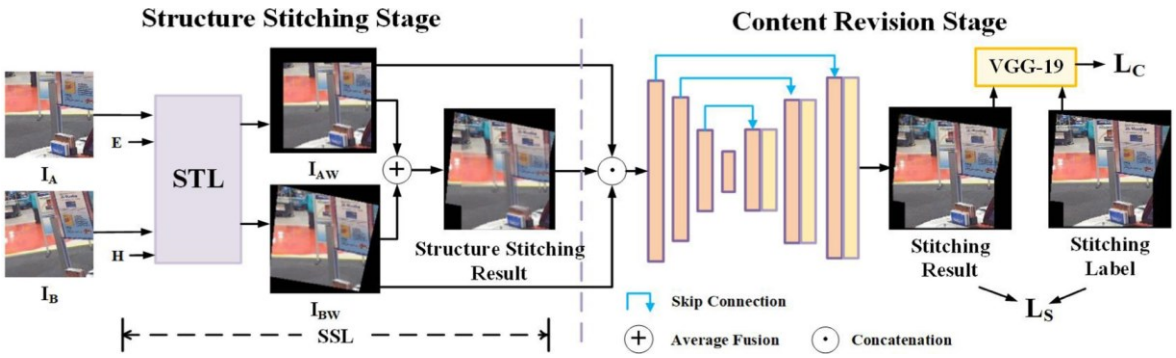


Figure 29: Stitching and content revision architecture – obtained from [73].

For the case of this work, a single video was converted to a total of 500 frames, and with the extracted retina crop from those, they were input in the Deep Stitching algorithm.

In the inference phase, 80 frames never seen before were inputted from the trained model. Figure 30 shows the results from the stitching method from 8 images that resulted in 4 pairs from different regions of the retina.

Figure 30: Example of 8 images paired with Deep Image Stitching.

It is possible to see that in this method, the stitching performed poorly compared to what was expected. Regarding usability, it suffers from detail gains, where the features obtained by the trained model fail to perceive the whole image features. Therefore, some adjustments may be needed for better performance stitching with the actual dataset.

4.4.2. **Super Retina**

In this method developed by Liu et al. [76], they proposed an approach based on training keypoint detectors and descriptors in a semi-supervised manner. They also proposed a Progressive Keypoint Expansion (KPE) to enhance keypoint labels for each epoch iteration and re-purposed and adapted a triplet loss as a keypoint-based description loss.

Figure 31 shows how their model performs compared to other models with the corresponding points for each pair of images, and with different cases for pairs such as large overlap pairs, small overlap, retinopathy incidence, and non-matching pairs. The green and red dots are considered valid or invalid matches respectively. The less bright green dots for the negative pair mean better accuracy since the pair isn't from the same image.

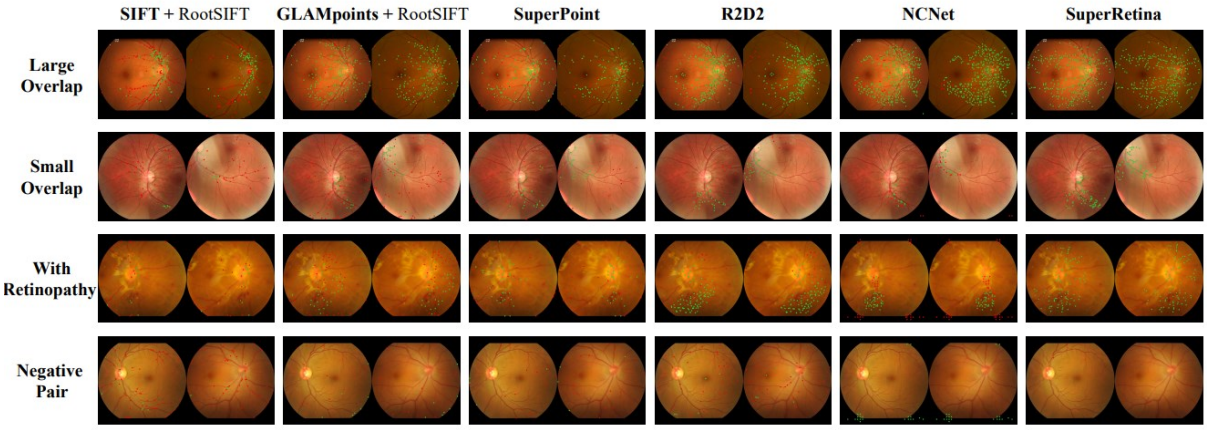


Figure 31: Retinal image matching comparison – obtained from [76].

For the network structure, they adopted the SuperPoint network [77], using an encoder to extract a down-sized feature F from a given image I, feeding the feature map in parallel into two decoders, one for keypoint detection and another to keypoint description, which was named Det-Decoder and Des-Decoder respectively. The function of the Det-Decoder is to produce a full probabilistic map P which indicates the probability of a pixel being a keypoint. The Des-Decoder produces a $h \times w \times d$ tensor D, which denotes a d-dimensional descriptor. Figure 32 shows the architecture of the proposed network.

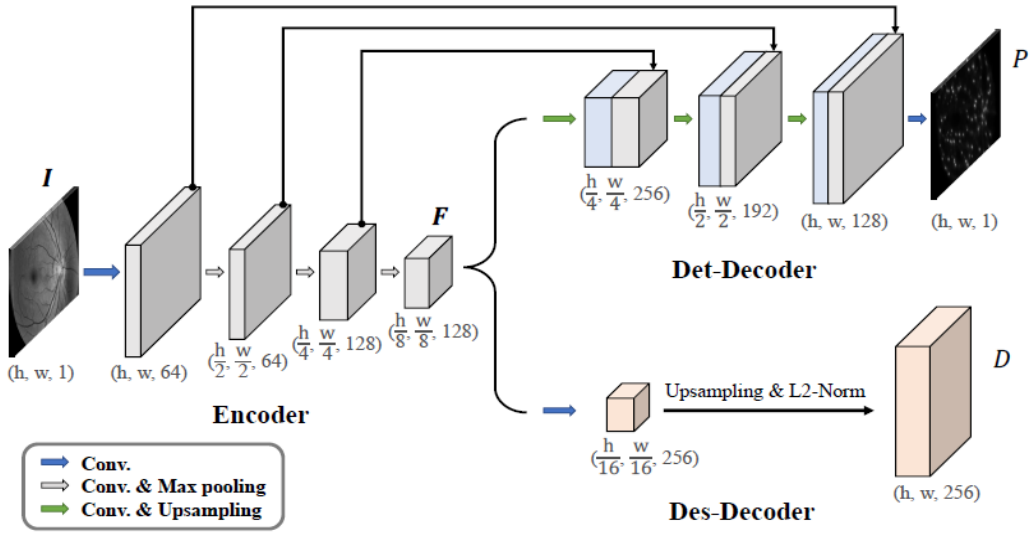


Figure 32: Super Retina network architecture – obtained from [76].

The authors also have a pre-trained model from retina matching available from their GitHub repository[2], which was helpful to perform some tests in common datasets. They also have the algorithm implementation to stitch pair images and see the results from their method. Figure 33 shows an example of image matching keypoints with a pair of images from the FIRE dataset [5] in their pre-trained model. Figure 34 shows the final image from their stitching.

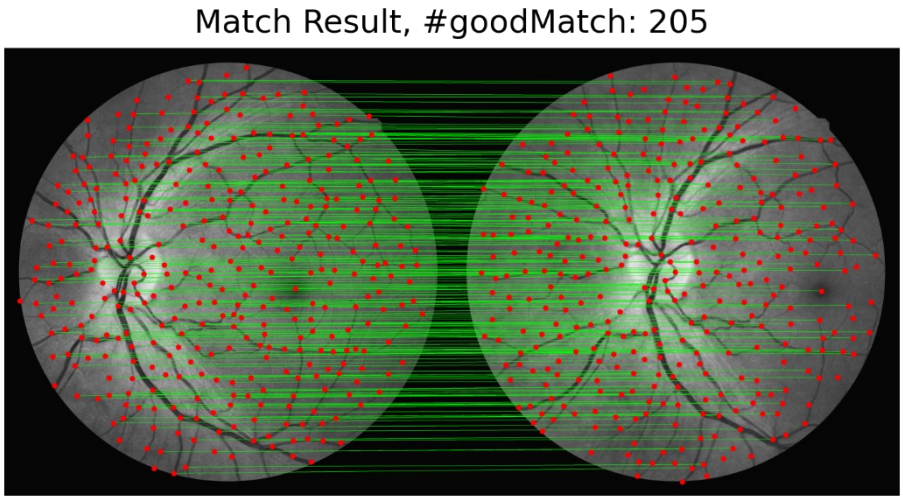


Figure 33: Matching keypoints from a pair of images.

[2] https://github.com/ruc-aimc-lab/SuperRetina

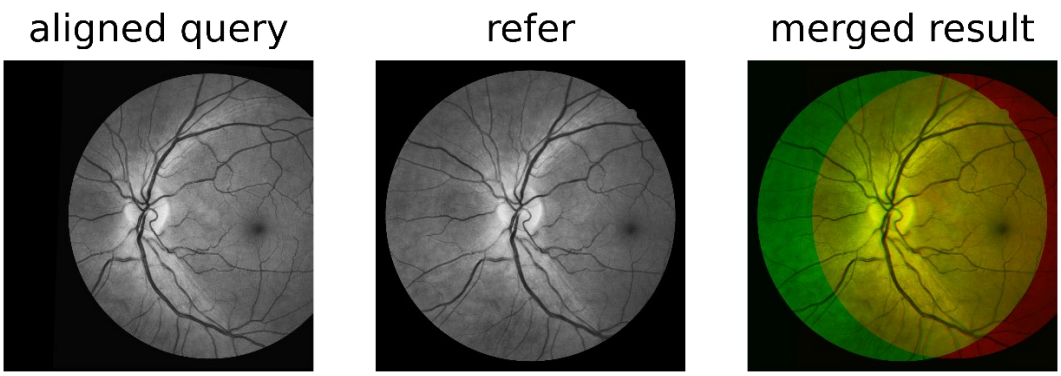


Figure 34: The final stitched image.

The methodology achieved a great number of matching points to enhance the feature correspondence, having 205 matches in the pair, resulting in a good stitched image. However, it was tested on high-resolution imaging. It is important to state that the stitched images are evaluated only in visual perception, a metric for evaluation may be implemented.

4.4.3. **Multi-image Stitching**

Multi-image stitching proposed by Hu et at. [78] shows a similar approach to the objective of this work. They intend to capture fundus retinal images from a device coupled to a smartphone and give a full image from these captures, except that the device used in this method can capture better images in terms of resolution. The proposed method seems promising to apply to D-Eye database and perceive if this method can be applied to lower-quality datasets. Figure 35 shows the workflow from this method.

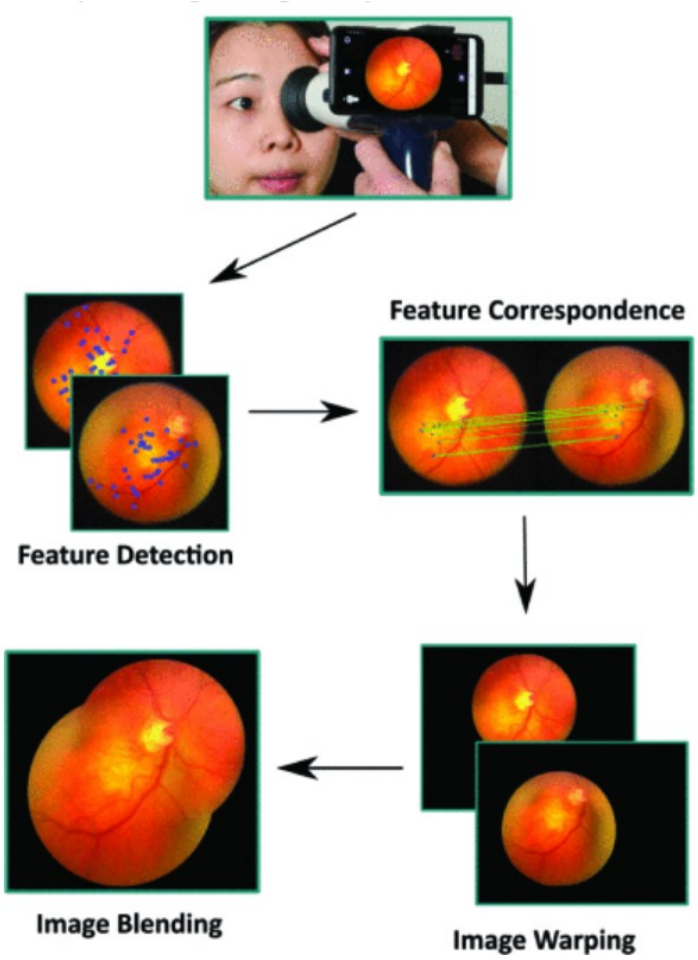


Figure 35: Retinal image stitching workflow – obtained from [78].

For starters, they choose UR-SIFT, an extended version of the scale-invariant feature transform (SIFT) [79], for feature detection since it is capable of better overall performance for the distribution of feature points.

SIFT uses scale-space to find scale-invariant features, the space-scale images are calculated by convolving an input image with a Gaussian kernel. After a Difference of Gaussian (DoG) pyramid is applied to detect regions that differ in color, to apply local extrema detection. The DoG image is calculated by subtracting the scale-space image from another scale-space with a different Gaussian kernel, as depicted in Figure 36, which illustrates this process.

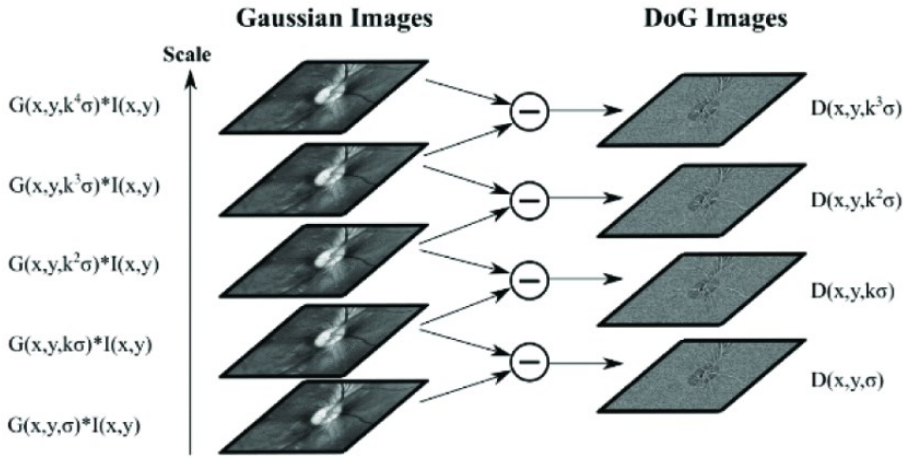


Figure 36: DoG feature extraction – obtained from [78].

The authors used Euclidian distance between the key point descriptors through a brute-force matching algorithm allied with the K-nearest neighbor (KNN) algorithm for the feature correspondence. In addition, they performed a ratio test for the matched keypoints to remove the bad matches, using the Euclidian distance between the closest neighbor to the second closest neighbor. The match will be discarded if the ratio for the neighbor distances is greater than a defined ratio threshold.

The next step is refining the matches with a random sample consensus (RANSAC) [81], removing outliers from discerning low-quality matches that correspond to a minimum quality threshold. Then, an warping image process is applied to match the images into a single one. Initially, the image with the highest sum of keypoint matches is selected to represent the anchor for all others. They will have to perform a transformation in their matrices to correctly match the points for the anchor image plane via homography.

The last step is to blend all warped images with the anchor image, after the previously mentioned process, resulting in a cleaner and seamless image. Two techniques are applied for the blending process: circular gradient masking [80] and Laplacian blending [81]. These two techniques are used together to smooth the final image, making it more natural by softening hard edges and blending them seamlessly with the surroundings. Figure 37 shows the result from a five-image input and the final result from the stitched image.

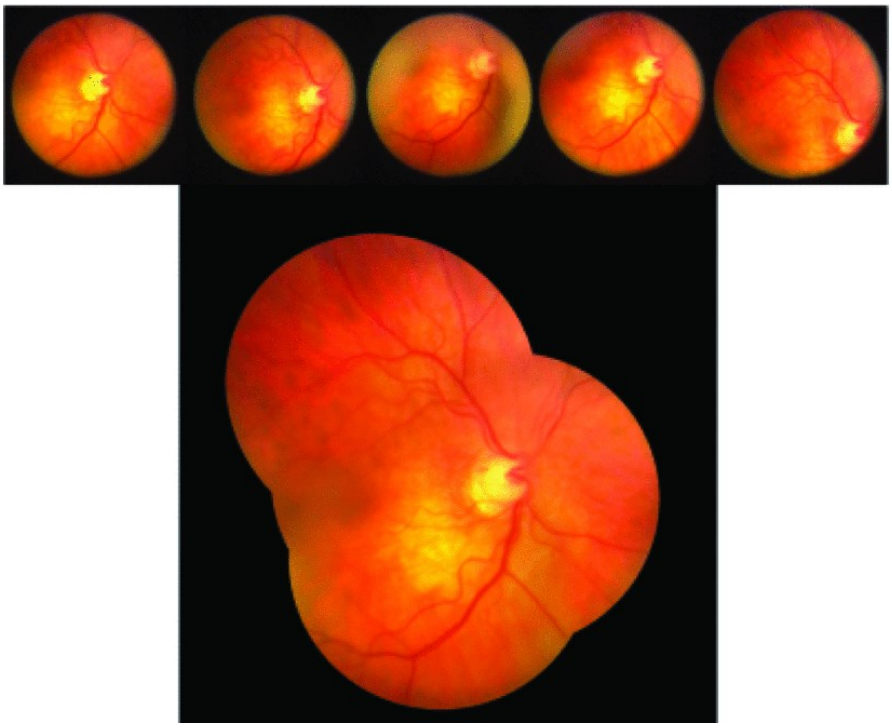


Figure 37: Final image stitched – obtained from [78].

Their proposed method makes it possible to realize that complete retina exposure is possible. However, their imaging samples were of high resolution and rich in detail. For the low-quality imaging that the D-Eye provides, some changes in the algorithm may be needed to perform a similar workflow and have a result close to Figure 37.

4.4.4. Unsupervised Deep Image Stitching (UDIS)

This method uses a framework to comprises an unsupervised coarse alignment stage and an unsupervised image reconstruction stage [82]. The pipeline in Figure 38 presents the main workflow of this framework.

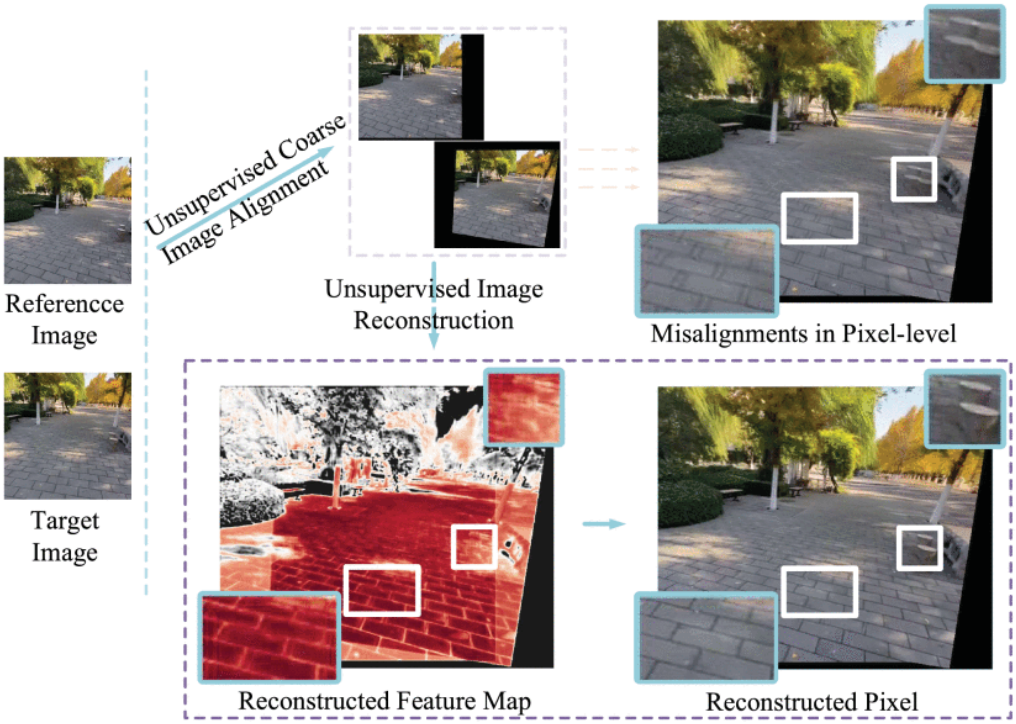


Figure 38: Pipeline from the Unsupervised Deep Image Stitching - obtained from [82].

To optimize the network, the initial phase consists of coarsely aligned input images using a single homography allied with an ablation-based loss. At the same time, a stitching-domain transformer layer is proposed to warp the images with less space occupation. Figure 39 and Figure 40 show how these methods behave.

Figure 39: Ablation-based strategy for homography – obtained from [82].

As shown in Figure 39, the target image can not relate to the reference image because of the no correlation between the two images. Therefore, an ablation process is applied to overcome this, allowing the images to perform homography estimation.

Figure 40: Stitching-domain transformer layer applied – obtained from [82].

Figure 40 shows that the spatial transformer layer has many black pixels for the two images on the left, wasting space in the process. To overcome this issue, they defined a stitching domain as the smallest bounding space of the stitched image, represented as the images on the right, saving space while preserving the quality of the images.

In the second phase, is presented a specific approach to reconstruct the stitched images from features to pixels, which eliminates artifacts through unsupervised image reconstruction. This network consists of a low-resolution deformation branch and a high-resolution refined branch, which respectively learn the rules of image stitching deformation and enhance the resolution. Figure 41 represents the method architecture.

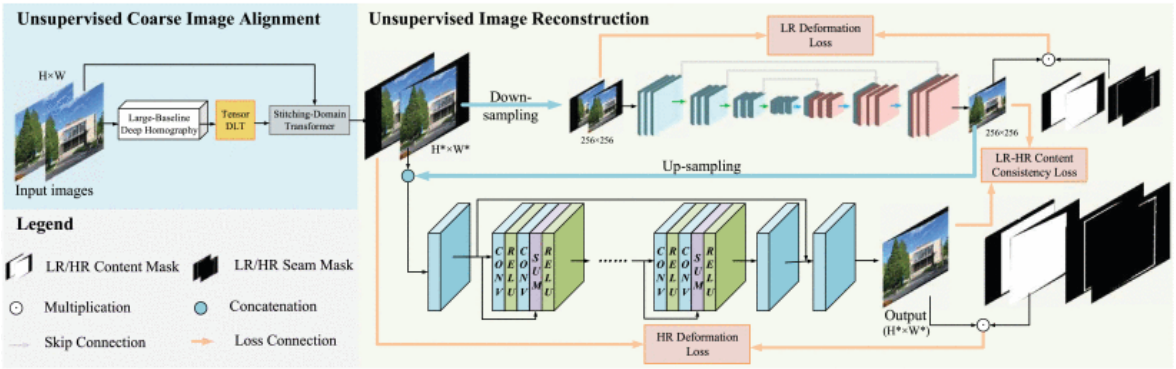


Figure 41: Unsupervised Deep Stitching architecture – obtained from [82].

The final stitched images from their method are compared with other methods for input images represented in Figure 42. The results show that they achieved success in their method by preserving the quality of the stitched images with a little loss on the spatial black pixel residue.

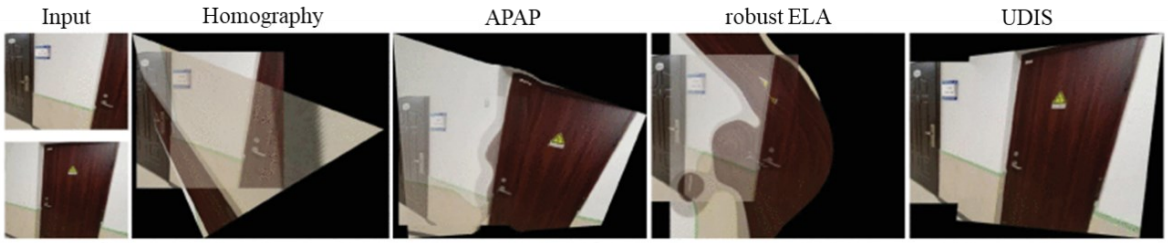


Figure 42: Image stitching comparison – obtained from [82].

From the comparison example given in Figure 42, visually speaking, UDIS has performed better than the other conventional methods. With this, the next chapter will sum up all these methods and evaluate which delivers better performance in summarizing retina images.

5. Results

This chapter presents the inference results from the methodologies presented in Chapter 4. The results from the retina detection will elect the best YOLO method to perform all the necessary detections, and the cropped retina images from them will fed the beforementioned summarization methods. As for the mosaicing, they will be evaluated from the visual results, aiming for the details and gains from the final images.

5.1. Retina Detection

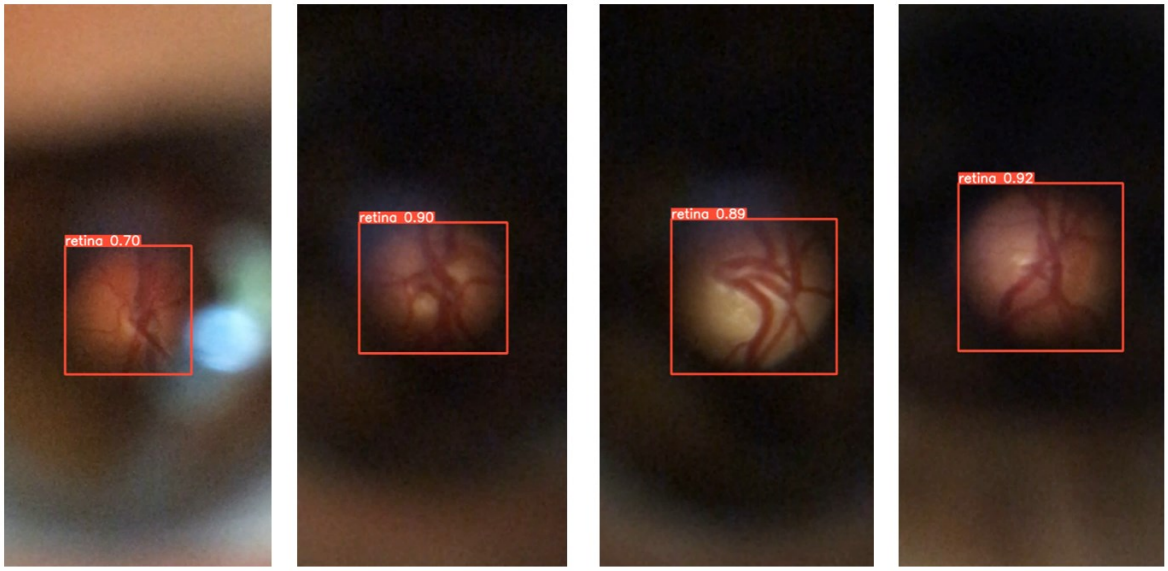
After analyzing the training and evaluation results, it has been determined that YOLOv8 performs better than other models in most cases, achieving high accuracy in detecting retinas while utilizing low computational power. As a result, it has been chosen to be integrated into the workflow. Table 8 displays the current best result from each model for the mean average precision. Although YOLOv5 had a similar result to YOLOv8, the latter has a significantly better user interface, more parameters for tuning, performing better in the inference and having a faster training process. However, YOLOv6 did not perform as well as expected, and a larger dataset may be necessary to realize its potential.

Table 8: YOLOs mean average precision overall results.

Model	mAP(0.5)	mAP(0.5:0.95)
YOLOv5	**99.13**	92.2
YOLOv6	-	-
YOLOv7	95.58	56.74
YOLOv8	98.89	**92.26**

The next step involves conducting new inferences using the same trained model on videos from the private D-Eye dataset to evaluate the model further. Then, using the cropped retina images from the detections, they will feed the summarization method to generate a full retina image. Figure 43 presents the results from the inference of retina detection in four videos from the private dataset. The videos were randomly selected and labeled as Samples 1 to 4.

Among the selected samples, Sample 1 had issues with its detection. This could be attributed to high luminosity during recording and the relatively large distance between the D-Eye lens and the eye, resulting in some frames being too bright and potentially interfering with the detection process. Figure 44 shows a frame with bad detection from the model. However, while the detection was unsuccessful, the cropped images from the correctly detected frames can still be used.

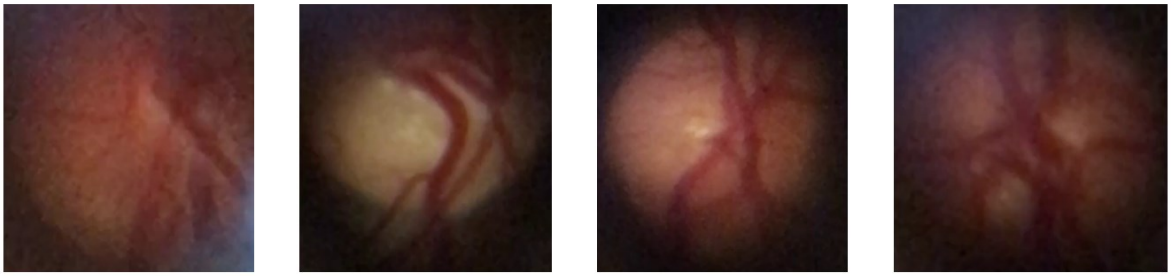


(a) Sample 1 (b) Sample 2 (c) Sample 3 (a) Sample 4

Figure 43: YOLOv8 detection inference

Figure 44: Wrong detection from the YOLOv8 model in Sample 1.

After the inference in the samples, a total of 2666 cropped images were acquired and are ready to feed the summarization methods. Figure 45 presents how the images are after being cropped.



(a) Sample 1 (b) Sample 2 (c) Sample 3 (a) Sample 4

Figure 45: Retina cropped images.

5.2. Retina Summarization

All previous methods were tested with the best YOLOv8 crops for summarization. To select these crops, a new inference was made in the Sample 3 video, and the chosen ones were evaluated with mAP(0.5:0.95) with a score of at least 0.95. After this process, only 8 crops, 1 to 8, came out as a result and were normalized to 512 pixels, as shown in Figure 46.

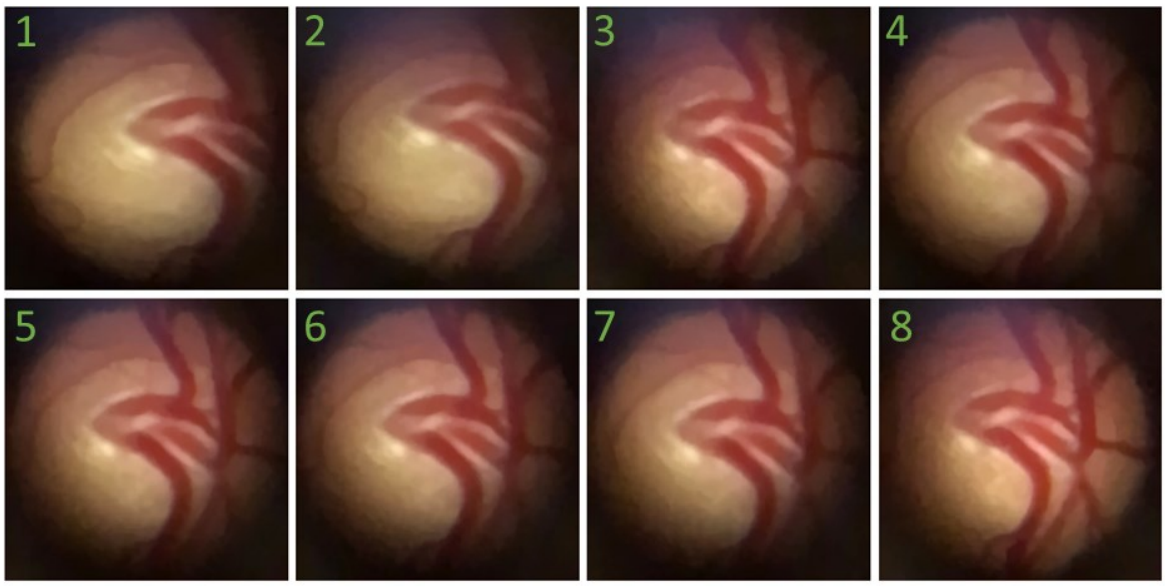


Figure 46: 8 final crops for summarization inference.

Applying the 8 crops together with Deep Image Stitching method [73], the results obtained from this inference are shown in Figure 47, and the method didn't perform well enough to fit the objective of this work.

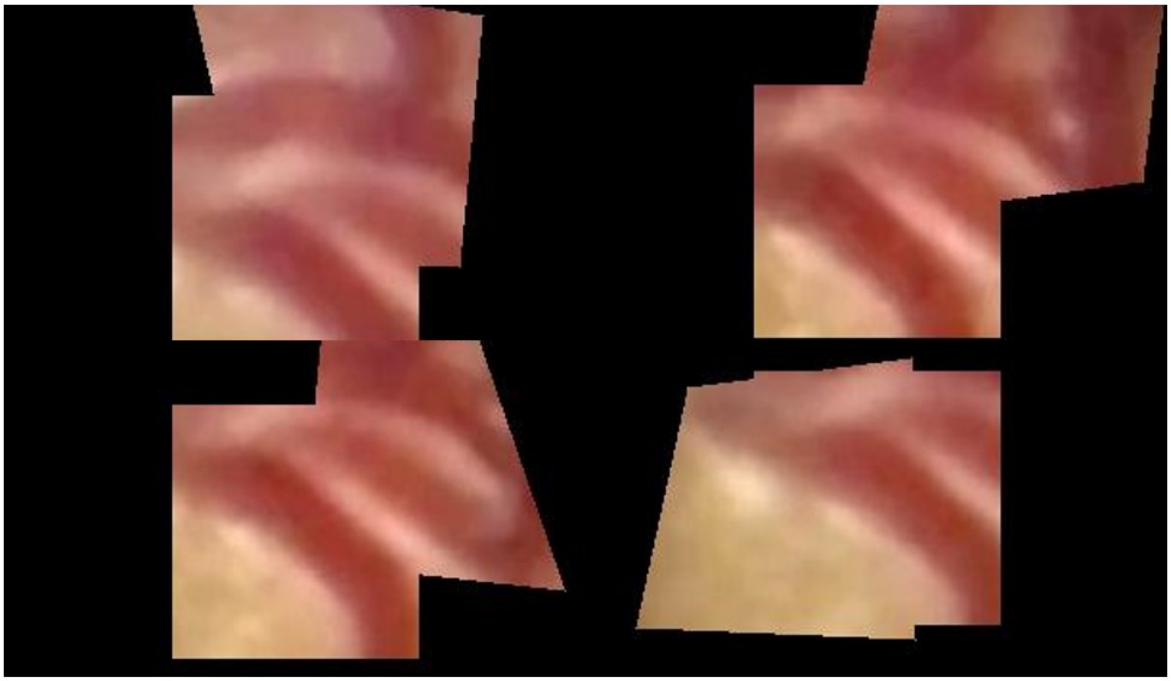


Figure 47: 4 stitches from the Deep Stitching method.

The method seems to find the keypoints only in the center of the images. As a result, the stitching always is performed around the center of the reference image, and this is happening because the refinement network probably is cutting part of the image to ensure that only the matched keypoints are present on the coarsely stitched images, hence, becoming inviable the usage of the final images for a better overview of the retina.

From Super Retina [76], pairs of crops must be inputted once per time. For this result the crops 1 and 8 were selected for being the most far apart from each other. The inference started promising, showing a good amount of keypoints being recorded. Figure 48 shows that this pair had 27 good matches between them, being a reasonable amount for the quality of the images. The final stitched image, Figure 49, could show details from the two images but still not enough to fully expand the retina. For now, it is the best result from the methods approached in this work.

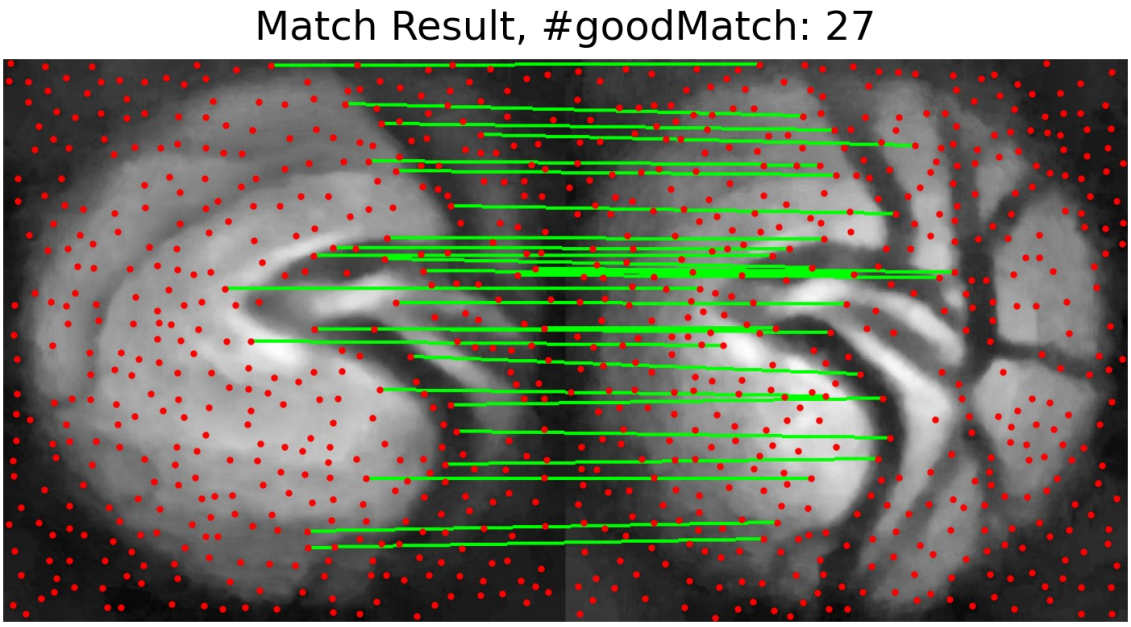


Figure 48: Keypoint registration on Super Retina.

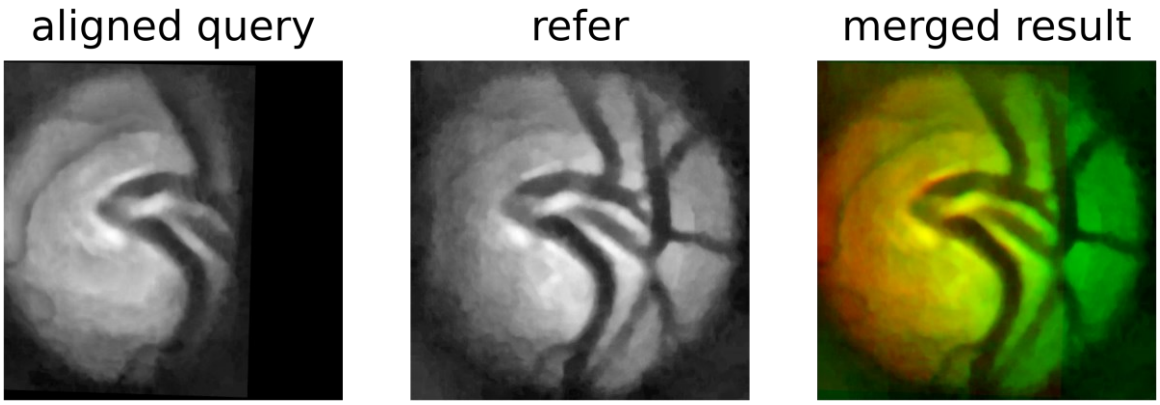


Figure 49: Final stitched image from pair.

The method presented in Multi-image Stitching [78] was the closest in similarity with this work, only diverging in the dataset. Unfortunately, its inference also didn't perform well in the D-Eye dataset since the algorithm could never find its homography due to low keypoint matches. Thus, the homography could not be computed. Figure 50 presents good stitching the method performs with a pair of images.

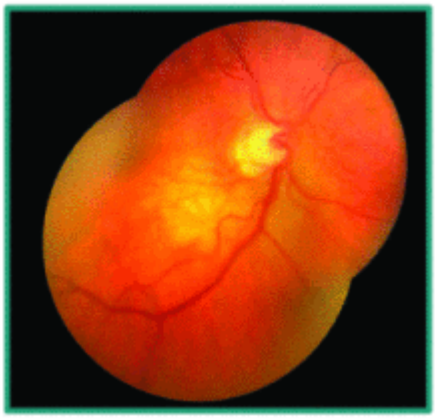


Figure 50: A good example of pair images stitching – obtained from [78].

The last inferred method was UDIS [82]. However, after inputting the crops into the unsupervised network, it also failed to deliver any significant result, giving pairs of densely blurred images, as shown in Figure 51, with no apparent gain of imaging detail from them.

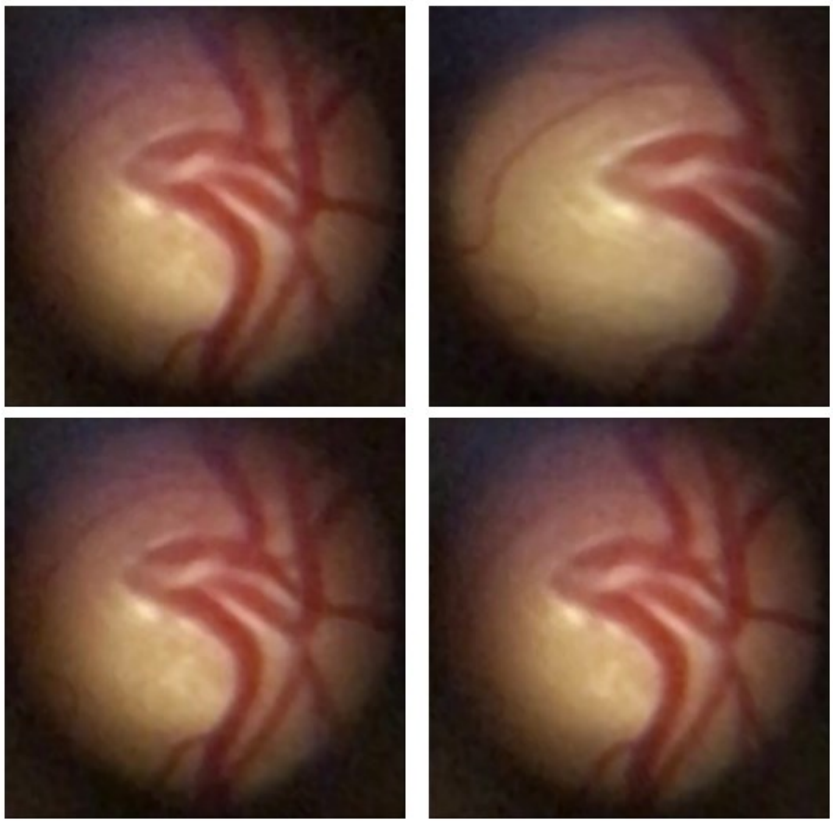


Figure 51: UDIS method for stitching pairs of images

From what was tested and evaluated, Super Retina is presented as the best method to perform stitching with the available D-Eye dataset. A more challenging display of pairs were

fed to the network to continue to evaluate the method even further. Two visually distant images were manually selected to perform stitching, as shown in Figure 52.

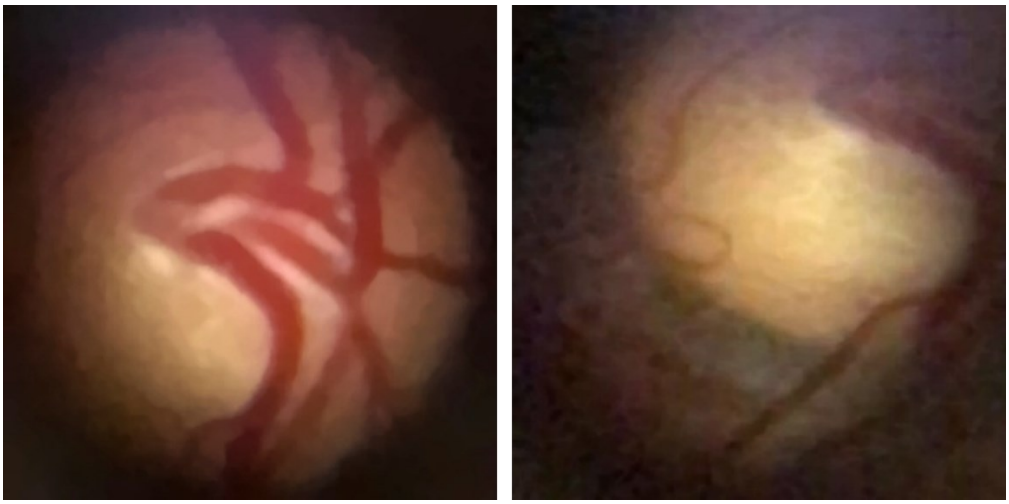


Figure 52: Selected pair of images.

As shown in Figure 53, only 4 keypoint matches were found. As a result, the homography wasn't performed well enough to stitch and provide an overview image, as presented in Figure 54.

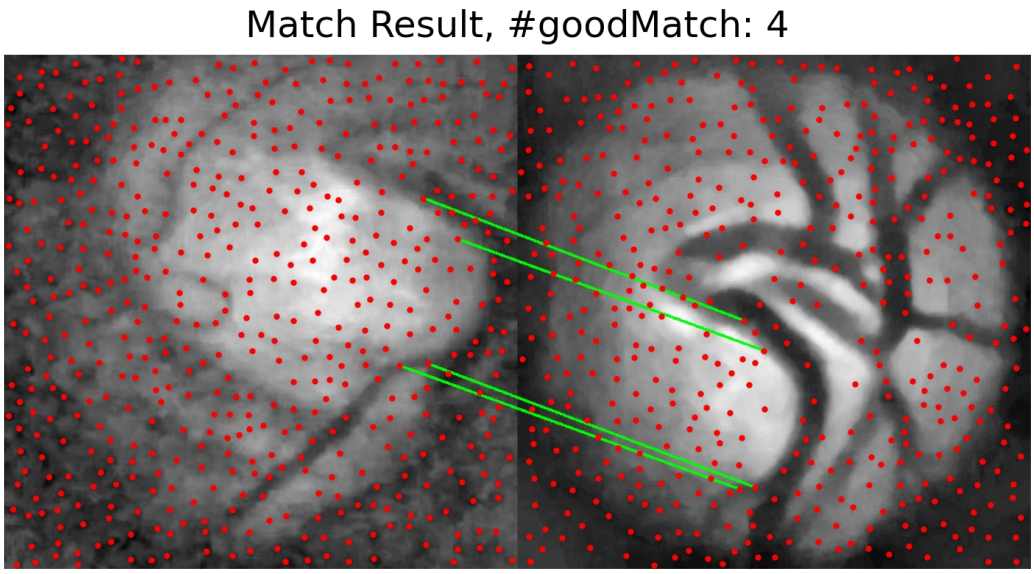


Figure 53: Keypoint match between selected images.

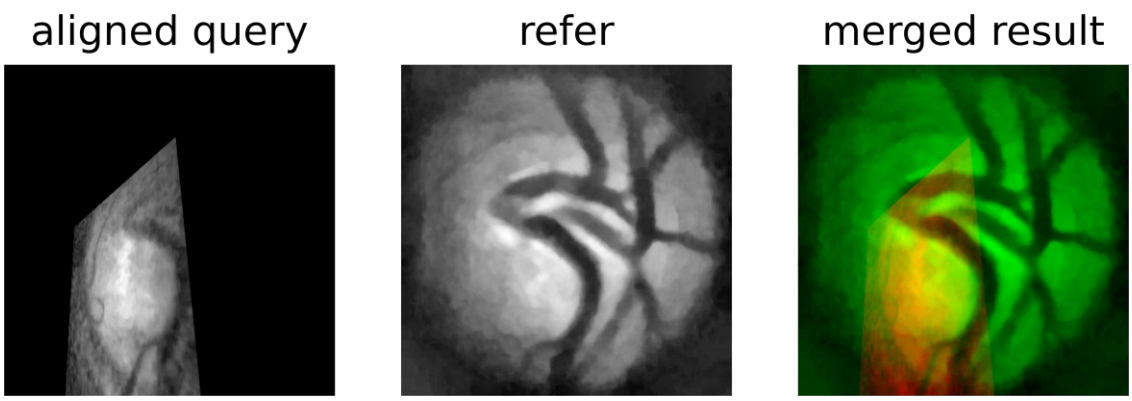


Figure 54: Stitching results from bad homography.

6. Conclusion and Future Work

This chapter will present the final thoughts and an overview of this work. The main idea is to summarize all researched, developed, and tested to simplify medical work on retina detection for glaucoma inference on a mobile device.

After conducting extensive research on image detection and video summarization for retina detection, some points were analyzed, and changes were applied to better fit the purpose of this work.

Initially, video summarization techniques use extensive neural networks that require significant computational power to execute, making it unfeasible for a mobile device to perform this type of activity. Thus, the summarization had to be executed in two stages: image acquisition through frame-by-frame retina detection, followed by an image stitching process presented in the methodology chapter.

The YOLO network [48] proved to be an extremely powerful tool for performing image detection, being able to train and change the conditions of the weights for the trained models in a simplistic way. From the various YOLOs available, the most recent release from Ultralitics, YOLOv8 [52], presented an impressive capability for training with such low dataset input and little computational power demand, making the best network to work in future projects.

Another important point was the difficulty in merging the images obtained from detection into a large final image capable of summarizing the entire video recording. This was largely due to the low resolution of the area of interest provided by the D-Eye dataset. Albeit some methods applied to perform this merge achieved interesting results, such as Super Retina [76], which can be adjusted with network training focused on the D-Eye dataset, and improved to be used in conjunction with other methods, such as Multi-image Stitching [78], which requires less computational power, making it possible to perform a sequential workflow for summarization on a mobile device.

Another way to evaluate the results is to ask for professional assistance regarding the final obtained images. This will help to optimize the methods to be chosen.

For what was proposed in this work, it is believed that a progress was made in terms of what the actual technology can perform with such low-resolution imaging. As stated before, adjustments are needed to acquire more significant results. For this to happen, some points may be in the discussion, such as:

- Develop a Deep Network, especially for D-Eye dataset;
- Enhance the actual available networks models to fit the D-Eye dataset;
- Enhance the quality of the captured videos;
- Train new models in the actual state of the art to perform better on the dataset;
- Develop a metric to estimate the success and quality of the merged images from the methods;
- Exploring other merging techniques for retinal image stitching.

7. Publications

From work developed and presented in this dissertation were also produced the following results:

Correia, T., Cunha, A., Coelho, P. (2023). A Review on the Video Summarization and Glaucoma Detection. In: Cunha, A., M. Garcia, N., Marx Gómez, J., Pereira, S. (eds) Wireless Mobile Communication and Healthcare. MobiHealth 2022. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 484. Springer, Cham. https://doi.org/10.1007/978-3-031-32029-3_14

Bibliography

[1] C. S. Cowan *et al.*, "Cell Types of the Human Retina and Its Organoids at Single-Cell Resolution," *Cell*, vol. 182, no. 6, pp. 1623-1640.e34, Sep. 2020, doi: 10.1016/j.cell.2020.08.013.

[2] A. Russo, F. Morescalchi, C. Costagliola, L. Delcassi, and F. Semeraro, "A Novel Device to Exploit the Smartphone Camera for Fundus Photography," *J. Ophthalmol.*, vol. 2015, pp. 1–5, Jun. 2015, doi: 10.1155/2015/823139.

[3] A. Malhotra, F. J. Minja, A. Crum, and D. Burrowes, "Ocular Anatomy and Cross-Sectional Imaging of the Eye," *Semin. Ultrasound CT MRI*, vol. 32, no. 1, pp. 2–13, Feb. 2011, doi: 10.1053/j.sult.2010.10.009.

[4] J. Zhu, E. Zhang, and K. Del Rio-Tsonis, "Eye Anatomy," in *eLS*, John Wiley & Sons, Ltd, Ed., 1st ed.Wiley, 2012. doi: 10.1002/9780470015902.a0000108.pub2.

[5] C. Hernandez-Matas, X. Zabulis, A. Triantafyllou, P. Anyfanti, S. Douma, and A. A. Argyros, "FIRE: Fundus Image Registration dataset," *Model. Artif. Intell. Ophthalmol.*, vol. 1, no. 4, Art. no. 4, Jul. 2017, doi: 10.35119/maio.v1i4.42.

[6] S. Beatty, M. Boulton, D. Henson, H.-H. Koh, and I. J. Murray, "Macular pigment and age related macular degeneration," *Br. J. Ophthalmol.*, vol. 83, no. 7, pp. 867–877, Jul. 1999, doi: 10.1136/bjo.83.7.867.

[7] A. W. Stitt *et al.*, "The progress in understanding and treatment of diabetic retinopathy," *Prog. Retin. Eye Res.*, vol. 51, pp. 156–186, Mar. 2016, doi: 10.1016/j.preteyeres.2015.08.001.

[8] O. Geyer and Y. Levo, "Glaucoma is an autoimmune disease," *Autoimmun. Rev.*, vol. 19, no. 6, p. 102535, Jun. 2020, doi: 10.1016/j.autrev.2020.102535.

[9] World Health Organization, *World report on vision*. Geneva: World Health Organization, 2019. Accessed: Mar. 16, 2023. [Online]. Available: https://apps.who.int/iris/handle/10665/328717

[10] G. Krishnan, N. Sivapriya, B. Muralidharan, and C. Newcomer, "Ocular Progenitor Cells and Current Applications in Regenerative medicines – Review," *Genes Dis.*, vol. 4, Feb. 2017, doi: 10.1016/j.gendis.2017.01.002.

[11] M. Karakaya and R. Hacisoftaoglu, "Comparison of smartphone-based retinal imaging systems for diabetic retinopathy detection using deep learning," *BMC Bioinformatics*, vol. 21, Jul. 2020, doi: 10.1186/s12859-020-03587-2.

[12] "Digital retinal camera | The Direct Ophthalmoscope for Your iPhone | Portable digital retinal camera | D-EYE." https://d-eyecare.com/ (accessed Mar. 16, 2023).

[13] F. Chollet, *Deep learning with Python*, Second edition. Shelter Island: Manning Publications, 2021.

[14] O. Atila and A. Şengür, "Attention guided 3D CNN-LSTM model for accurate speech based emotion recognition," *Appl. Acoust.*, vol. 182, p. 108260, Nov. 2021, doi: 10.1016/j.apacoust.2021.108260.

[15] J. Lin, S. Zhong, and A. Fares, "Deep hierarchical LSTM networks with attention for video summarization," *Comput. Electr. Eng.*, vol. 97, p. 107618, Jan. 2022, doi: 10.1016/j.compeleceng.2021.107618.

[16] B. Zhao, M. Gong, and X. Li, "Hierarchical multimodal transformer to summarize videos," *Neurocomputing*, vol. 468, pp. 360–369, Jan. 2022, doi: 10.1016/j.neucom.2021.10.039.

[17] S. S. Harakannanavar, S. R. Sameer, V. Kumar, S. K. Behera, A. V Amberkar, and V. I. Puranikmath, "Robust video summarization algorithm using supervised machine learning," *Glob. Transit. Proc.*, vol. 3, no. 1, pp. 131–135, Jun. 2022, doi: 10.1016/j.gltp.2022.04.009.

[18] P. Li, Q. Ye, L. Zhang, L. Yuan, X. Xu, and L. Shao, "Exploring global diverse attention via pairwise temporal relation for video summarization," *Pattern Recognit.*, vol. 111, p. 107677, Mar. 2021, doi: 10.1016/j.patcog.2020.107677.

[19] X. Feng, Y. Zhu, and C. Yang, "Video Summarization Based on Fusing Features and Shot Segmentation," in *2021 7th IEEE International Conference on Network Intelligence and Digital Content (IC-NIDC)*, Beijing, China: IEEE, Nov. 2021, pp. 383–387. doi: 10.1109/IC-NIDC54101.2021.9660579.

[20] S. R. Badre and S. D. Thepade, "Summarization with key frame extraction using thepade's sorted n-ary block truncation coding applied on haar wavelet of video frame," in *2016 Conference on Advances in Signal Processing (CASP)*, Pune, India: IEEE, Jun. 2016, pp. 332–336. doi: 10.1109/CASP.2016.7746190.

[21] M. Fei, W. Jiang, and W. Mao, "Memorable and rich video summarization," *J. Vis. Commun. Image Represent.*, vol. 42, pp. 207–217, Jan. 2017, doi: 10.1016/j.jvcir.2016.12.001.

[22] I. Mehmood, M. Sajjad, S. Rho, and S. W. Baik, "Divide-and-conquer based summarization framework for extracting affective video content," *Neurocomputing*, vol. 174, pp. 393–403, Jan. 2016, doi: 10.1016/j.neucom.2015.05.126.

[23] C. Huang and H. Wang, "A Novel Key-Frames Selection Framework for Comprehensive Video Summarization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 2, pp. 577–589, Feb. 2020, doi: 10.1109/TCSVT.2019.2890899.

[24] W. Zhu, J. Lu, Y. Han, and J. Zhou, “Learning multiscale hierarchical attention for video summarization,” *Pattern Recognit.*, vol. 122, p. 108312, Feb. 2022, doi: 10.1016/j.patcog.2021.108312.

[25] C. Chai *et al.*, “Graph-based structural difference analysis for video summarization,” *Inf. Sci.*, vol. 577, pp. 483–509, Oct. 2021, doi: 10.1016/j.ins.2021.07.012.

[26] S. E. F. de Avila, A. P. B. Lopes, A. da Luz, and A. de Albuquerque Araújo, “VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method,” *Pattern Recognit. Lett.*, vol. 32, no. 1, pp. 56–68, Jan. 2011, doi: 10.1016/j.patrec.2010.08.004.

[27] S. Huang, X. Li, Z. Zhang, F. Wu, and J. Han, “User-Ranking Video Summarization With Multi-Stage Spatio–Temporal Representation,” *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 2654–2664, Jun. 2019, doi: 10.1109/TIP.2018.2889265.

[28] R. Agyeman, R. Muhammad, and G. S. Choi, “Soccer Video Summarization Using Deep Learning,” in *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, San Jose, CA, USA: IEEE, Mar. 2019, pp. 270–273. doi: 10.1109/MIPR.2019.00055.

[29] A. Riahi, O. Elharrouss, and S. Al-Maadeed, “BEMD-3DCNN-based method for COVID-19 detection,” *Comput. Biol. Med.*, vol. 142, p. 105188, Mar. 2022, doi: 10.1016/j.compbiomed.2021.105188.

[30] E. Apostolidis, E. Adamantidou, A. I. Metsai, V. Mezaris, and I. Patras, “Video Summarization Using Deep Neural Networks: A Survey.” arXiv, Sep. 27, 2021. Accessed: Jan. 28, 2023. [Online]. Available: http://arxiv.org/abs/2101.06072

[31] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, “Creating Summaries from User Videos,” in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., in Lecture Notes in Computer Science, vol. 8695. Cham: Springer International Publishing, 2014, pp. 505–520. doi: 10.1007/978-3-319-10584-0_33.

[32] Yale Song, J. Vallmitjana, A. Stent, and A. Jaimes, “TVSum: Summarizing web videos using titles,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA: IEEE, Jun. 2015, pp. 5179–5187. doi: 10.1109/CVPR.2015.7299154.

[33] G. Liang, Y. Lv, S. Li, X. Wang, and Y. Zhang, “Video summarization with a dual-path attentive network,” *Neurocomputing*, vol. 467, pp. 1–9, Jan. 2022, doi: 10.1016/j.neucom.2021.09.015.

[34] T. Hussain, K. Muhammad, W. Ding, J. Lloret, S. W. Baik, and V. H. C. de Albuquerque, “A comprehensive survey of multi-view video summarization,” *Pattern Recognit.*, vol. 109, p. 107567, Jan. 2021, doi: 10.1016/j.patcog.2020.107567.

[35] H. Fu and H. Wang, “Self-attention binary neural tree for video summarization,” *Pattern Recognit. Lett.*, vol. 143, pp. 19–26, Mar. 2021, doi: 10.1016/j.patrec.2020.12.016.

[36] Z. Lei, C. Zhang, Q. Zhang, and G. Qiu, “FrameRank: A Text Processing Approach to Video Summarization.” arXiv, Apr. 12, 2019. Accessed: Jan. 28, 2023. [Online]. Available: http://arxiv.org/abs/1904.05544

[37] K.-H. Zeng, T.-H. Chen, J. C. Niebles, and M. Sun, “Title Generation for User Generated Videos.” arXiv, Sep. 08, 2016. Accessed: Mar. 16, 2023. [Online]. Available: http://arxiv.org/abs/1608.07068

[38] P. Mehta *et al.*, “Automated Detection of Glaucoma With Interpretable Machine Learning Using Clinical Data and Multimodal Retinal Images,” *Am. J. Ophthalmol.*, vol. 231, pp. 154–169, Nov. 2021, doi: 10.1016/j.ajo.2021.04.021.

[39] C. Sudlow *et al.*, “UK Biobank: An Open Access Resource for Identifying the Causes of a Wide Range of Complex Diseases of Middle and Old Age,” *PLOS Med.*, vol. 12, no. 3, p. e1001779, Mar. 2015, doi: 10.1371/journal.pmed.1001779.

[40] D. R. Nayak, D. Das, B. Majhi, S. V. Bhandary, and U. R. Acharya, “ECNet: An evolutionary convolutional network for automated glaucoma detection using fundus images,” *Biomed. Signal Process. Control*, vol. 67, p. 102559, May 2021, doi: 10.1016/j.bspc.2021.102559.

[41] L. Li, M. Xu, X. Wang, L. Jiang, and H. Liu, “Attention Based Glaucoma Detection: A Large-Scale Database and CNN Model,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA: IEEE, Jun. 2019, pp. 10563–10572. doi: 10.1109/CVPR.2019.01082.

[42] N. Venugopal, K. Mari, G. Manikandan, and K. R. Sekar, “Phase quantized polar transformative with cellular automaton for early glaucoma detection,” *Ain Shams Eng. J.*, vol. 12, no. 4, pp. 4145–4155, Dec. 2021, doi: 10.1016/j.asej.2021.04.018.

[43] A. Diaz-Pinto, S. Morales, V. Naranjo, T. Köhler, J. Mossi, and A. Navea, “CNNs for automatic glaucoma assessment using fundus images: An extensive validation,” *Biomed. Eng. OnLine*, vol. 18, Mar. 2019, doi: 10.1186/s12938-019-0649-y.

[44] F. Z. Zulfira, S. Suyanto, and A. Septiarini, “Segmentation technique and dynamic ensemble selection to enhance glaucoma severity detection,” *Comput. Biol. Med.*, vol. 139, p. 104951, Dec. 2021, doi: 10.1016/j.compbiomed.2021.104951.

[45] F. J. F. Batista, T. Diaz-Aleman, J. Sigut, S. Alayon, R. Arnay, and D. Angel-Pereira, “RIM-ONE DL: A Unified Retinal Image Database for Assessing Glaucoma

Using Deep Learning," *Image Anal. Stereol.*, vol. 39, no. 3, Art. no. 3, Nov. 2020, doi: 10.5566/ias.2346.

[46] G. García, A. Colomer, and V. Naranjo, "Glaucoma Detection from Raw SD-OCT Volumes: A Novel Approach Focused on Spatial Dependencies," *Comput. Methods Programs Biomed.*, vol. 200, p. 105855, Mar. 2021, doi: 10.1016/j.cmpb.2020.105855.

[47] N. Gupta, H. Garg, and R. Agarwal, "A robust framework for glaucoma detection using CLAHE and EfficientNet," *Vis. Comput.*, vol. 38, no. 7, pp. 2315–2328, Jul. 2022, doi: 10.1007/s00371-021-02114-5.

[48] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection." arXiv, May 09, 2016. Accessed: Jan. 29, 2023. [Online]. Available: http://arxiv.org/abs/1506.02640

[49] B. R. Silva, "FRAMEWORK FOR LOW-QUAL ITY RETINAL MOSAICING," masterThesis, 2021. Accessed: Mar. 28, 2023. [Online]. Available: https://iconline.ipleiria.pt/handle/10400.8/6752

[50] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Scaled-YOLOv4: Scaling Cross Stage Partial Network," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 13024–13033. doi: 10.1109/CVPR46437.2021.01283.

[51] M. Horvat, L. Jelečević, and G. Gledec, *A comparative study of YOLOv5 models performance for image localization and classification*. 2022.

[52] J. Solawetz, F. JAN 11, and 2023 10 Min Read, "What is YOLOv8? The Ultimate Guide.," *Roboflow Blog*, Jan. 11, 2023. https://blog.roboflow.com/whats-new-in-yolov8/ (accessed Feb. 17, 2023).

[53] C. Li *et al.*, "YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications." arXiv, Sep. 07, 2022. Accessed: Feb. 15, 2023. [Online]. Available: http://arxiv.org/abs/2209.02976

[54] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors." arXiv, Jul. 06, 2022. Accessed: Feb. 15, 2023. [Online]. Available: http://arxiv.org/abs/2207.02696

[55] G. Jocher, A. Chaurasia, and J. Qiu, "YOLO by Ultralytics." Jan. 2023. Accessed: Feb. 16, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[56] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010, doi: 10.1007/s11263-009-0275-4.

[57] "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Development Kit." http://host.robots.ox.ac.uk/pascal/VOC/voc2012/htmldoc/devkit_doc.html (accessed Mar. 16, 2023).

[58] T. Kanstrén, “A Look at Precision, Recall, and F1-Score,” *Medium*, May 19, 2021. https://towardsdatascience.com/a-look-at-precision-recall-and-f1-score-36b5fd0dd3ec (accessed Mar. 15, 2023).

[59] “Google Colaboratory.” https://colab.research.google.com/ (accessed Feb. 13, 2023).

[60] G. Jocher *et al.*, “ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation.” Zenodo, Nov. 22, 2022. doi: 10.5281/zenodo.7347926.

[61] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition”.

[62] Y. Li, J. Ma, Z. Zhao, and G. Shi, “A Novel Approach for UAV Image Crack Detection,” *Sensors*, vol. 22, p. 3305, Apr. 2022, doi: 10.3390/s22093305.

[63] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature Pyramid Networks for Object Detection.” arXiv, Apr. 19, 2017. Accessed: Feb. 14, 2023. [Online]. Available: http://arxiv.org/abs/1612.03144

[64] H. Li, P. Xiong, J. An, and L. Wang, “Pyramid Attention Network for Semantic Segmentation.” arXiv, Nov. 25, 2018. Accessed: Feb. 14, 2023. [Online]. Available: http://arxiv.org/abs/1805.10180

[65] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “YOLOX: Exceeding YOLO Series in 2021.” arXiv, Aug. 05, 2021. Accessed: Feb. 15, 2023. [Online]. Available: http://arxiv.org/abs/2107.08430

[66] Z. Gevorgyan, “SIoU Loss: More Powerful Learning for Bounding Box Regression.” arXiv, May 25, 2022. doi: 10.48550/arXiv.2205.12740.

[67] T.-Y. Lin *et al.*, “Microsoft COCO: Common Objects in Context.” arXiv, Feb. 20, 2015. Accessed: Feb. 15, 2023. [Online]. Available: http://arxiv.org/abs/1405.0312

[68] C.-Y. Wang, H.-Y. M. Liao, and I.-H. Yeh, “Designing Network Design Strategies Through Gradient Path Analysis.” arXiv, Nov. 09, 2022. Accessed: Feb. 15, 2023. [Online]. Available: http://arxiv.org/abs/2211.04800

[69] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, “RepVGG: Making VGG-style ConvNets Great Again.” arXiv, Mar. 29, 2021. Accessed: Feb. 15, 2023. [Online]. Available: http://arxiv.org/abs/2101.03697

[70] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition.” arXiv, Dec. 10, 2015. Accessed: Feb. 16, 2023. [Online]. Available: http://arxiv.org/abs/1512.03385

[71] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks.” arXiv, Jan. 28, 2018. Accessed: Feb. 16, 2023. [Online]. Available: http://arxiv.org/abs/1608.06993

[72] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-Supervised Nets." arXiv, Sep. 25, 2014. Accessed: Feb. 16, 2023. [Online]. Available: http://arxiv.org/abs/1409.5185

[73] L. Nie, C. Lin, K. Liao, M. Liu, and Y. Zhao, "A view-free image stitching network based on global homography," *J. Vis. Commun. Image Represent.*, vol. 73, p. 102950, Nov. 2020, doi: 10.1016/j.jvcir.2020.102950.

[74] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[75] M. Jaderberg, K. Simonyan, A. Zisserman, and koray kavukcuoglu, "Spatial Transformer Networks," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2015. Accessed: Mar. 18, 2023. [Online]. Available: https://proceedings.neurips.cc/paper/2015/hash/33ceb07bf4eeb3da587e268d663aba1a-Abstract.html

[76] J. Liu, X. Li, Q. Wei, J. Xu, and D. Ding, "Semi-Supervised Keypoint Detector and Descriptor for Retinal Image Matching." arXiv, Jul. 16, 2022. Accessed: Mar. 17, 2023. [Online]. Available: http://arxiv.org/abs/2207.07932

[77] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-Supervised Interest Point Detection and Description." arXiv, Apr. 19, 2018. Accessed: Mar. 17, 2023. [Online]. Available: http://arxiv.org/abs/1712.07629

[78] R. Hu, R. J. Chalakkal, G. Linde, and J. S. Dhupia, "Multi-image Stitching for Smartphone-based Retinal Fundus Stitching," in *2022 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, Jul. 2022, pp. 179–184. doi: 10.1109/AIM52237.2022.9863260.

[79] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004, doi: 10.1023/B:VISI.0000029664.99615.94.

[80] P. Cattin, H. Bay, L. Van Gool, and G. Székely, *Retina Mosaicing Using Local Features*, vol. 9. 2006, p. 92. doi: 10.1007/11866763_23.

[81] P. J. Burt and E. H. Adelson, "A multiresolution spline with application to image mosaics," *ACM Trans. Graph.*, vol. 2, no. 4, pp. 217–236, Oct. 1983, doi: 10.1145/245.247.

[82] L. Nie, C. Lin, K. Liao, S. Liu, and Y. Zhao, "Unsupervised Deep Image Stitching: Reconstructing Stitched Features to Images," *IEEE Trans. Image Process.*, vol. 30, pp. 6184–6197, 2021, doi: 10.1109/TIP.2021.3092828.