FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Characterization of Retinal Fluid in OCT Images

David Castanho Terroso

WORKING VERSION

Mestrado em Engenharia Biomédica

Supervisor: Prof. Tânia Melo

Co-Supervisor: Prof. Ana Maria Mendonça

June 24, 2025

Resumo

Abstract

UN Sustainable Development Goals

The United Nations Sustainable Development Goals (SDGs) provide a global framework to achieve a better and more sustainable future for all. It includes 17 goals to address the world's most pressing challenges, including poverty, inequality, climate change, environmental degradation, peace, and justice.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis.

The specific Sustainable Development Goals mentioned have the following names:

SDG 7 Ensure access to affordable, reliable, sustainable and modern energy for all

SDG 8 Promote sustained, inclusive and sustainable economic growth, full and productive employment and decent work for all

SGD	Target	Contribution	Performance Indicators and Metrics
7	7.1	Enhancing the efficiency of SCADA systems can help increase the reliability of solar energy production, facilitating universal access to clean energy.	Percentage of solar plants with improved ...
	7.2	Improving the management of solar plants helps enhance the efficiency and reliability of renewable energy ...	Increase in renewable energy share ...
8	8.1	Enhancing renewable energy infrastructure promotes resilience against climate-related hazards and supports sustainable energy sources.	Increase in resilience metrics ...

Acknowledgements

David Castanho Terroso

"Our greatest glory is not in never falling, but in rising every time we fall"

Confucius

Contents

List of Figures

List of Tables

List of Acronyms

AMD	Age-related macular degeneration
AMI	Anisotropic meta interpolation
BCE	Binary cross-entropy
BM	Brunch's membrane
CNN	Convolutional neural network
CS	Contrast sensitivity
CT	Computed tomography
DME	Diabetic macular edema
GAN	Generative adversarial network
GDL	Gradient difference loss
GT	Ground truth
HR	High-resolution
ILM	Internal limiting membrane
IRF	Intraretinal fluid
MAE	Mean absolute error
MRI	Magnetic resonance imaging
MSE	Mean square error
MS-SSIM	Multi-scale structural similarity index measure
LR	Low-resolution
OCT	Optical coherence tomography
ONL	Outer nuclear layer
PED	Pigment epithelial detachment
PSNR	Peak signal-to-noise ratio
RIFE	Real-time interpolation flow estimation
RGB	Red, green, and blue
ROI	Region of interest
RPE	Retinal pigment epithelium
RVO	Retinal vein occlusion
SR	Super-resolution
SRF	Subretinal fluid
SSIM	Structural similarity index measure

Chapter 1

Introduction

The vision is the human's most important and complex sense, playing a critical role in our orientation in the world [1]. However, the health of the retina, an important part of the eye, can be compromised by multiple diseases, that lead to fluid accumulation in it. The characterization of the fluid present in the retina is important to assess the progression of diseases such as age-related macular degeneration (AMD), diabetic macular edema (DME), and macular edema secondary to retinal vein occlusion (RVO) [2].

AMD affects the macular region of the retina, leading, in later stages, to a significant and permanent loss of central visual acuity, which has a severe impact on the patient's quality of life. In patients with AMD, the formation of new blood vessels can occur, which leak fluid, lipids, and blood into the retina, resulting in the formation of retinal fluid [3]. It is one of the leading causes of visual impairment with an expected effect on 300 million people by 2040 [4].

In patients with diabetes mellitus, DME represents the most common cause of visual impairment, affecting approximately 150 million people worldwide, as of 2015. It is anticipated that this number will increase as the prevalence of diabetes in developed countries is growing [5]. The fluid accumulation is caused by a disruption of the blood-retinal barrier, which allows fluid to accumulate in the intraretinal layers of the macula, resulting in retinal thickening (edema) [6, 7].

Affecting 16 million people worldwide, RVO represents a significant cause of vision loss in older individuals. The occlusion of the retinal vein can result in swelling of the optic disc, which leads to a reduction in visual acuity [8].

The presence of intraretinal fluid (IRF) is a defining criterion of DME and RVO, while two in every three patients with AMD present this type of fluid. The majority of patients with AMD and 30% of the patients with DME and RVO have subretinal fluid (SRF). Pigment epithelial detachments (PED) occurs more frequently in patients with AMD [2].

Therefore, retinal fluids are important for the classification and progression of these diseases, and can be observed through retinal optical coherence tomography (OCT) [2]. OCT is a non-invasive imaging technique that analyzes the light behavior (such as its reflection, absorption, and time-of-flight) to estimate the spatial dimensions of the tissue's structure [9]. This allows for *in vivo* visualization of the individual retinal layers within the posterior segment of the eye. An

OCT is composed of multiple consecutive cross-sectional 2D images that, when stacked, form a volumetric representation of the posterior segment. Each of these two-dimensional images is called a B-scan. In Figure 1.1, these concepts are illustrated, showing a B-scan and the three-dimensional representation of the posterior segment of the eye. The OCT resolution is sufficiently high to assess the tissue integrity, the retinal layers, and the fluids present [10, 11]. There are multiple devices used for the acquisition of OCT volumes, resulting in different image attributes across the same technique, such as inter-slice distance, image quality, and appearance [2].

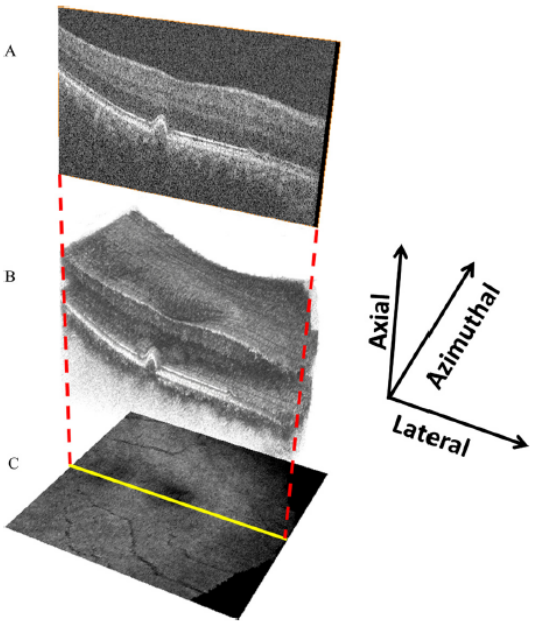

Figure 1.1: OCT B-scans (A), when taken at a fixed distance along the azimuthal axis, form a volume of the posterior segment of the eye (B) [12].

The classification of the fluid is dependent on its location within the retina. There are three different categories: IRF, which is situated in the inner and outer layers of the retina; SRF, positioned between the outer nuclear layer (ONL) and the retinal pigment epithelium (RPE); and PED, which appear beneath the RPE [2]. Figure 1.2 shows the characteristics and positions of these fluids on an OCT B-scan and Figure 1.3 exhibits the retinal layers in the OCT scan of a healthy patient.

By segmenting the fluids detected in the B-scans, their volume can be estimated and used as a progression marker of the mentioned retinal diseases. However, manual segmentation is laborious, expensive, and prone to bias, which motivates the search for automatic methods [11].

In OCT imaging, the precision of the estimated volume is not only dependent on the quality of the segmentation, but also on the inter-slice distance [14]. It is seen in other imaging techniques that the performance of the segmentation is improved when the neighboring slices are used as input. Consequently, the reduction in inter-slice distance and improvement of the resolution along this axis, betters the performance of models that include information from adjacent slices [15]. Given that the inter-slice space is reduced, the estimated segmented volume will also be closer to the real fluid volume.

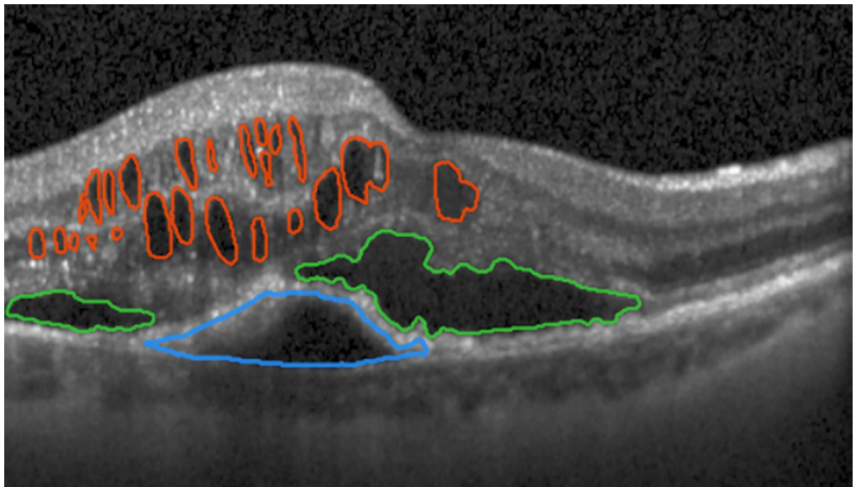

Figure 1.2: The three distinct fluid types on an OCT B-scan: IRF in red, SRF in green, and PED in blue [2].

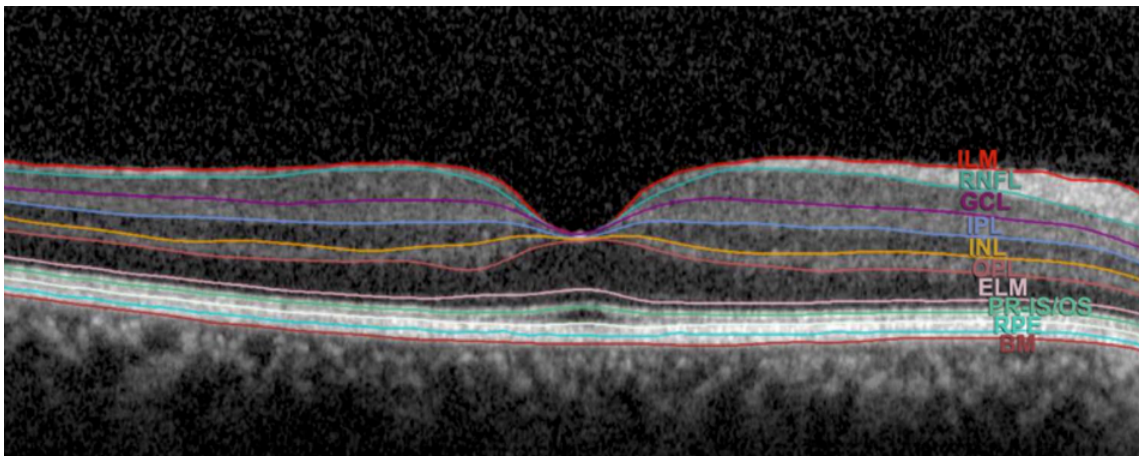

Figure 1.3: OCT scan of the retinal layers [13].

BM, Bruch's membrane; ELM, external limiting membrane; GCL, ganglion cell layer; ILM, internal limiting membrane; INL, inner nuclear layer; IPL, inner plexiform layer; ONL, outer nuclear layer; OPL, outer plexiform layer; PR-IS/OS, photoreceptor inner segment/outer segment; RNFL, retinal nerve fibre layer; RPE, retinal pigment epithelium.

Considering the previous statements, the dissertation general objective is to conduct an analysis of retinal OCT scans, classifying the retinal fluids in three distinct types (IRF, SRF, and PED) and quantifying their respective volumes. Another important objective is to increase the inter-slice resolution of the OCT volumes, with the aim of improving the fluid volume estimation. The specific objectives were determined as follows:

1. Develop different 2D deep learning models for multi-class segmentation of retinal fluids (IRF, SRF, and PED) in OCT volumes.

2. Evaluate the performance of the best segmentation model and estimate the volume of each fluid using the masks predicted by it.

3. Use a generative model for synthesizing intermediate slices in OCT volumes, generating one slice between two real slices in order to improve the inter-slice resolution of the volume, while assessing the quality of these generated images.

4. Investigate the impact of intermediate slices synthesis on the fluid volume estimation by the segmentation models.

Apart from the "Introduction", the dissertation is composed of the following chapters: "Literature Review", "Methods", "Results and Discussion", and "Conclusion". In the "Literature Review" chapter, an analysis is performed on the latest papers in the field of retinal fluid segmentation using 2D deep learning networks, as well as the latest publications on inter-slice resolution enhancement. The "Methods" chapter details the selection of the dataset for the experiments performed during the dissertation, alongside with an insightful description of the experiments performed on fluid segmentation, intermediate slice synthesis, and fluid volume estimation. In the "Results and Discussion" chapter, the results from each experiment are shared, showing the performance of each model in their respective task, while explaining the performance differences between experiments and the relationships between the variables changed and the obtained results, comparing them to the literature. Finally, the "Conclusion" shows the main findings from the experiments performed and suggests directions for further research, while exposing some limitations of the study.

Chapter 2

Literature Review

For this dissertation, research was conducted to find the most recent trends in 2D fluid segmentation of OCT volumes using deep learning and in the use of generative models in the intermediate slice synthesis.

2.1 Fluid Segmentation

In the fluid segmentation state-of-the-art research, articles were retrieved using the methodology of a systematic review. The next subsection details the retrieval process and the criteria for inclusion and exclusion of the articles. "2.1.2 Literature Review" shows the trends on the methodologies utilized for fluid segmentation.

2.1.1 Search Strategy

The search query was defined as: ""OCT" AND "segmentation" AND ("deep learning" OR "CNN" OR "neural network")". Using the query, papers were retrieved from four different databases: 398 articles from PubMed, 105 from IEEE, 125 from ScienceDirect, and 80 from ACM.

In the process of collecting the papers, those published over the previous five years and regarding 2D or 2.5D fluid segmentation in OCT volumes were included. Additionally, conferences proceedings, articles not written in English, and articles for which the full text was not accessible were excluded.

A total of 708 articles were initially identified, of which 133 were duplicates. Afterwards, 575 articles were subjected to screening, based on their titles and abstracts. These articles were analyzed in accordance with the inclusion and exclusion criteria, resulting in the removal of 499 papers. Of the remaining 76 articles for the full-text screening, 20 met the established criteria. These final articles represent the state-of-the art in 2D deep learning fluid segmentation in OCT volumes included in this dissertation.

2.1.2 Literature Review

The selected papers can be divided into two broad groups, according to the type of segmentation: binary segmentation [16, 17, 18, 19, 20, 21], where the fluid is classified in one whole class, and multi-class [22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35], where the segmented fluid is classified in two or more classes (namely IRF, SRF, and PED). We have also considered other criteria to group the papers, such as the segmentation architecture, and the use of retinal delimitation, as shown in Figure 2.1.

In binary segmentation, the approaches to the segmentation problem are simpler, but include both convolutional neural network (CNN) [17, 18, 19, 20, 21] and transformer solutions [16]. The CNN solutions differ among them, depending on the modules that constitute each network, but all are inspired by the U-Net [36]. In Figure 2.2, an instance of a CNN used for binary fluid segmentation is shown.

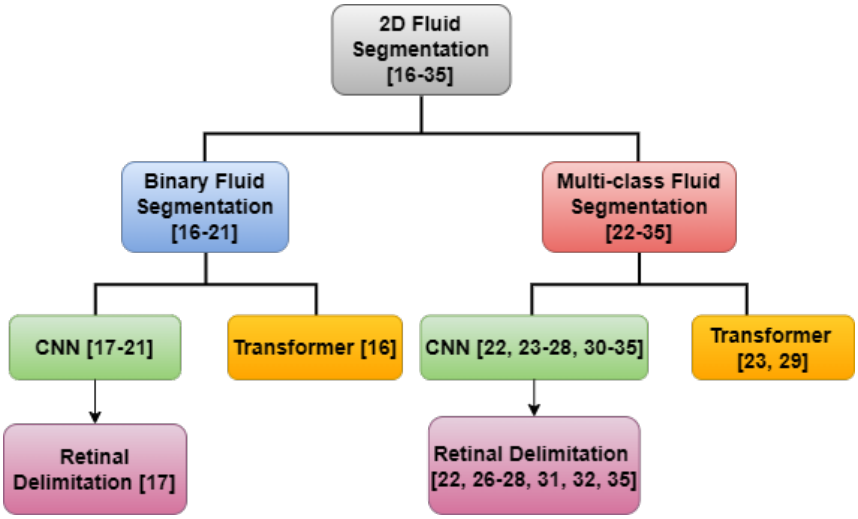

Figure 2.1: Grouping of the articles included in the literature review.

Pawan et al. [17] is the only paper in binary segmentation that restricts the input of the segmentation CNN to the content within the retinal layer. This approach is frequently observed in the papers focused on multi-class segmentation. In this article, this is achieved by performing a retinal layer segmentation and assigning all the values outside the boundaries to zero. The result of this operation is an input for the segmentation CNN. The removal of irrelevant information surrounding the retina simplifies the learning process and improves the model's focus on essential information [35].

In the framework proposed by Liu et al. [18], the slice's fluid mask and distance map are generated. The distance map consists of the predicted distance of each pixel to the background or retinal tissue, with only the values above a specified distance threshold being kept. This is achieved through the use of a double-branched network, where the encoder is the same, while the decoders vary. One encoder is responsible for generating the fluid segmentation map, while the

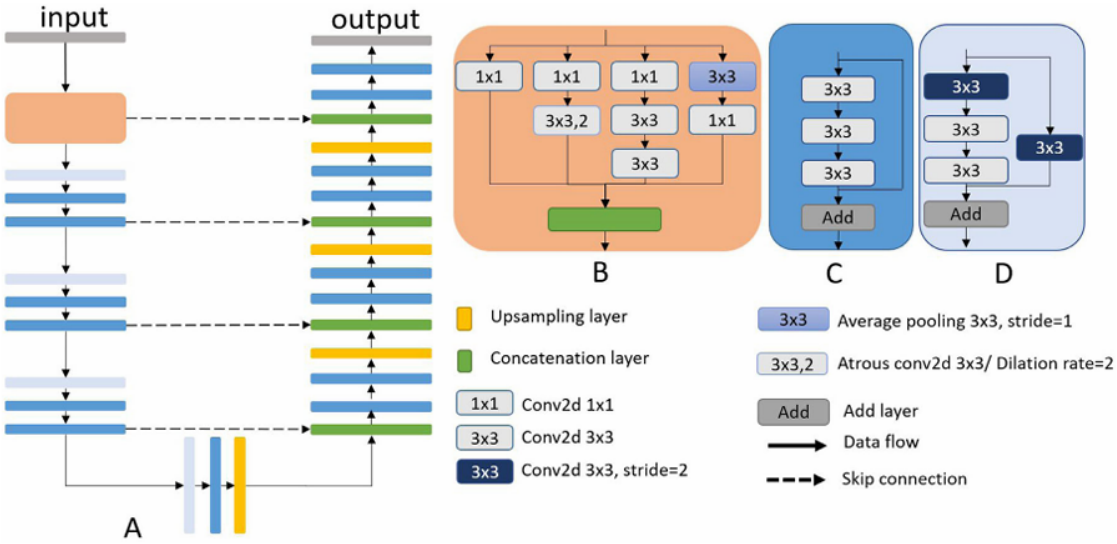

Figure 2.2: Example of a CNN architecture used in fluid binary segmentation. Image A depicts the neural network architecture. B shows the used multi-scale block, while C and D exhibit the residual convolutional blocks [19].

other predicts the distance map. The intersection between these outputs forms the final segmentation. This approach mitigates the issue of inappropriate merging of small and proximate fluid regions, as the distance map branch is better than the fluid segmentation network in discerning the boundaries that delineate fluid regions.

Resorting to generative adversarial networks (GANs), Wu et al. [21] make images from different vendors, visually similar to the images of a singular, specific vendor. Subsequently, a U-Net, which has extensively been trained on images from the specific vendor, is used for segmentation. This approach is intended to reduce the burden of learning the segmentation on multiple vendors by ensuring that all volumes are similar to one in which the segmentation model performs well. Similarly, the multi-class segmentation framework proposed by Li et al. [24] was designed based on the same idea.

CNNs inspired by the U-Net can also be combined with transformers in the context of image segmentation. While CNNs capture the information from local receptive fields, visual transformers integrate features from global receptive fields. Despite being more prevalent in multi-class segmentation frameworks, in this paper by Quek et al. [16], the visual transformers are located between the encoder and decoder paths, thus incorporating features from both receptive fields in the encoding branch.

The majority of the papers included in this review perform multi-class segmentation models, therefore presenting more diverse implementations. While all these articles segment two or more fluids, Hassan et al. [28] and Padilla-Pantoja et al. [33] also segment other biomarkers. Similarly to binary segmentation, the multi-class segmentation papers can also be divided according to the presence [23, 29] or absence [22, 24, 25, 26, 27, 28, 30, 31, 32, 33, 34, 35] of transformers in the segmentation network. All the papers that have transformers in their framework, combine them

with CNNs.

Similar to what was developed in Quek et al. [16], Liu et al. [23] have integrated transformers in the bottleneck section of a segmentation network inspired by the U-Net. Liu et al. [23] utilize two networks for the segmentation: one for coarse segmentation and other for the refinement of the results from the first. Both networks are similar to the U-Net, but in the refine branch, a transformer is included. Its purpose is to provide features from global fields, compensating for the deep features that are used as input in this branch. In contrast, Zhang et al. [29] replaced the CNN encoder with a transformer encoder, exploiting its modeling capacity with self-attention.

The limitation of the input to the region within the retinal layer, ignoring what is outside of it, is seen in many of the multi-class papers [22, 26, 27, 28, 31, 32, 35], similarly to what was done in Pawan et al. [17]. There are various approaches for this delimitation, with some using CNNs trained for the segmentation of the retinal layers or the retina [32, 35], and others using algorithms leveraging on the noticeable transition between the retinal layers and its background [17, 22, 26, 27, 28, 31].

The retinal delimitation is conducted as a separate process from the fluid segmentation. In [22, 26, 27, 28, 31, 32], the retinal layer is segmented prior to the fluid segmentation, conditioning the input of the fluid segmentation network and simplifying the learning process. However, the retinal delimitation can also limit the final segmentation by intersecting the network's output, limiting the segmentation results to the boundaries of the retinal layer, as observed in Mantel et al. [35].

The fluid segmentation network input is conditioned in multiple ways. In Xing et al. [31], the image is cropped to fit its region of interest. [22, 27, 32] combined the B-scan with the retinal delimitation result, either through concatenation or along another channel. In [17, 26, 28] the information outside the retinal layer is set to zero and ignored.

Contrasting with the work of Liu et al. [18] who used a CNN to output a distance map (relative to the background or the retinal tissue), Tang et al. [32], and Rahil et al. [22], inspired by the work of Lu et al. [27], calculate a relative distance map to combine with the input slice in a CNN. Starting with the retinal delimitation, the relative distance to the internal limiting membrane (ILM) is calculated for each pixel located between the ILM and the Bruch's membrane (BM) (see Figure 1.3). This map provides information about the relative position of each pixel to the ILM, influencing their classification. An example of such framework can be seen in Figure 2.3.

Regarding the segmentation CNNs adopted by the analyzed papers, most are directly inspired by the U-Net, to which changes are done, when considering the objectives of each study. Examples of such changes are the introduction of blocks (such as residual [23, 26, 28, 29, 33, 35]), and modules (like atrous sampling pyramid pooling [26, 28, 30, 34]), which makes the network distinctive. However, some papers use other variations of the U-Net that are also popular: the Deeplab [37] in Li et al. [24] and Hassan et al. [28], and the VGG [38] in Hassan et al. [26] and Padilla-Pantoja et al. [33].

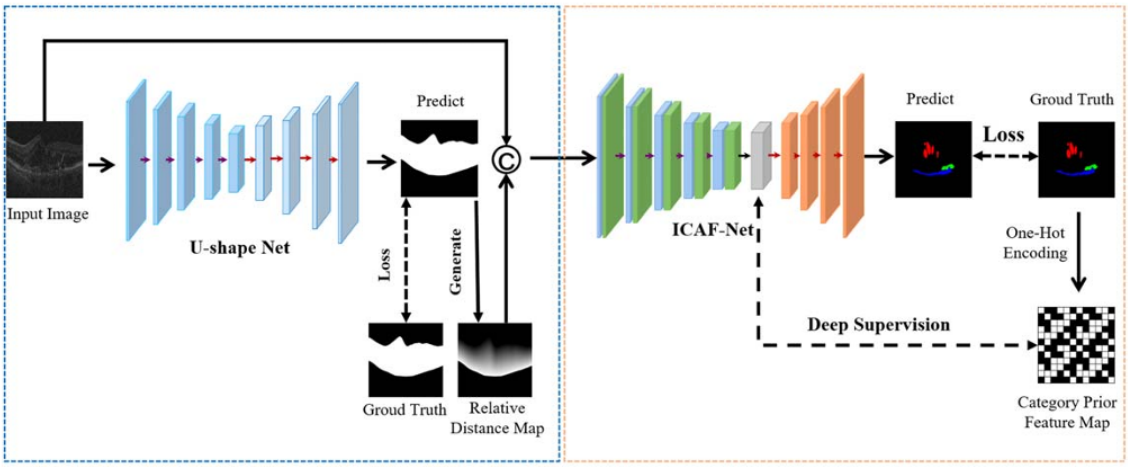

Figure 2.3: Example of a framework that includes delimitation of the retinal layer and a relative distance map (left side). The generated map is included in the segmentation network (denominated ICAF-Net, by the authors) [32].

2.2 Intermediate Slice Synthesis

For many years, there have been attempts to improve the resolution of OCT exams using computational methods, a process called super-resolution (SR). In 3D applications, such as magnetic resonance imaging (MRI), computed tomography (CT), and OCT, SR can be done intra-slice, which improves the resolution of each slice in the volume along one plane, or inter-slice, bettering the resolution of the volume along one axis, by generating one or more slices between a pair of original ones. Some frameworks may even contain both approaches [39].

The use of GANs to generate slices between other known slices is commonly used technique in MRI and CT, but with few examples in OCT [39]. The systematic literature review performed by Ibrahim et al. [40], which analyzed the latest trends in the use of generative models in medical data, only presents one example of GAN for inter-slice resolution improvement in OCT volumes [14]. In this imaging technique, the use of GANs is mainly done for the generation of OCT images and conversion between different vendors [40].

In the following subsection, the state-of-the-art architectures used for the improvement of inter-slice resolution are presented.

2.2.1 Architectures

Given the lack of examples in OCT imaging, it was considered appropriate to study works from other imaging techniques, such as CT, MRI, and even video, given that the working principle is the same across them. The selected papers that are applied to medical images can be classified into three distinct categories: inter-slice SR, which leverages information from adjacent slices to generate one or more intermediate slices [14, 41, 42, 43]; intra-slice SR combined with inter-slice SR, which improves the resolution of the slices from orthogonal planes and combines them with

the results of inter-slice SR [44, 45, 46, 47]; and SR applied directly in 3D volumes, utilizing three-dimensional convolutions in the generation process, which incorporates the information along all the axes from multiple slices simultaneously [48, 49, 50, 51].

López-Varela et al. [14], present an inter-slice SR framework based on a GAN (inspired by the ResNet) for the generation of three B-scan slices between two known slices. The GAN training process, as illustrated in Figure 2.4, begins with the generation of an intermediate slice (Central Fake) located between two original B-scans (Pre and Post), which are separated by another original one (Central). The Central B-scan will serve as the ground truth (GT) and will be used for the assessment of image quality generated by the network. Subsequently, the network generates other two slices: one between the Pre and Central slices, designated as Pre-Central fake, and another between the Central and Post slices, named Post-Central fake. As the mentioned generations lack a corresponding GT, the network performance is regulated by using these two new synthetic slices to generate an additional Central Fake (Central Fake 2), which is then compared to the true Central. Consequently, if the generation of Pre- and Post-Central fakes are inadequate, the Central Fake 2 will also be of poor quality, resulting in a higher loss value. During the inference process, one slice is synthesized for every two known B-scans, reducing the inter-slice distance to half of the original value.

The importance of this study comes not only from it being the only study in OCT but also from the approach selected, which is similar to the foundation of the frameworks implemented in other papers.

In a more straightforward approach, Nishimoto et al. [43] utilize a baseline U-Net that uses two spaced slices as input to generate the slices between them. This methodology was tested in the generation of three, four, and five intermediate slices, and obtained better outcomes than those generated through linear interpolation. This approach works particularly well due to the low noise present in the CT scans. In volumes with more noise, due to the presence of metal artifacts, the slices generated using the U-Net are of worse quality those obtained through linear interpolation.

The work by Xia et al. [41] demonstrates the enhancement of inter-slice resolution in MRI, through the utilization of multiple networks and a multi-scale discriminator that considers both the image from a large and a small field of view images. Therefore, the networks of this framework receive two consecutive slices ($x_{z\pm1}$) and attempt to generate one between them (x_z). The generator (G) in this framework attempts to generate the intermediate slice, while the discriminator (D) learns to distinguish the synthesized image from the real images. The loss of these networks is also determined by the similarity of the feature maps for different images (L_{FM}).

The framework contains two other U-shaped networks: one that learns to predict the optical flow and one that learns to predict the depth map for each of the two input images. Parting from the optical flow and depth map of the input images and their linear interpolation ($\bar{x_z}$), another U-shaped network generates the intermediate slice. The result from this network and the image output from the generator, to which is applied Gaussian blurring, are evaluated by the discriminator. Once again, the discriminator evaluates both images at different feature maps, comparing them.

The image generated by the network that uses optical flows and depth ($\hat{x_z}$) has special attention

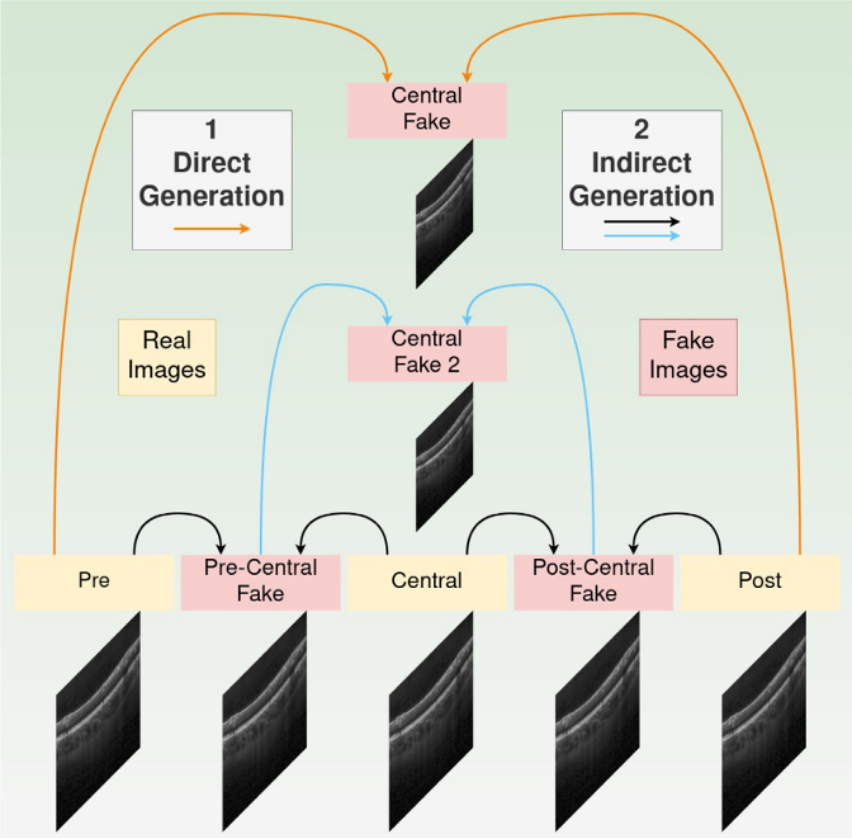

Figure 2.4: López-Varela et al. [14] training process.

to the transitions between images. The point of inputting these images to the discriminator is to incentive it to look for these characteristics in the images output from the generator, a network which has a larger capability of reproducing outputs more visually similar to real images. Similarly, the discriminator also receives blurred images to incentive the generator to output sharper images. A scheme of this framework can be seen in Figure 2.5.

Similarly, Wu et al. [42] improved the inter-slice resolution by training a generator network to output bi-directional spatial transformations instead of producing fake images. The advantage of this process is that it allows the same transformations to be applied to the segmentation masks from the surrounding slices, generating fake masks for the fake slices.

As in Xia et al. [41], the GAN's discriminators also judges the generated images in both a larger and smaller field of view. To evaluate the output at a larger field of view, a global discriminator classifies the image in real or fake. Meanwhile, the classification at a smaller field of view integrates an attention network, which focus on the most useful parts of the image that help the local discriminator and the object identifier.

The local discriminator classifies the image as real or fake based on these smaller features output from the attention network. Meanwhile, the object classifier, which is a much shallower network, verifies if certain structures that are present in the real image also appear in the fake one, parting from the output of the attention network.

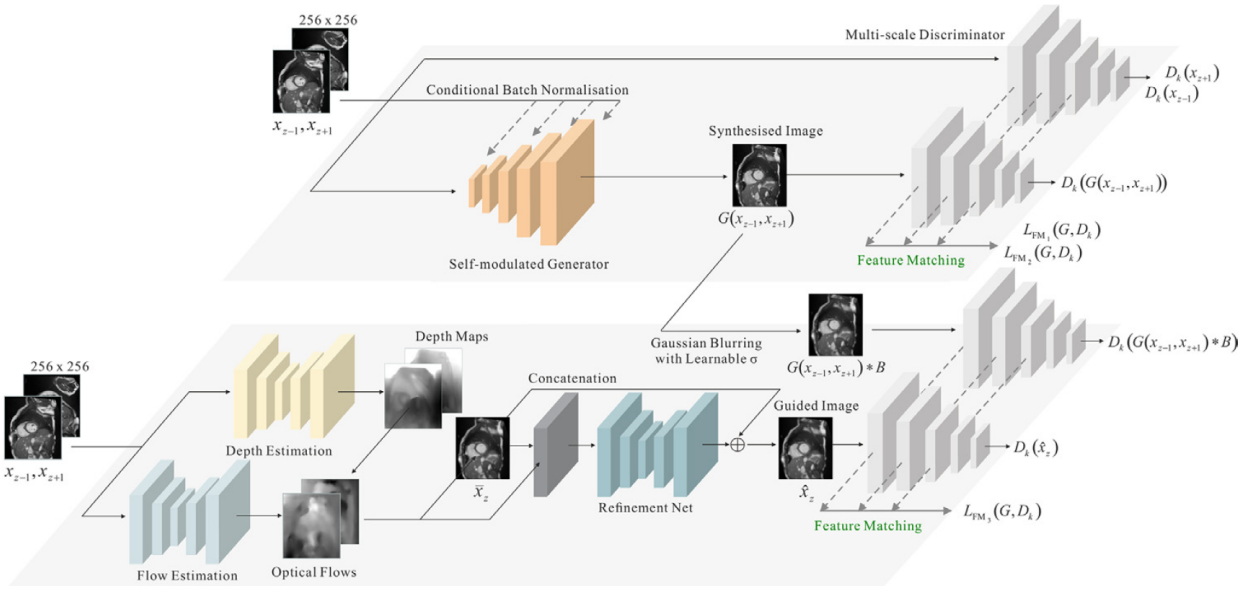

Figure 2.5: Framework developed by Xia et al. [41].

By using the global and local discriminator, the generator is encouraged to output images that closely resemble the real images, both in the overall structure and in the fine details. The object detector ensures that the objects present in the real image are represented in the generated one.

As an example of using intra-slice SR to improve inter-slice resolution, Zhang et al. [44] implemented two networks that enhance the resolution of CT volume's slices in the two planes with the lowest in-slice resolution: sagittal and coronal. These networks increase the resolution only along the axial direction. The models here utilized are based on the anisotropic meta interpolation (AMI) network developed by Peng et al. [52], a benchmark work in the field of medical image SR.

Peng et al. [52] introduced the AMI network, a single image SR network designed to enhance the slice's resolution along the axis of lowest spatial resolution, which is the axial axis, in this case. The AMI network is applied independently to the sagittal and coronal slices, producing in complementary interpolations along the axial direction. A fusion network then combines these two outputs to synthesize high-resolution slices along the axial plane.

The implementation by Zhang et al. [44] extends the use of the AMI network in the images of the sagittal and coronal planes by incorporating a GAN. This GAN receives two adjacent axial slices as input and attempts to generate the intermediate slice, with a framework that is similar to that of López-Varela et al. [14].

The three networks are trained on downsampled CT scans from which every other axial slice is removed. The outputs of the sagittal and coronal AMI networks are compared to the corresponding slices in the original CT scans. Meanwhile, the GAN's output is compared to the GT axial slices from the original CT scan.

Instead of using a dedicated fusion netowrk as in Peng et al. [52], Zhang et al. [44] introduce a loss function that directly compares the outputs of the AMI networks to the output of the GAN. This loss is backpropagated through the generator, encouraging it to output axial slices that are

coherent with the content inferred by the AMI networks. This results in axial images that are not only visually consistent with the original CT slices but also integrate structural information from other anatomical planes.

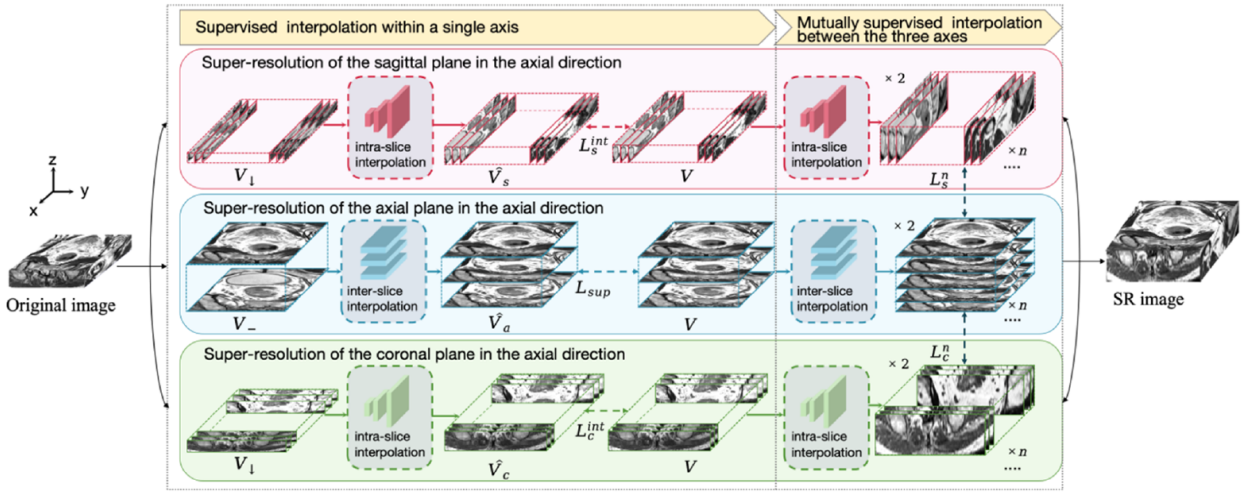

Figure 2.6: Architecture of the method developed by Zhang et al. [44].

A similar approach was done by Fang et al. [45], in which three networks (one for each axis) are trained to generate intermediate slices along one axis with a lower inter-slice resolution. However, during the unsupervised phase, upon each increase in resolution (a process that occurs twice), the information generated by the networks is compared between each other and a loss value that quantifies the performance is calculated, as illustrated in Figure 2.7.

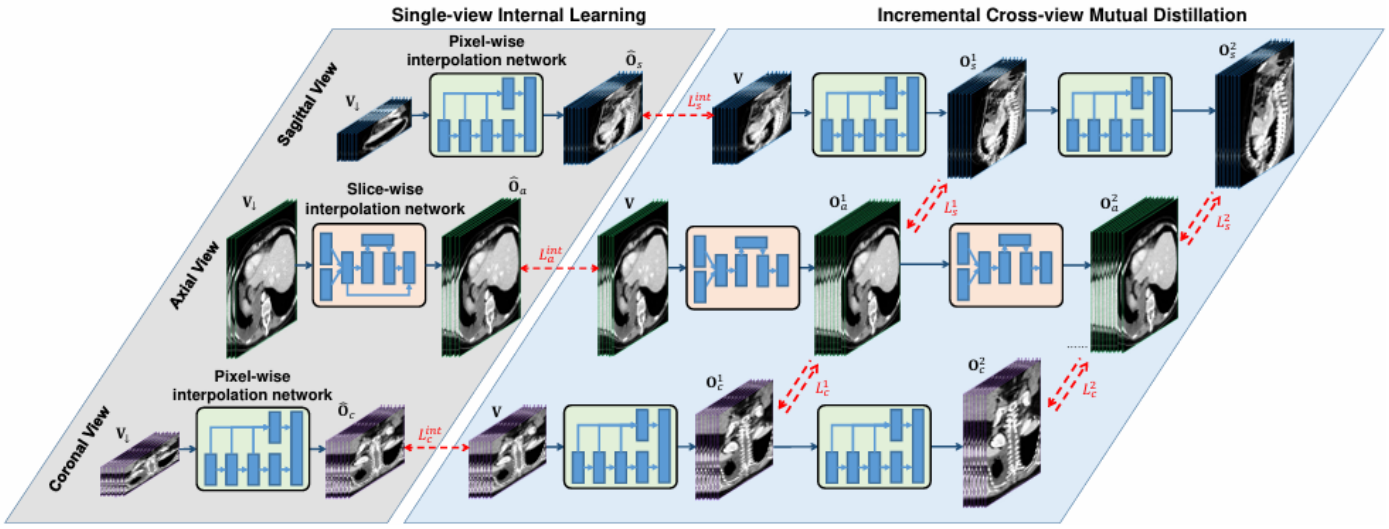

Figure 2.7: Pipeline of the methodology utilized by Fang et al. [45].

Similarly, Nimitha and Ameer [46] use GANs to improve intra-slice resolution and a CNN to improve inter-slice resolution. The method starts by increasing the resolution of low-resolution

(LR) slices, making them high-resolution (HR). Then an intermediate HR slice is generated between every set of two slices, using a CNN.

The same approach was also used by Georgescu et al. [47] to enhance the intra- and inter-slice resolution in CT and MRI scans. Two independently trained CNNs were used in LR volumes. One CNN was tasked with generating HR slices from the LR slices. Concurrently, the other CNN was utilized to reduce the distance between slices by increasing the resolution of the images from the orthogonal plane. By inferring an image with increased resolution along this plane, new intermediate slices were generated, improving the inter-slice resolution along the low resolution axis.

The methodologies utilizing 3D GANs are similar between each other, as they all apply networks based on the GANs implemented in 2D images. As this method already considers the information across all the axes simultaneously, there is no need to use multiple networks for each, as seen in some of the previous approaches. Therefore, the differences between papers mainly originate from the medical imaging technique to which it is applied, the modules that constitute the 3D GANs used, and the datasets used for evaluation [48, 49, 50, 51].

Applications to increase the number of frames per second in a video work based on the same principle as the implementations that increase the inter-slice resolution of the three-dimensional volumes in medical images. In order to increase the number of frames in a second of the video, these frameworks utilize two consecutive frames to generate an intermediate one, which is similar to what is seen in the previous papers, which perform intermediate slice generation in medical images. The key difference between these applications is that, in video, the physical quantity that separates the frames is time, while the physical quantity that separates slices in an OCT (or slices in any other medical imaging technique) is distance [53].

Believing that the concepts that work on video also work on CT and MRI, Gambini et al. [53] implemented a state-of-the-art method of video interpolation to generate intermediate slices in CT and MRI. The method used was real-time interpolation flow estimation (RIFE) [54]. The CNN used in RIFE learns the pixel movements between frames by seeing numerous examples. This approach, called contextual flow, appears as an alternative to optical flow, a method commonly utilized to describe the movement between frames which attempts to predict pixel movements by calculating their movement between consecutive frames. RIFE is also aware of the time difference between frames, which translates to the distance between slices in CT and MRI. To construct the middle image, the network learns how to blend the previous and following image so that the generated intermediate one looks more similar to the expected, combining it with the contextual flow. Lastly, a second network is used in the refinement of the generated image.

Both networks learn based on a loss function that has three components: a photometric loss that determines how close the generated image is to the ground truth; a perceptual loss which evaluates the generated image as the human perception would; and a smoothness loss that evaluates how smooth the generated image is [54].

Tran and Yang [55] present an alternative framework to the video frame interpolation. Instead of recurring to contextual or optical flow to understand how the pixels change between images,

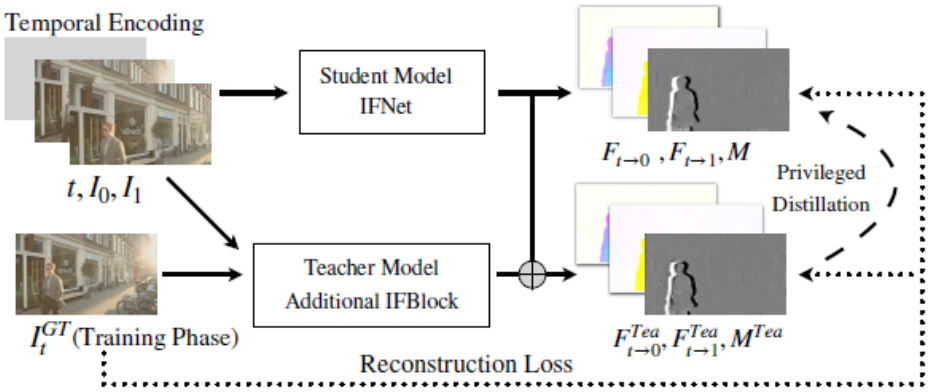

Figure 2.8: Pipeline that describes the RIFE framework. The student model attempts to generate the intermediate frame, while the teacher refines the frame generated by the student so that it looks more similar to the middle frame. The results from both networks are evaluated on the reconstruction loss [54].

two GANs are used to predict the intermediate frame. The first generator receives as input the previous and following slice of the one desired to segment. The resulting slice is evaluated using the generator loss, which is composed of four components. This loss evaluates the reconstruction of the image when compared to the true image and evaluates how well it fools the discriminator.

The image resulting from the generator is input to the discriminator which is responsible for correctly classifying it as real or fake. The prediction of the discriminator is compared to the true image's label and the loss that evaluates this performance is used in the adjustment of the discriminator weights.

After training the first GAN, the second one is trained, which is a pix2pix [56]. Similar to what is seen in the work of Gambini et al. [53] and Huang et al. [54], where a second network is used in the refinement of the output of the first one, this GAN is responsible for making the image output from the first more similar to the original image. Contrasting with the previous examples where both the generative and refining networks are trained at the same time, the refining network is trained independently and after the training of the first network. The pipeline that describes this framework is shown in Figure 2.9.

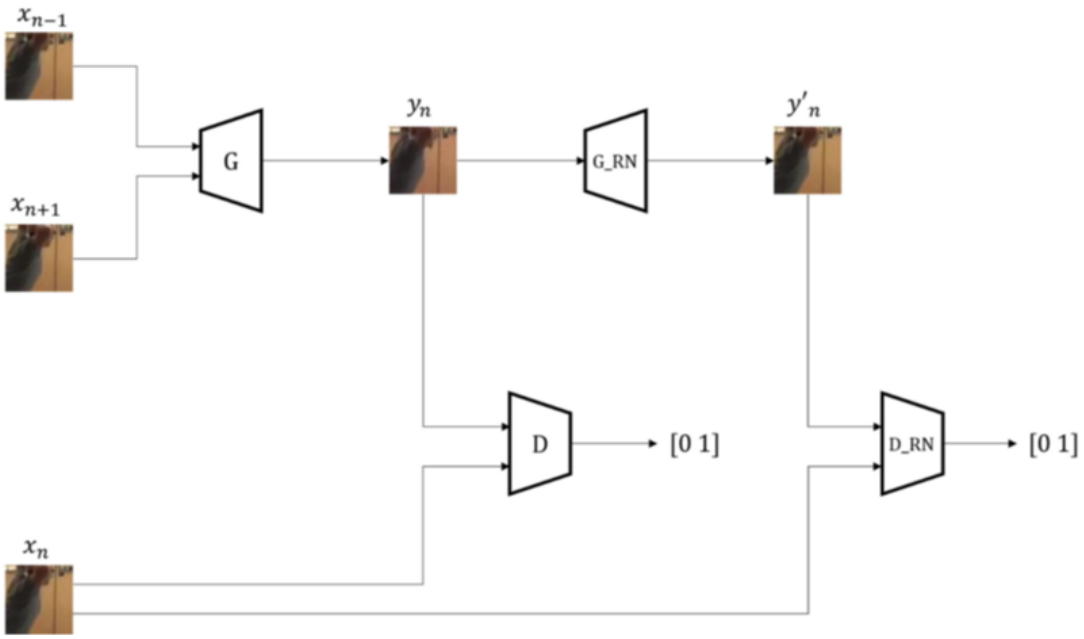

Figure 2.9: Pipeline representing the framework developed by Tran and Yang [55]. The generator that generates the intermediate frame is represented by *G*, while its discriminator is labeled *D*. The pix2pix generator is denoted by *G_RN* and the discriminator is represented by *D_RN*. x_{n-1} and x_{n+1} respectively represent the previous and following frames of the one that is being generated, x_n. y_n is the image generated by the first generator, while y'_n is the image refined by the pix2pix network [55].

Chapter 3

Methods

The Methods section starts with an overview of the dataset selected for the fluid segmentation and intermediate slice synthesis tasks, while regarding the requirements needed for the training of each model and the reasoning behind the selection. Afterwards, it provides an explanation of the experiments that were performed during the dissertation, regarding fluid segmentation, inter-slice generation, and fluid volume estimation, while explaining the methodologies that were implemented.

3.1 Dataset

The application of deep learning to fluid segmentation in OCT volumes requires a large number of images annotated with the three retinal fluids for the training process. The manual segmentation of large amounts of B-scans is a laborious process, which results in a shortage of publicly available annotated OCT datasets. Consequently, the majority of these datasets contain a limited quantity of images.

The dataset selected for this dissertation is the RETOUCH dataset [57]. This dataset consists of 112 OCT volumes, obtained with four different devices: 38 from the Cirrus HD-OCT (Zeiss Meditec), 38 from the Spectralis (Heidelberg Engineering), and 36 from the T-1000/T-2000 (Topcon). The 112 volumes are split into training (70 volumes) and testing (42 volumes). Only those in the training set have annotations of the retinal fluids (IRF, SRF, and PED). For the training and testing of the segmentation models, only the annotated volumes were used.

From the 70 volumes, 24 were obtained with the Cirrus, 24 volumes were acquired with the Spectralis, and 22 were obtained with the two Topcon devices. The number of B-scans per volume, the dimensions of the B-scans, and the axial resolutions vary according to the device utilized to obtain the OCT. The volumes acquired using the Cirrus have 128 B-scans, while those obtained with Spectralis have 49 B-scans. The volumes acquired using Topcon devices (T-1000 or T-2000) have 128 B-scans, but there are two volumes that only contain 64 B-scans. In total, 6838 B-scans were used on the train and test of the segmentation models.

When compared with other renown OCT datasets annotated with retinal fluid, such as the Duke dataset [58], the two datasets from the University of Minnesota [59, 60], and the Lu et al. [27] dataset, the RETOUCH presents a significantly larger quantity of annotated volumes. It also shows more variety since the volumes were obtained using four different devices instead of including volumes from just one device, as done in the mentioned datasets. In Table 3.1, a comparison between the number of annotated B-scans in each of the mentioned datasets is shown, as well as the devices utilized to obtain the OCT images, the diseases of the patients, and the distribution of annotated B-scans per OCT volume.

Table 3.1: Volumes, B-scans per volume, the total number of B-scans, and macular diseases in each dataset.

	DUKE2015 [58]	**UMN2017 [59]**	**UMN2018 [60]**	**LU2019 [27]**	**RETOUCH [57]**
Volumes	10	24	29	528	70[a]
B-scans/Volume	11	25	25	Variable	128 (Cirrus and Topcon[b]), 64 (Topcon[b]), 49 (Spectralis)
B-scans	110	600	725	750	6838
Device	Spectralis	Spectralis	Spectralis	Spectralis	Cirrus, Topcon and Spectralis
Disease	DME	AMD	DME	DME	AMD and RVO

[a] 24 volumes from Cirrus, 22 volumes from Topcon, and 24 volumes from Spectralis.
[b] Two of the training volumes obtained using the Topcon devices have only 64 slices.

For these reasons, the RETOUCH dataset is regarded as a diverse and large dataset, widely used in the literature that aims to perform fluid segmentation using deep learning, as done in [22, 23, 24, 26, 27, 29, 31, 32]. These aspects motivated the selection of the RETOUCH as the dataset that was used for implementing the models for fluid segmentation in OCT volumes in this dissertation.

In intermediate slice synthesis, the 112 OCT volumes that constitute the RETOUCH dataset were used for the training and evaluation of the models. The volumes that do not have segmentation masks can also be included since these masks are not necessary in the intermediate slice generation task.

The consistent number of slices per volume and large quantity of OCT volumes make the RETOUCH dataset suitable for the training and evaluation of the models developed to generate intermediate slices.

3.2 Experiments

In this subsection, the experiments conducted during the dissertation are explained in depth. The subsection begins with a description of how the data was split, followed by the experiments in fluid segmentation, intermediate slice generation, and in fluid volume estimation.

All the experiments were conducted using an NVIDIA GeForce RTX 3080 GPU and the PyTorch machine learning library (version 2.5.1).

3.2.1 Cross-validation

To promote consistency across all experiments, the conditions were held identical. In every experiment, the train-test split followed a 5-fold split, with different splits being used for the segmentation and generation tasks. During training, all the images in three folds were utilized to train the model, while the images from one fold were used in its validation. One fold was reserved and used to compare the performance between the best model from different experiments. Therefore, four training runs are completed in each experiment, rotating the validation fold across runs. The reserved fold consists of the same OCT volumes for all experiments, allowing for further comparisons on data not seen by any of the models.

The images in the fold that was used in validation allowed an insight of how the model was learning. In the segmentation experiments, the instance of the model that achieved the lowest loss on validation data was saved, as this typically indicates the best generalization performance on unseen data. Also, when the model was no longer improving, training could be stopped, saving computational resources.

The dataset was split so that the quantity of each fluid per vendor and the number of volumes per vendor was equally distributed across the folds. By equally distributing the volumes, it is easier to assess the model's learning capability and its behavior towards data with different characteristics (e.g. data from different vendors). To accomplish a fair data split, a custom algorithm was elaborated. This algorithm sought to divide the data in five folds while minimizing the differences of fluid per vendor and the number of slices per vendor in each fold.

A possible distribution of 70 OCT volumes from the RETOUCH dataset, which were used in the training of fluid segmentation, can be seen in Table 3.2. The split was applied to the volumes and not to the slices. The slices of the same volumes must be kept together to prevent data leakage, where similar images, obtained from the same patient, are present both in training and validation, leading to over-optimistic performance metrics.

Table 3.2: Number of OCT volumes per vendor in each fold, considering 5-fold validation.

Vendors	**1**st	**2**nd	**3**rd	**4**th	**5**th
Cirrus	5	5	5	5	4
Spectralis	5	5	5	5	4
Topcon	$4^a + 1^b$	$4^a + 1^b$	4^a	4^a	4^a

Volumes marked with ***a*** consist of 128 B-scans.

Volumes marked with ***b*** consist of 64 B-scans.

In 3.2.2.2 Experiment 2, where the model performs binary segmentation of each fluid, the volumes can be redistributed using the same algorithm, with less bounds. In this experiment, it is relevant to split the volumes in folds based only on their vendors and quantity of the fluid to segment, thus eliminating the restrictions imposed by the quantity of other two fluids. Nevertheless,

the volumes that are in the previously defined reserved fold can not be used nor in training nor in validation.

In the inter-slice generation experiments, the 5-fold split was not done by considering the fluids quantity in each fold. Since in this experiment the test volumes of the RETOUCH dataset were used and there are no fluid masks available, the quantity of fluid in each test volume is unknown. However, one of the folds in the inter-slice 5-fold split is the one reserved in the multi-class segmentation split.

The split was performed by taking into consideration solely the number of slices per device. In this experiments, the characteristic's of each device are important, since each device has a specific inter-slice distance, which is different even across devices of the same vendor and an important characteristic in image generation.

Considering both training and testing volumes of the RETOUCH dataset, there are 38 Cirrus, 38 Spectralis, 13 Topcon T-1000 and 23 Topcon T-2000 (two of which with 64 slices). The fold reserved in the multi-class segmentation task is composed of the following volumes: 4 Cirrus, 5 Spectralis, 3 Topcon T-1000, and 2 Topcon T-2000 (one of which with 64 slices). The volumes remaining for the four folds used in the generation task can be distributed as in the Table 3.3.

Table 3.3: Number of OCT volumes per device in each fold, in the four remaining folds.

Devices	**1**st	**2**nd	**3**rd	**4**th
Cirrus	9	9	8	8
Spectralis	9	8	8	8
T-1000	5	5	5	4
T-2000	3	2	2	2
T-2000^{a}	1	0	0	0

a: volumes with 64 B-scans.

Since the partition is not bounded by the quantity of fluid in each volume, it is possible to compute the best partition by iterating through all the possible combinations. In each combination, the standard deviation of the total number of B-scans in each fold is calculated. The combination with the smallest deviation was used.

Similar to what was done in the fluid segmentation task, three folds was used in training while one was used in validation. The reserved fold was used as a comparison between the best generative models from different experiments.

3.2.2 Fluid Segmentation

The initial experiments of this dissertation focused on training networks on the fluid segmentation task. The goal of these experiments is to determine which segmentation network performs the best in the considered task, which were later required for the fluid volume estimation.

In these experiments, the U-Net [36] was used in the multi-class segmentation of the fluid regions in each B-scan. The U-Net is distinguished by its encoder-decoder structure, which resembles the letter U (see Figure 3.1). In the encoder path, two 3x3 unpadded convolutions are applied to the input image, with each being followed by a rectified linear unit (ReLU) and a 2x2 max pooling operation with a stride of 2, downsampling the image. In each downsampling step, the number of channels is doubled. In the expanding path, a 2x2 up-convolution is used, resulting in the halving of the number of channels. The result is then concatenated with the cropped feature map from the respective contracting path. A 1x1 convolution is applied to the final layer.

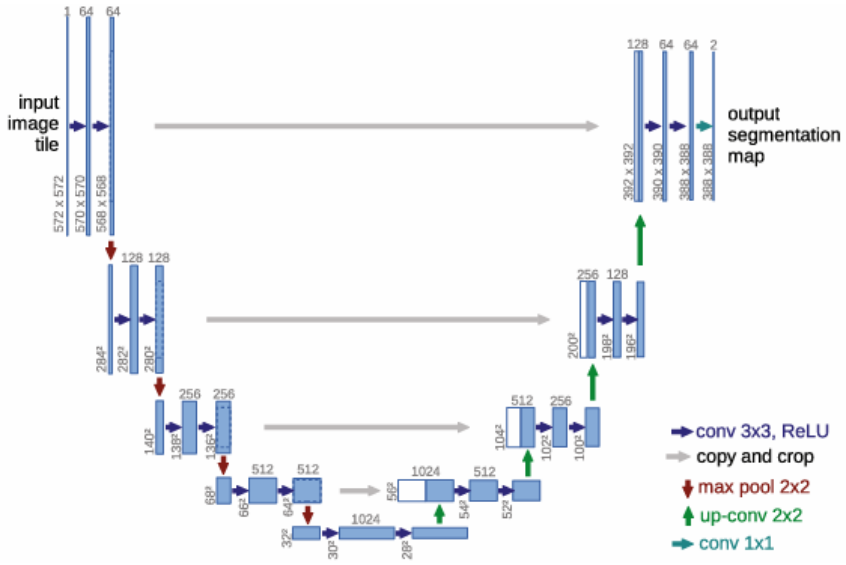

Figure 3.1: U-Net architecture [36].

The evaluation of all networks was conducted using the Dice coefficient. The Dice coefficient is a commonly used metric for evaluating the similarity between two sets. In this context, it was used for assessing the similarity between the segmentation mask generated by the segmentation network and the GT. The equation that describes the Dice coefficient can be seen in Equation 3.1, where A is a set that represents the GT binary mask of one fluid and B is another set that represents the predicted binary mask of the same fluid [61]. Considering a_i and b_i the binary pixels, the Dice coefficient can be rewritten as shown in Equation 3.2. The network that performed the best was selected to estimate the fluid volumes in the fluid volume estimation experiments.

$$\text{Dice}(A,B) = \frac{2|A \cap B|}{|A| + |B|} \tag{3.1}$$

$$\text{Dice}(A,B) = \frac{2\sum_i a_i b_i}{\sum_i a_i + \sum_i b_i} \tag{3.2}$$

The loss function that regularized the training in the fluid segmentation experiments was the same as the one used in Tennakoon et al. [62], whose segmentation model was previously implemented by the authors. This loss is described as seen in Equation 3.3, where λ_D is the weight of

the Dice component, $\mathscr{L}_D$, and λ_{CE} is the weight of the cross-entropy component, $\mathscr{L}_{CE}$, with both weights being 0.5.

$$\mathscr{L} = \lambda_D \mathscr{L}_D + \lambda_{CE} \mathscr{L}_{CE} \tag{3.3}$$

The component $\mathscr{L}_D$ is the Dice loss of the foreground. This translates to how good the model is at detecting and segmenting the fluid present in the B-scans. For any image, where each pixel is associated with an index i, the loss can be described through Equation 3.4, where $s_{i\bar{0}}$ is a binary variable that is 0 when the pixel i belongs to the class 0 (background) and is 1 whenever the pixel i belongs to any class that is not 0 (foreground). $p_{i\bar{0}}$ corresponds to the predicted probability of the pixel i belonging to the foreground. The ε constant is a small value utilized to prevent division by zero.

$$\mathscr{L}_D = 1 - \left(\frac{2 \sum_i s_{i\bar{0}} p_{i\bar{0}}}{\sum_i s_{i\bar{0}} + \sum_i p_{i\bar{0}} + \varepsilon} \right) \tag{3.4}$$

However, this loss component is not enough to correctly label the pixels in their respective classes and, for that reason, the cross-entropy component was used. Due to the large class imbalance in the images, with the background occupying the majority of them, the cross-entropy is balanced by taking into account the number of pixels belonging to each class. The cross-entropy is calculated for each pixel of index i belonging to the image. Then, for each class, the cross-entropy of all pixels in the image is summed, before being divided by the number of pixels that belong to the class. The mean of the values obtained for each class result finally in $\mathscr{L}_{CE}$, as can be seen in Equation 3.5. In this equation, $N = 4$ and is the number of classes, while C is the set of possible classes, $\{0, 1, 2, 3\}$, which corresponds, respectively, to background, IRF, SRF, and PED.

$$\mathscr{L}_{CE} = -\sum_{c \in C} \frac{1}{N} \left(\frac{1}{\sum_i s_{i,c}} \sum_i s_{i,c} \ln p_{i,c} \right) \tag{3.5}$$

3.2.2.1 Experiment 1

In the first experiment, the base U-Net model was trained to perform 2D multi-class segmentation of the retinal fluids in OCT volumes.

This was the most extensive set of experiments, where many variables were tested. Different patch shapes, transformations, and hyperparameters were experimented, until the best training settings were determined. The best settings were then used in 3.2.2.2 Experiment 2. All the experiments were done using the Adam optimizer [63] with a learning rate of 2×10^{-5}.

Experiment 1.1 The model was initially trained on patches of size 256 × 128 (H × W), following the same implementation as the one in Tennakoon et al. [62]. The extraction of patches aims at prioritizing the B-scan information relevant for the segmentation. To achieve this, the patches are not distributed uniformly. Instead, 10 patches are extracted from a random location inside the region of interest (ROI) of each image. The image's ROI is the part of the image where the entropy

is above a determined threshold or where retinal fluid is present. The patches are then randomly transformed by a rotation between 0 and 10 degrees, and horizontal flipping. Of the patches with no fluid, 75% of them were dropped. In Figure 3.2, it is possible to see the overlaying of the fluid masks and the ROI, with a red bounding box signaling a patch that would be used as input to train the U-Net.

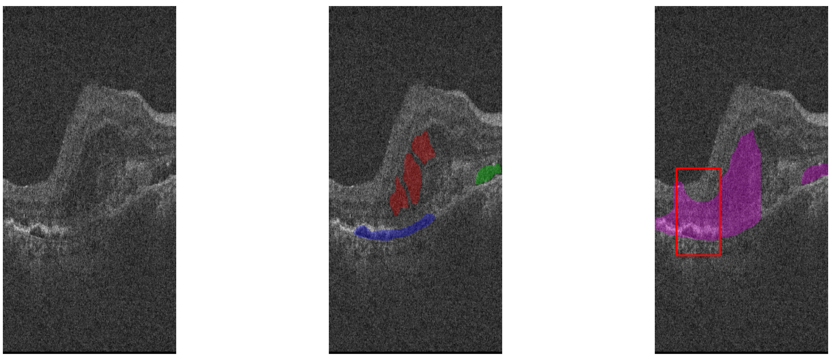

Figure 3.2: Cirrus B-scan (left), fluid masks overlay (middle) with IRF in red, SRF in green, and PED in blue, and the ROI mask overlaid in purple (right). The red bounding box signals a possible 256×128 patch that could be extracted.

In the Experiment 1.1, two sets of four training runs were performed, using each fold as validation. In both sets, the conditions were exactly the same, except the input patches, aiming to understand the effect that the random patch extraction has on the model's performances. The model was trained during 100 epochs with a batch size of 32, with no early stopping.

Experiment 1.2 In Experiment 1.2, the patch shape was changed from 256×128 to 496×512. By using such shape, the model receives a larger context of the B-scan as input, allowing it to learn the anatomic references that characterize and limit the fluids.

The patches used in this experiment were no longer randomly extracted from the ROI. Instead, the patches were extracted from top to bottom so that every section of the image would be present in at least one patch. In a Cirrus B-scan, with shape 1024×512, the first patch would correspond to section from $y = 528$ to $y = 1024$, while the second patch would be from $y = 32$ to $y = 528$. The last patch would then start on the bottom of the image, at $y = 0$, to $y = 496$. A representation of this process can be seen in Figure 3.3. The patches are extracted from top to bottom so that the retina and the fluid would not be split in two patches, damaging the quality of the input data.

In this experiment, due to the larger size of the patches that are being loaded, the batch size had to be reduced from 32 to 16. It was trained on 100 epochs with the same transformations as in the previous experiment and without early stopping.

Experiment 1.3 Another patch shape was experimented in Experiment 1.3. In this experiment, the images from different vendors were resized from their original dimensions to 496×512, the shape of the smaller images of the dataset, obtained with the Spectralis device.

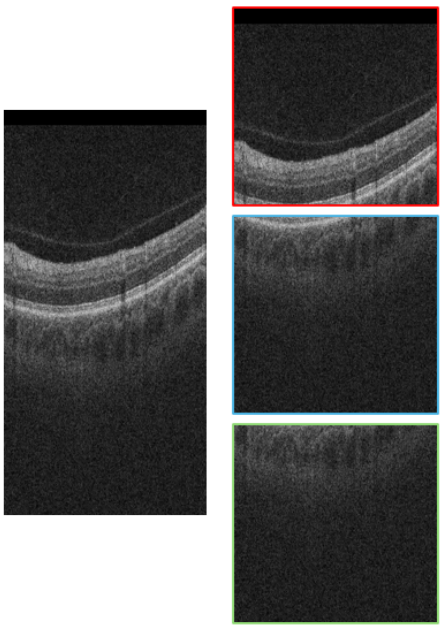

Figure 3.3: Cirrus B-scan and its respective three patches of shape 496×512.

The dimensions of the voxels in the OCT volumes change according to the device that was utilized to obtain the volume, resulting in images with different appearances across the vendors. For example, each voxel in the Cirrus volumes has an height of 1.95 μm, while each voxel in the Spectralis volumes has an height of 3.87 μm. For the same image, these differences in height lead to the same structures appearing bigger in Cirrus B-scans (see Figure 3.4). These differences across vendors makes the learning of the segmentation harder. Therefore, by resizing all the images to the same shape, the structures would have more consistent dimensions across vendors and the voxels roughly translated to the same dimensions, leading to a easier learning process for the model.

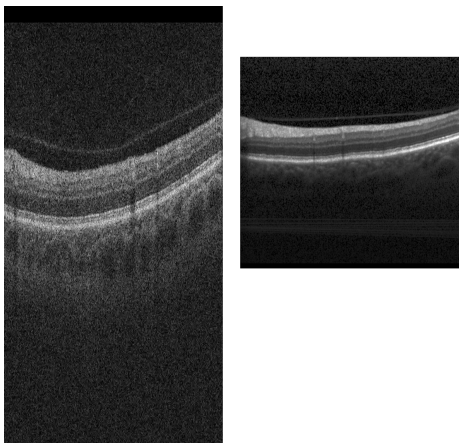

Figure 3.4: B-scan of the retinal layers in different patients, using Cirrus (left) and Spectralis (right) devices. In Cirrus, the retinal layers appear much larger than in Spectralis.

Then, vertical patches, of shape 496 × 128, were extracted from each B-scan. The number of patches extracted from each image was changed, experimenting with four (Figure 3.5), seven (Figure 3.6) and thirteen (Figure 3.7) patches. The advantage of extracting vertical patches is that each image contains both the complete retinal layer and the background. This does not happen in the previous experiments, where the patches are either too small to contain both background and the retinal layers (in Experiment 1.1) or the retinal layers are cropped during patch extraction (in Experiment 1.2, as seen in Figure 3.3).

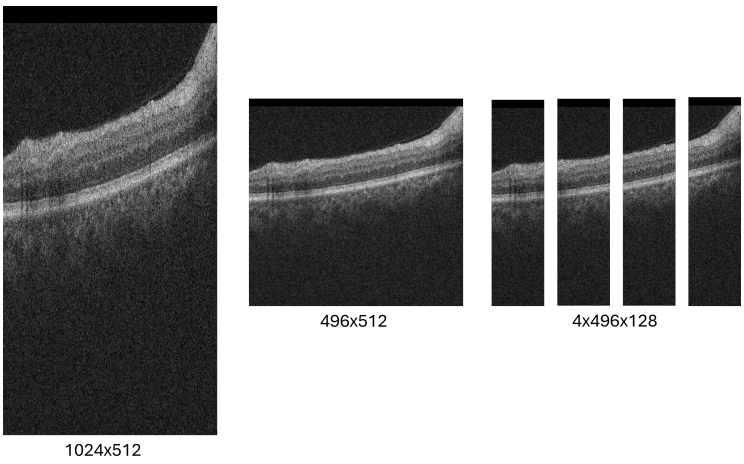

Figure 3.5: Four vertical patches of shape 496 × 128 extracted from a Cirrus B-scan.

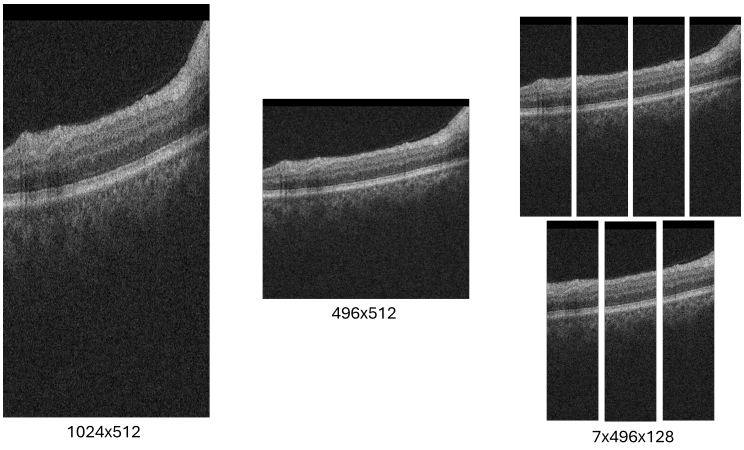

Figure 3.6: Seven vertical patches of shape 496 × 128 extracted from a Cirrus B-scan.

When using four vertical patches, the model was trained both for 100 and 200 epochs, maintaining a batch size of 32 and a maximum rotation of 10°, without early stopping.

Then, the model was trained using seven and thirteen patches for the best and worse performing folds when using four patches, while stopping early in case the validation loss did not progress 25 epochs after the minimum validation loss was encountered. This was used because the model progressed much faster when using seven and thirteen patches per B-scan, as it was trained in a

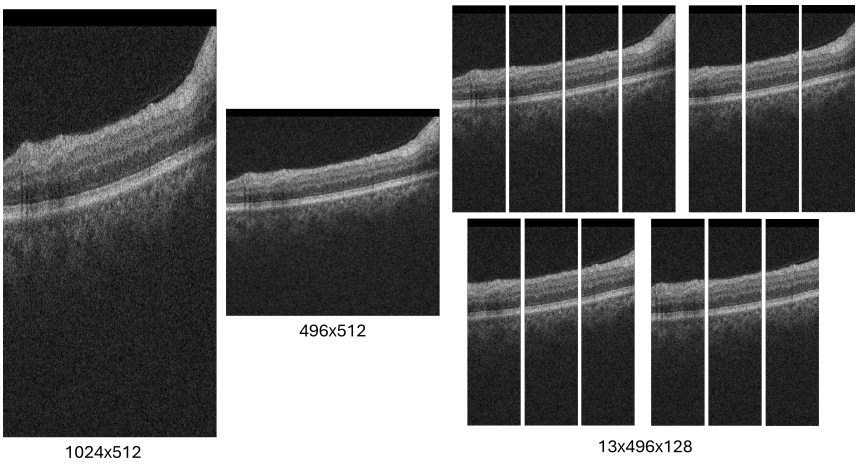

Figure 3.7: Thirteen vertical patches of shape 496 $\times$ 128 extracted from a Cirrus B-scan.

much larger number of images per epoch. Henceforth, it also required more computational power, which further motivated the early stopping.

Using seven vertical patches per image, which was the number of patches that performed best, three different rotations were experimented: no rotation, maximum rotation of 5°, and maximum rotation of 10°. These values were tested for a better understanding of how the rotation of the image affects the segmentation and the model's understanding of anatomic references. The model was trained on a minimum of 100 epochs, after which a patience of 25 epochs was applied. Therefore, if after 100 epochs, the model did not improve its validation loss for 25 consecutive epochs, then training would be interrupted.

Lastly, the model was trained using four patches and a maximum rotation of 5°, using the same early stopping criteria as in the last seven patches runs. This allowed for one last comparison between the two number of patches used in training, under the same conditions. The best model between those trained with four patches and those trained with seven patches was selected to infer on the reserved fold.

3.2.2.2 Experiment 2

The second experiment also involved multi-class segmentation of the retinal fluids. Contrasting with the first experiment, where the segmentation was done using a U-Net, three U-Nets were used in this experiment, one for each fluid. Each U-Net model focused only on the segmentation of one fluid, with one model for IRF, one for SRF, and one for PED.

One of the main problems with multi-class segmentation performed by binary models is the merging of the multiple masks. In this context, multiple fluid classes can be predicted for the same voxel, while only one of those can be correct. In this experiment, two alternatives were explored: order of priority, where the merging of the fluid masks follows a predefined hierarchy that determines which fluid takes precedence, and highest probability, where the assigned class is the one that was predicted with highest probability by its model.

Two different losses were used to regulate the model. Initially, the same loss as the one used in 3.2.2.1 Experiment 1, with only two classes (background and the fluid that would be segmented), and then, the weighted cross-entropy, using weights that balance the larger quantity of background voxels. This last loss is described in Equation 3.5, with $N = 2$.

When using the loss from 3.2.2.1 Experiment 1, each model was trained on two different splits: the split used in the multi-class segmentation experiments and a split created specifically for the segmentation of the fluid that was being segmented.

The balanced cross-entropy loss was tested on the best performing folds of the multi-class and IRF splits, for the segmentation of IRF, in the same conditions. However, since the results were much worse than those obtained with the initial loss, no more folds or fluids were considered.

All the models were trained with seven vertical patches extracted from each B-scan, on a minimum of 100 epochs, after which a patience of 25 epochs was applied, like what was done in the last runs of 3.2.2.1 Experiment 1. Similarly, the random transformations applied to the images consisted of horizontal flipping and a maximum rotation of 5°.

3.2.3 Intermediate Slice Synthesis

The objective of the subsequent experiments is to improve the resolution between slices, thus approximating the estimated fluid volume to the true value.

The intermediate slices were generated using the RETOUCH dataset as training and validation data. In this experiment, subvolumes that consist of overlapping triplets of consecutive slices, sampled with a step size of 1, were used, extracted as shown in Figure 3.8. The first and the last slice of these triplets were used for the generation of the middle slice. Consequently, it is possible to evaluate the generated slice in comparison to the original one, as done in other examples of the literature. For each volume, the number of potential subsets is then determined to be $n-2$, where n represents the number of slices within the same volume.

The generation of slices can be evaluated in specific metrics, as well as through qualitative assessment. To assess the efficacy of the generation model, the model utilized for fluid segmentation could be used for the estimation of the fluid's area in the generated image and to compare the resulting mask with the original image's mask. This comparison can be conducted using the Dice coefficient [14] (see Equation 3.2). However, this metric is insufficient for evaluating the generation performance, as it requires comparisons that encompass the entire slice and not just the fluid region. Examples of such metrics include the mean absolute error (MAE) [14, 42, 51], the peak signal-to-noise ratio (PSNR) [39, 41, 44, 45, 46, 48, 49, 50, 51], and the structural similarity index measure (SSIM) [39, 44, 45, 46, 48, 49, 50, 51].

The MAE and mean squared error (MSE) quantify the errors between the original image and the generated image. For every pixel, the difference between the value in the original image and the generated image is calculated. In MAE, the absolute value of this difference is calculated, and then the mean of all pixels in the image is computed. Meanwhile, in MSE, the difference is squared before computing the mean. In Equation 3.6 and Equation 3.7, MAE and MSE are described, respectively, where x_i is the intensity of the pixel of index i in the predicted image,

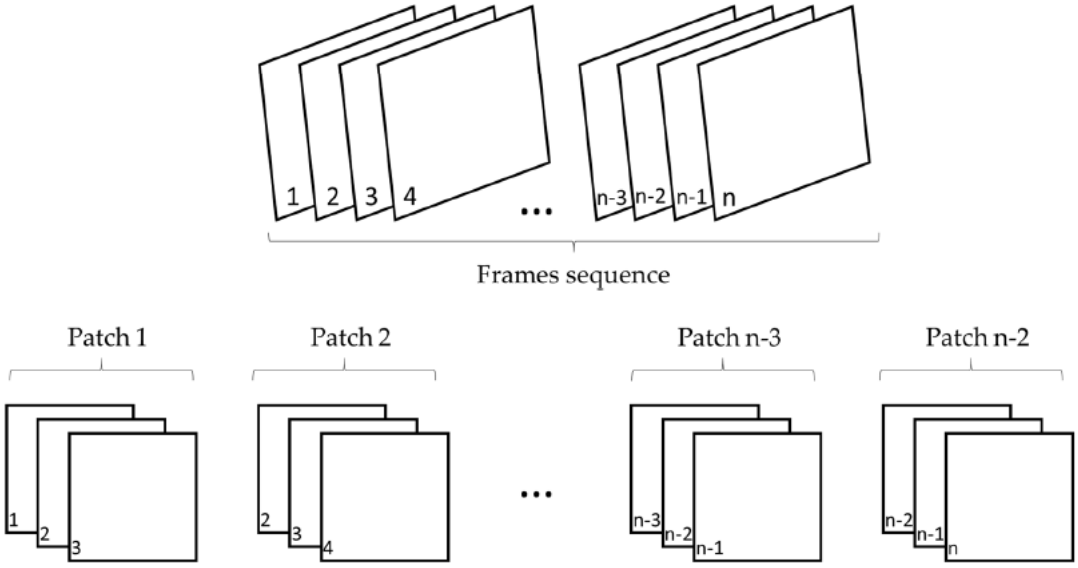

Figure 3.8: Scheme explaining the input data of the generative models. Each frame refers to B-scan from an OCT volume. Extracted from Tran and Yang [55].

while y_i is the intensity of the pixel with index i in the original image, and N is the number of pixels in the image [64, 65].

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|x_i - y_i| \tag{3.6}$$

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2 \tag{3.7}$$

PSNR is a metric used to calculate the ratio between the maximum signal power (which corresponds to the maximum value of a pixel in the image) and the power of the distorting noise, which affects the quality of its representation. Therefore, the PSNR, described in Equation 3.8, is inversely proportional to the mean squared error. PSNR can also be understood as the representation of absolute error in dB [64]. It is important to note that in OCT the signal-to-noise ratio is low, due to the speckle present in the images. Therefore, a smaller PSNR is also expected, when compared with other imaging techniques that do not present as much speckle [2].

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{L^2}{\text{MSE}}\right) \tag{3.8}$$

While the previous metrics focus on the differences between two images at a pixel level, the SSIM is based on the perception of the image. This metric considers the change of perception in structural information, estimating the perceived quality of images and videos. SSIM measures the similarity between the original image and the generated. The SSIM is calculated as shown in Equation 3.9, where x and y represent the generated and the original images, respectively, so that

μ_x and μ_y are their local means, σ_x and σ_y are their standard deviations, while C_1 and C_2 are small constants that stabilize the division. The contrast sensitivity (CS) between the images x and y is represented by $CS(x,y)$ [64].

$$\text{SSIM}(x,y) = \left(\frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}\right) \cdot \left(\frac{2\sigma_{xy} + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}\right) = \left(\frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}\right) \cdot CS(x,y) \tag{3.9}$$

Since the best performing models in segmentation resized the images to 496 $\times$ 512, the images were generated to match those dimensions. Therefore, regardless of the device utilized to obtain the OCT volume, all its B-scans were resized to 496 $\times$ 512 for both experiments of image generation.

3.2.3.1 Experiment 3

In the first experiment focused on intermediate slice synthesis, a GAN was used. The underlying principle of a GAN, originally proposed by Goodfellow et al. [66], is based on a competitive game between two networks. The generator network starts with the first and last slice of a subvolume, which is composed of three consecutive B-scans from an OCT scan, and aims to generate the intermediate slice. In contrast, the discriminator network is trained to distinguish between the generated and real slices. When the discriminator correctly labels generated slices as fake, the generator is penalized, motivating it to fool the discriminator and consequently improving its generation, resulting in outputs more similar to the real inputs. However, the discriminator network loss also penalizes misclassifications, dependent on the probability of the prediction. As a result, as the generator improves, so does the discriminator [67]. The overall framework for GANs is illustrated in Figure 3.9.

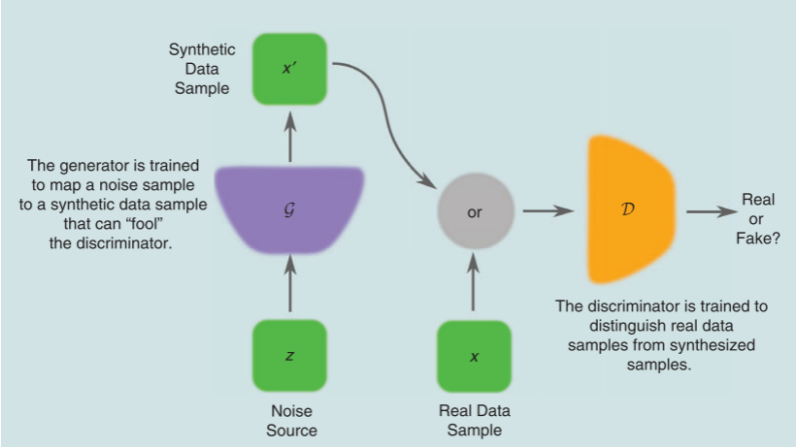

Figure 3.9: Example of a GAN framework, where $\mathscr{D}$ is the discriminator and $\mathscr{G}$ is the generator [68].

The GAN implemented by Tran and Yang [55] was used to generate the intermediate slices of the OCT volumes. This framework is used in the interpolation of intermediate slices in video and is trained in patches of 64 × 64.

In the original implementation, one patch is randomly extracted from the triplet of images and used in training. Due to the much smaller quantity of data available in OCT, all the possible disjoint 64 × 64 patches were extracted from each B-scan. The extraction was done from top to bottom and in the last row of slices, the image was padded until it had 64 pixels. An example of the patches extracted from a Cirrus B-scan can be seen in Figure 3.10. By methodically extracting the patches, triplets are easily created by accessing patches of the same index in the three consecutive images.

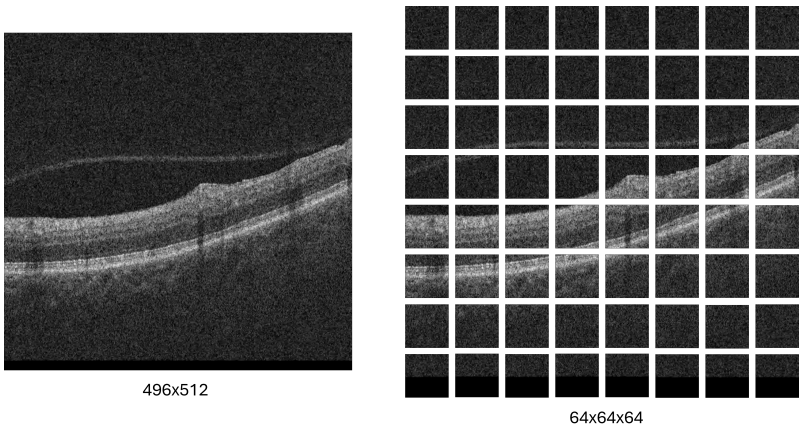

Figure 3.10: Patches with shape 64 × 64 extracted from a Cirrus B-scan which was resized to 496 × 512.

The generator of the GAN is composed of contracting and expanding path. In the contracting path, convolutions are applied to the images and feature maps, followed by batch normalization, and a leaky ReLU activation function. After the images are downsampled to 512 × 8 × 8, reaching the bottleneck, the expanding path begins, where deconvolutions are applied, followed by batch normalization, and a leaky ReLU. Finally, after the last deconvolution, the hyperbolic tangent function is used as the final activation function, resulting in an output of size 64 × 64, in range -1 to 1. An illustrative scheme of the generator can be seen in Figure 3.11.

To match the needs of our application, some changes had to be made in the generator regarding the input shape. As shown in Figure 3.11, the input has six channels, one red, one green, and one blue (RGB) for each input patch. Similarly, the output has three channels, for the generated middle patch. In our application, each patch in the input only had one channel, since the OCT B-scans are images in gray scale, instead of RGB, as in the original implementation. Therefore, the output only had one channel.

The final activation function, the hyperbolic tangent, outputs values between -1 and 1. Since the output images were compared to images in range 0 to 1, the final activation function was changed to the sigmoid. This function converts the values output from the last convolution to the range of 0 to 1, where it can be compared to the GT images.

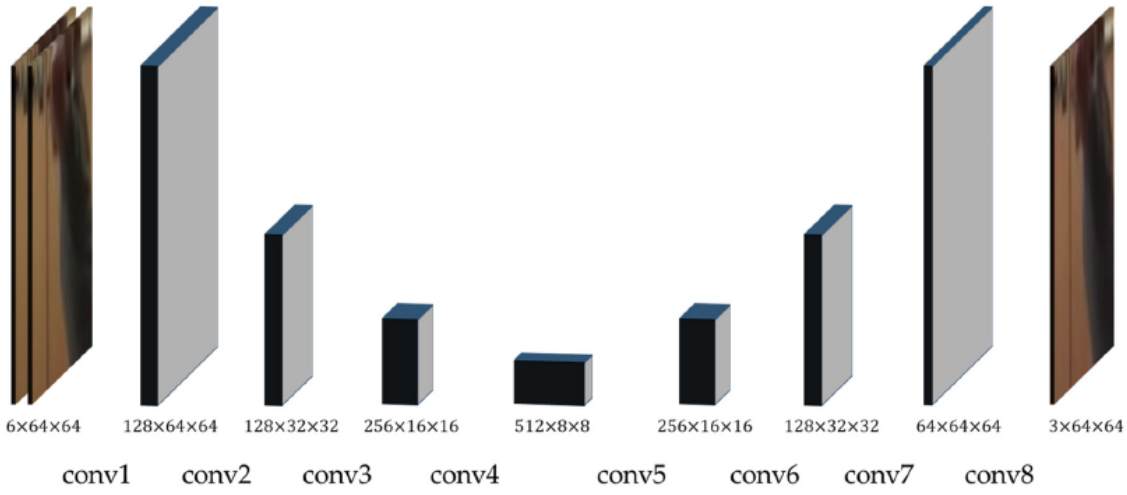

Figure 3.11: Architecture of the generator used in the GAN. It has a contracting and an expanding path, making it a U-Net like network [55].

The GAN's discriminator receives as input a patch of shape 64×64, and outputs a probability of the input patch being real. The discriminator is composed of five consecutive convolutions. The first convolution is followed by a leaky ReLU activation function. The second, third, and fourth convolutions are followed by a batch normalization and a leaky ReLU activation function. After the last convolution, the sigmoid is applied and converts the final output to a value between 0 and 1 that represents the probability of the image being real. In Table 3.4, the layers that compose the discriminator and the generator are explained, including the shape of the outputs.

In the training of a GAN, both the generator and the discriminator are being trained sequentially and independently. First, the generator, which receives the previous and following patch of the input triplet, attempts to generate the intermediate patch. The generated image is then compared to the original image, using the generator loss, which is then used in the updating of the generator weights. The generator loss is composed of four components: the adversarial, the MAE, the multi-scale SSIM (MS-SSIM), and the gradient difference loss (GDL). The overall loss function is described in Equation 3.10, where $\lambda_{\text{adv}} = 0.05$, $\lambda_{\text{MAE}} = 1.0$, $\lambda_{\text{MS-SSIM}} = 6.0$, and $\lambda_{\text{GDL}} = 1.0$, representing the weights of each loss component.

$$\mathscr{L}_{\text{Gen}} = \lambda_{\text{adv}} \times \mathscr{L}_{\text{adv}} + \lambda_{\text{MAE}} \times \mathscr{L}_{\text{MAE}} + \lambda_{\text{MS-SSIM}} \times \mathscr{L}_{\text{MS-SSIM}} + \lambda_{\text{GDL}} \times \mathscr{L}_{\text{GDL}} \tag{3.10}$$

The adversarial loss is used in the evaluation of how good the output from the generator fools the discriminator. This evaluation is done using the binary cross-entropy (BCE). To calculate this, the generated image is input into the discriminator, which then outputs the probability of being real. Afterwards, the BCE is calculated for the predicted probability and 1, the label of a real image. The better the generator fools the discriminator, the closer the output of the discriminator is to one, and, therefore, the closer the adversarial loss is to 0. The adversarial loss is explained in Equation 3.11, where $\mathscr{D}$ is the discriminator, x is the generated image, and y is the label that

Table 3.4: Layers that compose the generator and the discriminator. Each convolution is represented by Conv2d(K, OC, S), where K is the kernel size, OC is the number of output channels, and S is the stride. The same notation is used in deconvolutions, represented by TransposedConv2d. The output size is shown following C × H × W notation, where C is the number of channels, H is the height, and W is the width. The inputs have shape $1 \times 64 \times 64$. Adapted from Tran and Yang [55].

Generator		
Layers	**Details**	**Output Size (C × H × W)**
1	Conv2d(3, 128, 1), BatchNorm2d, LeakyReLU	128 × 64 × 64
2	Conv2d(4, 128, 2), BatchNorm2d, LeakyReLU	128 × 32 × 32
3	Conv2d(4, 256, 2), BatchNorm2d, LeakyReLU	256 × 16 × 16
4	Conv2d(4, 512, 2), BatchNorm2d, LeakyReLU	512 × 8 × 8
5	TransposedConv2d(4, 256, 2), BatchNorm2d, LeakyReLU	256 × 16 × 16
6	TransposedConv2d(4, 128, 2), BatchNorm2d, LeakyReLU	128 × 32 × 32
7	TransposedConv2d(4, 64, 2), BatchNorm2d, LeakyReLU	64 × 64 × 64
8	TransposedConv2d(1, 1, 1), Sigmoid	1 × 64 × 64
Discriminator		
Layers	**Details**	**Output Size (C × H × W)**
1	Conv2d(4, 64, 2), LeakyReLU	64 × 32 × 32
2	Conv2d(4, 128, 2), BatchNorm2d, LeakyReLU	128 × 16 × 16
3	Conv2d(4, 256, 2), BatchNorm2d, LeakyReLU	256 × 8 × 8
4	Conv2d(4, 512, 2), BatchNorm2d, LeakyReLU	512 × 4 × 4
5	Conv2d(4, 1, 1), Sigmoid	1 × 1 × 1

indicates that the image is real. It is important to note that in the generator training, the images that are input to the discriminator are detached, not contributing to the updating of the discriminator weights in this step.

$$\mathscr{L}_{\text{adv}}(\mathscr{D}(x), y) = \mathscr{L}_{\text{BCE}}(\mathscr{D}(x), y) = -\left[y\log(\mathscr{D}(x)) + (1-y)\log(1-\mathscr{D}(x))\right] \tag{3.11}$$

The MAE loss, also referred to as L_1 loss, performs a pixel-by-pixel comparison between the generated image, x, and the real image, y, as described in Equation 3.6, where i is the index of a pixel. While this loss gives an insight of how similar the images are, on average, this loss can be deceiving, since the model can blur the output to attain better MAE values. For this reason, this loss must be combined with other reconstructive losses, such as the MS-SSIM and the GDL.

The MS-SSIM loss seeks to preserve the structural similarity, at different scales, between the real and the generated image, facilitating a smoother output. This loss, originally suggested by Wang et al. [69], is described in Equation 3.12, while the MS-SSIM used in this implementation is explained in Equation 3.13, using the concepts of SSIM and CS explained in Equation 3.9. This

version of the MS-SSIM is faster than the original implementation [69], while using the same array of weights β and number of levels M, which is set to 5. Therefore, the MS-SSIM corresponds to the product of the contrast sensitivity in the image for first four levels and the SSIM of the image in the last level, all raised to the power of the respective level's weights.

$$\mathscr{L}_{\text{MS-SSIM}}(x,y) = 1 - \text{MS-SSIM}(x,y) \tag{3.12}$$

$$\text{MS-SSIM}(x,y) = \prod_{j=1}^{M-1} [\text{CS}_j(x,y)]^{\beta_j} \cdot [\text{SSIM}_M(x,y)]^{\beta_M} \tag{3.13}$$

The last component of the loss is the GDL, proposed originally by Mathieu et al. [70]. This component is used to reduce the motion blur in the generated images, a problem in video datasets. In this loss, the relative difference of neighboring pixels between the generated and true images is considered, as shown in Equation 3.14. In this equation, i and j are the index of row and column, respectively, that identify a pixel of the image, with α set to 2.

$$\mathscr{L}_{\text{GDL}}(x,y) = \sum_{i,j} \left(\left| |x_{i,j} - x_{i-1,j}| - |y_{i,j} - y_{i-1,j}| \right|^{\alpha} + \left| |x_{i,j} - x_{i,j-1}| - |y_{i,j} - y_{i,j-1}| \right|^{\alpha} \right) \tag{3.14}$$

After the images are generated and evaluated using the previously defined generator loss, two images are input, subsequently, to the discriminator, with one of them being fake while the other is real. The discriminator outputs the probability of each image being real and its result is compared to the true label of each image using the BCE. The BCE is calculated for the probabilities predicted by the discriminator and the image's respective label as described in Equation 3.11. This is done for the fake image and for the real image. The mean between the BCE calculated for the fake image and the BCE computed for the real image is the discriminator loss.

The GAN was trained in 250 epochs, using a batch size of 32 and 2×10^{-4} as the learning rate. The selected optimizer was Adam, with $\beta_1 = 0.5$ and $\beta_2 = 0.999$.

3.2.3.2 Experiment 4

As in the previous experiment, the intermediate slice was generated using the first and last slices of a subvolume that consists of three consecutive B-scans from an OCT scan. However, in this experiment, inspired by the work of Nishimoto et al. [43], the intermediate slice was generated using a U-Net. While the U-Net is more commonly applied in segmentation, as seen in the reviewed literature, Nishimoto et al. [43] apply it to generate the intermediate slices of a subvolume. The U-Net receives the edge slices as input and forces the output of the intermediate ones. In the paper [43], this was tested for three, four, and five slices. However, in this experiment it was utilized to generate a single intermediate slice.

In this experiment, the whole image is input to the network and for that reason the batch size was set to 8. The optimizer utilized was Adam with a learning rate of 2×10^{-4} and the model was trained for 200 epochs.

3.2.4 Fluid Volume Estimation

The estimation of fluid volume was done using the optimal segmentation and intermediate slice generation models. The GAN was used in the generation of the intermediate slices in the OCT volumes, while the best performing multi-class segmentation U-Net model inferred the segmentation masks to both the unaltered volumes and the volumes with generated slices.

The OCT scans used in the fluid volume estimation experiments were the ones from the RETOUCH dataset that composed the reserved fold in the segmentation and generation experiments and the ones from the private dataset obtained in Hospital São João. These volumes were selected because no model was trained or validated on them, allowing an insight of both models generalization on unseen data and how the increase in resolution affects the total fluid volume.

The area of each fluid in each OCT scan was estimated considering the resolution of each OCT scan, which varies according to the device utilized to obtain the OCT volume. Afterwards, the area is multiplied by the axial distance (half the axial distance to the previous slice plus half the axial distance to the following slice) to obtain the volume of fluid per slice. In the first and last slice of an OCT volume, the area is multiplied by half of the axial distance (half the axial distance to the neighboring slice). The total volume of fluid in an OCT scan can be estimated by summing the fluid volumes of all individual B-scans. This allows the volume estimation of IRF, SRF, and PED, as well as the overall fluid volume in the OCT scan.

The total volume of fluid from class c in a slice of index s is defined in Equation 3.15. The slice belongs to an OCT volume obtained using device D and its total number of B-scans is defined as S. In this equation, H_D and W_D are the height and width of a voxel, respectively, obtained with device D, while $d_{D,s,s+1}$ is the axial distance between the slice of index s and the slice of index $s+1$, a value that depends on the device D characteristics. The variable l_i is the label attributed to the voxel of index i. Like the variable c, l can be one of the following classes: $\{0,1,2,3\}$, which respectively correspond to background, IRF, SRF, and PED. Meanwhile, the total volume of fluid from a class c in an OCT scan obtained with device D is described by Equation 3.16, and consists of the sum of the fluid's volume obtained in each B-scan that compose the OCT.

$$
\begin{aligned}
f_{c,s,D} &= \sum_i (v_{i,s,D} \times y_{i,c}) \quad \text{where:} \\
v_{i,s,D} &= \begin{cases} 0.5 \times H_D \times W_D \times d_{D,s,s+1} & \text{if } s = 0 \\ 0.5 \times H_D \times W_D \times d_{D,s,s-1} & \text{if } s = S \\ 0.5 \times H_D \times W_D \times d_{D,s,s-1} + 0.5 \times H_D \times W_D \times d_{D,s,s+1} & \text{otherwise} \end{cases} \\
y_{i,c} &= \begin{cases} 1 & \text{if } l_i = c \\ 0 & \text{if } l_i \neq c \end{cases}
\end{aligned} \tag{3.15}
$$

$$
F_{c,D} = \sum_s^S f_{c,s,D} \tag{3.16}
$$

The fluid volumes resulting from both experiments were compared. Since there is no true value for the fluid quantity in the OCT scans, the results were compared with each other. Therefore, the results would be deemed satisfying in case they do not vary more than an order of magnitude between each other. In case a significant difference was observed, the generated images and their respective masks were analyzed, in order to understand what is causing the observed difference between experiments.

3.2.4.1 Experiment 5

In this experiment, the fluid volumes were calculated for the OCT scans without the generated slices. The best segmentation model was utilized to segment the fluid in three classes and the volume was estimated for each class as described. The results from this experiment allow the comparison with the values obtained in the following experiment, where slice generation was used.

3.2.4.2 Experiment 6

This experiment consisted of the fluid volume estimation in OCT scans with generated images. The model used in segmentation was the same as in the previous experiment, which predicted the fluid masks for all the slices. From the predicted fluid masks, the fluid volume was estimated and compared with those obtained in the previous experiment.

References

[1] F. Hutmacher, "Why Is There So Much More Research on Vision Than on Any Other Sensory Modality?" *Frontiers in Psychology*, vol. 10, 2019, ISSN: 1664-1078. DOI: 10.3389/fpsyg.2019.02246.

[2] H. Bogunović, W.-D. Vogl, S. M. Waldstein, and U. Schmidt-Erfurth, "Chapter 14 - OCT fluid detection and quantification," in E. Trucco, T. MacGillivray, and Y. Xu, Eds. Academic Press, 2019, pp. 273–298, ISBN: 978-0-08-102816-2. DOI: 10.1016/B978-0-08-102816-2.00015-0.

[3] L. S. Lim, P. Mitchell, J. M. Seddon, F. G. Holz, and T. Y. Wong, "Age-related macular degeneration," *The Lancet*, vol. 379, pp. 1728–1738, 9827 2012, ISSN: 0140-6736. DOI: 10.1016/S0140-6736(12)60282-7.

[4] P. Mitchell, G. Liew, B. Gopinath, and T. Y. Wong, "Age-related macular degeneration," *The Lancet*, vol. 392, pp. 1147–1159, 10153 Sep. 2018, ISSN: 0140-6736. DOI: 10.1016/S0140-6736(18)31550-2.

[5] O. Musat et al., "Diabetic Macular Edema," *Rom J Ophthalmol*, vol. 59, pp. 133–136, 3 Jul. 2015.

[6] N. Bhagat, R. A. Grigorian, A. Tutela, and M. A. Zarbin, "Diabetic Macular Edema: Pathogenesis and Treatment," *Survey of Ophthalmology*, vol. 54, pp. 1–32, 1 2009, ISSN: 0039-6257. DOI: 10.1016/j.survophthal.2008.10.001.

[7] F. Bandello, R. Lattanzio, I. Zucchiatti, A. Arrigo, M. Battista, and M. V. Cicinelli, "Diabetic Macular Edema," in M. Attilio, L. Rosangela, Z. I. B. Francesco, and Zarbin, Eds. Springer International Publishing, 2019, pp. 97–183, ISBN: 978-3-319-96157-6. DOI: 10.1007/978-3-319-96157-6_3.

[8] T. Y. Wong and I. U. Scott, "Retinal-Vein Occlusion," *New England Journal of Medicine*, vol. 363, pp. 2135–2144, 22 2010. DOI: 10.1056/NEJMcp1003934.

[9] D. Huang et al., "Optical coherence tomography," *Science*, vol. 254, pp. 1178–1181, 5035 Nov. 1991. DOI: 10.1126/science.1957169.

[10] W. Drexler and J. G. Fujimoto, "State-of-the-art retinal optical coherence tomography," *Progress in Retinal and Eye Research*, vol. 27, pp. 45–88, 1 2008, ISSN: 1350-9462. DOI: 10.1016/j.preteyeres.2007.07.005.

[11] I. A. Viedma, D. Alonso-Caneiro, S. A. Read, and M. J. Collins, "Deep learning in retinal optical coherence tomography (OCT): A comprehensive survey," *Neurocomputing*, vol. 507, pp. 247–264, 2022, ISSN: 0925-2312. DOI: 10.1016/j.neucom.2022.08.021.

[12] N. Jain et al., "Quantitative Comparison of Drusen Segmented on SD-OCT versus Drusen Delineated on Color Fundus Photographs," *Investigative Ophthalmology & Visual Science*, vol. 51, pp. 4875–4883, 10 May 2010, ISSN: 1552-5783. DOI: 10.1167/iovs.09-4962.

[13] M. Almonte, P. Capellàn, T. Yap, and M. F. Cordeiro, "Retinal correlates of psychiatric disorders," *Therapeutic Advances in Chronic Disease*, vol. 11, p. 204 062 232 090 521, Dec. 2020. DOI: 10.1177/2040622320905215.

[14] E. López-Varela, N. Barreira, N. O. Pascual, M. R. A. Castillo, and M. G. Penedo, "Generation of synthetic intermediate slices in 3D OCT cubes for improving pathology detection and monitoring," *Computers in Biology and Medicine*, vol. 163, p. 107 214, 2023, ISSN: 0010-4825. DOI: 10.1016/j.compbiomed.2023.107214.

[15] E. Selvi, M. Özdemir, and M. A. Selver, "Performance Analysis of Distance Transform Based Inter-Slice Similarity Information on Segmentation of Medical Image Series," *Mathematical and Computational Applications*, vol. 18, pp. 511–520, 3 2013, ISSN: 2297-8747. DOI: 10.3390/mca18030511.

[16] T. C. Quek et al., "Predictive, preventive, and personalized management of retinal fluid via computer-aided detection app for optical coherence tomography scans," *EPMA Journal*, vol. 13, pp. 547–560, 4 2022, ISSN: 1878-5085. DOI: 10.1007/s13167-022-00301-5.

[17] S. J. Pawan et al., "Capsule Network–based architectures for the segmentation of sub-retinal serous fluid in optical coherence tomography images of central serous chorioretinopathy," *Medical & Biological Engineering & Computing*, vol. 59, pp. 1245–1259, 6 2021, ISSN: 1741-0444. DOI: 10.1007/s11517-021-02364-4.

[18] X. Liu, S. Wang, Y. Zhang, D. Liu, and W. Hu, "Automatic fluid segmentation in retinal optical coherence tomography images using attention based deep learning," *Neurocomputing*, vol. 452, pp. 576–591, 2021, ISSN: 0925-2312. DOI: 10.1016/j.neucom.2020.07.143.

[19] Y. Guo, T. T. Hormel, H. Xiong, J. Wang, T. S. Hwang, and Y. Jia, "Automated Segmentation of Retinal Fluid Volumes From Structural and Angiographic Optical Coherence Tomography Using Deep Learning," *Translational Vision Science & Technology*, vol. 9, p. 54, 2 Oct. 2020, ISSN: 2164-2591. DOI: 10.1167/tvst.9.2.54.

[20] Z. Wang et al., "Automated segmentation of macular edema for the diagnosis of ocular disease using deep learning method," *Scientific Reports*, vol. 11, p. 13 392, 1 2021, ISSN: 2045-2322. DOI: 10.1038/s41598-021-92458-8.

[21] Y. Wu et al., "Training Deep Learning Models to Work on Multiple Devices by Cross-Domain Learning with No Additional Annotations," *Ophthalmology*, vol. 130, pp. 213–222, 2 2023, ISSN: 0161-6420. DOI: 10.1016/j.ophtha.2022.09.014.

[22] M. Rahil, B. N. Anoop, G. N. Girish, A. R. Kothari, S. G. Koolagudi, and J. Rajan, "A Deep Ensemble Learning-Based CNN Architecture for Multiclass Retinal Fluid Segmentation in OCT Images," *IEEE Access*, vol. 11, pp. 17 241–17 251, 2023, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2023.3244922.

[23] H. Liu et al., "Semantic uncertainty Guided Cross-Transformer for enhanced macular edema segmentation in OCT images," *Computers in Biology and Medicine*, vol. 174, p. 108 458, 2024, ISSN: 0010-4825. DOI: 10.1016/j.compbiomed.2024.108458.

[24] X. Li, S. Niu, X. Gao, X. Zhou, J. Dong, and H. Zhao, "Self-training adversarial learning for cross-domain retinal OCT fluid segmentation," *Computers in Biology and Medicine*, vol. 155, p. 106 650, 2023, ISSN: 0010-4825. DOI: 10.1016/j.compbiomed.2023.106650.

[25] K. Gao et al., "Double-branched and area-constraint fully convolutional networks for automated serous retinal detachment segmentation in SD-OCT images," *Computer Methods and Programs in Biomedicine*, vol. 176, pp. 69–80, 2019, ISSN: 0169-2607. DOI: 10.1016/j.cmpb.2019.04.027.

[26] B. Hassan et al., "Deep learning based joint segmentation and characterization of multi-class retinal fluid lesions on OCT scans for clinical use in anti-VEGF therapy," *Computers in Biology and Medicine*, vol. 136, p. 104 727, 2021, ISSN: 0010-4825. DOI: 10.1016/j.compbiomed.2021.104727.

[27] D. Lu et al., "Deep-learning based multiclass retinal fluid segmentation and detection in optical coherence tomography images using a fully convolutional neural network," *Medical Image Analysis*, vol. 54, pp. 100–110, 2019, ISSN: 1361-8415. DOI: 10.1016/j.media.2019.02.011.

[28] B. Hassan, S. Qin, T. Hassan, R. Ahmed, and N. Werghi, "Joint Segmentation and Quantification of Chorioretinal Biomarkers in Optical Coherence Tomography Scans: A Deep Learning Approach," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–17, 2021. DOI: 10.1109/TIM.2021.3077988.

[29] H. Zhang, J. Yang, C. Zheng, S. Zhao, and A. Zhang, "Annotation-efficient learning for OCT segmentation," *Biomed. Opt. Express*, vol. 14, pp. 3294–3307, 7 Jul. 2023. DOI: 10.1364/BOE.486276.

[30] L. B. Sappa et al., "RetFluidNet: Retinal Fluid Segmentation for SD-OCT Images Using Convolutional Neural Network," *Journal of Digital Imaging*, vol. 34, pp. 691–704, 3 2021, ISSN: 1618-727X. DOI: 10.1007/s10278-021-00459-w.

[31] G. Xing et al., "Multi-Scale Pathological Fluid Segmentation in OCT With a Novel Curvature Loss in Convolutional Neural Network," *IEEE Transactions on Medical Imaging*, vol. 41, pp. 1547–1559, 6 2022. DOI: 10.1109/TMI.2022.3142048.

[32] W. Tang et al., "Multi-class retinal fluid joint segmentation based on cascaded convolutional neural networks," *Physics in Medicine & Biology*, vol. 67, p. 125 018, 12 Jun. 2022. DOI: 10.1088/1361-6560/ac7378.

[33] F. D. Padilla-Pantoja, Y. D. Sanchez, B. A. Quijano-Nieto, O. J. Perdomo, and F. A. Gonzalez, "Etiology of Macular Edema Defined by Deep Learning in Optical Coherence Tomography Scans," *Translational Vision Science & Technology*, vol. 11, p. 29, 9 Sep. 2022, ISSN: 2164-2591. DOI: 10.1167/tvst.11.9.29.

[34] J. Hu, Y. Chen, and Z. Yi, "Automated segmentation of macular edema in OCT using deep neural networks," *Medical Image Analysis*, vol. 55, pp. 216–227, 2019, ISSN: 1361-8415. DOI: 10.1016/j.media.2019.05.002.

[35] I. Mantel et al., "Automated Quantification of Pathological Fluids in Neovascular Age-Related Macular Degeneration, and Its Repeatability Using Deep Learning," *Translational Vision Science & Technology*, vol. 10, p. 17, 4 Apr. 2021, ISSN: 2164-2591. DOI: 10.1167/tvst.10.4.17.

[36] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., Springer International Publishing, 2015, pp. 234–241, ISBN: 978-3-319-24574-4. DOI: 10.1007/978-3-319-24574-4_28.

[37] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 834–848, 4 2018. DOI: 10.1109/TPAMI.2017.2699184.

[38] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv e-prints*, arXiv:1409.1556, Sep. 2014. DOI: 10.48550/arXiv.1409.1556.

[39] C. You et al., "CT Super-Resolution GAN Constrained by the Identical, Residual, and Cycle Learning Ensemble (GAN-CIRCLE)," *IEEE Transactions on Medical Imaging*, vol. 39, pp. 188–203, 1 2020. DOI: 10.1109/TMI.2019.2922960.

[40] M. Ibrahim et al., "Generative AI for Synthetic Data Across Multiple Medical Modalities: A Systematic Review of Recent Developments and Challenges," *arXiv e-prints*, arXiv:2407.00116, Jun. 2024. DOI: 10.48550/arXiv.2407.00116.

[41] Y. Xia, N. Ravikumar, J. P. Greenwood, S. Neubauer, S. E. Petersen, and A. F. Frangi, "Super-Resolution of Cardiac MR Cine Imaging using Conditional GANs and Unsupervised Transfer Learning," *Medical Image Analysis*, vol. 71, p. 102 037, 2021, ISSN: 1361-8415. DOI: 10.1016/j.media.2021.102037.

[42] Z. Wu, J. Wei, J. Wang, and R. Li, "Slice imputation: Multiple intermediate slices interpolation for anisotropic 3D medical image segmentation," *Computers in Biology and Medicine*, vol. 147, p. 105 667, 2022, ISSN: 0010-4825. DOI: 10.1016/j.compbiomed.2022.105667.

[43] S. Nishimoto, K. Kawai, K. Nakajima, H. Ishise, and M. Kakibuchi, "Generating intermediate slices with U-nets in craniofacial CT images," *medRxiv*, p. 2024.05.08.24307089, Jan. 2024. DOI: 10.1101/2024.05.08.24307089.

[44] H. Zhang, X. Yang, Y. Cui, Q. Wang, J. Zhao, and D. Li, "A novel GAN-based three-axis mutually supervised super-resolution reconstruction method for rectal cancer MR image," *Computer Methods and Programs in Biomedicine*, vol. 257, p. 108 426, 2024, ISSN: 0169-2607. DOI: 10.1016/j.cmpb.2024.108426.

[45] C. Fang, L. Wang, D. Zhang, J. Xu, Y. Yuan, and J. Han, "Incremental Cross-View Mutual Distillation for Self-Supervised Medical CT Synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 20 677–20 686. DOI: 10.48550/arXiv.2112.10325.

[46] U. Nimitha and P. Ameer, "MRI super-resolution using similarity distance and multi-scale receptive field based feature fusion GAN and pre-trained slice interpolation network," *Magnetic Resonance Imaging*, vol. 110, pp. 195–209, 2024, ISSN: 0730-725X. DOI: 10.1016/j.mri.2024.04.021.

[47] M.-I. Georgescu, R. T. Ionescu, and N. Verga, "Convolutional Neural Networks With Intermediate Loss for 3D Super-Resolution of CT and MRI Scans," *IEEE Access*, vol. 8, pp. 49 112–49 124, 2020. DOI: 10.1109/ACCESS.2020.2980266.

[48] Y. Chen, F. Shi, A. G. Christodoulou, Y. Xie, Z. Zhou, and D. Li, "Efficient and Accurate MRI Super-Resolution Using a Generative Adversarial Network and 3D Multi-level Densely Connected Network," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, A. F. Frangi, S. Julia A., C. Davatzikos, C. Alberola-López, and G. Fichtinger, Eds., Springer International Publishing, 2018, pp. 91–99, ISBN: 978-3-030-00928-1. DOI: 10.1007/978-3-030-00928-1_11.

[49] I. Sanchez and V. Vilaplana, "Brain MRI super-resolution using 3D generative adversarial networks," *arXiv e-prints*, arXiv:1812.11440, Dec. 2018. DOI: 10.48550/arXiv.1812.11440.

[50] A. Kudo, Y. Kitamura, Y. Li, S. Iizuka, and E. Simo-Serra, "Virtual Thin Slice: 3D Conditional GAN-based Super-Resolution for CT Slice Interval," in *Machine Learning for Medical Image Reconstruction*, Andreas, R. Daniel, Y. J. C. K. Florian, and Maier, Eds., Springer International Publishing, 2019, pp. 91–100, ISBN: 978-3-030-33843-5. DOI: 10.1007/978-3-030-33843-5_9.

[51] K. Zhang et al., "SOUP-GAN: Super-Resolution MRI Using Generative Adversarial Networks," *Tomography*, vol. 8, pp. 905–919, 2 2022, ISSN: 2379-139X. DOI: 10.3390/tomography8020073.

[52] C. Peng, W.-A. Lin, H. Liao, R. Chellappa, and S. K. Zhou, "SAINT: Spatially Aware Interpolation NeTwork for Medical Slice Synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020. DOI: 10.48550/arXiv.2001.00704.

[53] L. Gambini et al., "Video frame interpolation neural network for 3D tomography across different length scales," *Nature Communications*, vol. 15, p. 7962, 1 2024, ISSN: 2041-1723. DOI: 10.1038/s41467-024-52260-2.

[54] Z. Huang, T. Zhang, W. Heng, B. Shi, and S. Zhou, "Real-Time Intermediate Flow Estimation for Video Frame Interpolation," in *Computer Vision – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., Springer Nature Switzerland, 2022, pp. 624–642, ISBN: 978-3-031-19781-9. DOI: 10.1007/978-3-031-19781-9_36.

[55] Q. N. Tran and S.-H. Yang, "Efficient Video Frame Interpolation Using Generative Adversarial Networks," *Applied Sciences*, vol. 10, 18 2020, ISSN: 2076-3417. DOI: 10.3390/app10186245.

[56] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-To-Image Translation With Conditional Adversarial Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017. DOI: 10.48550/arXiv.1611.07004.

[57] H. Bogunović et al., "RETOUCH: The Retinal OCT Fluid Detection and Segmentation Benchmark and Challenge," *IEEE Transactions on Medical Imaging*, vol. 38, pp. 1858–1874, 8 2019. DOI: 10.1109/TMI.2019.2901398.

[58] S. J. Chiu, M. J. Allingham, P. S. Mettu, S. W. Cousins, J. A. Izatt, and S. Farsiu, "Kernel regression based segmentation of optical coherence tomography images with diabetic macular edema," *Biomed. Opt. Express*, vol. 6, pp. 1172–1194, 4 Apr. 2015. DOI: 10.1364/BOE.6.001172.

[59] A. Rashno et al., "Fully-automated segmentation of fluid regions in exudative age-related macular degeneration subjects: Kernel graph cut in neutrosophic domain," *PLOS ONE*, vol. 12, pp. 1–26, 10 Dec. 2017. DOI: 10.1371/journal.pone.0186949.

[60] A. Rashno et al., "Fully Automated Segmentation of Fluid/Cyst Regions in Optical Coherence Tomography Images With Diabetic Macular Edema Using Neutrosophic Sets and Graph Algorithms," *IEEE Transactions on Biomedical Engineering*, vol. 65, pp. 989–1001, 5 2018. DOI: 10.1109/TBME.2017.2734058.

[61] R. R. Shamir, Y. Duchin, J. Kim, G. Sapiro, and N. Harel, "Continuous Dice Coefficient: a Method for Evaluating Probabilistic Segmentations," *arXiv e-prints*, arXiv:1906.11031, Jun. 2019. DOI: 10.48550/arXiv.1906.11031.

[62] R. Tennakoon, A. K. Gostar, R. Hoseinnezhad, and A. Bab-Hadiashar, "Retinal fluid segmentation in OCT images using adversarial loss based convolutional neural networks," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, 2018, pp. 1436–1440. DOI: 10.1109/ISBI.2018.8363842.

[63] D. P. Kingma, "Adam: A method for stochastic optimization," in *3rd International Conference for Learning Representations*, 2015. DOI: 10.48550/arXiv.1412.6980.

[64] U. Sara, M. Akter, and M. S. Uddin, "Image quality assessment through FSIM, SSIM, MSE and PSNR—a comparative study," *Journal of Computer and Communications*, vol. 7, pp. 8–18, 3 2019. DOI: 10.4236/jcc.2019.73002.

[65] S. Rajkumar and G. Malathi, "A comparative analysis on image quality assessment for real time satellite images," *Indian J. Sci. Technol*, vol. 9, pp. 1–11, 34 2016. DOI: 10.17485/ijst/2016/v9i34/96766.

[66] I. Goodfellow et al., "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds., vol. 27, Curran Associates, Inc., 2014. DOI: 10.48550/arXiv.1406.2661.

[67] I. Goodfellow et al., "Generative adversarial networks," *Commun. ACM*, vol. 63, pp. 139–144, 11 Oct. 2020, ISSN: 0001-0782. DOI: 10.1145/3422622.

[68] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative Adversarial Networks: An Overview," *IEEE Signal Processing Magazine*, vol. 35, pp. 53–65, 1 2018. DOI: 10.1109/MSP.2017.2765202.

[69] Z. Wang, E. Simoncelli, and A. Bovik, "Multiscale structural similarity for image quality assessment," in *Conference Record of the Asilomar Conference on Signals, Systems and Computers*, vol. 2, May 2003, 1398–1402 Vol.2, ISBN: 0-7803-8104-1. DOI: 10.1109/ACSSC.2003.1292216.

[70] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," in *4th International Conference on Learning Representations, ICLR 2016*, 2016. DOI: 10.48550/arXiv.1511.05440.

Appendix A

Lorem Ipsum