FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Characterization of Retinal Fluid in OCT Images

David Castanho Terroso



Mestrado em Engenharia Biomédica

Supervisor: Prof. Tânia Melo

Co-Supervisor: Prof. Ana Maria Mendonça

July 22, 2025

© David Castanho Terroso, 2025

Resumo

Danos causados na retina podem resultar em graves deficiências visuais, limitando significativamente a qualidade de vida dos pacientes. Doenças como a degenerescência macular da idade (AMD), edema macular diabético (DME) e a oclusão da veia central da retina (RVO) estão entre as principais causas de lesão da retina e caracterizam-se normalmente pela presença e acumulação de fluído na retina.

As características destes fluídos, visíveis em *scans* de tomografia de coerência ótica (OCT), são importantes biomarcadores para a progressão de doenças. Contudo, a anotação manual destes fluídos é laboriosa, demorada e propensa a variabilidade entre observadores, o que motiva a necessidade de métodos de caracterização automáticos.

Esta dissertação explora técnicas atuais de aprendizagem profunda para a segmentação semântica de fluídos retinais, visando a segmentação de fluído intra-retiniano (IRF), fluído sub-retiniano (SRF), e do descolamento do epitélio pigmentar (PED) em dois conjuntos de dados diferentes.

Adicionalmente, o trabalho investiga o uso de modelos generativos, como redes adversariais generativas (GANs) para sintetizar *scans* intermédios de OCT, com o objetivo de melhorar a resolução entre *scans* num volume de OCT. Os resultados de segmentação no conjunto de dados com resolução melhorada são utilizados para calcular o volume de fluído nestes exames, um biomarcador importante na progressão das doenças na retina mencionadas anteriormente.

Os resultados experimentais mostram que o modelo de segmentação alcançou resultados comparáveis aos da literatura, particularmente quando se utilizam *patches* maiores que contribuem com contexto anatómico suficiente para a segmentação. Por outro lado, os modelos generativos, embora menos explorados em OCT, obtiveram resultados promissores na síntese de *slices* intermédias realistas. Esta combinação de segmentação e geração permite uma caracterização mais robusta dos fluídos da retina, assegurando uma maior confiança nos volumes estimados.

Apesar dos resultados obtidos serem encorajadores, existem ainda bastantes limitações, particularmente na generalização do modelo de segmentação para imagens externas e nos detalhes das imagens geradas e das máscaras de fluído previstas. Trabalhos futuros devem assim explorar arquiteturas avançadas, como mecanismos de atenção, e funções de penalização percetuais, com o objetivo de melhorar os detalhes e a robustez dos modelos em diversas condições de imagem.

Abstract

The damage of the retina can result in significant visual impairment, severely limiting the quality of life of patients. Diseases such as age-related macular degeneration (AMD), diabetic macular edema (DME), and retinal vein occlusion (RVO) are among the leading causes of retinal damage and are typically characterized by the presence and accumulation of fluid in the retina.

The characteristics of these fluids, visible in retinal optical coherence tomography (OCT) scans, serve as important biomarkers for disease progression. However, the manual annotation of these fluids is laborious, time consuming, and prone to inter-observer variability, which motivates the need for automatic characterization approaches.

This dissertation explores the state-of-the-art deep learning techniques for semantic segmentation of retinal fluids, targeting the segmentation of intra-retinal fluid (IRF), sub-retinal fluid (SRF), and pigment epithelial detachment (PED) in two different datasets.

Additionally, the work investigates the use of generative models, such as the generative adversarial networks (GANs), for synthesizing intermediate OCT slices, aiming to enhance the inter-slice resolution throughout an OCT volume. The segmentation results in the enhanced dataset are used in the for computing the fluid volume in the scans, an important biomarker in the progression of the aforementioned retinal diseases.

The experimental results show that the segmentation model achieved comparable outcomes to existing approaches in the literature, particularly when using as input larger patches that contribute with enough anatomic context for the segmentation. Meanwhile, the generative models, though less explored in OCT, demonstrate promising results in the synthesis of realistic interpolated slices. This combination of segmentation and generation supports a more robust characterization of the retinal fluids, providing more confidence to the fluid volumes estimated.

Despite the encouraging obtained results, several limitations remain, particularly in the segmentation model's generalization to external datasets and in the fine-detail realism of the generated slices and segmentation masks. Future works should explore advanced architectures, such as attention mechanism, and perceptual loss functions, to improve detail and robustness across diverse imaging conditions.

UN Sustainable Development Goals

The United Nations Sustainable Development Goals (SDGs) provide a global framework to achieve a more equitable, healthy, and sustainable future for all. Comprising 17 different goals, this framework addresses pressing world's challenges, including poverty, hunger, health, and education.

This dissertation contributes to these SDGs by advancing automated tools for the detection and characterization of retinal diseases, which are among the leading causes of vision impairment and blindness. The proposed methods support the "SDG 3: Ensure healthy lives and promote well-being for all at all ages", by developing fast, automatic, scalable, and accurate diagnostic tools that assist in the early identification and monitoring of retinal conditions. Since these solutions are fast, accessible, and easily deployed, they have the potential to be used in developing countries, where resources and access to trained specialists are scarce, supporting Target 3.d., which aims to support the capacity of all countries for early detection and monitoring of health risks. Additionally, through the prediction and monitoring of retinal diseases, this technology can also reduce the load on the healthcare systems, while improving decision-making and access to more equitable healthcare, once again addressing Target 3.d.

In addition to the healthcare contributions, the studied technology can also be implemented as a educational material, providing an alternative tool for the understanding and visualization of retinal fluid pathologies, aligning with "SDG 4: Ensure inclusive and equitable quality education and promote lifelong learning opportunities for all". This addresses Target 4.4, which focuses on increasing of the number of young adults with relevant technical skills, and Target 3.c, which aims to improve the training of healthcare personnel.

The relevant SGDs, their associated contributions, and performance metrics are explained in the following table:

SDG 3 Ensure healthy lives and promote well-being for all at all ages

SDG 4 Ensure inclusive and equitable quality education and promote lifelong learning opportunities for all

SGD	Target	Contribution	Performance Indicators and Metrics
3	3.d	Developed a GAN that generates intermediate slices between two known scans, increasing the spatial resolution and continuity of OCT volumes.	Reduction in misdiagnosis cases; Increased detection of early-stage retinal conditions.
	3.d	Designed an automatic fluid volume estimation pipeline that integrates retinal fluid segmentation and slice generation, and can accurately calculate the quantity of each fluid in type (IRF, SRF, and PED) an OCT volume, enabling disease progression monitoring and reducing clinician workload.	Reduction in clinician time spent per patient; Improved diagnostic consistency across clinicians.
	3.c	Developed a segmentation model that performs accurate retinal fluid segmentation in OCT B-scans automatically, and can be used as an educational tool for understanding of retinal pathologies.	Improvement in diagnostic skills before and after using this tool; Increased acceptability of AI tools in the clinical workflow.
4	4.4		

Acknowledgements

As this dissertation marks the end of my studies, it is of my responsibility to thank and acknowledge all of those who shaped the path that lead me here.

First, I would like to thank my supervisor Prof. Tânia Melo and my co-supervisor Prof. Ana Maria Mendonça for all the support, dedication, and patience throughout this dissertation. They motivated me to dig deeper in the topics here approached, inspiring me and forging my knowledge, while sharing their experienced and useful insights.

As I thank my supervisors, I must also thank all the professors that motivated, inspired, and, most importantly, taught me along my academic path. From English to equations and programming, the knowledge transmitted by them fills the pages of this document.

I am also grateful for all the friends that made part of this process. Therefore, I thank those that I met at FEUP and made the experience in this university more fun and enjoyable. The burdens shared with them will never be forgotten, as they made the ride of the last two years smoother. I also thank those that have accompanied me since high school, who showed me the corners of this university and shared their technical insights in the topics approached in this dissertation, despite sharing a different background. It has been a pleasure to watch you grow beside me.

I can not write the acknowledgments without thanking my family, especially my parents and my sister Sofia, who have been supporting me all the way through the academic path that lead me to this master's. No dissertation could have been done without your unconditional support, even when you had no clue what I was writing about.

Lastly, I have to thank Mariana for all the support and companionship she provided during this journey. Especially in the last weeks, where the pressure and uncertainty started to accumulate, she made sure I kept going and gracefully completed this document. Without you, this dissertation would never happened, as you continuously encouraged me to pursuit it, believing in me more than I did. For that faith, I thank you. The growth I have experienced in these last weeks, months, and years are a reflection of your motivation to keep going and chasing my dreams, pushing me to make the most out of my capabilities. Sharing this journey and helping you whenever I could were some of the best moments I have ever lived, I am very delighted to see you succeed by my side. Without a doubt, the best and most important part of this path.

David Castanho Terroso

"The cosmos is within us. We are made of star-stuff.
We are a way for the universe to know itself."

Carl Sagan

Contents

1 Introduction **1**

2 Literature Review **5**
2.1 Fluid Segmentation in OCT Scans . . . 5
2.1.1 Search Strategy . . . 5
2.1.2 Trends in Segmentation Methodologies . . . 6
2.2 Intermediate Slice Synthesis . . . 9
2.2.1 Architectures for Medical Image Super-Resolution . . . 9

3 Materials and Methods **18**
3.1 Datasets . . . 18
3.1.1 RETOUCH Dataset . . . 18
3.1.2 Centro Hospitalar Universitário de São João OCT Dataset . . . 20
3.2 Methodology for Fluid Segmentation . . . 21
3.3 Methodology for Intermediate Slice Synthesis . . . 26
3.4 Methodology for Fluid Volume Estimation . . . 32

4 Experiments **35**
4.1 Data Partition . . . 35
4.2 Experiment 1 - Multi-class Fluid Segmentation using a single U-Net . . . 37
4.2.1 Experiment 1.1 - Random Patch Extraction . . . 37
4.2.2 Experiment 1.2 - Large Patch Extraction . . . 37
4.2.3 Experiment 1.3 - Vertical Patch Extraction . . . 38
4.3 Experiment 2 - Multi-class Fluid Segmentation using Three Separate U-Net models 39
4.3.1 Experiment 2.1 - Multi-class Segmentation Loss . . . 39
4.3.2 Experiment 2.2 - Balanced Cross-entropy Loss . . . 39
4.4 Experiment 3 - Intermediate Slices Synthesis Using a GAN . . . 39
4.5 Experiment 4 - Intermediate Slices Synthesis Using a U-Net . . . 39
4.6 Experiment 5 - Fluid Volume Estimation Using Predicted Masks in Real OCT Volumes . . . 40
4.7 Experiment 6 - Fluid Volume Estimation Using Predicted Masks in Generated and Super-resolved OCT Volumes . . . 40

5 Results and Discussion **42**
5.1 Fluid Segmentation . . . 42
5.1.1 Experiment 1 - Multi-class Fluid Segmentation using a single U-Net . . . 42
5.1.2 Experiment 2 - Multi-class Fluid Segmentation using Three Separate U-Net models . . . 57

5.1.3 Experiment 1 vs Experiment 2: Comparison in the RETOUCH Dataset . 69
5.1.4 Experiment 1 vs Experiment 2: Comparison in the CHUSJ Dataset . . . 73
5.2 Intermediate Slice Synthesis 80
5.2.1 Experiment 3 - Intermediate Slices Synthesis Using a GAN 81
5.2.2 Experiment 4 - Intermediate Slices Synthesis Using a U-Net 86
5.3 Fluid Volume Estimation 88
5.3.1 Experiment 5 - Fluid Volume Estimation Using Predicted Masks in Real OCT Volumes 88
5.3.2 Experiment 6 - Fluid Volume Estimation Using Predicted Masks in Generated and Super-resolved OCT Volumes 90

6 Conclusion **98**
6.1 Limitations 103
6.2 Future Research 105

References **107**

A Impact of Random Initialization of the U-Net in Segmentation Performance **114**

List of Figures

1.1 OCT B-scans (A), acquired at fixed intervals along the azimuthal axis, form a volume representation of the posterior segment of the eye (B) [12]. 2
1.2 The three distinct fluid types on an OCT B-scan: IRF in red, SRF in green, and PED in blue [2]. 3
1.3 OCT scan of the retinal layers [13]. 3

2.1 Grouping of the articles included in the literature review. 6
2.2 Example of a CNN architecture used for binary fluid 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]. 7
2.3 Example of a framework that includes retinal layer delimitation and construction of a relative distance map (left side). The generated map, together with the original image, serves as input to the segmentation network (denominated ICAF-Net, by the authors) [32]. 9
2.4 López-Varela et al. [14] training process. 11
2.5 Framework developed by Xia et al. [41]. 12
2.6 Architecture of the method developed by Zhang et al. [44]. 13
2.7 Pipeline of the methodology proposed by Fang et al. [46]. 14
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 [57]. 16
2.9 Pipeline representing the framework developed by Tran and Yang [58]. The generator that produces the intermediate frame is represented by G, while the 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 [58]. . 17

3.1 Retinal layers B-scans from different patients, using Cirrus (left), Topcon (middle), and Spectralis (right) devices. Despite representing the same structure, the Cirrus scan, the retinal layers appear much thicker than in the scans obtained with other vendors. 19
3.2 Three B-scans with different dimensions (496 $\times$ 1024, 496 $\times$ 768, and 496 $\times$ 512 pixels, respectively) from the CHUSJ dataset. As indicated by scale bars in the bottom left corners of each B-scan, the resolution also differs across slices. 20
3.3 U-Net architecture [36]. 22

3.4 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. 23
3.5 Cirrus B-scan with its three corresponding patches of size 496 × 512. 24
3.6 Four vertical patches of dimension 496 × 128 extracted from a Cirrus B-scan. . . 24
3.7 Seven vertical patches of dimension 496 × 128 extracted from a Cirrus B-scan. . 25
3.8 Thirteen vertical patches of size 496 × 128 extracted from a Cirrus B-scan. . . . 25
3.9 Scheme explaining the input data of the generative models. Each frame refers to a B-scan from an OCT volume. Extracted from Tran and Yang [58]. 27
3.10 Example of a GAN framework, where $\mathscr{D}$ is the discriminator and $\mathscr{G}$ is the generator [71]. 28
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 [58]. 29
3.12 Patches of size 64 × 64 extracted from a Cirrus B-scan previously resized to 496 × 512. 30

5.1 Example of a patch extracted from a Cirrus OCT volume used in Experiment 1.1. In this patch, the background is distinguishable due to its darker intensity; however, the choroid is harder to be identified by an observer (or a model) due to the lack of anatomical context. 44
5.2 Example of a poor segmentation made by the model trained in Run 1 (right). In the left, the GT mask for the same image is shown. IRF is represented in red, SRF in green, and PED in blue. 44
5.3 Example of the segmentation performed by the model trained in Run 9 (right) and its respective GT mask (left). The OCT B-scan segmented is the same as in Figure 5.2, and the predicted IRF is represented in red. 46
5.4 Example of the segmentation performed by the model trained in Run 12 (right) and its respective GT mask (left). It is evident that the model confuses the choroid with the retina, as it segments IRF and PED within the choroid. IRF is represented in red, SRF in green, and PED in blue. 47
5.5 Training and validation loss curves for Run 13 (left two plots) and Run 17 (right two plots). Despite reaching the minimum validation loss at a similar number of epochs and presenting comparable training and validation loss curves, the final performance, shown in Table 5.3, differs significantly. 48
5.6 Predicted masks by the models trained on Run 16 (middle) and Run 20 (right) and their respective GT (left). SRF is represented in green and PED in blue. 49
5.7 Validation loss curves for models validated on fold 2. The curve on the left is from the model trained in Run 17 with four patches, while the middle curve is from the model trained in Run 21 with seven patches. The curve on the right is from the model trained in Run 23 with thirteen patches. 50
5.8 Segmentation errors in Cirrus B-scans. The GT fluid mask, seen on the left, shows a large region of PED fluid. Meanwhile, the predictions made by the models trained in Run 21 and 23, which respectively correspond to the B-scans in the middle and right, classify the center of this region as IRF. IRF is represented in red, SRF in green, and PED in blue. 51

5.9 Transformations applied to a vertical patch of a Cirrus B-scan. The original vertical patch (left) can be rotated a maximum of 10°, a transformation that is shown in the middle image. The image on the right represents the combination of a 10° rotation and an horizontal flip. It is seen that the rotation transform pads a significant portion of the image, reducing the information in the input. 51
5.10 Example of two OCT B-scans in which the retina is oriented differently. The retina in the image on the right is significantly more inclined. Both B-scans were obtained using a Cirrus device and do not present fluid. 53
5.11 Segmentation errors in Cirrus B-scans when trained without random rotations, using the same B-scan as in Figure 5.8. On the left, the GT mask is shown, while on the right, the predicted masks are displayed. IRF is represented in red, SRF in green, and PED in blue. 53
5.12 Predictions by the models trained using 5° (middle image) and 10° (right image) rotations, compared with the respective GT (left image). IRF is represented in red, SRF in green, and PED in blue. 55
5.13 Predictions made by the model trained on Run 32 on unseen Cirrus volumes of fold 1. IRF is represented in red, SRF in green, and PED in blue. 57
5.14 Predictions made by the model trained on Run 32 on unseen Spectralis volumes of fold 1. IRF is represented in red, SRF in green, and PED in blue. 58
5.15 Predictions made by the model trained on Run 32 on unseen Topcon volumes of fold 1. IRF is represented in red, SRF in green, and PED in blue. 58
5.16 Predictions (right) made by the binary IRF segmentation model and their respective GT (left). The prediction represented in the top shows an example of an accurate IRF segmentation, represented in red, while the bottom prediction reveals an oversegmentation in a slice that does not contain fluid. 60
5.17 Predictions (right) made by the binary PED segmentation model and their respective GT (left). In the top-right image, the model predicted PED fluid, represented in blue, in a region where it does not exist. The bottom images show undersegmentation by the model trained in Run 59. While this model is capable of detecting the fluid's location, it fails to segment it completely. 63
5.18 Comparison between the priority (left) and probability (right) merging approach. IRF is represented in red, SRF in green, and PED in blue. 64
5.19 Predictions made by the models trained in Run 42, 52, and 59 on unseen Cirrus volumes from fold 1 (last row), contrasted with the predictions made by the multi-class model from Experiment 1. IRF is represented in red, SRF in green, and PED in blue. 66
5.20 Predictions made by the models trained in Run 42, 52, and 59 on unseen Spectralis volumes from fold 1 (last row), contrasting with the predictions made by the multi-class segmentation model from Experiment 1. IRF is represented in red, SRF in green, and PED in blue. 67
5.21 Predictions made by the models trained in Run 42, 52, and 59 on unseen Topcon volumes of fold 1 (last row), contrasting with predictions made by the multi-class segmentation model from Experiment 1. IRF is represented in red, SRF in green, and PED in blue. 67
5.22 Small IRF (red) oversegmentation performed in a Spectralis B-scan. This type of predictions appear commonly across multiple B-scans. 69

5.23 IRF (red) segmentation in a Cirrus slice, represented in red, performed by a model trained using BCE (right). The corresponding GT is shown on the left. Although this example presents a decent segmentation, significant oversegmentation is seen on the top left of the image. 69
5.24 Three B-scans from the CHUSJ dataset with characteristics different from those in RETOUCH. The left scan shows definition on the choroid, while the middle and right scans present an odd orientation and significant noise. 74
5.25 Significantly different segmentations (middle images) of the same B-scan (left image) by different evaluators in the RETOUCH dataset. IRF is represented in red, SRF in green, and PED in blue. The regions shown in cyan in the rightmost image represent the regions where the two evaluators diverge [60]. 75
5.26 Visually similar PED regions are segmented in one dataset but not in the other. These inconsistencies can translate to lead to poorer model performances during testing. IRF is represented in red, SRF in green, and PED in blue. 75
5.27 SRF (green) and PED (blue) segmentation in two OCT B-scans from CHUSJ. The regions segmented by the model as PED were not considered in the GT, but present characteristics similar to regions segmented as PED in the RETOUCH dataset (used for training). 76
5.28 SRF (green) and PED (blue) segmentations (right) in a B-scan from CHUSJ, compared to its GT (left). This shows the model's tendency to oversegmentation. . . . 77
5.29 Segmentation of IRF regions (right) compared to its respective GT (left). The prediction of fluid regions in the choroid is also noticeable. IRF is represented in red, SRF in green, and PED in blue. 77
5.30 Segmentation of multiple fluids in the choroid region of the B-scans. It is a common problem and significantly reduces the Dice coefficients, as reported in Table 5.18. IRF is represented in red, SRF in green, and PED in blue. 78
5.31 Fluid segmentations performed by the multi-class model in noisy B-scans from the CHUSJ dataset. IRF is represented in red, SRF in green, and PED in blue. . . 79
5.32 Processed fluid segmentations performed by the multi-class model in the examples of Figures 5.29 and 5.30. IRF is represented in red, SRF in green, and PED in blue. 81
5.33 Processed fluid segmentations performed by the multi-class model, in the noisy scans shown in Figure 5.31. IRF is represented in red, SRF in green, and PED in blue. 82
5.34 Example of two pairs of generated slices, one from Cirrus and one from Spectralis, which capture the retinal layers. 83
5.35 Surrounding slices of the B-scans generated in Figure 5.34. 84
5.36 Examples of generated B-scans with fluid. 85
5.37 OCT B-scans generated using the U-Net. The images on the top row show the original B-scans while those on the bottom row are the corresponding generated slices. 86
5.38 Volume estimated for each fluid in OCT volume using the masks predicted by the segmentation model compared with the volumes calculated using the GT masks. The OCT scans considered are the same as in 5.21 and the gray line marks the points in which the fluid volume in the GT is equal to the predicted. 90
5.39 Volume estimated for each fluid in the OCT volumes using the GT masks, compared with the volumes estimated in columns "P", "G", and "E". 92

5.40 Volume estimated for each fluid in the OCT volumes using the masks predicted by the segmentation model compared with the volumes calculated using the masks from columns “G” and “E”. 93
5.41 Volume estimated for each fluid in the OCT volumes using the masks predicted by the segmentation model in the generated dataset, compared with the volumes calculated in the enhanced dataset. 93
5.42 Example of IRF (red) segmentation in the enhanced volume TRAIN039. 96
5.43 Example of SRF (green) and PED (blue) segmentation in the enhanced volume TRAIN012. 97

List of Tables

3.1 Summary of the number of OCT volumes, B-scans per volume, total number of B-scans, and associated macular diseases in each dataset. 19

3.2 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 $\times$ H $\times$ W notation, where C is the number of channels, H is the height, and W is the width. The inputs have a size of $1 \times 64 \times 64$. Adapted from Tran and Yang [58]. . 31

4.1 Number of OCT volumes per vendor in each fold, considering a 5-fold split. . . . 36

4.2 Device-wise distribution of OCT volumes across the four folds used for training and validation in OCT slice synthesis. 37

4.3 Summary of the experiments performed in this dissertation. 41

5.1 Dice coefficients for every vendor and fluid for the runs done in Experiment 1.1. The conditions were the same for both sets except the extracted patches, which are different in every run due to the random patch extraction method used. 43

5.2 Dice coefficients for every vendor and fluid for the runs performed in the Experiment 1.2. 45

5.3 Dice coefficients for every vendor and fluid obtained from runs using four vertical patches extracted from each B-scan. In "Set 4", the models were trained for 100 epochs, while in "Set 5" they were trained for up to 200 epochs with a 100-epoch patience. The transformations applied in these sets were the same: horizontal flipping and a maximum rotation of 10°. 47

5.4 Dice coefficients for every vendor and fluid from runs performed using seven (Runs 21 and 22) and thirteen (Runs 23 and 24) vertical patches extracted from each B-scan. Only two folds were used in order to understand the viability of using this number of patches, which are represented in the column "P". Runs 17 and 18 are shown here to make the comparison between the number of patches easier. The values shown in bold represent the best value for the models validated on the same fold. 49

5.5 Dice coefficients for every vendor and fluid using seven vertical patches per image for training. The only transformation applied to these runs was horizontal flipping, without rotation. Runs 21 and 22 are also shown, promoting an easier comparison. 52

5.6 Dice coefficients for every vendor and fluid using seven vertical patches per image for training. In these runs, horizontal flipping and 5° rotation were used, instead of the usual 10° rotation. The results obtained in Runs 21 and 22 are also shown, enabling a direct comparison. 53

5.7 Dice coefficients for every vendor and fluid using seven vertical patches per image for training. The mean scores are calculated only for the B-scans that contain that fluid. For example, Cirrus IRF is the mean of the IRF Dice coefficient across all the Cirrus slices in the validation fold that contain IRF. The transformations utilized were the same as in Table 5.6. 54
5.8 Dice scores for every vendor and fluid using four vertical patches per image for training. The rotation applied in these runs was of 5°, instead of the previously used 10°. "Set 7" and the best-performing run in this set (Run 32) are also shown, promoting an easier comparison between the use of four and seven patches. The underlined values correspond to the best performances between the models trained in Run 32 and Run 33. 55
5.9 Dice coefficients for every vendor and fluid in the reserved fold (fold 1). The first row reports the mean Dice coefficient across all slices. The second row shows the mean for slices containing the fluid specified in the respective column, while the third row shows the mean for the slices without that fluid type. 56
5.10 IRF Dice coefficients for every vendor. Runs 37 to 40 utilize the multi-class 5-fold split from Experiment 1, while the Runs 41 to 44 use the fluid-specific 5-fold split. The results are presented both for every slice and for only the slices containing IRF. 59
5.11 SRF Dice coefficients for every vendor. Runs 45 to 48 utilize the multi-class 5-fold split from Experiment 1, while the Runs 49 to 52 use the fluid-specific 5-fold split. The results are presented both for every slice and only for the slices which present SRF. 61
5.12 PED Dice coefficients for every vendor. Runs 53 to 56 utilize the multi-class 5-fold split from Experiment 1, while the Runs 57 to 60 use the fluid-specific 5-fold split. The results are presented both for every slice and only for the slices which present PED. 62
5.13 Dice coefficients for every vendor and fluid in the reserved fold (fold 1), using the models from Runs 42 (IRF), 52 (SRF), and 59 (PED). The first block shows results using the priority-based merging strategy, while the second block presents the results using the probability-based merging strategy. In each block, the first row reports the mean Dice score across all slices. The second row shows the mean Dice score for slices containing the specified fluid, while the third row shows the mean Dice score for slices without that fluid. Bold values highlight the best performance for each corresponding row between merging strategies (e.g., the "All" row in the first block is compared with the "All" row in the second block). 65
5.14 IRF Dice coefficients for every vendor, using the binary segmentation U-Net for IRF. The loss function used in Runs 61 and 62 is the weighted cross-entropy. This model was trained on two different validation folds: validation fold 0 from the multi-class 5-fold split, and validation fold 4 from the IRF-specific 5-fold split. These results can be compared, respectively, to those obtained in Runs 40 and 43. The metrics are compared between rows with the same name, so that the values in bold are the best across the four experiments. 68
5.15 Dice scores for every vendor and fluid in the reserved fold (fold 1), using the best models from Experiment 1 and Experiment 2. In Experiment 1, the model trained on Run 32 was selected, while in Experiment 2 the chosen models were those from Runs 42 (IRF), 52 (SRF), and 59 (PED). 70
5.16 Dice scores for every fluid in "Set 7" and "Set 10, 12, and 14", compared with the results obtained in the two networks proposed by Alsaih et al. [77]. 71

5.17 Dice scores for every vendor and fluid in "Set 7", "Set 10, 12, and 14", compared with the results obtained in the authors' previous implementation of the work of Tennakoon et al. [65]. . . . 73
5.18 Dice scores for every vendor and fluid in the CHUSJ dataset, using the best models from Experiment 1 and Experiment 2. . . . 74
5.19 Dice coefficients for every fluid in the CHUSJ dataset, using the best models from Experiment 1 and Experiment 2, after discarding the fluids predicted outside the retina. . . . 79
5.20 PSNR and SSIM values in every device, based on the generated slices in each validation fold. . . . 81
5.21 Volume of each fluid calculated for the masks segmented in the RETOUCH (GT) and the masks segmented by the best segmentation model from Experiment 1 and 2 (P). The volumes correspond to the total fluid volume in the OCT scan, in mm^3. 89
5.22 Volume of each fluid calculated for the masks segmented in the RETOUCH (GT) and for the masks predicted by the best segmentation model in the original RETOUCH images (P), in the generated slices (G), and when the original dataset is enhanced, combining alternating the real images with B-scans generated between them (E). The values correspond to the total fluid volume in the OCT volume, measured in mm^3, with the relative error to the GT shown in parentheses below each volume. . . . 91
5.23 Comparison in Dice scores for every vendor and fluid in the reserved fold (fold 1), using the best segmentation model to segment the original B-scans and their generated equivalent. . . . 95

A.1 Dice scores for every vendor and fluid from multiple runs performed in the same conditions. "Set 1" includes Runs 1 to 5, all validated on fold 2. Meanwhile, "Set 2" includes Runs 6 to 14, validated on fold 3. All runs in each set were trained under the same conditions for 100 epochs. The random transformations applied were horizontal flipping and a maximum rotation of 10°. This provides insight into how the inherent randomness of the U-Net affects the segmentation performance. 114

List of Acronyms

AMD	Age-related macular degeneration
AMI	Anisotropic meta interpolation
BCE	Binary cross-entropy
BM	Bruch's membrane
CDFI	Compression-driven network design for frame interpolation
CHUSJ	Centro Hospitalar Universitário de São João
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 squared 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
VF	Validation fold

Chapter 1

Introduction

The vision is the most important and complex sense in humans, 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. 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, and 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) occur 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].

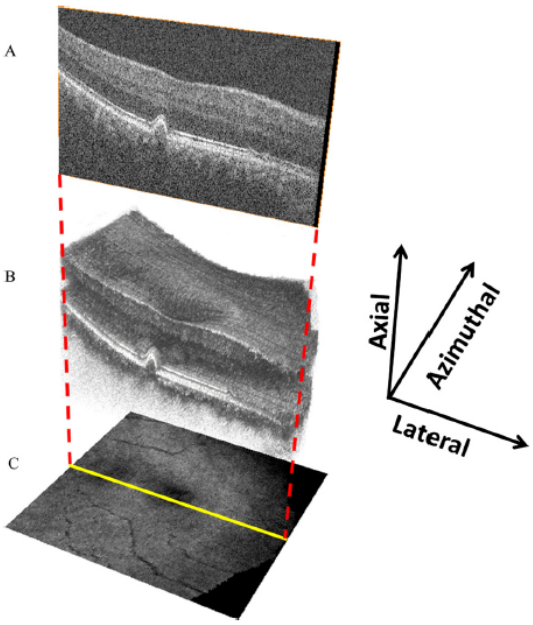


Figure 1.1: OCT B-scans (A), acquired at fixed intervals along the azimuthal axis, form a volume representation of the posterior segment of the eye (B) [12].

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]. By reducing this distance and improving the OCT volume's resolution along this axis, the fluid segmentation becomes more accurate, when done both by clinicians and by segmentation models that leverage multiple consecutive slices as input [15]. Given that the inter-slice space is reduced, the estimated segmented volume will also be closer to the real fluid volume.

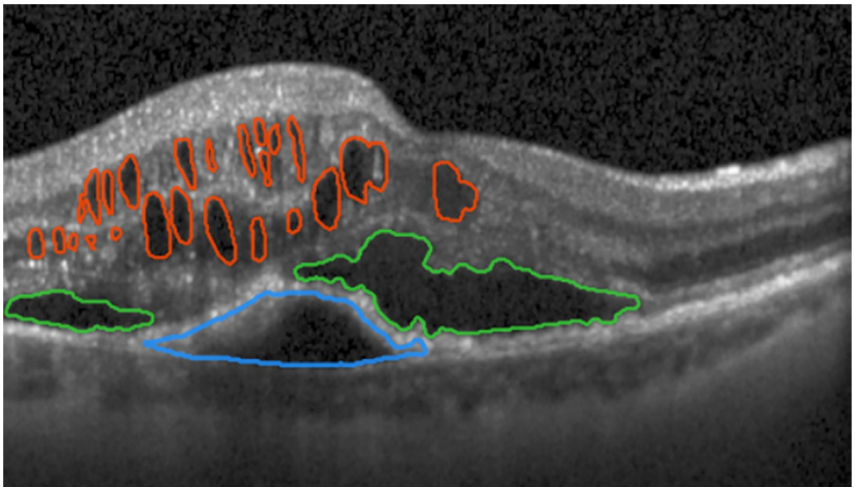


Figure 1.2: The three distinct fluid types on an OCT B-scan: IRF in red, SRF in green, and PED in blue [2].

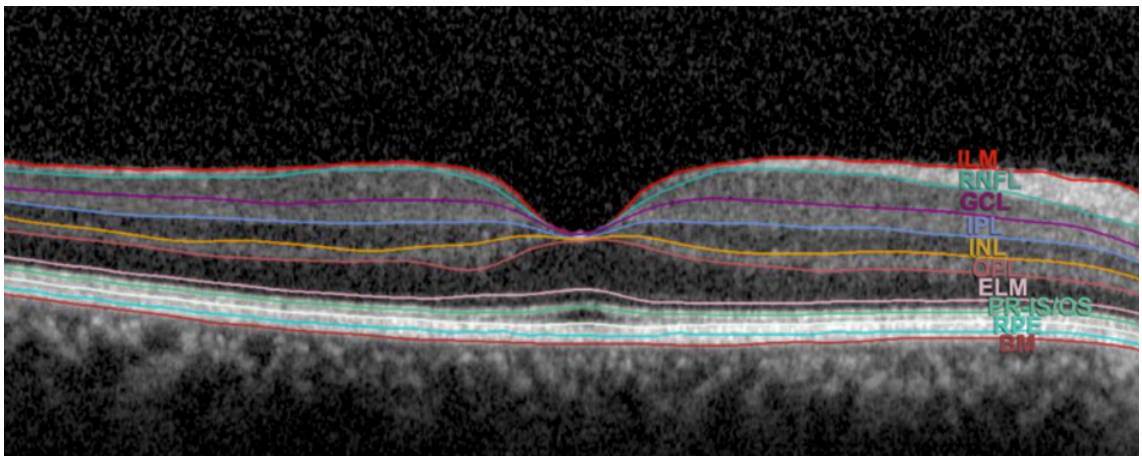


Figure 1.3: OCT scan of the retinal layers [13].
ILM, internal limiting membrane; RNFL, retinal nerve fibre layer; GCL, ganglion cell layer; IPL, inner plexiform layer; INL, inner nuclear layer; OPL, outer plexiform layer; ELM, external limiting membrane; PR-IS/OS, photoreceptor inner segment/outer segment; RPE, retinal pigment epithelium; BM, Bruch's membrane.

Considering the previous statements, the general objectives of this dissertation 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 are 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 its predicted masks.

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", this dissertation is composed of the following chapters: "Literature Review", "Methods and Materials", "Experiments", "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 and Materials" chapter details the selection of the dataset for the methods applied in the experiments performed during the dissertation. The "Experiments" chapter presents how the methods introduced in the previous chapter were applied in practice, through experiments 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 its respective task. The differences in performance between experiments, the relationships with the implemented changes, and comparisons with results from the literature are also discussed. Finally, the "Conclusion" summarizes 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 identify the most recent trends in 2D fluid segmentation of OCT volumes using deep learning, as well as the application of generative models for intermediate slice synthesis.

2.1 Fluid Segmentation in OCT Scans

In the fluid segmentation state-of-the-art research, articles were retrieved using a systematic review methodology. The next subsection details the retrieval process and the criteria for inclusion and exclusion of the articles. Section 2.1.2 presents the trends in methodologies used 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, conference proceedings, non-English articles, and articles without the full text accessible were excluded.

A total of 708 articles were initially identified, of which 133 were duplicates. Afterwards, 575 articles were screened, based on 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-based fluid segmentation in OCT volumes included and form the foundation of the literature reviewed in this dissertation.

2.1.2 Trends in Segmentation Methodologies

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 segmentation [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). Additional grouping criteria include 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 networks (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.

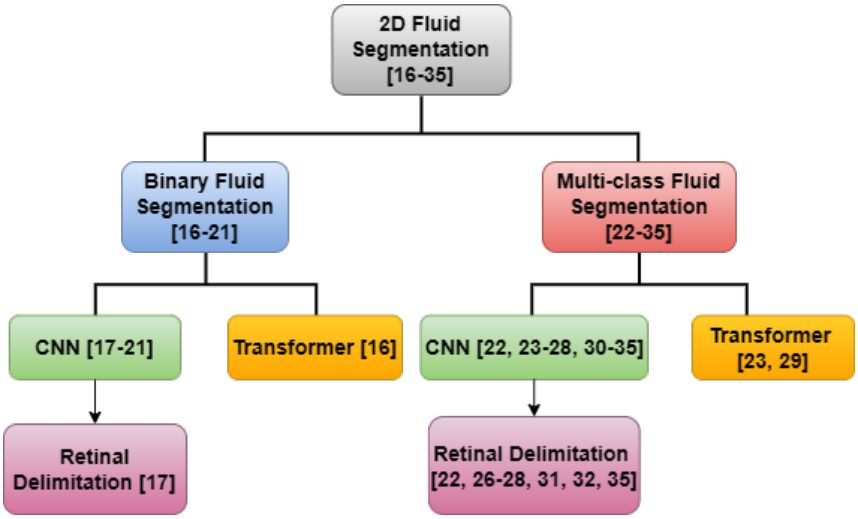


Figure 2.1: Grouping of the articles included in the literature review.

The work of Pawan et al. [17] is the only one in binary segmentation that restricts the input of the CNN to the content within the retinal layer. This approach is frequently observed in the papers focused on multi-class segmentation. In [17], this is achieved by performing a retinal layer segmentation and assigning all the values outside the boundaries to zero. The result of this operation serves as input to the 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 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 decoder is responsible for generating the fluid segmentation map, while the other predicts the distance map. This distance map is then converted to a binary mask by applying a threshold value: pixels with greater distance than this value are set to one, and all others to zero. The intersection

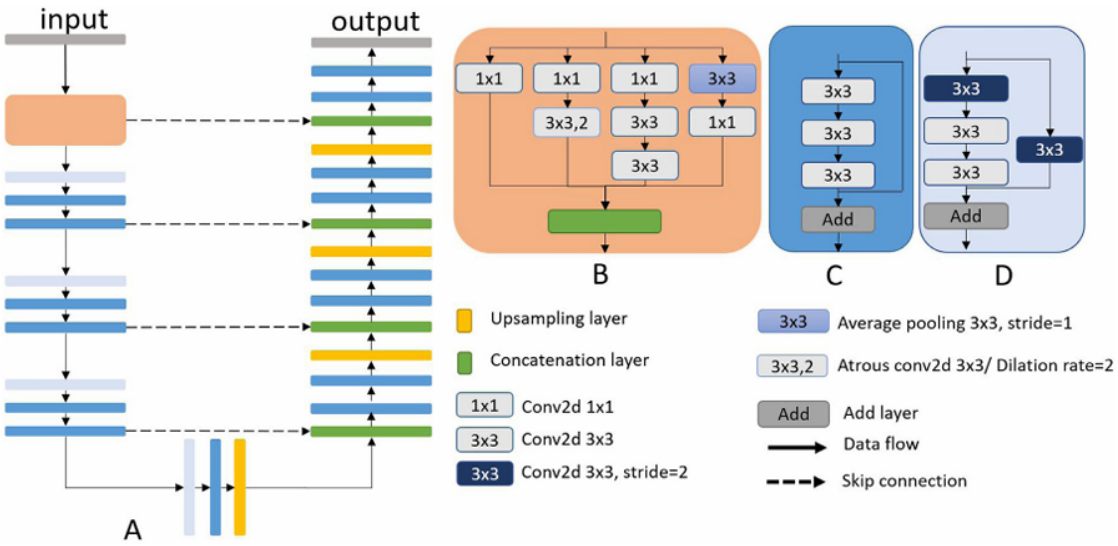


Figure 2.2: Example of a CNN architecture used for binary fluid segmentation. Image A depicts the neural network architecture, B shows the used multi-scale block, while C and D exhibit the residual convolutional blocks [19].

between the fluid segmentation map and this binary mask 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 that 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. 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 binary fluid segmentation [16]. While CNNs capture the information from local receptive fields, visual transformers integrate features from global receptive fields. In the neural network proposed 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, which results in a broader range of implementation strategies. While all of them segment two or more types of fluids, Hassan et al. [28] and Padilla-Pantoja et al. [33] also segment other retinal 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 by Quek et al. [16], Liu et al. [23] have integrated transformers in the bottleneck section of a segmentation network inspired by the U-Net. The authors 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 a transformer is included in the refinement branch. 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 relying on algorithms that detect the distinct transition in intensity or texture between the retina and the surrounding tissue or 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 constrain the final output by limiting segmentation to the retinal layer zone, as observed in Mantel et al. [35].

The input can be conditioned in multiple ways. Xing et al. [31] crop the image to fit the region of interest. In [22, 27, 32], the OCT B-scan is combined with the retinal delimitation result through concatenation. Conversely, 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 for returning a distance map (relative to 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 the internal limiting membrane (ILM), which is concatenated with the input B-scan in a CNN. Starting with the retinal delimitation, the relative distance to the 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 each pixel's relative position 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].

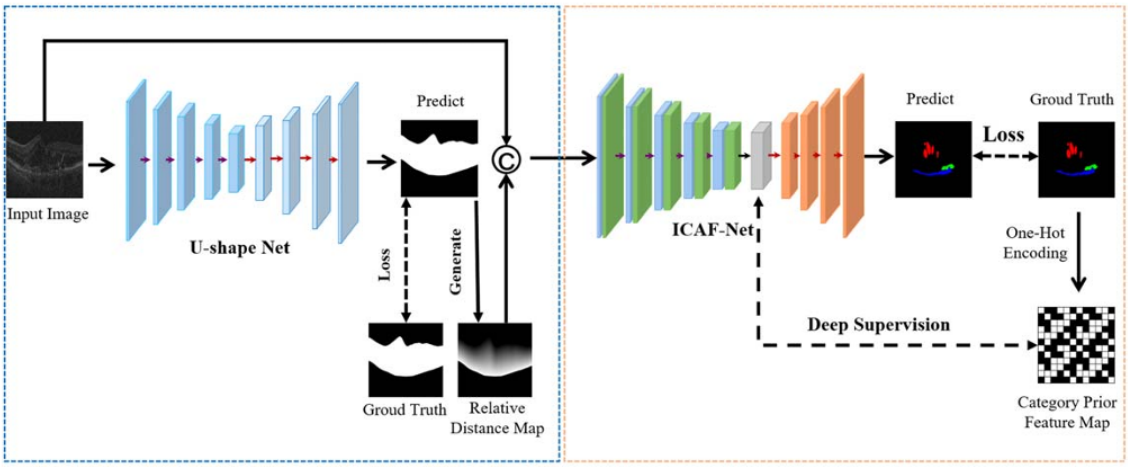


Figure 2.3: Example of a framework that includes retinal layer delimitation and construction of a relative distance map (left side). The generated map, together with the original image, serves as input to the segmentation network (denominated ICAF-Net, by the authors) [32].

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 applied either intra-slice, improving the resolution of each slice in the volume along one plane, or inter-slice, enhancing the resolution of the volume along one axis by generating one or more slices between a pair of original ones. Some frameworks may even combine both approaches [39].

The use of GANs for generating slices between pre-existing slices is a 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 for Medical Image Super-Resolution

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, 48]; 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 [49, 50, 51, 52].

2.2.1.1 Inter-slice SR

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 serves as the ground truth (GT) and is used in the assessment of the quality of the image 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. Since these generated slices lack corresponding GT, the network's performance is regulated by using these two new synthetic slices for generating an additional intermediate slice (Central Fake 2), which is then compared to the true Central. Consequently, if the generation of Pre- and Post-Central fake images 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 than those obtained through linear interpolation.

The work by Xia et al. [41] demonstrates the enhancement of inter-slice resolution in MRI through the use of multiple networks and a multi-scale discriminator that considers both global and local features by processing different image scales and a scheme that illustrates this framework can be seen in Figure 2.5. The networks in this framework receive two slices ($x_{z\pm1}$) and generate a new one between them (x_z). The generator (G) is responsible for producing the intermediate slice, while the discriminator (D) learns to distinguish the synthesized image from the real image. The loss of these networks is also determined by the similarity of feature maps between 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 input image. Based on the optical flow, depth maps, and their linear interpolation ($\bar{x_z}$), another U-shaped network generates the intermediate slice. The result from this network and the generator's output (to which Gaussian blurring is

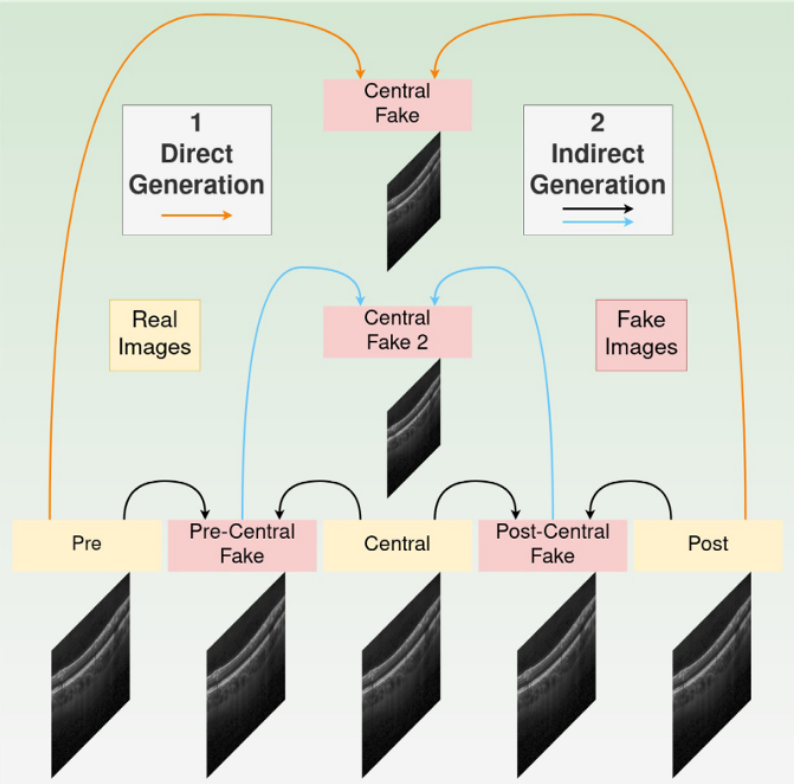


Figure 2.4: López-Varela et al. [14] training process.

applied) are evaluated by the discriminator. Once again, the discriminator evaluates both images in separate feature spaces, processing each input at a different scale to extract both global and local representations for comparison.

The image generated by the network using optical flow and depth ($\hat{x_z}$) pays special attention to the transitions between the two input slices. The point of inputting these images to the discriminator is to encourage it to recognize these characteristics in the generator's output. Additionally, the discriminator is exposed to blurred images to further incentive the generator to produce sharper images.

On the other hand, 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 evaluate 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 as real or fake. Meanwhile, the evaluation at a smaller field of view integrates an attention network, which focuses on the most informative regions of the image to support the local discriminator and the object identifier.

The local discriminator classifies the image as real or fake based on fine-grained features extracted by the attention network. Meanwhile, the object classifier, which is a much shallower

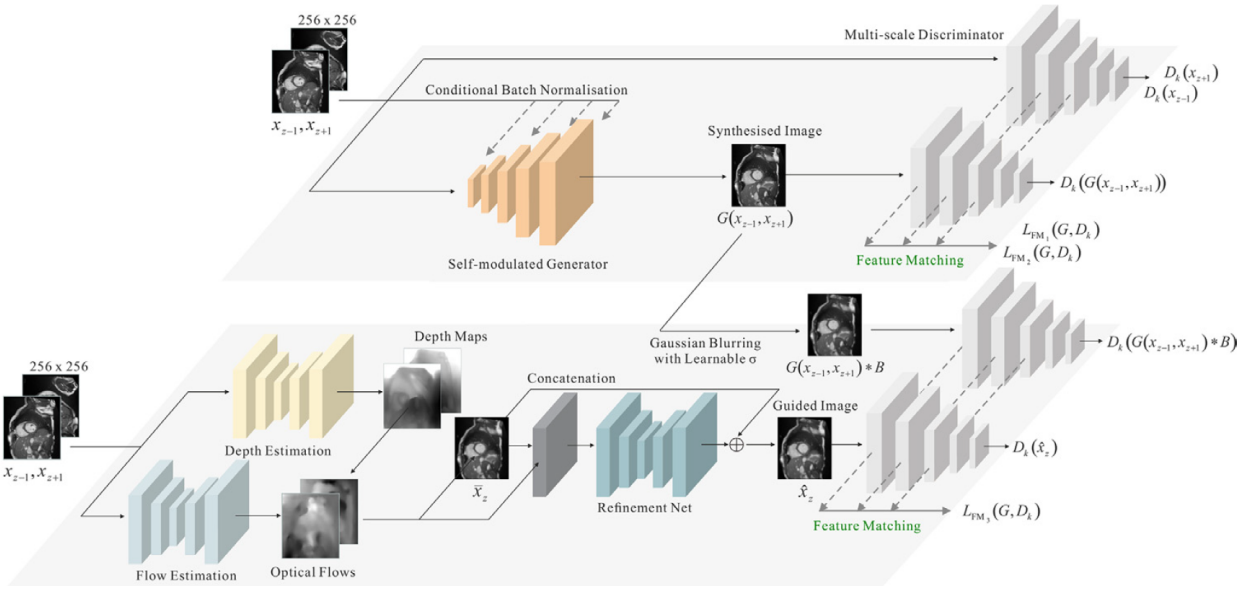


Figure 2.5: Framework developed by Xia et al. [41].

network, verifies whether certain structures present in the real image also appear in the fake one, using the output of the attention network.

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

2.2.1.2 Intra- and Inter-slice SR

As an example of combining intra-slice SR with inter-slice SR, 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. [45], a benchmark work in the field of medical image SR.

Peng et al. [45] introduced the AMI network as a single-image SR method designed to enhance the slice's resolution along the axis with the lowest spatial resolution - in this case, the axial axis. The AMI network is applied independently to the sagittal and coronal slices, generating 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 takes two adjacent axial slices as input and generates an 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 network as in Peng et al. [45], 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. A scheme of this method is shown in Figure 2.6.

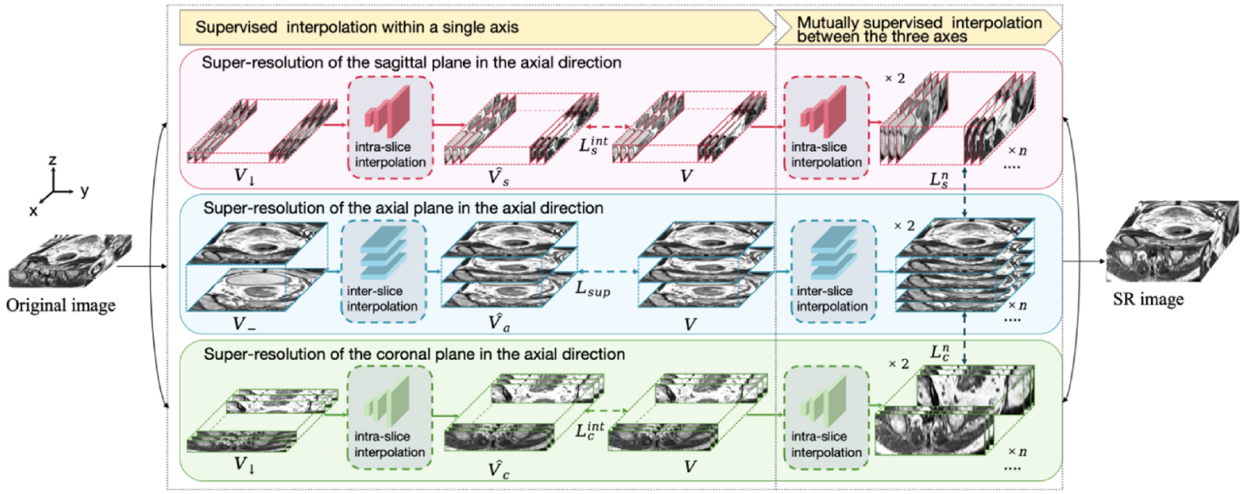


Figure 2.6: Architecture of the method developed by Zhang et al. [44].

A similar approach was done by Fang et al. [46]. In this framework, three CNNs (one for each axis) are trained with the goal of generating the intermediate slices along the axis with a lower inter-slice resolution, which is the axial axis. Similarly to what is done in Zhang et al. [44], two CNNs (one for the sagittal plane and one for the coronal plane) are trained to increase the resolution along the axial axis in the images of their respective plane. Both CNNs used in this work correspond to the single-image SR model developed by Niu et al. [53]. This CNN is an holistic attention network which improves the image's resolution by capturing both local and global contextual information. It uses holistic attention modules that combine channel attention (which emphasizes informative feature maps) and spatial attention (which highlights the most important regions of the image). This attention mechanism allows an image reconstruction with finer textures and sharper details in the super-resolved output.

Instead of using a GAN to generate the intermediate slices along the axial plane, the authors apply another CNN. This network begins by extracting feature maps through space-to-depth transformation operations, which reorganize the spatial information into the channel dimension. The resulting features are then processed by a U-shaped architecture which captures both local and global context. Finally, a depth-to-space transformation is applied at the output layer, to reconstruct the intermediate axial slice at the original spatial resolution.

All the networks are trained on downsampled CT volumes. After the first interpolation, all the outputs are compared to their respective GT and their loss is calculated. Then, another interpolation is performed in each axis, and a loss compares the generated images not with the GT, but with each other. This ensures coherence in the outputs predicted on different axis and transference of knowledge between them. During testing, inference is performed using a weighted prediction, with each axis' network contributing equally. In Figure 2.7, the pipeline describing this implementation is shown.

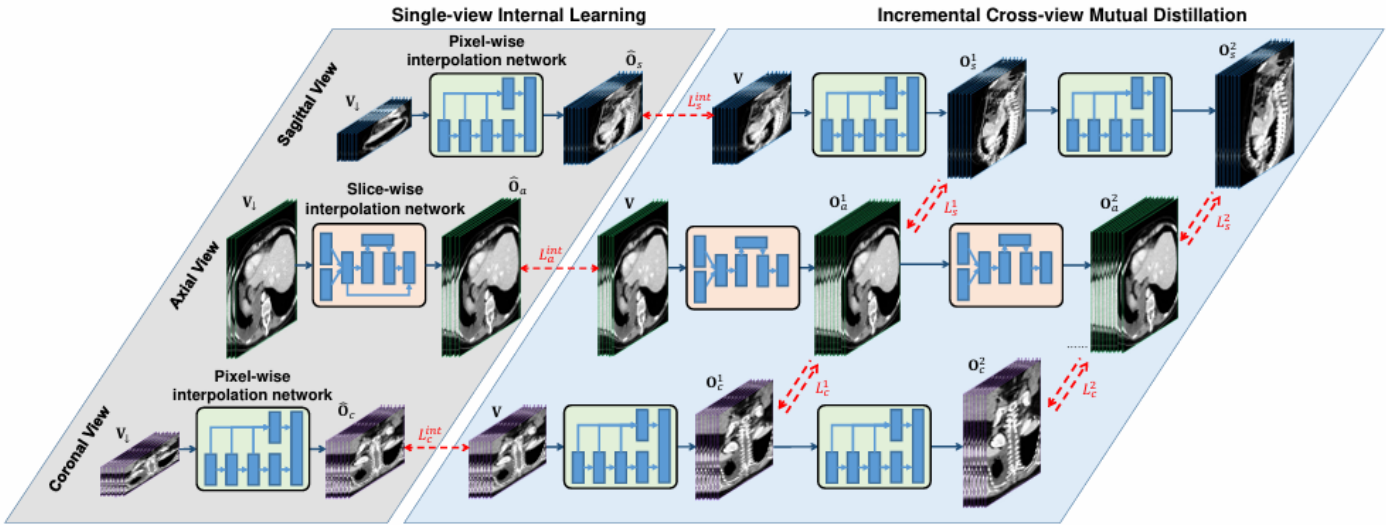


Figure 2.7: Pipeline of the methodology proposed by Fang et al. [46].

Nimitha and Ameer [47] also perform intra-slice and inter-slice resolution enhancement. However, instead of improving both resolutions simultaneously as seen in Zhang et al. [44] and Fang et al. [46], the enhancement is done sequentially. The pipeline starts with a low-resolution (LR) volume, in which axial slices are firstly enhanced to becoming high-resolution (HR). Based on these HR axial slices, an intermediate slice is generated between each pair of consecutive axial slices.

The network used in the enhancement of intra-slice resolution is a GAN. The generator starts with an in-plane and out-of-plane attention modules that extract features from the MRI slices adjacent to the target one. These extracted features are then passed through a U-shaped network that reconstructs the intermediate image. However, the number of upsampling blocks is higher than the number of downsampling blocks to ensure that the output has higher spatial resolution than the input.

To generate the intermediate slices, the compression-driven frame interpolation (CDFI) network, a state-of-the-art video frame interpolation CNN developed by Ding et al. [54], was utilized. CDFI is built on top of the AdaCoF module [55], which is particularly powerful at handling a wide range of motions between images.

The CDFI network first extracts features using a U-Net architecture. These features are passed from the U-Net encoder to a pyramid network with the same number of levels as the encoder. At

each level, a 1x1 convolution is applied to the feature maps to reduce dimensionality and prepare them for motion estimation.

Two sub-networks take the U-Net's output and estimate the parameters of the AdaCoF operation. The U-Net's output is also combined with an occlusion mask to create the first candidate intermediate slice.

The pyramid network's outputs are warped using AdaCoF and then passed into a synthesis network, which generates the second candidate intermediate slice. The two candidate intermediate slices are combined through a second occlusion mask to generate the final output. This model maintains visual realism from the first slice while ensuring a coherent pixel movement by using the second slice.

Georgescu et al. [48] used a similar approach for enhancing intra- and inter-slice resolution in CT and MRI scans through two separate CNNs. One CNN was tasked with enhancing the resolution of 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 CNNs for intra- and inter-slice resolution are similar. Both networks consist of 10 consecutive convolutional layers, with the first 5 having 32 filters. The number of filters in the following layers changes according to the output size. This output size differs between the two CNNs because the first increases the resolution in two directions, while the other only increases the resolution in one direction.

2.2.1.3 Three-dimensional SR

The methodologies that use 3D GANs are similar to one another, as they all employ network architectures based on those used in 2D images. As the 3D GANs 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 [49, 50, 51, 52].

2.2.1.4 Video Frame Interpolation for Inter-slice SR in Medical Imaging

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 video, these frameworks utilize two consecutive frames to generate an intermediate one, which is similar to what is seen in the previous papers that 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 medical imaging volumes is distance [46, 56].

Believing that the concepts that work on video also work on CT and MRI, Gambini et al. [56] implemented a state-of-the-art method of video interpolation to generate intermediate slices in CT and MRI. The method used was the real-time interpolation flow estimation (RIFE) [57]. The CNN incorporated in RIFE learns the pixel movements between frames from 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 GT; 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 [57].

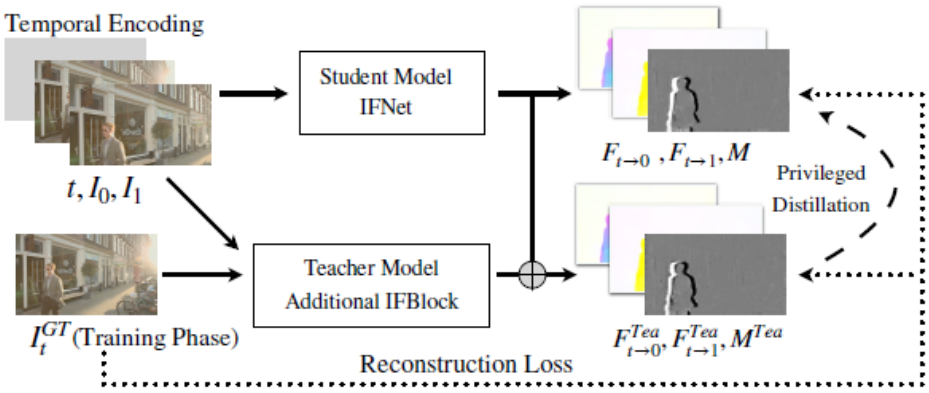


Figure 2.8: Pipeline that describes the RIFE framework. The student model attempts to generate the intermediate frame, while the teacher refines the frame generated by the student so that it looks more similar to the middle frame. The results from both networks are evaluated on the reconstruction loss [57].

Tran and Yang [58] present an alternative framework to the video frame interpolation. Instead of relying on contextual or optical flow to model pixel changes, two GANs are used for predicting 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 assesses how accurately the image is reconstructed and evaluates how well it fools the discriminator.

The image resulting from the generator is given as input to the discriminator, which receives real and fake images as input and is responsible for correctly classifying it. The discriminator's prediction is compared to the true image's label, and the resulting loss is used in the adjustment of the generator weights.

After training the first GAN, the second one is trained, which is a pix2pix [59]. Similar to what is seen in the work of Gambini et al. [56] and Huang et al. [57], 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 GAN 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, after the training of the first network. The pipeline that describes this framework is shown in Figure 2.9.

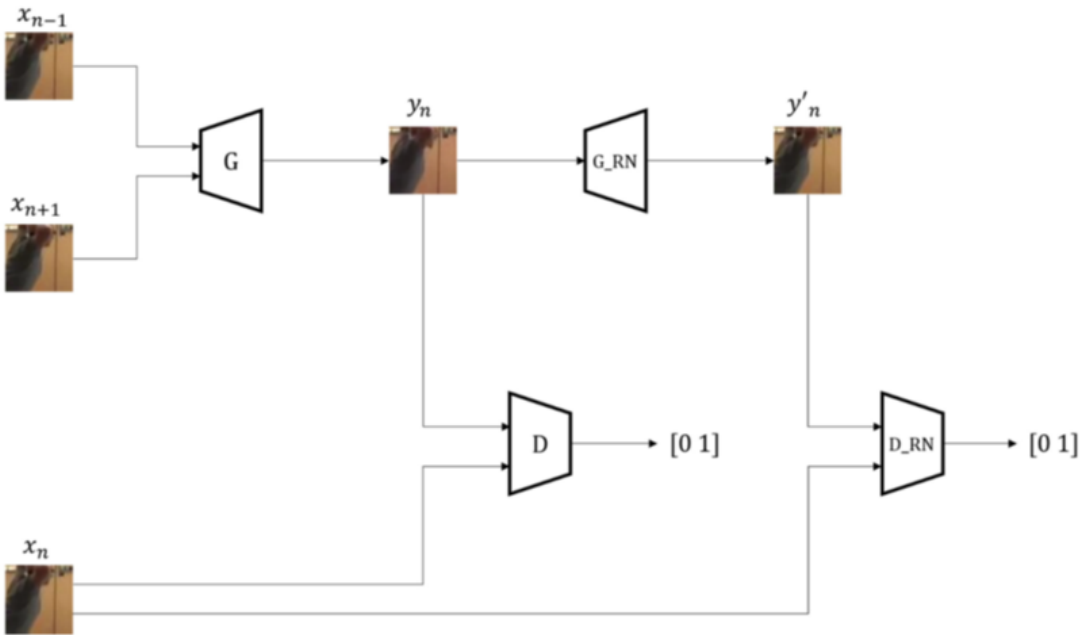


Figure 2.9: Pipeline representing the framework developed by Tran and Yang [58]. The generator that produces the intermediate frame is represented by *G*, while the discriminator is labeled *D*. The pix2pix generator is denoted by *G_RN* and the discriminator is represented by *D_RN*. x_{n-1} and x_{n+1} 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 [58].

Chapter 3

Materials and Methods

This chapter starts with an overview of the datasets selected for the fluid segmentation and intermediate slice synthesis tasks, along with the requirements needed for the training of each model and the reasoning behind the selection.

3.1 Datasets

The training of a model for the segmentation of different types of fluid in OCT volumes requires a large number of images annotated with those types of fluids. However, the manual segmentation of large amounts of B-scans is a laborious process, which results in a shortage of publicly available annotated OCT datasets. Furthermore, the majority of these datasets contain a limited quantity of images.

3.1.1 RETOUCH Dataset

The dataset selected to be used in this dissertation was the RETOUCH dataset [60]. It 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 a training (70 volumes) and a test set (42 volumes). Only the OCT volumes in the training set have annotations of the retinal fluids (IRF, SRF, and PED). The annotations present in this dataset were performed by two clinicians and both segmentations were considered correct. In this dissertation, only the annotated volumes were used for training and testing the fluid segmentation.

From those 70 OCT 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. Figure 3.1 illustrates how different axial resolutions affect the visualization of the retinal layers. For example, each voxel in the Cirrus volumes has a height of 1.95 μm, while each voxel in the Spectralis volumes has an height of 3.87 μm. These differences in height cause the same structures to appear larger in Cirrus B-scans, as seen in this figure.

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) typically contain 128 B-scans; however, two volumes include only 64 B-scans. In total, 6838 B-scans were used on the training and testing of the segmentation models.

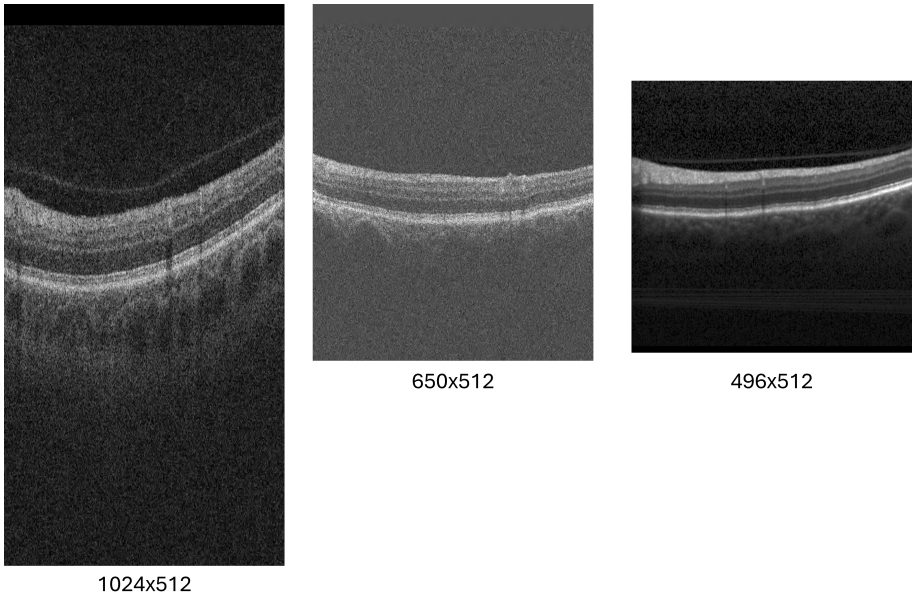


Figure 3.1: Retinal layers B-scans from different patients, using Cirrus (left), Topcon (middle), and Spectralis (right) devices. Despite representing the same structure, the Cirrus scan, the retinal layers appear much thicker than in the scans obtained with other vendors.

When compared with other renowned OCT datasets annotated with retinal fluid, such as the Duke dataset [61], the two datasets from the University of Minnesota [62, 63], and the Lu et al. [27] dataset, the RETOUCH presents a significantly larger quantity of annotated volumes. It also offers greater variety, as the volumes were obtained using four different devices, unlike the other datasets, which include scans from only one device. 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: Summary of the number of OCT volumes, B-scans per volume, total number of B-scans, and associated macular diseases in each dataset.

	DUKE2015 [61]	**UMN2017 [62]**	**UMN2018 [63]**	**LU2019 [27]**	**RETOUCH [60]**
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.

The diversity and dimension of the RETOUCH dataset make it one of the most widely used datasets for the development of fluid segmentation models [22, 23, 24, 26, 27, 29, 31, 32]. These aspects also motivated the selection of the RETOUCH dataset for the implementation of fluid segmentation models in this dissertation.

Regarding 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.

3.1.2 Centro Hospitalar Universitário de São João OCT Dataset

A private dataset provided by the Centro Hospitalar Universitário de São João (CHUSJ) was also used for external evaluation of the proposed segmentation models. This dataset comprises six Spectralis OCT volumes, containing 19 slices each. The volumes vary in resolution, dimensions, noise levels, and contrast, offering diverse testing scenarios. All volumes include at least one type of fluid, though the overall fluid presence is limited. Unlike RETOUCH, which includes the annotations from two clinicians, this dataset was segmented by a single one.

Despite the scans in this dataset also being obtained with a Spectralis device, the images have different dimensions. In RETOUCH, all the Spectralis scans present a size of 496 $\times$ 512 pixels (height $\times$ width; this notation is used for images shapes throughout this dissertation), while in CHUSJ dataset the images differ in width, depending on the OCT volume. The dimensions of the OCT B-scans vary between 496 $\times$ 1024, 496 $\times$ 768, and 496 $\times$ 512 pixels. Since the images in this dataset do not contain metadata specifying the resolution of each slice, the slices had to be observed individually in order to determine whether the pixels in each B-scan corresponded to the same physical dimensions. Figure 3.1 shows three OCT B-scans with different dimensions, highlighting the variation in resolution across the images.

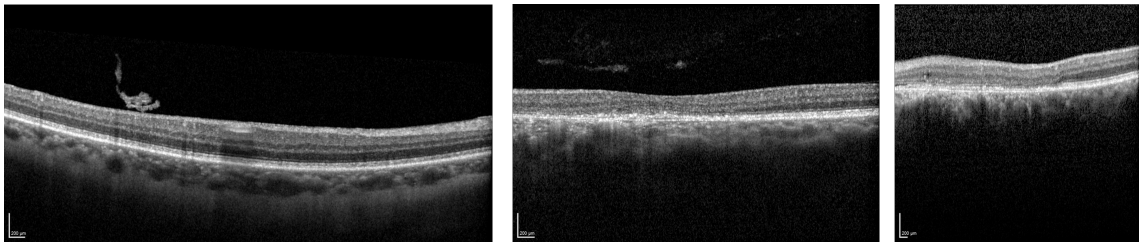


Figure 3.2: Three B-scans with different dimensions (496 $\times$ 1024, 496 $\times$ 768, and 496 $\times$ 512 pixels, respectively) from the CHUSJ dataset. As indicated by scale bars in the bottom left corners of each B-scan, the resolution also differs across slices.

In the same figure, it is also possible to see different image characteristics. For example, in the left image, the choroid is very well-defined, while in the middle image it looks more blurry. The

noise levels also differ depending on the image. The background in the left image exhibits minimal noise, and the retinal layers are easily identifiable. In the middle image, some noise appears in the background and the retinal layers become harder to differentiate. Lastly, the image on the right contains noticeable background noise, while the retinal layers appear brighter than in the other images.

3.2 Methodology for Fluid Segmentation

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

For its influence and significance for fluid segmentation papers, the U-Net [36] was selected to perform 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 3.3). In the encoder path, two 3x3 unpadded convolutions are applied to the input image, each followed by a rectified linear unit (ReLU) and a 2x2 max pooling operation with a stride of 2, which downsamples the image. At each downsampling step, the number of channels is doubled. In the expanding path, a 2x2 up-convolution is applied, halving the number of channels. The result is then concatenated with the cropped feature map from the corresponding contracting path block. A 1x1 convolution is applied in the final layer.

The number of output channels in the U-Net can be adapted to the number of classes the network is segmenting. For example, if the model aims to segment the three types of retinal fluid, the number of output channels is set to 4 (background, IRF, SRF, and PED). However, in case the model is trained only for the segmentation of a single fluid, the number of output channels is then 2 (background and selected fluid).

The evaluation of all segmentation 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 formula for the Dice coefficient is shown 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 [64].

Considering the multi-class segmentation scenario, the Dice coefficient can be computed for each class independently as shown in Equation 3.2, by treating the corresponding class labels as binary masks, where a_i and b_i represent the predicted and GT binary values for a specific class i. The network that achieved the highest Dice score was selected for use in the fluid volume estimation experiments.

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

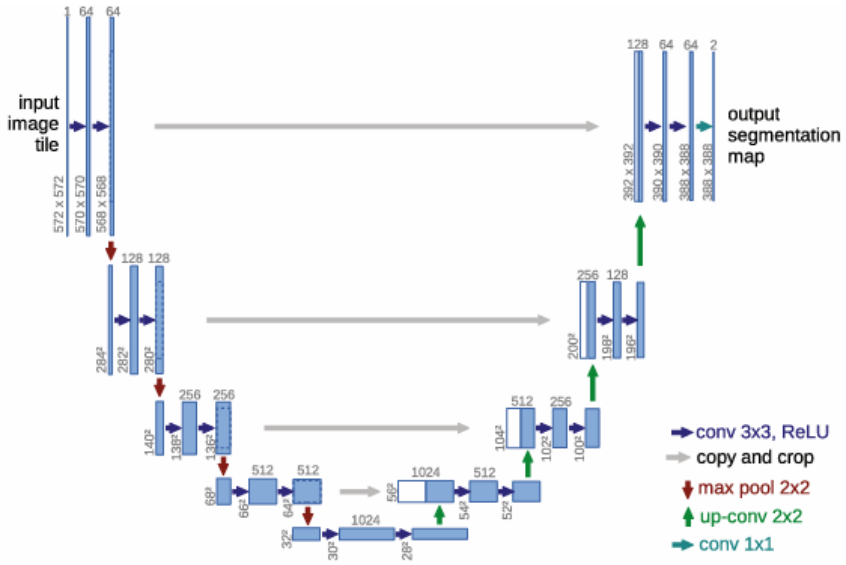


Figure 3.3: U-Net architecture [36].

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

The loss function that regularized the training in the fluid segmentation experiments was the same as the one used by Tennakoon et al. [65]. This loss is described 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 set to 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 for the foreground. This reflects how well 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 is described in 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 when it belongs to any class other than 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 alone is not enough to correctly label the pixels into their respective classes; for this reason, the cross-entropy component was used. Due to the large class imbalance in the images, with the background occupying the majority of each image, the cross-entropy loss is balanced by taking into account the number of pixels belonging to each class. The cross-entropy is calculated for each pixel i in the image. Then, for each class, the cross-entropy of all pixels belonging to that class is summed and divided by the number of pixels in 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 represents 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.

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

Since the U-Net is a fully convolutional network, it is not dependent on the shape of the input to perform segmentation and can be trained on patches of multiple sizes. The model was initially trained on patches of size 256 × 128 pixels, following the same implementation as that in Tennakoon et al. [65]. The extraction of patches aimed to prioritize 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 defined threshold or where retinal fluid is present. Of the patches with no fluid, 75% were dropped. In Figure 3.4, 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.

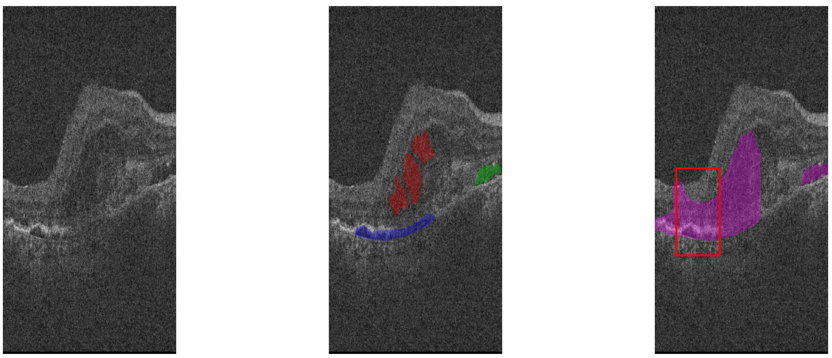


Figure 3.4: Cirrus B-scan (left), fluid masks overlay (middle) with IRF in red, SRF in green, and PED in blue, and the ROI mask overlaid in purple (right). The red bounding box signals a possible 256 × 128 patch that could be extracted.

The second patch size experimented was significantly larger. They were no longer randomly extracted from the ROI. Instead, the patches were extracted from top to bottom so that every section of the image was present in at least one patch. A representation of this process can be seen in Figure 3.5. The patches were extracted this way with the intent of including the entire retina and fluid regions within a single patch, in order to preserve spatial coherence and avoid degrading input quality.

In the third patch extraction method tested, all the B-scans images were resized to the same size. As seen in Figure 3.1, the images present different appearances across the vendors, since the real dimensions of each voxel are different, depending on the device used to obtain the volume. These differences across vendors makes the learning of the segmentation harder. Therefore, by resizing all the images to the same shape, the structures would present more consistent dimensions

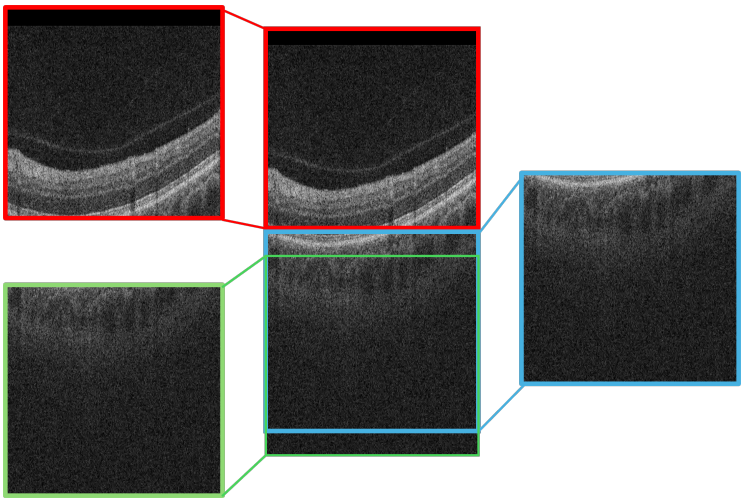


Figure 3.5: Cirrus B-scan with its three corresponding patches of size 496 × 512.

across vendors and the voxels would roughly translate to the same dimensions, leading to an easier learning process for the model.

After image resizing, vertical patches of dimension 496 × 128 pixels, were extracted from each resized B-scan. The number of patches extracted from each image was changed, experimenting with four (Figure 3.6), seven (Figure 3.7), and thirteen (Figure 3.8) patches. To extract seven and thirteen patches, they were sampled at regular intervals along the horizontal axis, starting from the left of the image and ending at the last position where a full-width patch could be extracted.

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 the first patch shape) or the retinal layers are cropped during patch extraction (in the second patch shape, as seen in Figure 3.5).

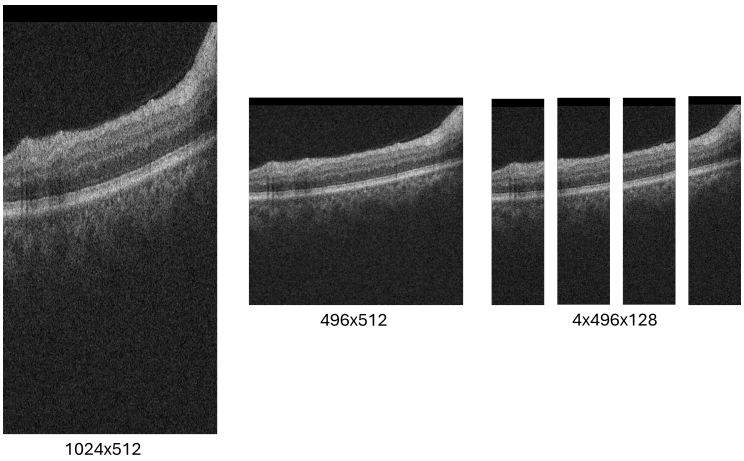


Figure 3.6: Four vertical patches of dimension 496 × 128 extracted from a Cirrus B-scan.

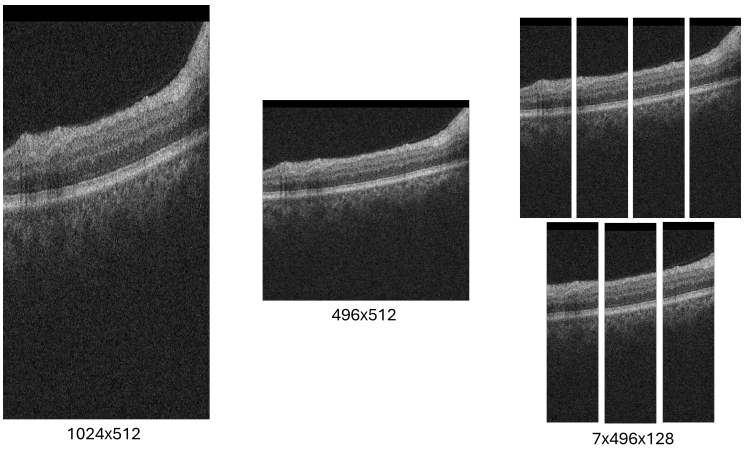


Figure 3.7: Seven vertical patches of dimension 496 × 128 extracted from a Cirrus B-scan.

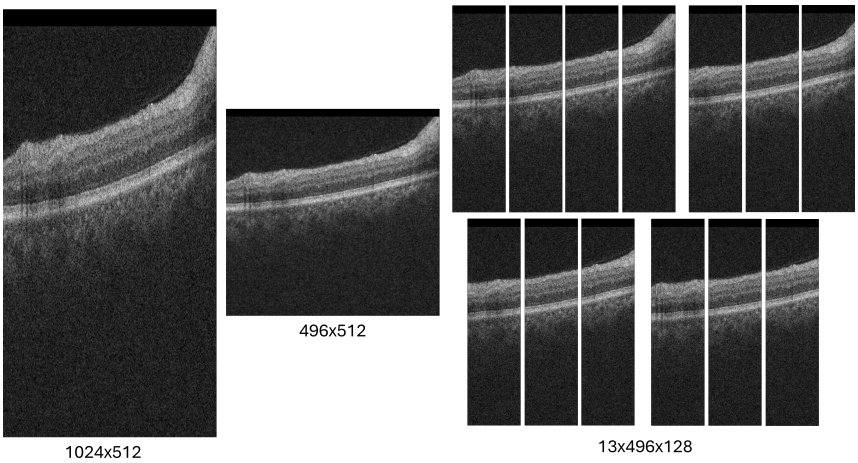


Figure 3.8: Thirteen vertical patches of size 496 × 128 extracted from a Cirrus B-scan.

In the first segmentation experiment, a single U-Net was utilized to perform multi-class segmentation, outputting a four channel mask. However, in the second experiment, three binary U-Nets were used in the segmentation of all fluids, with one network being trained for the segmentation of each fluid.

One of the main problems with multi-class segmentation performed by separate binary models is the merging of the multiple masks. In this context, multiple fluid classes can be predicted for the same pixel, although 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;
- highest probability, where the assigned class is the one predicted with the highest probability by its model.

Two different losses were used for training the models. Initially, the same loss as the one used in multi-class experiment was applied, using only two classes (background and the fluid that would be segmented). Later, a weighted cross-entropy loss was used, with weights balancing the larger proportion of background voxels. This second loss is described in Equation 3.5, with $N = 2$.

After selecting the best segmentation model, the inference in CHUSJ required the support of a retinal layer segmentation model, developed by Melo et al. [66], to ensure anatomically consistent predictions. This necessity comes from the differences in image quality between the CHUSJ and the RETOUCH dataset, on which the model was trained. For instance, in the RETOUCH, the Spectralis B-scans present the choroid region blurred, while in the CHUSJ, this region appears with much greater quality and definition. This discrepancy confuses the segmentation model, which often misinterprets the well-defined choroid region as IRF, leading to incorrect segmentations.

The retinal layer segmentation model was used exclusively to post-process the masks predicted by the segmentation model, removing any fluid that appeared outside the retina, as a way to mitigate the oversegmentations that occurred in the choroid region.

3.3 Methodology for Intermediate Slice Synthesis

The objective of the experiments in intermediate slice synthesis was to improve the inter-slice resolution of OCT scans in order to obtain more accurate fluid volume estimates.

The intermediate slices were produced using a generative model also trained on the RETOUCH dataset. In this experiment, subvolumes consisting of overlapping triplets of consecutive slices, sampled with a step size of 1, were used, as shown in Figure 3.9. The first and last slices of these triplets were used for the generation of the middle slice. Therefore, the quality of the generated slice can be evaluated by comparing it with to the original one, as done in similar studies in 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 that volume.

The generation of slices can be evaluated using specific metrics, as well as through qualitative assessment. To assess the performance of the generative model, the fluid segmentation model could also be used for the estimation of the fluid area in the generated image, and the resulting mask could then be compared to the original image's fluid mask. This comparison could be conducted using the Dice coefficient [14] (see Equation 3.2). However, this metric is insufficient for fully evaluating the generative model performance, as it only compares the segmented fluid region and does not assess the entire slice. Examples of other metrics include the mean absolute error (MAE) [14, 42, 52], the peak signal-to-noise ratio (PSNR) [39, 41, 44, 46, 47, 49, 50, 51, 52], and the structural similarity index measure (SSIM) [39, 44, 46, 47, 49, 50, 51, 52].

The MAE and mean squared error (MSE) quantify the differences 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 computed, and then the mean over all pixels in the image is obtained. In contrast, in MSE, the difference is squared before computing the mean. Equation 3.6 and Equation 3.7 describe MAE and MSE,

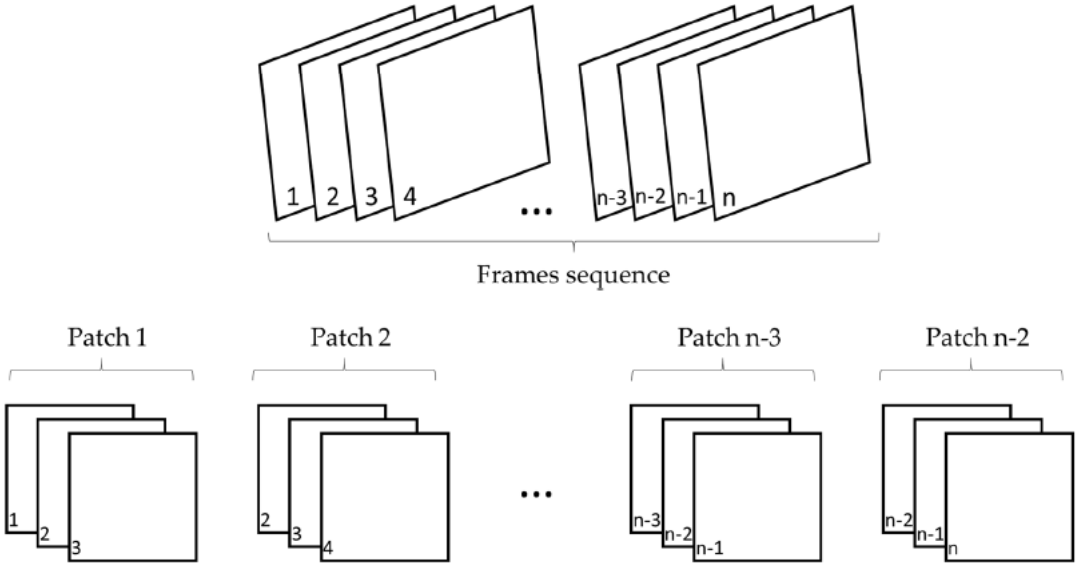


Figure 3.9: Scheme explaining the input data of the generative models. Each frame refers to a B-scan from an OCT volume. Extracted from Tran and Yang [58].

respectively, where x_i is the intensity of the pixel of index i in the predicted image, 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 [67, 68].

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

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

PSNR is a metric used for computing 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, where L is the pixel's maximum intensity in the image. It can also be interpreted as a representation of absolute error in decibels [67]. It is important to note that in OCT imaging the signal-to-noise ratio is relatively low due to the presence of speckle in the images. Therefore, lower PSNR values are expected, compared to other imaging techniques that do not exhibit such noise [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, SSIM is based on the perceptual quality of the image. This metric considers changes in perceived structural information, estimating the visual quality of images and videos. SSIM measures the similarity between the original image and the generated one, and it is calculated as shown in Equation

3.9. Here, x and y represent the generated and original images, respectively. Here, μ_x and μ_y are their local means, σ_x and σ_y are their standard deviations, and 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 $CS(x,y)$ [67].

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

Since the best-performing models in segmentation resized the images to 496×512, the images were generated to match those dimensions. Therefore, regardless of the device used for acquiring the OCT volume, all B-scans were resized to 496×512 for both image generation.

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. [69], is based on a competitive game between two networks. In a scenario where a GAN is employed for inter-slice SR, the generator network takes the first and last slices of a subvolume, 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, improve its output, resulting in slices more similar to the real ones. However, the discriminator network loss also penalizes misclassifications, depending on the probability of the prediction. As a result, as the generator improves, so does the discriminator [70]. The overall framework for GANs is illustrated in Figure 3.10.

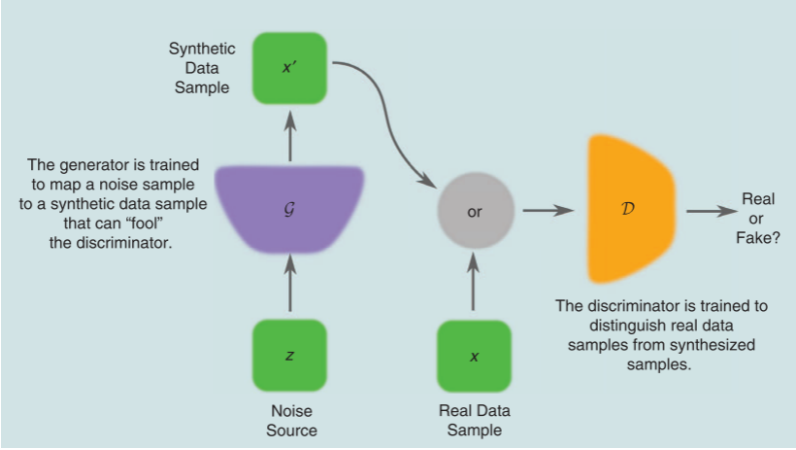


Figure 3.10: Example of a GAN framework, where $\mathscr{D}$ is the discriminator and $\mathscr{G}$ is the generator [71].

The GAN originally developed by Tran and Yang [58] for interpolating intermediate frames in video was adapted in this work to generate the intermediate slices in OCT volumes. The network training was performed on image patches of 64×64 pixels.

The generator of the GAN is composed of a contracting and an 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 $\times$ 8 $\times$ 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 $\times$ 64, in the range of -1 to 1. An illustrative scheme of the generator is shown in Figure 3.11.

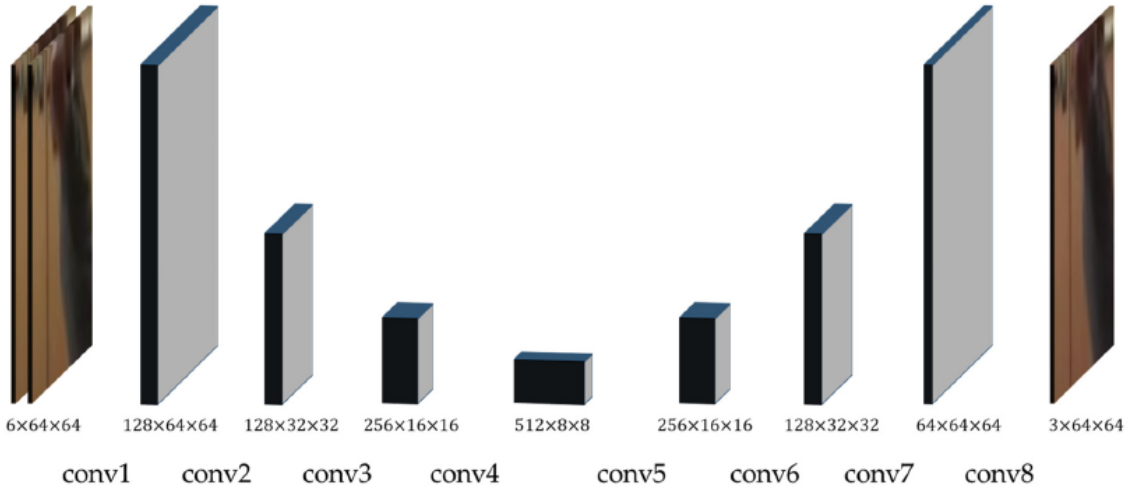


Figure 3.11: Architecture of the generator used in the GAN. It has a contracting and an expanding path, making it a U-Net like network [58].

In the original implementation, one patch is randomly extracted from the triplet of images and used in training. Due to the much smaller amount of data available in OCT, all possible disjoint 64 $\times$ 64 patches were extracted from each B-scan. The extraction was performed from top to bottom, and in the last row of slices, the image was padded with zeros until it reached 64 pixels. An example of the patches extracted from a Cirrus B-scan is shown in Figure 3.12. By methodically extracting the patches, triplets could be easily formed by accessing patches at the same index in the three consecutive images.

To match the needs of our application, some changes had to be made in the generator regarding the input size. As shown in Figure 3.11, the input had six channels - one red, one green, and one blue (RGB) for each input patch. Similarly, the output had three channels, for the generated middle patch. In our application, each input patch has one channel, since the OCT B-scans are grayscale images, rather than RGB, as in the original implementation. Therefore, the output has one channel. The final activation function, the hyperbolic tangent, outputs values in the range of -1 and 1. After applying the hyperbolic tangent, the results had to be converted to the 0 to 1 range, making them comparable to the GT images.

The GAN's discriminator receives a 64 $\times$ 64 patch as input and outputs a probability of the input patch being real. The discriminator is composed of five consecutive convolutional layers. The first convolution is followed by a leaky ReLU activation function. The second, third, and

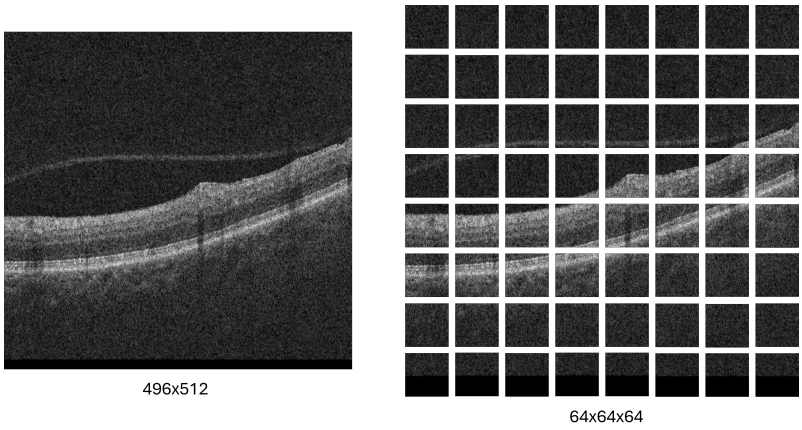


Figure 3.12: Patches of size 64 × 64 extracted from a Cirrus B-scan previously resized to 496 × 512.

fourth convolutions are followed by 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 3.2, the layers that compose the discriminator and the generator are explained, including the dimensions of the outputs.

In the training of a GAN, both the generator and the discriminator are trained sequentially and independently. First, the generator, which receives the first and last slice 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 for updating the generator weights. The generator loss is composed of four components: the adversarial loss, the MAE, the multi-scale SSIM (MS-SSIM) loss, 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 evaluates of how effectively 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 passed to the discriminator, which then outputs the probability that it is real. 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 indicates that the image is real. It is important to note that during 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.

Table 3.2: 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 a size of $1 \times 64 \times 64$. Adapted from Tran and Yang [58].

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

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

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. Although this loss gives insight into how similar the images are, on average, it can be deceiving, since the model can blur the output to attain better MAE values. For this reason, it 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 images, facilitating a smoother output. This loss, originally suggested by Wang et al. [72], 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 introduced in Equation 3.9. This version of the MS-SSIM is faster than the original implementation [72], 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 the first four levels and the SSIM of the image at 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 generator loss (Equation 3.10) is the GDL, originally proposed by Mathieu et al. [73]. This component is used to reduce for reducing 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 pixels coordinates, 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, sequentially, 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 this 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 of the BCE values for the fake and real images is the discriminator loss.

In the second intermediate slice synthesis experiment, the selected network was the U-Net, inspired by the work of Nishimoto et al. [43]. While the U-Net is more commonly applied in segmentation, as seen in the reviewed literature, Nishimoto et al. [43] applied it to generate the intermediate slices of a subvolume. The U-Net receives the edge slices as input and produces the intermediate slices as output. In their paper [43], this was tested for three, four, and five slices. However, in this experiment it was utilized to generate a single intermediate slice. The loss used to regularize the network training was the MAE, previously described in Equation 3.6.

3.4 Methodology for Fluid Volume Estimation

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

The OCT scans used in the fluid volume estimation experiments were those from the RETOUCH dataset that composed the reserved fold in the segmentation. These volumes were used because

no model was trained or validated on them, allowing an insight into 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 used for obtaining 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 attributed to all individual B-scans, which results from the fluid area in each slice. 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 characteristics of the device D. 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 attributed to each B-scan that composes the OCT, which is obtained from the fluid area in each slice.

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

$$
F_{c,D} = \sum_{s=0}^{S} f_{c,s,D} \tag{3.16}
$$

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 satisfactory 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 caused the observed difference between experiments.

Using the GAN, the corresponding fake OCT volumes of the reserved fold were created. In these volumes, the segmentation model inferred the fluid masks and they were compared to the GT masks of the original volumes, using the Dice coefficient.

Also utilizing the GAN, the OCT volumes were super-resolved, as a slice was generated between each pair of two known slices. To calculate the fluid volume in the super-resolved volumes, the OCT's inter-slice distance was reduced in half.

Chapter 4

Experiments

Building upon the methods and materials exposed, the experiments conducted during this dissertation are explained in depth in this chapter. It begins with a description of how the data were split, followed by the experiments on fluid segmentation, intermediate slice generation, and fluid volume estimation.

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

4.1 Data Partition

To promote consistency across all experiments, the conditions were held identical. In every experiment, the data were split into five folds, with different splits being used for the segmentation and generation tasks. From these five folds, one was reserved until the end to compare the performance of the models trained in the different experiments and identify the one that generalizes best to an unseen image set. The remaining four folds were used for training the models through a 4-fold cross-validation procedure, in which three folds were used for training and one for validation. 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. Fold 1 was randomly selected as the reserved fold.

For the multi-class segmentation models, the data were split so that the quantity of each fluid and the number of volumes per vendor were evenly distributed across the folds - referred to as a multi-class 5-fold split. This balanced distribution allows for a fairer assessment of the model's learning ability and its performance on data with varying characteristics (e.g., from different vendors). To achieve this, a custom algorithm was developed to divide the data into five folds while minimizing discrepancies in fluid distribution and the number of slices per vendor within each fold.

A possible distribution of the 70 OCT volumes from the RETOUCH dataset, which were used in the training of the fluid segmentation models, can be seen in Table 4.1. The split was applied to

the volumes and not to the slices. Slices from the same volume must be kept together to prevent data leakage, where similar images obtained from the same patient appear in both the training and validation sets, potentially resulting in overly optimistic performance metrics.

Table 4.1: Number of OCT volumes per vendor in each fold, considering a 5-fold split.

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.

For experiments using a binary segmentation approach, the volumes can be redistributed using the same algorithm, but with fewer constrains - a fluid-specific 5-fold split. In this case, the folds are created based only on the vendor and amount of the target fluid, thus eliminating the restrictions imposed by the quantities of the other two fluids. Nevertheless, the volumes that are in the previously defined reserved fold must not be used in either training in validation.

For the experiments related with inter-slice generation, the 5-fold split of the data was not done by considering the quantity of fluid in each fold. Since the test volumes of the RETOUCH dataset were used in this experiment and fluid masks are not available, the quantity of fluid in each test volume is unknown. Please note that, for comparison purposes, one of the folds in this 5-fold split is the one reserved in the multi-class segmentation split for testing.

The split was performed by taking into consideration solely the number of slices per device. In these experiments, the characteristics of each device are important, since each device has a specific inter-slice distance. This distance varies even across devices from the same vendor and is an important factor 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 for testing 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 are distributed as shown in Table 4.2.

Since the partition is not constrained by the quantity of fluid in each volume, it is possible to compute the optimal 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 were used in training while one was used in validation.

Table 4.2: Device-wise distribution of OCT volumes across the four folds used for training and validation in OCT slice synthesis.

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

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

4.2 Experiment 1 - Multi-class Fluid Segmentation using a single U-Net

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

This was the most extensive set of experiments, where many variables were tested. Different patch shapes, transformations, and hyperparameters were explored, until the best training settings were determined. The best settings were then used in Experiment 2 (section 4.3). All the experiments were conducted using the Adam optimizer [74] with a learning rate of 2×10^{-5}.

In each epoch, the model was trained on three folds and validated on one. The performance of the model in the validation fold allowed an insight into how the model was learning. Therefore, 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.

4.2.1 Experiment 1.1 - Random Patch Extraction

In Experiment 1.1, the patches, of size 256×128, were extracted randomly, following the same procedure as implemented by Tennakoon et al. [65], explained in the previous chapter. For this reason, two sets of four training runs were performed, considering the 4-fold split mentioned in section 4.1. In both sets, all conditions were kept identical, with the only differences being caused by the random extraction of training patches. This setup was designed to evaluate how the random nature of the patch extraction affected the model's performance.

The model was trained for 100 epochs with a batch size of 32, with no early stopping. The input patches were transformed by a rotation between 0 and 10 degrees, and horizontal flipping.

4.2.2 Experiment 1.2 - Large Patch Extraction

In this experiment, the patch size was changed from the 256×128 pixels used in Experiment 1.1 to 496×512 pixels and was no longer extracted randomly. By using these dimensions, the model

receives a larger context of the B-scan as input, allowing it to learn the anatomical references that characterize and limit the fluids.

Since larger images were loaded to the model, the batch size had to be changed from the usual 32 to 16, due to memory constrains. Nevertheless, the model was trained for 100 epochs while using the same transformations as in Experiment 1.1.

4.2.3 Experiment 1.3 - Vertical Patch Extraction

In Experiment 1.3, all the images were initially resized from their original dimensions to 496 $\times$ 512 pixels, which was the shape of the smaller images in the dataset. Then, vertical patches were extracted from each B-scan. The number of patches extracted was changed, varying between four, seven, and thirteen. Regarding the transformations applied, all the models were trained with horizontal flipping and multiple rotations were tested.

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 the sets of seven and thirteen patches on the best- and worse-performing folds from the four-patch setup, with early stopping applied if the validation loss did not improve within 25 epochs after reaching its minimum. This stopping criteria was only applied after the model trained for 25 epochs. This approach was adopted because the model trained much faster when using seven and thirteen patches per B-scan, due to the increased number of images processed per epoch. Henceforth, it also required more computational power, which further motivated the use of early stopping.

Using seven vertical patches per image, which was the number of patches that performed best, three different rotation settings were tested: 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 models where rotation was applied were 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, training would be interrupted. The models trained without rotation converged much faster and, for that reason, the minimum number of epochs used was 25, while still keeping the same patience. All models were trained for up to 200 epochs.

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 using this rotation was selected to infer on the reserved test fold and in the CHUSJ dataset.

4.3 Experiment 2 - Multi-class Fluid Segmentation using Three Separate U-Net models

The second experiment also involved multi-class segmentation of the retinal fluids. In contrast with the first experiment, where the segmentation was done using a single U-Net, three binary U-Nets were used in this experiment, with one model for IRF, one for SRF, and one for PED.

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 - similar to what was done in the last runs of Experiment 1 (section 4.2). Similarly, the random transformations applied to the images consisted of horizontal flipping and a maximum rotation of 5°. Two losses were used in the training of these models: the loss used in the multi-class experiments and the balanced cross-entropy loss.

4.3.1 Experiment 2.1 - Multi-class Segmentation Loss

When using the loss from Experiment 1 (section 4.2), 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. All the remaining conditions were equal.

4.3.2 Experiment 2.2 - Balanced Cross-entropy Loss

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

4.4 Experiment 3 - Intermediate Slices Synthesis Using a GAN

In the first intermediate slice synthesis experiment, the GAN developed by Tran and Yang [58] was used in the generation of intermediate slices. The GAN was trained for 250 epochs, using a batch size of 32. The generator's learning rate was 2×10^{-5}, while the discriminator's was 2×10^{-6}. The selected optimizer for both networks was Adam, with $\beta_1 = 0.5$ and $\beta_2 = 0.999$.

In every run, the generator model was saved every 10 epochs. Of the saved models, the one that obtained the minimal adversarial loss was selected for evaluation. Since this loss is perceptual, the model that performs the best usually generates images that look more real.

4.5 Experiment 4 - Intermediate Slices Synthesis Using a U-Net

In the second experiment, a U-Net was selected to generate the intermediate slice between a pair of two known slices.

While in the previous experiment the model was trained with patches of size 64×64 pixels, this model was trained with the full images, which were resized to 496×512 pixels. Since this requires a larger use of memory, the batch size was changed from 32 to 8.

The model was trained for up to 200 epochs with a minimum of 100, after which a patience of 25 epochs was applied. The optimizer used was Adam with a learning rate of 2×10^{-4}.

4.6 Experiment 5 - Fluid Volume Estimation Using Predicted Masks in Real OCT Volumes

In this experiment, the fluid volumes were calculated for the OCT volumes without generated slices. Two volumes were calculated for each OCT volume: one using the GT masks in the RETOUCH dataset and one using the masks predicted by the best segmentation model, using the equations described in the previous chapter. The results from this experiment allow comparison with values obtained in the subsequent experiment, which used slice generation.

4.7 Experiment 6 - Fluid Volume Estimation Using Predicted Masks in Generated and Super-resolved OCT Volumes

This experiment involved estimating fluid volume in OCT scans containing generated images and the model used for segmentation was the same as in the previous experiment. This model inferred the segmentation masks on a fake OCT volume, where the all the real OCT B-scans, except the first and the last from each volume, were substituted with the B-scans generated using the GAN. The segmentation model was also used in the prediction of masks in super-resolved OCT volumes. These volumes had a generated slice between every pair of known slices.

From the predicted fluid masks, the fluid volumes were estimated using the equation explained in the previous chapter, and compared between each other and with the values obtained in the previous experiment.

Table 4.3 provides a summary of the experiments performed in this dissertation is shown, allowing for easier reference and comparison.

Table 4.3: Summary of the experiments performed in this dissertation.

Experiment	**Description**
Experiment 1 - Fluid Segmentation with Multi-class U-Net	**1.1** – Trained with patches of size 256 × 128 pixels randomly extracted from the ROI. **1.2** – Trained with large patches of size 496 × 512 pixels. **1.3** – Trained with a varying number of vertical patches (four, seven, or thirteen) of size 496 × 128 pixels, extracted from B-scans resized to 496 × 512 pixels.
Experiment 2 - Fluid Segmentation with Binary U-Net	**2.1** – Trained with the same loss as in Experiment 1, using two different data partitions. **2.2** – Trained with the balanced cross-entropy loss.
Experiment 3 - Synthesis of Intermediate Slices using a GAN	**3** – Trained a GAN to generate a fake B-scan between a pair of two known B-scans.
Experiment 4 - Synthesis of Intermediate Slices using a U-Net	**4** – Trained a U-Net to generate a fake B-scan between a pair of two known B-scans.
Experiment 5 - Fluid Volume Estimation in Real OCT Volumes	**5** – Estimated fluid volumes using the GT and the fluid masks predicted by the segmentation model.
Experiment 6 - Fluid Volume Estimation in Super-resolved OCT Volumes	**6** – Estimated fluid volumes for generated and super-resolved OCT volumes, synthesized by the GAN, using the masks predicted by the segmentation model.

Chapter 5

Results and Discussion

This chapter presents the results from the experiments described in Chapter 4. These outcomes are organized into three sections: “Fluid Segmentation”, “Intermediate Slice Synthesis”, and “Fluid Volume Estimation”. After showing the results from the experiments, the factors influencing them are discussed, and visual examples of the models’ performance are shown. The results are also compared with those from literature, with an explanation of how different methods lead to different results.

5.1 Fluid Segmentation

In this section, the results from the experiments performed on multi-class fluid segmentation are shown. This includes all the runs described in Section 4.2 (Experiment 1) and 4.3 (Experiment 2). In these sections, the resulting Dice coefficients are displayed in tables.

Jointly with the quantitative results, images illustrating the segmentation results of the models and the corresponding GT are also presented.

5.1.1 Experiment 1 - Multi-class Fluid Segmentation using a single U-Net

5.1.1.1 Experiment 1.1 - Random Patch Extraction

The results from the Experiment 1.1 are shown in Table 5.1. In this experiment, two runs were performed for each validation fold (VF), using the same conditions. The only difference was the patches used for training: as they were extracted randomly (as described in Section 3), they varied between runs.

Each value corresponds to the mean Dice coefficient computed across all slices in the validation OCT volumes. The results are shown for each fluid (IRF, SRF, and PED) both grouped by vendor and aggregated across all vendors. There is also a column that contains the Dice coefficient when considering all the fluids as a single binary label. Unless otherwise specified, the Dice coefficient calculated for a fluid considers all the slices and not just those that contain the specified fluid.

These values are presented for each run, following the conditions described in the Fluid Segmentation section. Every four runs completed using a different validation fold but following the same conditions are grouped into a single row, by calculating their mean (row "Set"). From the 5-fold split, fold 1 was reserved, while the remaining four (0, 2, 3, and 4) were used in training and validation. The fold selected for validation in each run is listed in the table's VF column. For example, if the fold in the VF column is 2, then the folds used for training were 0, 3, and 4.

Table 5.1: Dice coefficients for every vendor and fluid for the runs done in Experiment 1.1. The conditions were the same for both sets except the extracted patches, which are different in every run due to the random patch extraction method used.

Runs	**VF**	**Cirrus**			**Spectralis**			**Topcon**			**IRF**	**SRF**	**PED**	**Fluid**
		IRF	**SRF**	**PED**	**IRF**	**SRF**	**PED**	**IRF**	**SRF**	**PED**				
Run 1	2	0.138	0.089	0.072	0.255	0.331	0.163	0.259	0.446	0.056	0.200	0.254	0.083	0.163
Run 2	3	0.138	0.290	0.236	0.264	0.670	0.652	0.389	0.532	0.284	0.252	0.445	0.327	0.303
Run 3	4	0.209	0.158	0.024	0.255	0.451	0.310	0.286	0.595	0.151	0.249	0.386	0.117	0.173
Run 4	0	0.166	0.179	0.041	0.307	0.343	0.243	0.371	0.376	0.052	0.266	0.280	0.081	0.165
Set 1	-	0.163	0.179	0.093	0.270	0.449	0.342	0.326	0.487	0.136	0.242	0.341	0.152	0.201
Run 5	2	0.106	0.152	0.085	0.290	0.406	0.256	0.433	0.443	0.065	0.250	0.296	0.110	0.175
Run 6	3	0.370	**0.297**	0.454	**0.410**	**0.735**	**0.673**	0.205	**0.821**	**0.672**	0.317	**0.566**	0.572	0.299
Run 7	4	**0.590**	0.265	**0.666**	0.296	0.372	0.498	**0.687**	0.683	0.555	**0.593**	0.460	**0.596**	**0.420**
Run 8	0	0.321	0.259	0.067	0.352	0.514	0.467	0.410	0.610	0.152	0.249	0.386	0.117	0.173
Set 2	-	**0.347**	**0.243**	**0.318**	**0.337**	**0.507**	**0.474**	**0.434**	**0.639**	**0.361**	**0.380**	**0.439**	**0.362**	**0.274**

By looking at the table, a few conclusions can be drawn. First, the difference in performance between runs using the same training OCT volumes is evident. Despite some values being similar for the same training volumes, this variability becomes clear when looking at the mean values in the "Set" rows, where a significant difference is noted, especially for the Cirrus vendor and the PED fluid. It is also apparent that some VFs are more consistent than others. For example, with different randomly extracted patches, the models evaluated on validation folds 0 and 2 presented similar results, whereas those evaluated on validation fold 3 showed significant variation when the training patches were different.

The second conclusion is that, overall, the models did not perform well, as indicated by the low Dice coefficients across all VFs. These values were specially low for Cirrus, but were comparatively better for Spectralis and Topcon. One of the reasons the model performed so poorly is related to the training data. Most of the extracted patches failed to capture the transitions between the background and the retina, and between the retina and the choroid, due to their small patch size. These transitions are of great importance in OCT fluid segmentation, as they often define the anatomical boundaries of the region where fluid can appear. In case the model does not understand these anatomic features, it may segment fluid outside the retina, which leads to a lower the Dice coefficient.

Although performance was suboptimal across all vendors, it was notably worse for Cirrus. In the literature, it is common to observe poorer performance with images from Cirrus and Topcon, due to the higher levels of speckle noise. However, the poor results observed for Cirrus were largely attributed to the patch size. The B-scans in these volumes present a higher vertical resolution compared to those from Topcon and Spectralis. Therefore, when the same patch size is used

across devices, each patch from a Cirrus scan covers a smaller retinal area. This leads to a poorer understanding of the transitions between background and retina, and between retina and choroid, as shown in Figure 5.1, ultimately causing oversegmentation beyond the retinal region.



Figure 5.1: Example of a patch extracted from a Cirrus OCT volume used in Experiment 1.1. In this patch, the background is distinguishable due to its darker intensity; however, the choroid is harder to be identified by an observer (or a model) due to the lack of anatomical context.

In Figure 5.2, the segmentation result of the model trained in Run 1 is shown. In this figure, it becomes evident that the model learned which areas can be segmented inside the retina, but does not understand how to process the regions further away. For example, in the area significantly above the retina, the background is labeled as SRF. However, the background region closer to the retina is not frequently labeled as fluid, since it rarely appears in the training patches. The same problem occurs with the oversegmentation of PED in the choroid region, which does not appear in the training patches with sufficient frequency. The fact that the patches are extracted exclusively from the ROI makes the model more prone to misclassify the background regions, as it did not see any patch that contained solely background.

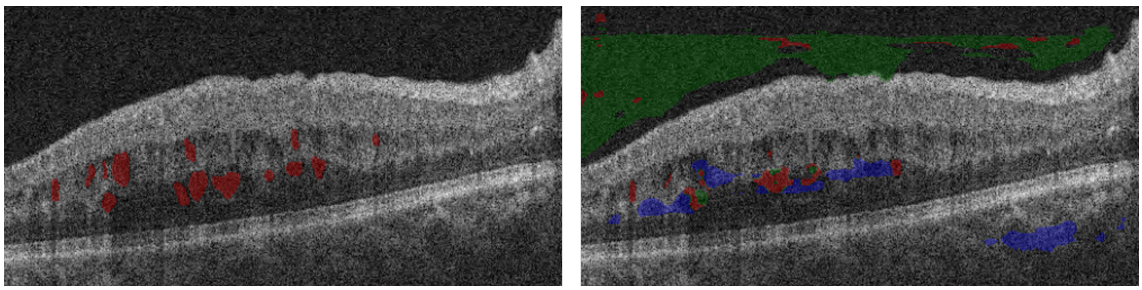


Figure 5.2: Example of a poor segmentation made by the model trained in Run 1 (right). In the left, the GT mask for the same image is shown. IRF is represented in red, SRF in green, and PED in blue.

Oversegmentation beyond the retinal bounds is not exclusive to the Cirrus volumes, as it also appears in the OCT scans from other vendors. This suggests that the issue is primarily due to the small patch size and lack of background representativeness in the patches rather than just the smaller retinal area captured in Cirrus patches, despite this amplifying the problem.

In the same figure, it is evident that the model has not learned the anatomical relationships between fluids, since it segmented IRF and PED close to each other. This occurs because the model has not learned the relationship between the retinal landmarks and the fluid types due to the small dimensions of the patches.

5.1.1.2 Experiment 1.2 - Large Patch Extraction

The Dice coefficient values obtained in Experiment 1.2, where patches of size 496 $\times$ 512 were used, are shown in Table 5.2.

Table 5.2: Dice coefficients for every vendor and fluid for the runs performed in the Experiment 1.2.

Runs	**VF**	**Cirrus**			**Spectralis**			**Topcon**			**IRF**	**SRF**	**PED**	**Fluid**
		IRF	**SRF**	**PED**	**IRF**	**SRF**	**PED**	**IRF**	**SRF**	**PED**				
Run 9	2	0.291	0.450	0.281	0.472	0.638	0.394	0.505	0.647	0.573	0.396	0.551	0.400	0.393
Run 10	3	**0.586**	**0.619**	**0.727**	0.482	**0.780**	**0.698**	**0.749**	**0.793**	**0.788**	**0.627**	**0.711**	**0.744**	**0.667**
Run 11	4	0.281	0.453	0.415	0.429	0.503	0.322	0.228	0.532	0.324	0.278	0.494	0.363	0.296
Run 12	0	0.242	0.334	0.336	**0.551**	0.564	0.423	0.321	0.643	0.407	0.325	0.488	0.377	0.339
Set 3	-	0.350	0.464	0.440	0.484	0.621	0.459	0.451	0.654	0.523	0.407	0.561	0.471	0.424
Set 2	-	0.347	0.243	0.318	0.337	0.507	0.474	0.434	0.639	0.361	0.380	0.439	0.362	0.274

The information summarized in this table allows for a comparison between the performance of models that used smaller patches in Experiment 1.1 and those that used larger patches in this experiment.

By comparing the "Set" rows, an overall increase in performance becomes evident when using larger patches. In fact, almost all the mean values are higher compared to the best-performing set in Experiment 1.1. This increase exceeds one decimal point in some columns, with the most significant improvements observed in volumes obtained using the Cirrus device and in the segmentation of SRF or PED.

The improvement in performance for Cirrus volumes can be attributed to the greater contextual information provided by the larger patches. In Experiment 1.1, the Cirrus patches used for training the model were large big enough for the network to capture and understand the retinal boundaries or the relationships between them and the fluids. Additionally, the model was trained with limited background representation, which led to oversegmentation beyond the retina, as previously explained. In Experiment 1.2, each extracted patch covers both the retinal layer and the background simultaneously, enabling to learn not to segment beyond the retina. In Figure 5.3, the same OCT B-scan shown in Figure 5.2 is segmented by the model trained in Run 9, and it is clear that background regions are no longer labeled as fluid. This also partially justifies the significant improvement in the SRF and PED Dice coefficient. By providing inputs with enough context, the model is no longer segmenting these fluids outside the retina.

Furthermore, Figure 5.3 shows that the model no longer confuses the fluids within the retina, as it no longer segmented the PED in the area typically occupied by IRF, as it did in Figure 5.2.

Lastly, it is important to note that the IRF did not improve as much as the other fluids. This happens because the training patches in Experiment 1.1 were already centered on the retina, providing sufficient anatomical context for IRF segmentation. By training the model with larger patches, its attention to the finer details may be reduced, resulting in less precise segmentations of small fluid regions.

In Figure 5.4, the model confuses the choroid with the retina, labeling parts of it as IRF and PED. The likely cause for this confusion is that the patches used for training are often cropped in the middle of the retinal layer (as illustrated in Figure 3.5), preventing the model from fully understanding the choroid's position relative to the retina. The inaccurate segmentation of IRF and PED in this region is probably due to the visual similarities between the structures observed in the choroid and the fluid regions in the retina. This indicates that the model does not correctly recognize the anatomical location of the choroid.

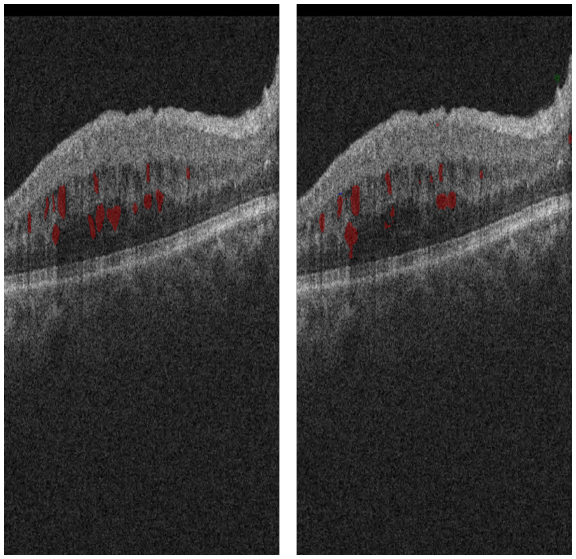


Figure 5.3: Example of the segmentation performed by the model trained in Run 9 (right) and its respective GT mask (left). The OCT B-scan segmented is the same as in Figure 5.2, and the predicted IRF is represented in red.

5.1.1.3 Experiment 1.3 - Vertical Patch Extraction

Experiment 1.3 contains all the runs that were performed using vertical patches extracted from resized B-scans for training the multi-class segmentation U-Net. In the first two sets, shown in Table 5.3, the models were trained using four vertical patches, obtained as explained in Chapter 3. In this table, the "Set 4" is composed of runs that were trained for 100 epochs, while the runs in "Set 5" were trained for up to 200 epochs with a 100-epoch patience.

The results presented in Table 5.3 can be compared with each other and with the results obtained in the sets from previous experiments.

When comparing the results between sets, it is evident that the runs with the best scores are mostly located in the set that was trained for fewer epochs. For example, the model validated on

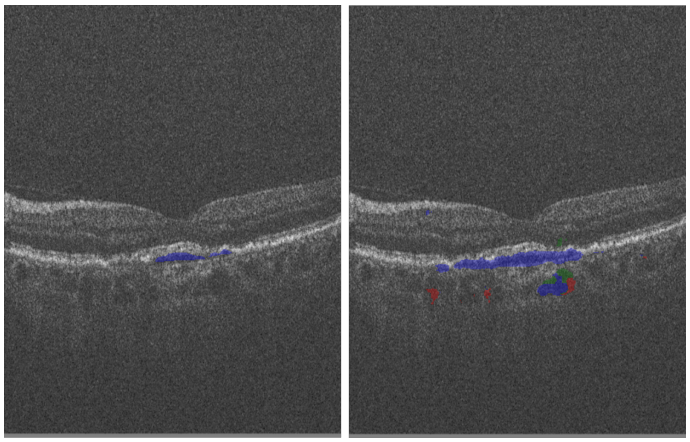


Figure 5.4: Example of the segmentation performed by the model trained in Run 12 (right) and its respective GT mask (left). It is evident that the model confuses the choroid with the retina, as it segments IRF and PED within the choroid. IRF is represented in red, SRF in green, and PED in blue.

Table 5.3: Dice coefficients for every vendor and fluid obtained from runs using four vertical patches extracted from each B-scan. In "Set 4", the models were trained for 100 epochs, while in "Set 5" they were trained for up to 200 epochs with a 100-epoch patience. The transformations applied in these sets were the same: horizontal flipping and a maximum rotation of 10°.

Runs	**VF**	**Cirrus**			**Spectralis**			**Topcon**			**IRF**	**SRF**	**PED**	**Fluid**
		IRF	**SRF**	**PED**	**IRF**	**SRF**	**PED**	**IRF**	**SRF**	**PED**				
Run 13	2	0.411	0.665	0.498	0.654	0.735	0.611	0.731	0.743	0.519	0.563	0.704	0.526	0.581
Run 14	3	0.390	0.520	0.217	0.403	0.564	0.429	0.378	0.754	0.477	0.388	0.614	0.350	0.386
Run 15	4	0.792	**0.846**	**0.883**	0.732	**0.901**	0.733	0.615	0.758	0.656	0.707	0.815	0.765	0.652
Run 16	0	**0.810**	0.834	0.715	0.779	0.857	0.753	0.783	**0.924**	**0.882**	**0.795**	**0.871**	**0.783**	**0.709**
Set 4	-	**0.601**	**0.716**	0.578	0.642	**0.764**	0.632	0.627	**0.795**	0.634	0.613	**0.751**	0.606	0.582
Run 17	2	0.522	0.784	0.703	**0.805**	0.765	**0.865**	**0.828**	0.879	0.835	0.677	0.813	0.777	0.684
Run 18	3	0.509	0.654	0.598	0.590	0.806	0.755	0.777	0.787	0.792	0.621	0.729	0.697	0.622
Run 19	4	0.377	0.387	0.398	0.478	0.624	0.398	0.483	0.729	0.448	0.436	0.567	0.420	0.399
Run 20	0	0.753	0.697	0.696	0.779	0.844	0.783	0.691	0.720	0.671	0.735	0.731	0.702	0.637
Set 5	-	0.540	0.631	**0.599**	**0.663**	0.760	**0.700**	**0.695**	0.779	**0.687**	**0.617**	0.710	**0.649**	**0.586**

fold 4 (Run 15) sees a significant decrease in performance when trained for 100 more epochs (Run 19). When looking at the model's validation loss per epoch, the lowest value is commonly attained before 100 epochs. Since the model saved in each run is the one that achieves the lowest loss on validation data, this means that models trained for 100 or 200 epochs should perform equally, as long as the lowest validation loss occurs prior to the 100-epoch mark.

However, looking at Figure 5.5, which shows the training and validation loss curves for Run 13 and Run 17 are shown, it becomes clear that although both models reach their minimum validation loss in a similar number of epochs and follow comparable learning curves, their final performance is markedly different, as shown in Table 5.3.

To justify this behavior, it is important to remember that training a neural network involves some randomness. One aspect that affects the model performance, despite using the same data and architecture, is the weight initialization. The random initialization of weights affects the tra-

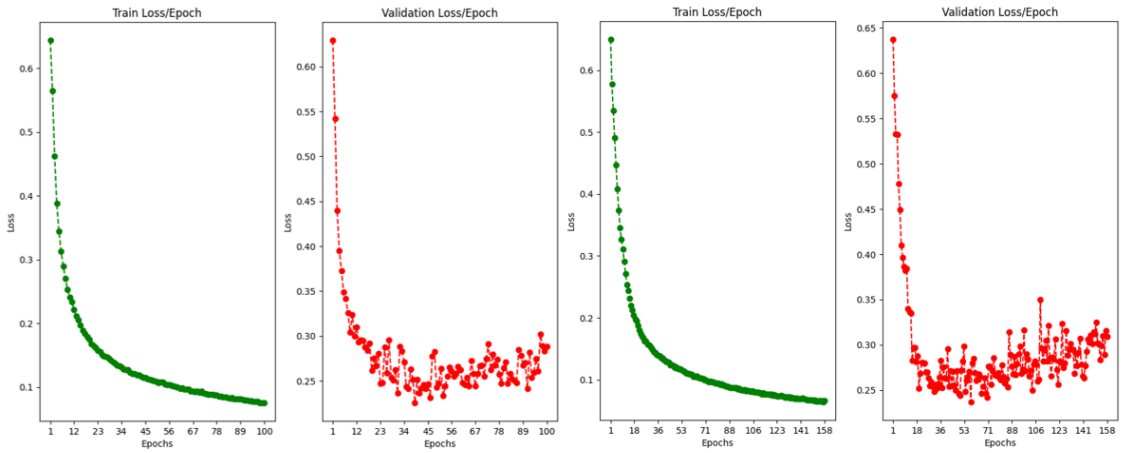


Figure 5.5: Training and validation loss curves for Run 13 (left two plots) and Run 17 (right two plots). Despite reaching the minimum validation loss at a similar number of epochs and presenting comparable training and validation loss curves, the final performance, shown in Table 5.3, differs significantly.

jectory of gradient descent, resulting in different convergences for different runs. Another factor that affects the model learning is the data that composes each batch. The images that form a batch are randomly fetched from the available training images. This creates batches with different distributions of characteristics, affecting the model's perception of the data. Data augmentation, which randomly selects images to apply transformations as they are fetched, also influences performance. Lastly, there are other factors that have a much smaller impact, such as the optimizer's behavior and non-deterministic GPU operations [75, 76].

In Appendix A, Table A.1 shows the results of multiple models trained under the same conditions, aiming to provide insight into how the randomness associated with the U-Net affects the model performance. Five models were validated on fold 2, while nine models were validated on fold 3. More models were trained on fold 3 because the performance gap between the models for 100 epochs and 200 epochs was significantly larger compared to fold 2. The results support the differences observed in Table 5.3, where models trained under identical conditions, but for more epochs, can achieve significantly worse Dice scores during validation.

Despite performance variability observed in Table 5.3, both sets achieved better results than the best outcomes from previous experiments. The improvements were significant, with increases above 0.2 in some metrics when compared to the results from Experiment 1.2. Every vendor and every fluid type showed a considerable improvement in performance.

This overall increment can be attributed to two main changes: training on vertical patches and resizing the images. Training with vertical patches enables the model to focus on finer details within each region. This is important because the retinal fluid segmentation relies heavily on the structural features of the retina, a region that occupies only a small portion of the entire image. Although the patch size in this experiment is smaller than in Experiment 1.2, the patches were still able to capture the transitions between the retina and the surrounding regions such as choroid and background. A smaller patch size also requires less memory, therefore allowing an increase in the

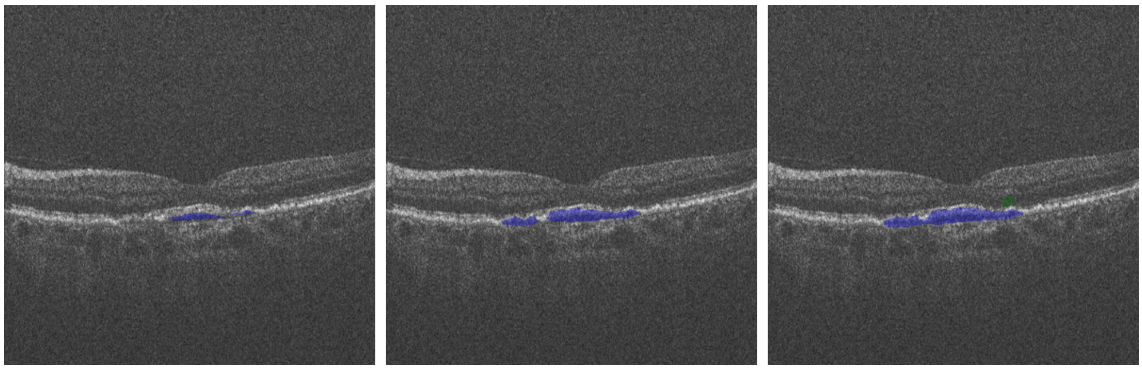


Figure 5.6: Predicted masks by the models trained on Run 16 (middle) and Run 20 (right) and their respective GT (left). SRF is represented in green and PED in blue.

batch size, leading to faster and more stable convergence during training. Meanwhile, resizing the input images makes them more homogeneous across different vendors. This results in the same anatomical structures being represented with consistent shapes and sizes across OCT B-scans from different vendors, simplifying the learning process considerably.

As observed in Figure 5.4, the model previously predicted fluid masks in the choroid region. However, in Figure 5.6, the models trained on vertical patches no longer make such predictions, even when applying to the same B-scan. This suggests that, by using vertical patches, the model learns not to segment below the retina. This behavior was not observed when larger patches were used, since those often failed to simultaneously capture both the retinal layer and the choroid in the same patch, restraining the model from learning their spatial relationship, which led to incorrect fluid predictions in the choroid region.

Afterwards, the use of seven and thirteen patches extracted from each B-scan was tested, and the results are shown in Table 5.4. Two models were trained for each number of patches extracted, one validated on fold 2, and another validated on fold 3. The models were trained for up to 200 epochs, with a 25-epoch patience applied after the first 100 epochs.

Table 5.4: Dice coefficients for every vendor and fluid from runs performed using seven (Runs 21 and 22) and thirteen (Runs 23 and 24) vertical patches extracted from each B-scan. Only two folds were used in order to understand the viability of using this number of patches, which are represented in the column "P". Runs 17 and 18 are shown here to make the comparison between the number of patches easier. The values shown in bold represent the best value for the models validated on the same fold.

Runs	**VF**	**P**	**Cirrus**			**Spectralis**			**Topcon**			**IRF**	**SRF**	**PED**	**Fluid**
			IRF	**SRF**	**PED**	**IRF**	**SRF**	**PED**	**IRF**	**SRF**	**PED**				
Run 17	2	4	0.522	0.784	**0.703**	**0.805**	0.765	**0.865**	0.828	0.879	0.835	0.677	0.813	**0.777**	0.684
Run 21	2	7	**0.556**	**0.837**	0.672	0.761	**0.853**	0.848	**0.829**	0.908	**0.858**	**0.685**	**0.864**	0.767	**0.681**
Run 23	2	13	0.500	0.762	0.635	0.672	0.790	0.773	0.819	**0.936**	0.805	0.639	0.826	0.718	0.637
Run 18	3	4	0.509	0.654	0.598	0.590	0.806	**0.755**	**0.777**	**0.787**	**0.792**	0.621	0.729	0.697	0.622
Run 22	3	7	**0.734**	**0.855**	0.836	**0.636**	**0.846**	0.689	0.686	0.781	0.731	**0.700**	**0.822**	**0.771**	**0.672**
Run 24	3	13	0.612	0.648	**0.882**	0.455	0.636	0.548	0.499	0.593	0.668	0.542	0.622	0.745	0.536

Training the models on more patches per B-scan, significantly increases the amount of data that

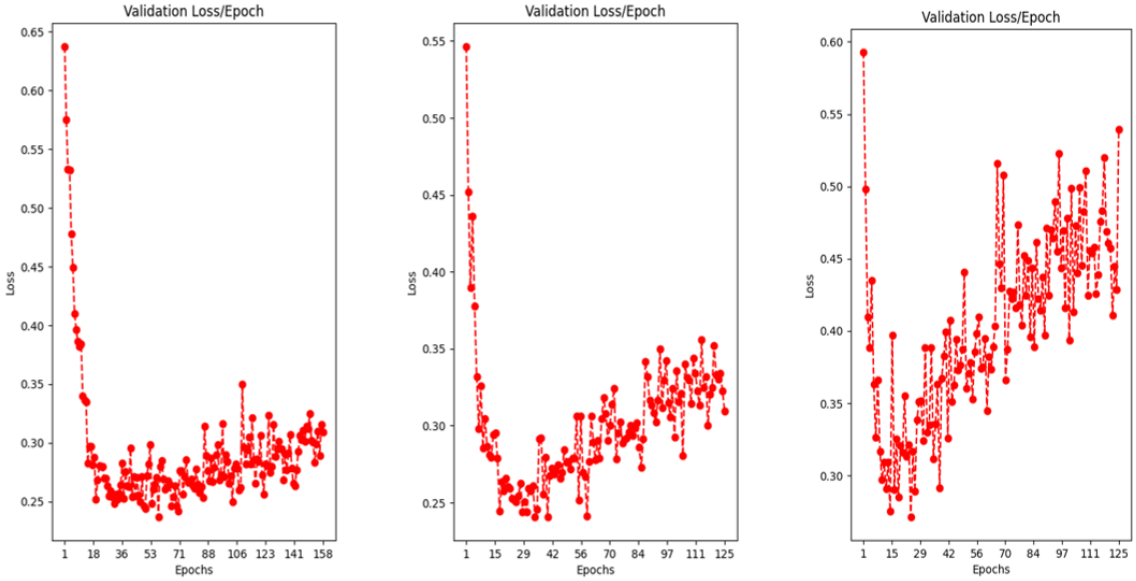


Figure 5.7: Validation loss curves for models validated on fold 2. The curve on the left is from the model trained in Run 17 with four patches, while the middle curve is from the model trained in Run 21 with seven patches. The curve on the right is from the model trained in Run 23 with thirteen patches.

is seen by the model during each epoch. As a result, each epoch takes much longer to complete, requiring more computational power and often leading to bottlenecks. However, since the model sees many more patches per epoch, it reaches the validation loss minimum much earlier than models trained with only four patches.

In Figure 5.7, a comparison is shown between the validation losses of the models validated on fold 2, using four, seven, and thirteen vertical patches per OCT B-scan. Despite following the same trends, the model learns significantly faster as the number of patches increases, resulting in a faster progression of the validation loss. As a result the models starts to overfit much earlier and aggressively with larger number of patches, which is signaled by the increase in validation loss after reaching low values.

When comparing the models' performances (Table 5.3), the models trained with thirteen patches extracted from each training image are outperformed by those trained with fewer patches. These results suggest that the faster progression in validation loss for the model trained with thirteen patches from each training image inhibits its ability to converge to a solution as optimal as those achieved by models trained more slowly with fewer patches.

The performance differences between the models trained with four and seven patches are smaller than the difference between either of those models and one model trained with thirteen patches. However, the model trained with seven patches consistently outperforms the one trained with four in most metrics. In the few cases where the model trained with four patches performs better, the other model still achieves comparable results. The reverse is not true: when the model trained with seven patches outperforms the one trained with four patches, the performance gap

is significantly larger. For these reasons, the model trained with seven patches was considered superior.

Nevertheless, both models still make incorrect predictions that do not respect the anatomical context of the image. For example, in Figure 5.8, the IRF fluid is predicted below the PED, which is not anatomically possible, and there are no examples in the dataset that exhibit this relationship. While this unexpected behavior could be attributed to the B-scan differing from the examples seen during training, the subsequent runs aimed to investigate whether it was caused by the rotation transformation. It was hypothesized that random rotations could alter the vertical order of fluid regions, potentially leading the model to learn incorrect patterns. Additionally, the padding applied to fill the rotated areas, as illustrated in Figure 5.9, could also influence the model's performance.

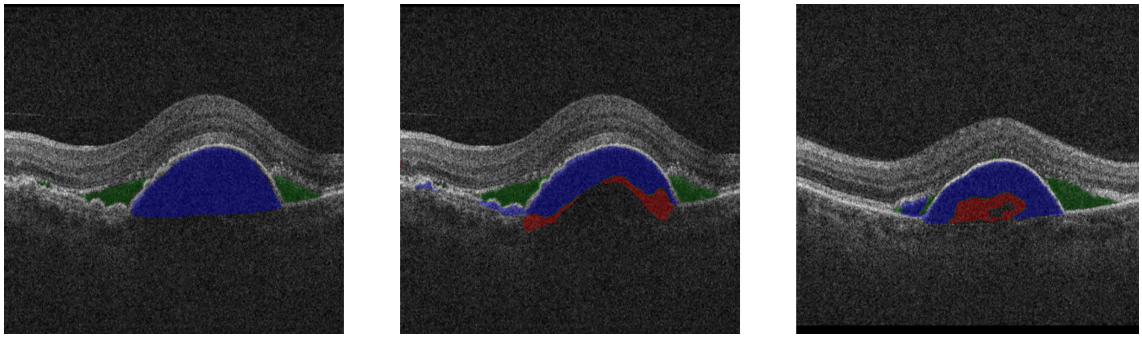


Figure 5.8: Segmentation errors in Cirrus B-scans. The GT fluid mask, seen on the left, shows a large region of PED fluid. Meanwhile, the predictions made by the models trained in Run 21 and 23, which respectively correspond to the B-scans in the middle and right, classify the center of this region as IRF. IRF is represented in red, SRF in green, and PED in blue.

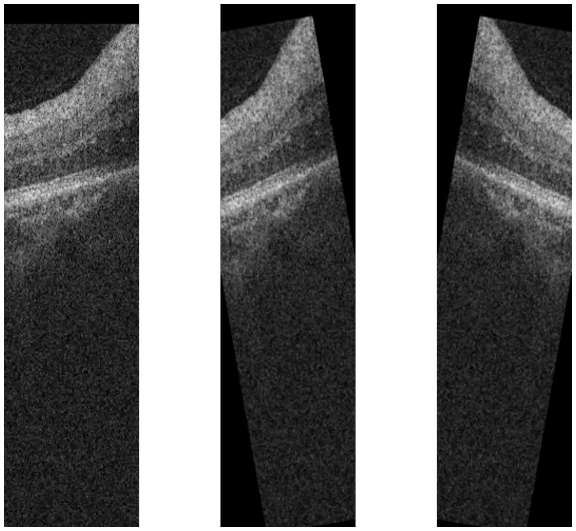


Figure 5.9: Transformations applied to a vertical patch of a Cirrus B-scan. The original vertical patch (left) can be rotated a maximum of 10°, a transformation that is shown in the middle image. The image on the right represents the combination of a 10° rotation and an horizontal flip. It is seen that the rotation transform pads a significant portion of the image, reducing the information in the input.

Table 5.5: Dice coefficients for every vendor and fluid using seven vertical patches per image for training. The only transformation applied to these runs was horizontal flipping, without rotation. Runs 21 and 22 are also shown, promoting an easier comparison.

Runs	VF	Cirrus			Spectralis			Topcon			IRF	SRF	PED	Fluid
		IRF	SRF	PED	IRF	SRF	PED	IRF	SRF	PED				
Run 25	2	0.406	0.789	0.656	0.595	**0.805**	**0.786**	0.762	0.841	0.797	0.560	0.809	0.727	0.620
Run 26	3	0.589	0.775	0.553	0.586	0.803	0.766	0.769	**0.827**	0.755	0.654	0.799	0.664	0.617
Run 27	4	**0.797**	**0.845**	**0.827**	**0.734**	0.782	0.659	0.533	0.756	0.553	0.674	0.798	0.686	0.626
Run 28	0	0.677	0.842	0.699	0.681	0.799	0.732	**0.791**	0.846	**0.854**	**0.719**	**0.836**	**0.762**	**0.665**
Set 6	-	0.617	0.813	0.684	0.649	0.797	0.736	0.714	0.818	0.740	0.652	0.811	0.710	0.632
Run 21	2	0.556	0.837	0.672	0.761	0.853	0.848	0.829	0.908	0.858	0.685	0.864	0.767	0.681
Run 22	3	0.734	0.855	0.836	0.636	0.846	0.689	0.686	0.781	0.731	0.700	0.822	0.771	0.672

Table 5.5 shows the results obtained from the runs performed using only horizontal flipping as the transformation, without applying any rotation. In these runs, seven input patches were extracted from each B-scan. Due to the faster progression of models trained using seven vertical patches per B-scan, the model was trained on a minimum for 25 epochs with a 25-epoch patience applied afterward. Since the minimum validation is usually attained early in models trained with more patches, as seen in Figure 5.7, applying early stopping shortens the training duration while maintaining the same performance.

Removing the rotation from the transformations applied to the input data, significantly worsened the models performances. In Table 5.5, when comparing Run 21 to Run 25, both validated on the same fold, the model trained in Run 21, with rotation, outperformed the model trained in Run 25 in every metric. Following the same trend, the model trained in Run 26 outperformed the model trained in Run 22 in only four metrics: Spectralis PED and all the fluids in Topcon.

The performance differences observed here highlight the importance of applying transformations to the inputs. The introduction of variability to the input data significantly improves the model generalization, preventing the model from overfitting so early on the training stage. In OCT imaging, the rotation is particularly important since different images exhibit different orientations of the retinal layer. By introducing rotation to the inputs, the model becomes more robust to varied input data and less specific to the orientations seen during training. This transformation particularly enhances the performance of the models trained on B-scans where the retinal layer appears horizontally. Figure 5.10 two B-scans that illustrate the variability in retinal orientation.

Nevertheless, the absence of rotation in the training data mitigated the poor segmentation shown in Figure 5.8. As shown in Figure 5.11, the model trained without rotation predicts less IRF in the PED region compared to models trained with rotation. However, it also predicted IRF in a region that none of the other models did, exemplifying why this model still underperformed relative to the others.

In order to balance the robustness introduced by rotation and the preservation of vertical anatomical relationships, new runs were performed using a maximum 5° rotation combined with horizontal flipping. This smaller rotation does not remove as much information from the patches as a 10° rotation. The results obtained in these runs are presented in Table 5.6.

When looking at the results in Table 5.6, the runs trained using a maximum 10° rotation attain

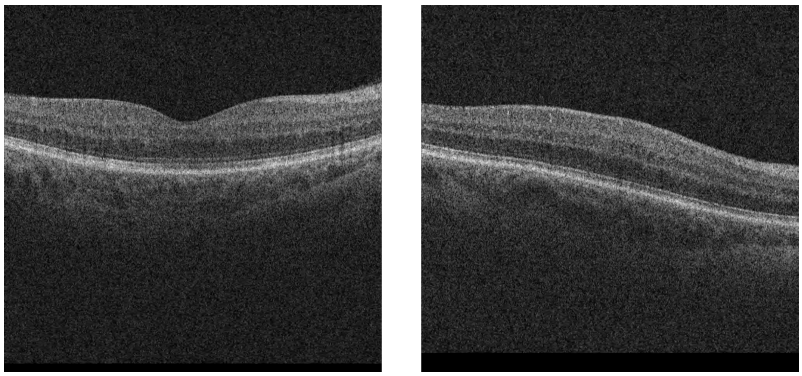


Figure 5.10: Example of two OCT B-scans in which the retina is oriented differently. The retina in the image on the right is significantly more inclined. Both B-scans were obtained using a Cirrus device and do not present fluid.

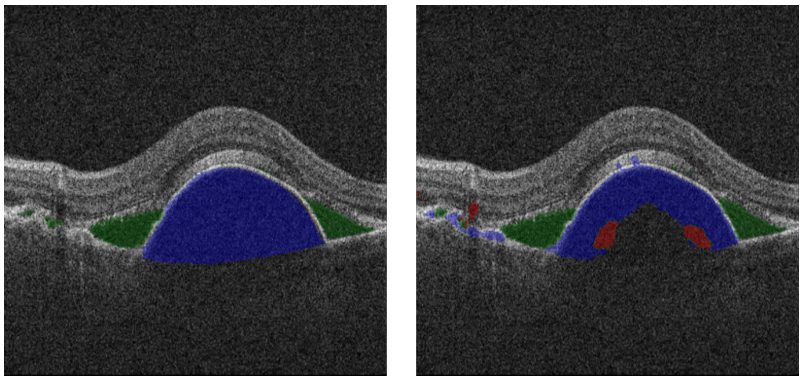


Figure 5.11: Segmentation errors in Cirrus B-scans when trained without random rotations, using the same B-scan as in Figure 5.8. On the left, the GT mask is shown, while on the right, the predicted masks are displayed. IRF is represented in red, SRF in green, and PED in blue.

Table 5.6: Dice coefficients for every vendor and fluid using seven vertical patches per image for training. In these runs, horizontal flipping and 5° rotation were used, instead of the usual 10° rotation. The results obtained in Runs 21 and 22 are also shown, enabling a direct comparison.

Runs	VF	Cirrus			Spectralis			Topcon			IRF	SRF	PED	Fluid
		IRF	SRF	PED	IRF	SRF	PED	IRF	SRF	PED				
Run 29	2	0.497	0.783	0.609	0.763	0.851	0.822	0.832	**0.904**	0.799	0.658	0.836	0.712	0.666
Run 30	3	0.475	0.747	0.613	0.656	0.846	0.757	**0.856**	0.869	**0.835**	0.646	0.809	0.720	0.646
Run 31	4	0.643	0.709	0.747	0.627	0.800	0.635	0.457	0.599	0.529	0.560	0.673	0.638	0.523
Run 32	0	**0.794**	**0.878**	**0.770**	**0.807**	**0.902**	**0.862**	0.796	0.842	0.830	**0.797**	**0.869**	**0.808**	**0.698**
Set 7	-	0.602	0.779	0.685	0.713	0.850	0.769	0.735	0.804	0.748	0.665	0.797	0.720	0.633
Run 21	2	0.556	0.837	0.672	0.761	0.853	0.848	0.829	0.908	0.858	0.685	0.864	0.767	0.681
Run 22	3	0.734	0.855	0.836	0.636	0.846	0.689	0.686	0.781	0.731	0.700	0.822	0.771	0.672

a better Dice coefficient than those that are trained using 5°. However, it is important to also compare the performance specifically on slices containing fluid (Table 5.7).

The results shown in Table 5.7 depict comparable results between the models trained with rotations of 5° and 10°. While the results still slightly favor those trained with 10° rotation, the difference between models is much smaller than it was in Table 5.6. The differences observed

Table 5.7: Dice coefficients for every vendor and fluid using seven vertical patches per image for training. The mean scores are calculated only for the B-scans that contain that fluid. For example, Cirrus IRF is the mean of the IRF Dice coefficient across all the Cirrus slices in the validation fold that contain IRF. The transformations utilized were the same as in Table 5.6.

Runs	**VF**	**Cirrus**			**Spectralis**			**Topcon**			**IRF**	**SRF**	**PED**	**Fluid**
		IRF	**SRF**	**PED**	**IRF**	**SRF**	**PED**	**IRF**	**SRF**	**PED**				
Run 29	2	0.499	0.730	**0.603**	0.596	0.854	0.583	**0.696**	0.629	0.577	0.575	0.731	0.588	0.645
Run 30	3	0.535	0.675	0.584	0.519	0.837	0.592	0.646	**0.735**	0.605	0.561	0.721	0.595	0.623
Run 31	4	**0.691**	0.637	0.602	0.619	**0.899**	0.623	0.506	0.651	**0.656**	0.603	0.709	**0.635**	**0.646**
Run 32	0	0.627	**0.815**	0.590	**0.705**	0.808	**0.646**	0.563	0.489	0.521	**0.627**	**0.761**	0.585	0.645
Set 7	-	0.608	0.742	0.586	0.647	0.848	0.649	0.582	0.620	0.552	0.605	0.746	0.590	0.643
Run 21	2	0.626	0.795	0.449	0.681	0.842	0.771	0.613	0.697	0.524	0.636	0.791	0.518	0.636
Run 22	3	0.720	0.687	0.554	0.653	0.914	0.543	0.569	0.679	0.615	0.647	0.738	0.583	0.651

indicate that the Dice coefficient measured in the models trained with 10° rotation leverages the model's high capability of detecting fluid.

This model is particularly good at detecting which slices have or do not have fluid. When the model does not detect fluid in a slice that does not have fluid, the Dice coefficient for that slice is 1. Meanwhile, in case the slices do not have fluid but the model still detects fluid, the Dice coefficient for that fluid is 0.

Due to the large number of slices that do not have fluid, which is larger than the slices that have fluid, a segmentation model can considerably improve its Dice performance by enhancing its capability of detecting fluid. Despite the model trained with 5° rotation performing worse at detecting fluid, most of these incorrect predictions involve only a small number of sparse pixels (less than 100 pixels in each 496×512 image).

It is also of interest to look into the predictions made by each model. While the Dice coefficient is a good metric to represent the models performance, some relevant segmentation characteristics cannot be expressed by this measurement.

When looking at the predictions made by both models, the model trained with 5° rotations achieves results that are closer to the GT. An example can be seen in Figure 5.12, where a comparison between the predictions made by different models is shown. In this image, the mask predicted by the model trained with 10° rotation (middle image) segments all the PED region, but also segments much more PED overall. Meanwhile, the model trained with 5° rotation correctly identifies the PED region, but segments less PED in the image (right image). The segmentation of IRF and SRF is very similar between both models.

This illustrates how the model trained with 5° rotations tends to be more conservative in the fluid segmentation, despite being less effective at identifying fluid in the B-scans. For this reason, it was expected that this model would perform better on unseen data than the model trained with 10° rotation, which tends to oversegment. Therefore, the maximum rotation selected for further experiments was 5°.

Following the experiments that applied a 5° rotation to the seven vertical patches extracted from each B-scan, additional runs were conducted using the same rotation, but with only four patches extracted from each B-scan instead. The objective of these runs was to verify whether the

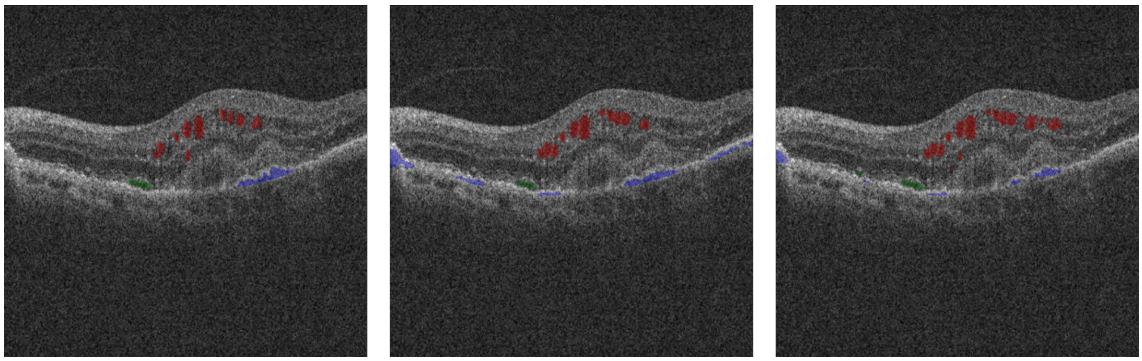


Figure 5.12: Predictions by the models trained using 5° (middle image) and 10° (right image) rotations, compared with the respective GT (left image). IRF is represented in red, SRF in green, and PED in blue.

model still performs better with seven patches than with four patches when the applied rotation is changed from 10° to 5°. The results are shown in Table 5.8.

Contrasting with the results shown in Table 5.3 and Table 5.4, which show a significant improvement when using seven vertical patches over four, the results presented in Table 5.8 show a similar performance between the models trained on four and seven vertical patches. The most notable performance differences appear in the IRF segmentation, where the difference is more significant depending on the number of patches used, favoring the models trained with seven patches.

However, it is necessary to select a single segmentation model to perform inference on the reserved test fold. The best-performing model among those trained with seven vertical patches obtained from each B-scan and a maximum rotation of 5° is the model trained in Run 32. Meanwhile, the best-performing model from "Set 8" is Run 33.

In most metrics, the model trained in Run 32 performs better than the model trained on Run 33. Nevertheless, both models perform very similarly, with the only significant difference being the IRF segmentation, which is superior in Run 32. It is important to note that the models were neither trained nor validated on the same data, and an assumption was made that this would directly translate to performance on unseen data. However, it is possible that the data on which one model

Table 5.8: Dice scores for every vendor and fluid using four vertical patches per image for training. The rotation applied in these runs was of 5°, instead of the previously used 10°. "Set 7" and the best-performing run in this set (Run 32) are also shown, promoting an easier comparison between the use of four and seven patches. The underlined values correspond to the best performances between the models trained in Run 32 and Run 33.

Runs	**VF**	**Cirrus**			**Spectralis**			**Topcon**			**IRF**	**SRF**	**PED**	**Fluid**
		IRF	**SRF**	**PED**	**IRF**	**SRF**	**PED**	**IRF**	**SRF**	**PED**				
Run 33	2	0.598	**0.824**	0.719	**0.805**	0.873	**0.887**	**0.848**	**0.947**	**0.863**	**0.720**	**0.874**	**0.798**	**0.733**
Run 34	3	0.298	0.701	0.436	0.442	0.785	0.720	0.717	0.814	0.746	0.477	0.757	0.599	0.530
Run 35	4	0.670	0.796	**0.850**	0.531	0.825	0.701	0.509	0.703	0.674	0.582	0.759	0.754	0.578
Run 36	0	**0.695**	0.816	0.737	0.761	**0.882**	0.838	0.728	0.867	0.775	0.719	0.846	0.769	0.635
Set 8	-	0.565	**0.784**	**0.686**	0.635	0.841	**0.787**	0.701	**0.833**	**0.765**	0.625	**0.809**	**0.730**	0.619
Run 32	0	0.794	0.878	0.770	0.807	0.902	0.862	0.796	0.842	0.830	0.797	0.869	0.808	0.698
Set 7	-	**0.602**	0.779	0.685	**0.713**	**0.850**	0.769	**0.735**	0.804	0.748	**0.665**	0.797	0.720	**0.633**

was trained prepared it better for the data in the reserved fold. Therefore, the U-Net model selected to represent the multi-class segmentation approach was the one trained in Run 32. This model subsequently performed the inference on the unseen fold (fold 1), and its results are shown in Table 5.9. In this table, the rows correspond to the model's performance across different sets of slices. In the first row the performance across all slices is considered.

The results shown in the table indicate that the model generalized well. The performance obtained in the reserved fold, whose data had not been used for training or validation, reveals the model's robustness across different fluids and vendors. Since the model performs equally well across the three vendors, a good cross-device generalization has been achieved. Therefore, the model is not dependent on the B-scan image quality to produce accurate predictions.

The model behaved differently depending on the fluid it was expected to segment. The Dice coefficient was considerably higher for SRF and PED than for IRF. This is associated with the greater variability in IRF shape and position within the retina. IRF regions are usually irregular in shape, smaller, and less well-defined than the other fluids, often appearing as small structures. This fluid is also located between the retinal layers, making it harder for the model to distinguish from the surrounding tissue, often leading to labeling ambiguity. Still, the model attained a better performance in fluid segmentation than those reported in Table 5.7, where only slices containing fluid are considered.

When considering all slices, the Dice coefficient improves significantly compared to its performance on slices containing fluid only. At the same time, the Dice coefficient obtained in the slices without fluid is also higher than for slices containing fluid. This reflects the greater difficulty of accurately segmenting fluid regions compared to simply detecting their presence. Since most slices in the dataset do not contain fluid, an accurate detection of it can significantly increase the mean Dice across all slices, as explained previously.

Overall, the performance obtained on the unseen fold was significantly better than any other validation result previously shown in Table 5.6. This indicates that the model did not overfit the training data and benefited from the diversity introduced during training - both through image variation and applied transformations. This performance also benefited from selecting the best-performing model during cross-validation, which allows the choice of a robust and generalizable model.

In Figures 5.13, 5.14, and 5.15, some examples of predictions made by the model on the unseen data are shown. In the IRF segmentations, it is observed that the model cannot correctly

Table 5.9: Dice coefficients for every vendor and fluid in the reserved fold (fold 1). The first row reports the mean Dice coefficient across all slices. The second row shows the mean for slices containing the fluid specified in the respective column, while the third row shows the mean for the slices without that fluid type.

Runs	**Slices**	**VF**	**Cirrus**			**Spectralis**			**Topcon**			**IRF**	**SRF**	**PED**	**Fluid**
			IRF	**SRF**	**PED**	**IRF**	**SRF**	**PED**	**IRF**	**SRF**	**PED**				
Run 32	All	1	0.852	0.918	0.934	0.645	0.821	0.852	0.818	0.929	0.755	0.799	0.905	0.841	0.742
	Fluid	1	0.654	0.799	0.767	0.645	0.687	0.703	0.624	0.744	0.588	0.641	0.756	0.716	0.702
	No Fluid	1	0.941	0.972	0.978	0.646	0.889	0.885	0.884	0.960	0.766	0.873	0.952	0.862	0.786

identify the retinal tissue that separates and connects nearby fluid regions. The PED segmentation is characterized by accurate delineation of large PED regions, although some predictions were made in B-scans without PED. Lastly, SRF is the fluid for which the predicted segmentation most closely resembles the GT.

The performance of this model on the reserved test fold was subsequently compared with the best-performing models from the following experiment.

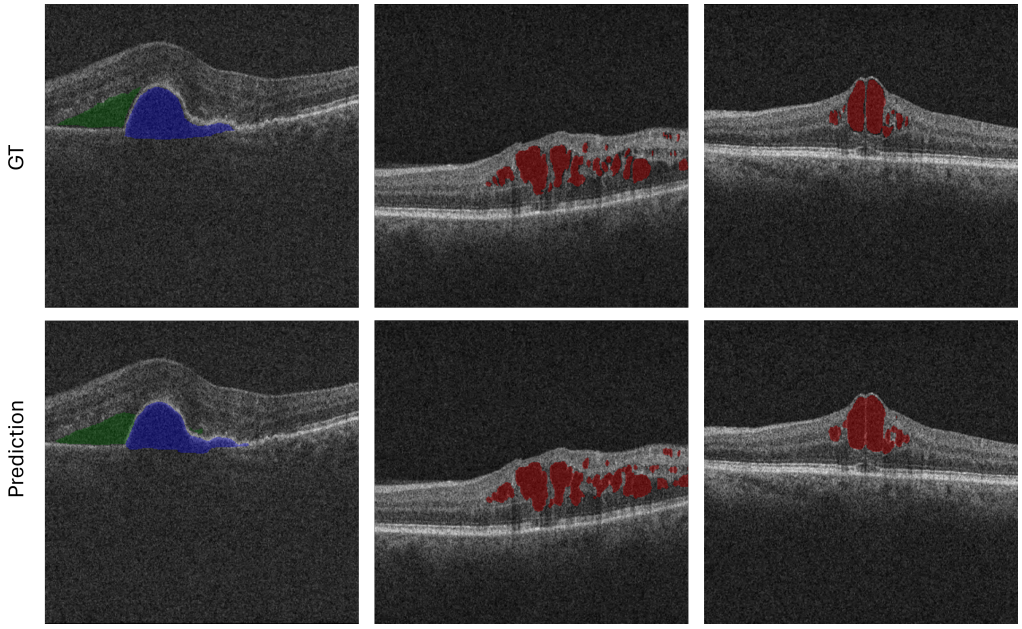


Figure 5.13: Predictions made by the model trained on Run 32 on unseen Cirrus volumes of fold 1. IRF is represented in red, SRF in green, and PED in blue.

5.1.2 Experiment 2 - Multi-class Fluid Segmentation using Three Separate U-Net models

The results presented in this subsection correspond to the use of three separate U-Net models, one for each fluid, to perform multi-class fluid segmentation. In Experiment 2.1, the loss function was the same as in Experiment 1, while in Experiment 2.2, the loss function was changed to the weighted cross-entropy. The results from these experiments were compared with each other.

5.1.2.1 Experiment 2.1 - Multi-class Segmentation Loss

In this experiment, eight models were trained for each fluid type: one model per fold in the multi-class 5-fold split, and one model per fold in a fluid-specific 5-fold split.

Table 5.10 illustrates the performance of the models trained to segment IRF, with the upper block corresponding to the models trained on the multi-class split, while the second block shows the performance of models trained on the IRF-specific split.

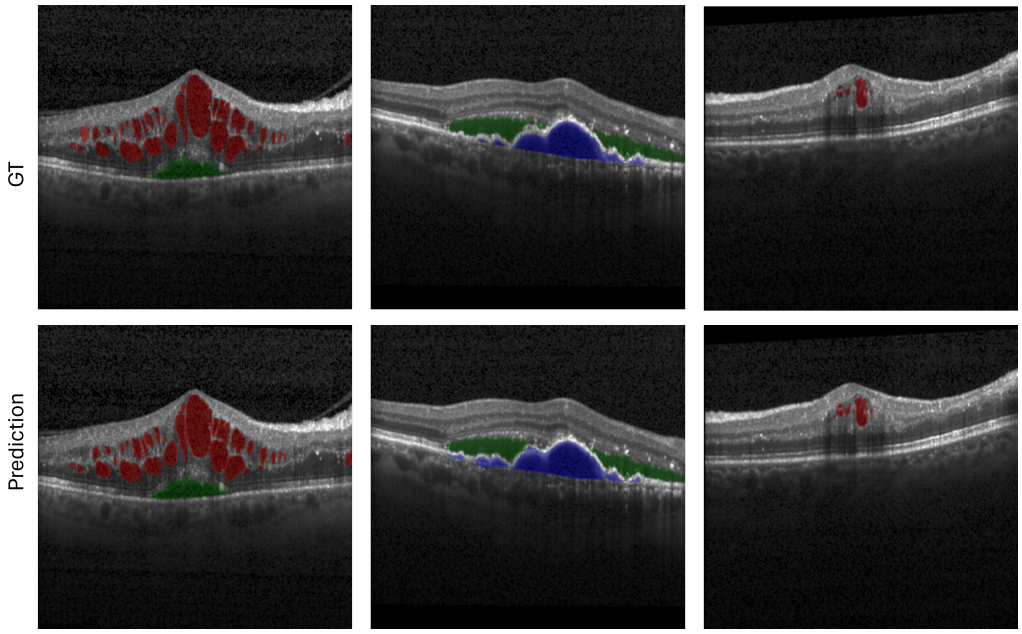


Figure 5.14: Predictions made by the model trained on Run 32 on unseen Spectralis volumes of fold 1. IRF is represented in red, SRF in green, and PED in blue.

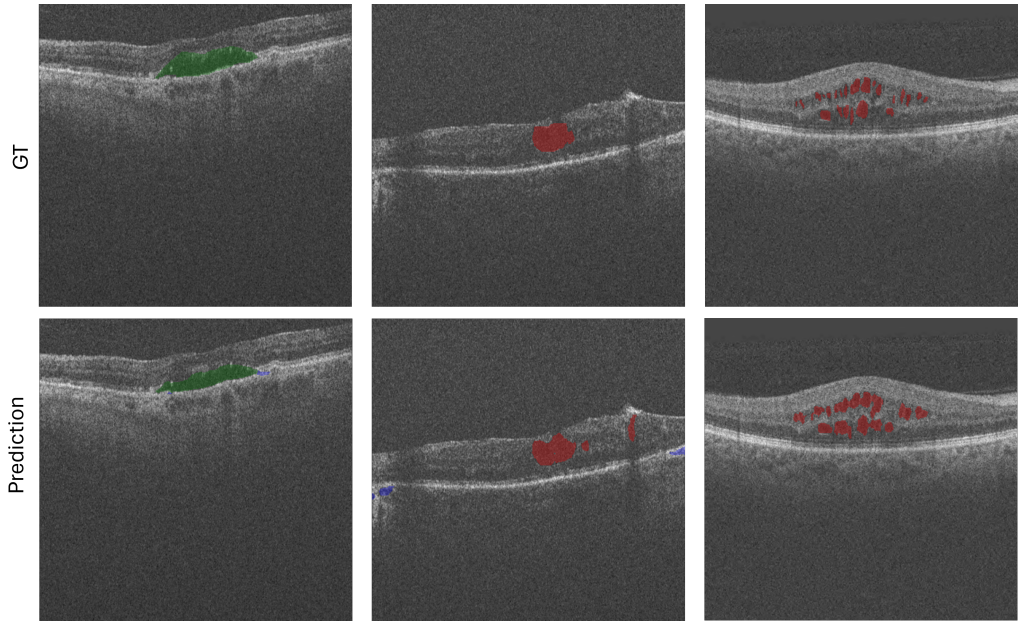


Figure 5.15: Predictions made by the model trained on Run 32 on unseen Topcon volumes of fold 1. IRF is represented in red, SRF in green, and PED in blue.

The results shown in this table follow a well-defined trend: the IRF segmentation in the slices which contain this fluid performs similarly to the segmentation predicted by multi-class models, whereas the performance when considering all slices is significantly worse.

The Dice coefficients obtained for slices containing IRF are really similar to those obtained when training multi-class models, with the difference between the mean scores of the runs being

Table 5.10: IRF Dice coefficients for every vendor. Runs 37 to 40 utilize the multi-class 5-fold split from Experiment 1, while the Runs 41 to 44 use the fluid-specific 5-fold split. The results are presented both for every slice and for only the slices containing IRF.

Runs	VF	Cirrus		Spectralis		Topcon		IRF	
		All	Fluid	All	Fluid	All	Fluid	All	Fluid
Run 37	2	0.478	0.634	0.695	0.669	0.735	0.569	0.604	0.625
Run 38	3	0.536	0.452	0.514	0.588	**0.820**	**0.706**	0.637	0.553
Run 39	4	0.545	**0.683**	0.617	0.670	0.540	0.484	0.552	0.600
Run 40	0	0.576	0.619	0.695	0.682	0.661	0.599	0.628	0.628
Set 9	-	0.534	0.597	0.630	0.652	0.689	**0.590**	0.605	0.602
Run 41	2	**0.654**	0.663	0.541	0.606	0.648	0.619	0.632	0.635
Run 42	3	0.507	0.641	**0.741**	0.670	0.775	0.675	**0.649**	**0.656**
Run 43	4	0.583	0.516	0.689	0.660	0.310	0.451	0.502	0.546
Run 44	0	0.618	0.566	0.644	**0.686**	0.588	0.600	0.611	0.594
Set 10	-	0.591	0.597	0.654	**0.656**	0.580	**0.586**	0.599	**0.608**
Set 7	-	**0.602**	**0.608**	**0.713**	0.647	**0.735**	0.582	**0.665**	0.605

around 0.01 across all vendors. Changing the data split on which the models were trained made little difference in the average performance. However, while the segmentation performance in slices containing IRF remained unchanged, the mean performance across all slices showed a significant improvement in Cirrus, a slight increase in Spectralis, and a significant decrease in Topcon.

The reason Topcon behaves differently is that, overall, the volumes obtained with devices from this vendor contain less fluid than those from other devices. Therefore, the model has to analyze the Topcon images more carefully and be more attentive to smaller details, as the fluid regions are not as big as those seen in images from other vendors. This leads to the model being better at separating the slices with fluid from those that do not present fluid. As seen in Experiment 1, an effective detection of fluid can significantly improve the model's Dice coefficient. The implications of this on model performance are also visible in the multi-class segmentation results, as demonstrated by the performance differences in Topcon B-scans between Table 5.6 and Table 5.7.

When comparing the results across all slices between these binary segmentation models with the multi-class segmentation models, a significant decrease in Dice coefficient is observed in the models tasked with segmenting IRF exclusively. There are two main causes for this decrease. The first is an increment in the number of pixels labeled as background. As the model attempts to exclusively segment IRF, all the pixels that are not labeled as IRF are classified as background. This means that regions corresponding to SRF and PED are also interpreted as background. Although the images in the dataset are predominantly composed of non-fluid pixels, re-assigning the pixels corresponding to other fluids in the retina as background reduces the proportion of fluid pixels in the image, and particularly in the retina, even further. As the number of background pixels increases, the more chances the model has of mislabeling them as fluid.

This increased number of background pixels is further compounded by the lack of inter-class

competition. This inter-class competition is particularly helpful for segmenting other ambiguous regions, such as other fluids, which are not contested by other classes, in binary segmentation. Without this competiton, the model lacks motivation to learn the anatomical characteristics of the retina that are essential for identifying the region corresponding to each fluid type, often leading to over-segmentation.

This translates to a decrease in the IRF segmentation model's performance, particularly in the slices that do not present this fluid, with two examples shown in Figure 5.16. The same performance decrease is also observed in SRF and PED, as shown later, due to the same reasons.

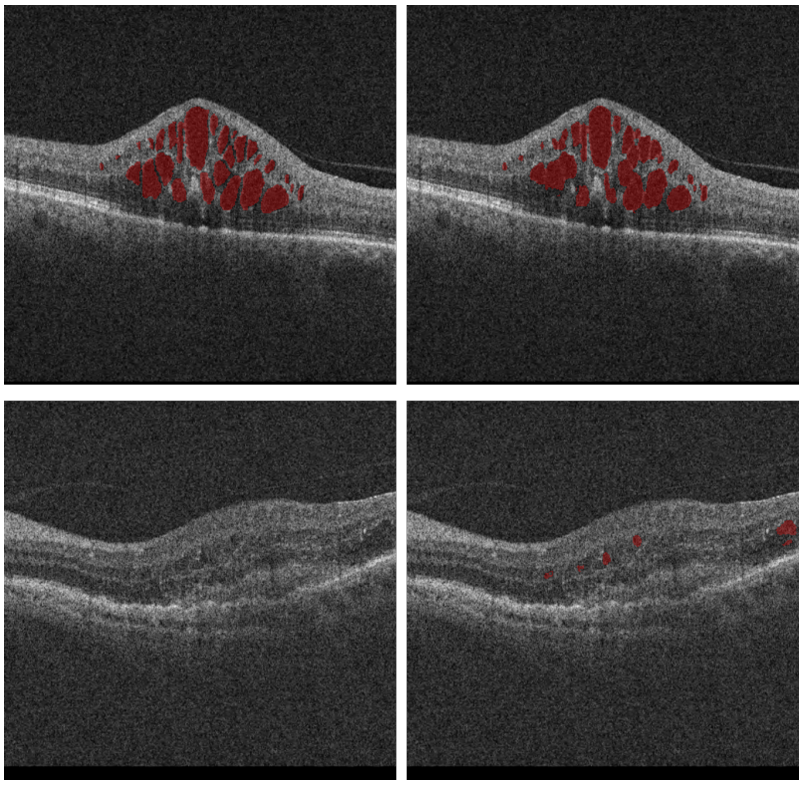


Figure 5.16: Predictions (right) made by the binary IRF segmentation model and their respective GT (left). The prediction represented in the top shows an example of an accurate IRF segmentation, represented in red, while the bottom prediction reveals an oversegmentation in a slice that does not contain fluid.

Table 5.11 shows the performance of the models trained under the same conditions as those used for segmenting IRF, but now with the goal of segmenting SRF. Similarly, the second block of this table shows the results of the models trained using a 5-fold split specifically designed to evenly distribute the OCT volumes according to the amount of SRF they contain.

As shown in this table, the segmentation performance for SRF in slices that contain this fluid type improved when compared to the multi-class model, especially when using the SRF-specific volume distribution. Under these conditions, the binary model significantly outperformed the multi-class model in the segmenting SRF slices from Cirrus and Topcon, while performing almost equally in the Spectralis slices.

Table 5.11: SRF Dice coefficients for every vendor. Runs 45 to 48 utilize the multi-class 5-fold split from Experiment 1, while the Runs 49 to 52 use the fluid-specific 5-fold split. The results are presented both for every slice and only for the slices which present SRF.

Runs	VF	Cirrus		Spectralis		Topcon		SRF	
		All	Fluid	All	Fluid	All	Fluid	All	Fluid
Run 45	2	**0.866**	0.784	0.882	0.826	**0.956**	0.645	**0.899**	0.774
Run 46	3	0.599	0.574	0.741	0.843	0.820	0.737	0.705	0.666
Run 47	4	0.769	0.645	**0.874**	0.873	0.845	0.689	0.816	0.728
Run 48	0	0.667	0.795	0.765	0.776	0.773	0.643	0.723	0.765
Set 11	-	0.725	0.700	0.816	0.830	**0.849**	0.679	0.786	0.733
Run 49	2	0.509	0.758	0.617	0.793	0.759	0.607	0.613	0.745
Run 50	3	0.743	0.733	0.776	**0.885**	0.901	0.801	0.807	0.779
Run 51	4	0.720	0.722	0.864	0.875	0.816	**0.803**	0.781	0.809
Run 52	0	0.775	**0.872**	0.810	0.826	0.879	0.676	0.819	**0.826**
Set 12	-	0.687	**0.771**	0.767	0.845	0.839	**0.722**	0.755	**0.790**
Set 7	-	**0.779**	0.742	**0.850**	**0.848**	0.804	0.620	**0.797**	0.746

When looking at the models trained on the SRF 5-fold split, a significant performance drop is again observed when all slices are considered (similar to what occurred with the IRF model). In these overall metrics, the binary model is outperformed by the multi-class model in every column except for the Topcon slices, where performance improves. As seen in the IRF segmentation, the performance on the Topcon volumes is less affected by binary segmentation due to the low amount of fluid present in the OCT scans from this vendor.

The results obtained using the multi-class split follow the same trend in Topcon. However, in Cirrus, a small increase in the Dice coefficient is observed when considering all the slices. This is caused by an uneven distribution of volumes, resulting in some folds having containing easier cases for fluid detection, which have less fluid. This is reflected in the high variance observed in performance when considering all slices in Cirrus. In contrast, when the data is split evenly, the dispersion is less pronounced, with only one outlier in that device (Run 49). Despite being more noticeable in Cirrus, Spectralis presents the same issue, where the results using the multi-class split differ from what is expected and have sparser values.

Overall, the Dice coefficient obtained in the SRF binary segmentation is higher than that of IRF. This trend was also observed in Experiment 1, where SRF was the best-performing fluid.

The runs in which a binary segmentation U-Net was trained to segment PED are shown in Table 5.12, and the behavior is comparable to that observed for IRF and SRF.

In the binary segmentation of PED, the Dice coefficients obtained for slices containing this fluid type are slightly better results than those obtained in Experiment 1. However, this improvement is only seen when using the 5-fold split specifically designed for this fluid. As observed in the previous experiments, performance declines when all slices are considered.

In the context of PED, it is important to understand how the fluid is distributed across the OCT volumes. While all volumes contain at least some IRF or SRF, PED appears only in a smaller

Table 5.12: PED Dice coefficients for every vendor. Runs 53 to 56 utilize the multi-class 5-fold split from Experiment 1, while the Runs 57 to 60 use the fluid-specific 5-fold split. The results are presented both for every slice and only for the slices which present PED.

Runs	**VF**	**Cirrus**		**Spectralis**		**Topcon**		**PED**	
		All	**Fluid**	**All**	**Fluid**	**All**	**Fluid**	**All**	**Fluid**
Run 53	2	0.305	0.482	0.555	0.623	0.626	0.552	0.459	0.522
Run 54	3	0.447	0.508	0.437	0.628	**0.769**	0.623	0.563	0.583
Run 55	4	0.368	0.594	0.346	0.566	0.328	0.618	0.348	0.600
Run 56	0	0.484	0.547	0.326	0.639	0.697	0.460	0.534	0.545
Set 13	-	0.401	0.533	0.416	0.614	0.605	0.563	0.476	0.563
Run 57	2	0.357	0.561	0.304	0.720	0.456	0.593	0.381	0.594
Run 58	3	0.232	0.581	0.305	0.574	0.361	0.568	0.292	0.574
Run 59	4	0.456	**0.648**	0.345	0.601	0.588	**0.775**	0.499	**0.703**
Run 60	0	**0.515**	0.620	**0.558**	**0.769**	0.672	0.529	**0.580**	0.630
Set 14	-	0.390	**0.603**	0.378	**0.666**	0.519	**0.616**	0.438	**0.625**
Set 7	-	**0.685**	0.586	**0.769**	0.647	**0.748**	0.552	**0.720**	0.590

number of volumes and, generally, in large quantities. This makes an even division across folds more difficult. In each fold, the OCT volumes are typically either severely deformed by PED or contain no PED at all. This happens because most OCT volumes with PED are come from severe cases, where large quantities of all fluids appear. This high intra-class variability increases the difficulty of segmentation and often results in oversegmentation.

This uneven distribution contributes to poor performance when evaluating on all slices. As noted before, the models performed better at fluid detection when trained on volumes with smaller quantities of fluid, such as Topcon, where IRF and SRF appear in smaller areas. A similar trend is observed for PED. However, in the other vendors, where PED is more abundant, the regions with this fluid generally occupy larger areas or are completely absent. This confuses the model, often leading it to predict PED in slices where it is not present. This issue did not affect the multi-class segmentation as significantly, because it had a better understanding of retina anatomy. That model was required to distinguish the three fluid types between them, which encouraged a better association between PED regions and the structural deformations seen of the retina.

As a result, the segmentation performance is good in slices with fluid but poor when all slices are considered. When the data partition is tailored specifically for PED, the quantity of fluid is more balanced between training and validation sets, leading to a performance that aligns more closely with expectations: a significant decrease when all slices are evaluated, compared to the decent performance on slices that contain fluid.

When using the multi-class split, the fluid is not as evenly distributed. This results in folds with more slices containing PED and others with fewer or none. In such cases, especially for Topcon, models trained on folds with less PED perform better at detecting it. For example, in Run 54, the model was trained on Topcon volumes with lower quantities of PED, while its validation fold contained larger amounts of fluid. This results in a correct detection of fluid in the slices

without PED. Inversely, in Run 55, the validation fold is composed of Topcon volumes with small amounts of PED, but the model was trained on volumes with higher PED quantities. This results in oversegmentation during validation, thus attaining low Dice values when considering all the slices, but a decent score on the slices that have PED. Two examples of the segmentation performed by the binary PED segmentation model are shown in Figure 5.17.

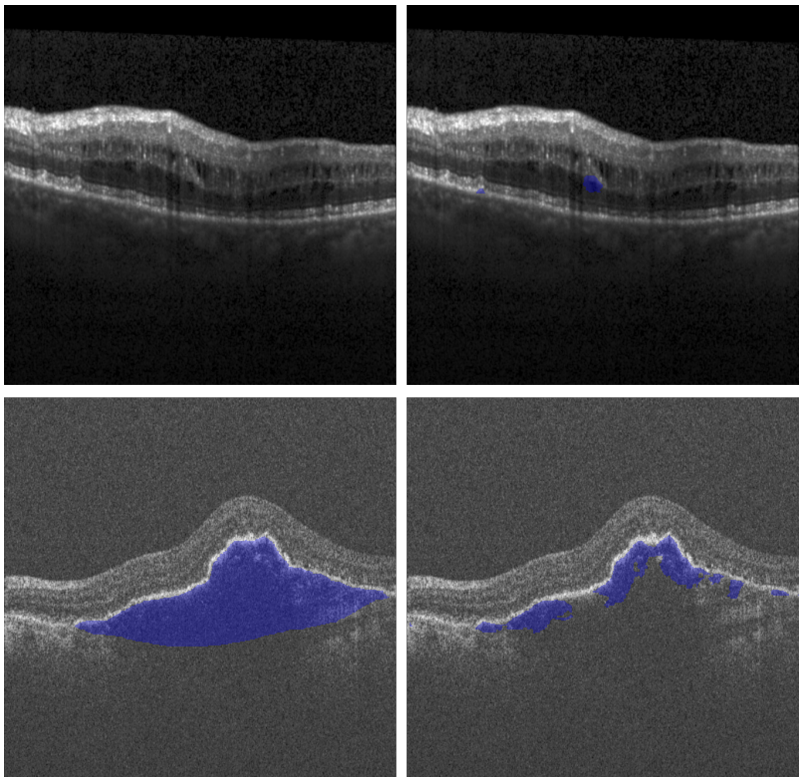


Figure 5.17: Predictions (right) made by the binary PED segmentation model and their respective GT (left). In the top-right image, the model predicted PED fluid, represented in blue, in a region where it does not exist. The bottom images show undersegmentation by the model trained in Run 59. While this model is capable of detecting the fluid's location, it fails to segment it completely.

In order to compare the results from this experiment to those obtained in Experiment 1, three models had to be selected: one for IRF, one for SRF, and one for PED. Since the results obtained in Experiment 2.2 were much worse than those achieved in this experiment, the models selected for inference on the unseen fold were chosen from the runs performed in Experiment 2.1.

From the models trained for IRF segmentation, ranging from Run 37 to Run 44, the model which obtained the best performance was the one trained in Run 42, as shown in Table 5.10. This model reached the best average performance in IRF segmentation both across all slices and when only slices with IRF were considered. In the other metrics, this model's performance was consistently similar to that of the best performing model in each metric. When compared to the average model from "Set 7", this model outperformed it in most metrics.

The model selected to infer the SRF masks in the unseen fold was one model trained in Run 52. This model was the best-performing SRF model in fluid segmentation for slices where SRF

is present. In the other metrics, it performed similarly to the best models, surpassing the average model from "Set 7" in most metrics.

For PED, the selected model was the one trained in Run 59. This model was the best-performing PED segmentation model for slices that contain this fluid type, with a significant difference from the others. When considering all slices, the model was the second-best, but with a weak performance. Regarding the slices with fluid, this model outperformed the average from "Set 7" across all vendors. However, similar to all other models trained for the binary segmentation of PED, it performed significantly worse performance than the multi-class segmentation models.

When merging the masks from three independent binary segmentation models with the goal of outputting a multi-class segmentation, overlaps may occur, where multiple models assign different fluid labels to the same pixel. To resolve this conflict, two alternatives were explored in the reserved test fold: a priority-based approach, where SRF takes precedence over IRF, which in turn takes precedence over PED; and a probability-based approach, where the predicted probabilities for each pixel are compared, assigning the label with highest predicted probability to it. An example comparing the probability-based approach to the priority-based approach is shown in Figure 5.18.

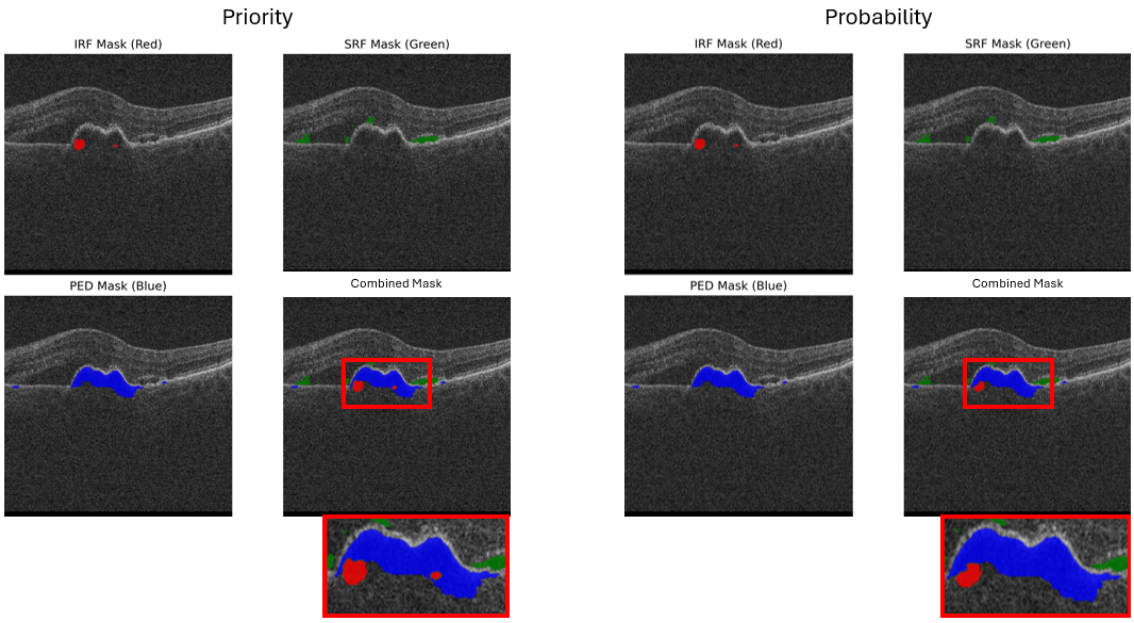


Figure 5.18: Comparison between the priority (left) and probability (right) merging approach. IRF is represented in red, SRF in green, and PED in blue.

The results obtained using the three selected models are shown in Table 5.13, where the first block corresponds to the results when using the priority-based merging approach and the second block presents the results when using the probability-based approach.

The performances of the models shown in this table indicate that the best merging strategy is the probability-based approach. In the slices that contain the target fluid, the priority-based approach performs better for SRF, since it is the fluid with the highest priority. Regarding IRF, in

Table 5.13: Dice coefficients for every vendor and fluid in the reserved fold (fold 1), using the models from Runs 42 (IRF), 52 (SRF), and 59 (PED). The first block shows results using the priority-based merging strategy, while the second block presents the results using the probability-based merging strategy. In each block, the first row reports the mean Dice score across all slices. The second row shows the mean Dice score for slices containing the specified fluid, while the third row shows the mean Dice score for slices without that fluid. Bold values highlight the best performance for each corresponding row between merging strategies (e.g., the "All" row in the first block is compared with the "All" row in the second block).

Merging Strategy	**Slices**	**VF**	**Cirrus**			**Spectralis**			**Topcon**			**IRF**	**SRF**	**PED**	**Fluid**
			IRF	**SRF**	**PED**	**IRF**	**SRF**	**PED**	**IRF**	**SRF**	**PED**				
Priority-	All	1	0.693	0.776	0.425	0.587	0.623	0.177	0.808	0.858	**0.531**	0.723	0.783	0.425	**0.544**
based	Fluid	1	0.687	**0.670**	0.573	**0.660**	**0.659**	0.632	**0.634**	**0.560**	**0.484**	**0.661**	**0.640**	0.570	**0.639**
Approach	No Fluid	1	0.695	0.824	**0.387**	**0.520**	0.605	**0.075**	0.867	0.907	**0.534**	0.752	0.830	**0.402**	**0.440**
Probability-	All	1	**0.697**	**0.787**	**0.428**	**0.628**	**0.631**	**0.178**	**0.810**	**0.864**	**0.531**	**0.733**	**0.792**	**0.427**	**0.544**
based	Fluid	1	**0.688**	0.663	**0.586**	**0.660**	0.646	**0.636**	0.632	0.537	**0.484**	**0.661**	0.627	**0.578**	**0.639**
Approach	No Fluid	1	**0.701**	**0.844**	**0.387**	**0.520**	**0.623**	**0.075**	**0.870**	**0.917**	**0.534**	**0.766**	**0.844**	**0.402**	**0.440**

the slices where this fluid is present, the performance is equal for both approaches, while for PED, the probability-based approach performs better.

However, when considering all slices, the probability-based merging approach performs significantly better. This due to its superior handling of slices that do not contain at least one type of fluid. In such slices, this approach prevents the segmentation of fluid regions which are not present. Oversegmentations can be incorrectly predicted in the regions belonging to other fluids. However, since the models responsible for correctly segmenting those regions are typically more confident than the ones making incorrect predictions, the labels are attributed to the correct class.

It is important to note that the "Fluid" column does not change between merging approaches. This is because the segmented fluid region remains the same, since the only changes occur in pixels contested by multiple fluids. If all fluids are treated as a single class, as considered in this column, these contested pixels do not affect the results.

The overall segmentation performance is comparable to the values obtained during training on the validation folds. For IRF, the model trained in Run 42 performed better on the unseen fold than on the fold where it was originally validated. This improvement holds when considering all slices, only those with IRF, and those without IRF, indicating good generalization by the model.

In contrast, the SRF segmentation model did not perform as well on the unseen fold. While fluid detection improved - evident from the higher Dice coefficient when considering all slices compared to just those with fluid - this contrasts with its performance during validation in Run 52, where performance dropped in that scenario. However, the segmentation worsened significantly in slices containing SRF, with the Dice coefficient dropping from 0.819 in validation to 0.627 in the unseen fold. Nonetheless, when considering all slices, the Dice coefficients are still comparable, largely due to the improved performances on slices without SRF.

For IRF and SRF, the models benefited from a good generalization in the fluid detection task, outperforming their validation results. However, in the case of PED, where the models were most sensitive, the performance declined. PED is the only fluid for which the Dice coefficient is lower when considering all slices than when considering only slices with the fluid. This drop in

performance is directly related to the model selected for PED segmentation. As seen in Table 5.12, the model trained in Run 59 already shows a subpar performance in fluid detection, as indicated by the decrease in Dice coefficient when considering all slices instead of only those that contain PED. Therefore, the poor detection in unseen images is not unexpected. It is worth noting that the selected model performs worse in the Spectralis volumes, which mirrors what was observed during validation. Similarly, the vendor in which the best performance is attained is the Topcon, where the model also achieved the highest scores during validation. Moreover, all PED segmentation models were expected to generalize poorly on the unseen fold in the fluid detection task, due to the previously discussed reasons that explain the lack of robustness in these models, such as the uneven distribution of PED in OCT volumes.

In Figures 5.19, 5.20, and 5.21, the predictions made by the multi-class segmentation using fluid-specific binary models, in Cirrus, Spectralis, and Topcon are shown, respectively. The B-scans presented in these figures are the same as those shown in Experiment 1.

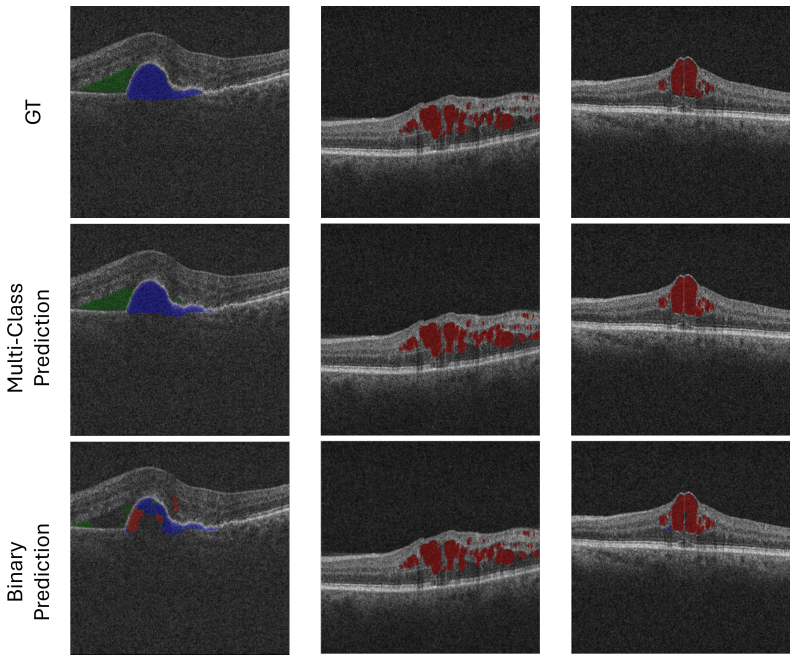


Figure 5.19: Predictions made by the models trained in Run 42, 52, and 59 on unseen Cirrus volumes from fold 1 (last row), contrasted with the predictions made by the multi-class model from Experiment 1. IRF is represented in red, SRF in green, and PED in blue.

5.1.2.2 Experiment 2.2 - Balanced Cross-entropy Loss

In this experiment, the loss function used for the training of the segmentation models was changed from a combination of a weighted cross-entropy and Dice coefficient of the foreground to the weighted cross-entropy. The results are shown in Table 5.14, where only two runs were performed for the IRF segmentation, one in each 5-fold split.

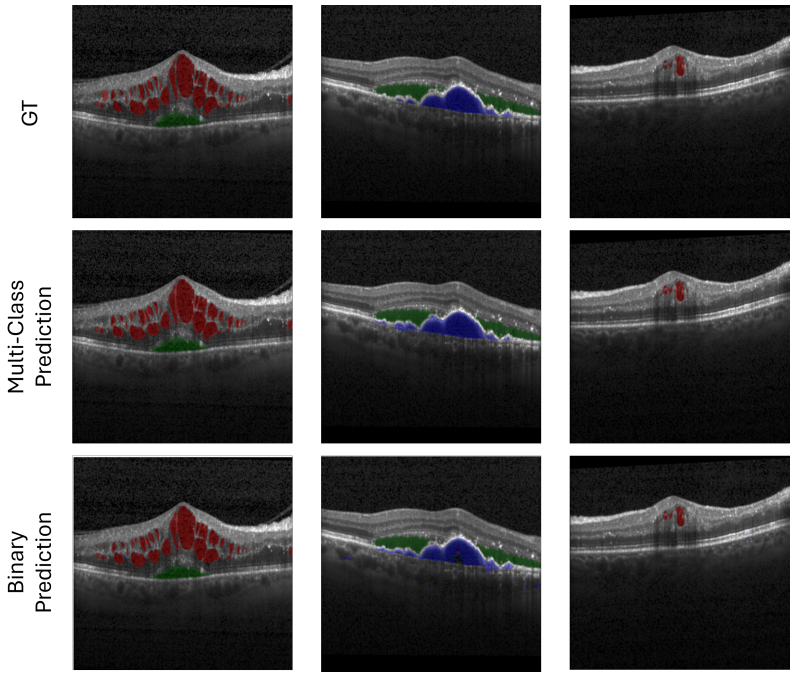


Figure 5.20: Predictions made by the models trained in Run 42, 52, and 59 on unseen Spectralis volumes from fold 1 (last row), contrasting with the predictions made by the multi-class segmentation model from Experiment 1. IRF is represented in red, SRF in green, and PED in blue.

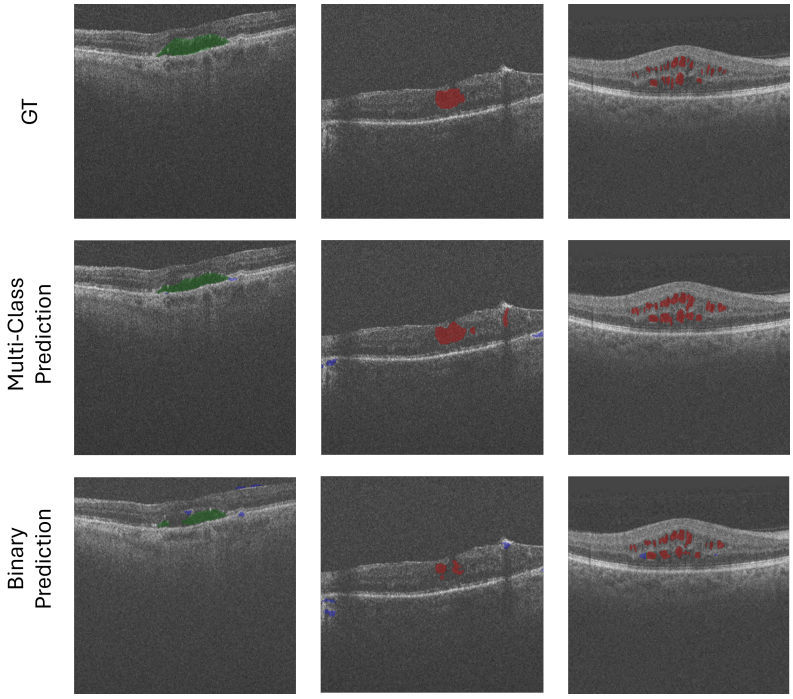


Figure 5.21: Predictions made by the models trained in Run 42, 52, and 59 on unseen Topcon volumes of fold 1 (last row), contrasting with predictions made by the multi-class segmentation model from Experiment 1. IRF is represented in red, SRF in green, and PED in blue.

Table 5.14: IRF Dice coefficients for every vendor, using the binary segmentation U-Net for IRF. The loss function used in Runs 61 and 62 is the weighted cross-entropy. This model was trained on two different validation folds: validation fold 0 from the multi-class 5-fold split, and validation fold 4 from the IRF-specific 5-fold split. These results can be compared, respectively, to those obtained in Runs 40 and 43. The metrics are compared between rows with the same name, so that the values in bold are the best across the four experiments.

Runs	Slices	VF	Cirrus	Spectralis	Topcon	IRF
Run 61	All	0	0.088	0.164	0.121	0.113
	Fluid	0	0.320	0.390	0.472	0.378
	No Fluid	0	0.002	0.044	0.041	0.024
Run 62	All	4	0.130	0.250	0.142	0.151
	Fluid	4	0.399	0.531	0.549	0.466
	No Fluid	4	0.000	0.052	0.053	0.030
Run 40	All	0	0.576	**0.695**	**0.661**	**0.628**
	Fluid	0	**0.619**	**0.682**	**0.599**	**0.628**
	No Fluid	0	0.555	0.705	**0.686**	**0.628**
Run 43	All	4	**0.583**	0.689	0.310	0.502
	Fluid	4	0.516	0.660	0.451	0.546
	No Fluid	4	**0.601**	**0.719**	0.272	0.486

The results shown in the table are much worse than those obtained in the same conditions with the previous loss. In the slices with IRF, the segmentation was poor, but still had a performance almost comparable to the values obtained in Runs 40 and 43.

However, what really worsens the overall performance is fluid detection. All the Dice coefficients in the slices with no fluid were lower than 0.05, while most of the models studied in Experiment 2.1 reached values higher than 0.5.

The reason these values were so low is related to the loss function used. By keeping the Dice loss as a component of the training loss, the model is highly penalized when predicting fluid in an image that does not have fluid. Whenever any fluid is predicted in such an image, the Dice coefficient is 0. This motivates the model to carefully predict pixels in regions of the image where it highly believes there is fluid.

When removing the Dice component from the loss, the wrongful predictions are only penalized by the cross-entropy component. This component does not penalize incorrect predictions of fluid in empty images as heavily. The cross-entropy of each B-scan is calculated based on the number of pixels belonging to a class in the image, as shown in Chapter 3. Whenever a background pixel is wrongfully classified as fluid, the resulting penalty is very small, due to the large quantity of background pixels in the mask. Therefore, the motivation to not segment fluid in images composed solely of background is much smaller than it is when the Dice is considered. This results in segmentations such as the one seen in Figure 5.22.

The cross-entropy is much more useful when regularizing the mislabeling, especially on fluid. Since this loss component is adjusted according to the representation of each class in the image, wrongfully labeling a fluid region as another fluid or background brings a large penalty to the

model. This is why it is combined with the Dice loss in the original loss: while the Dice ensures that the correct areas of fluid are segmented, the cross-entropy makes sure that the correct labels are assigned to each pixel. For this reason, most of the segmentation errors seen in the predictions made by the models trained in Runs 61 and 62 are oversegmentations, and not as many incorrect labelings. An example of how this translates to the model's segmentation is shown in Figure 5.23.

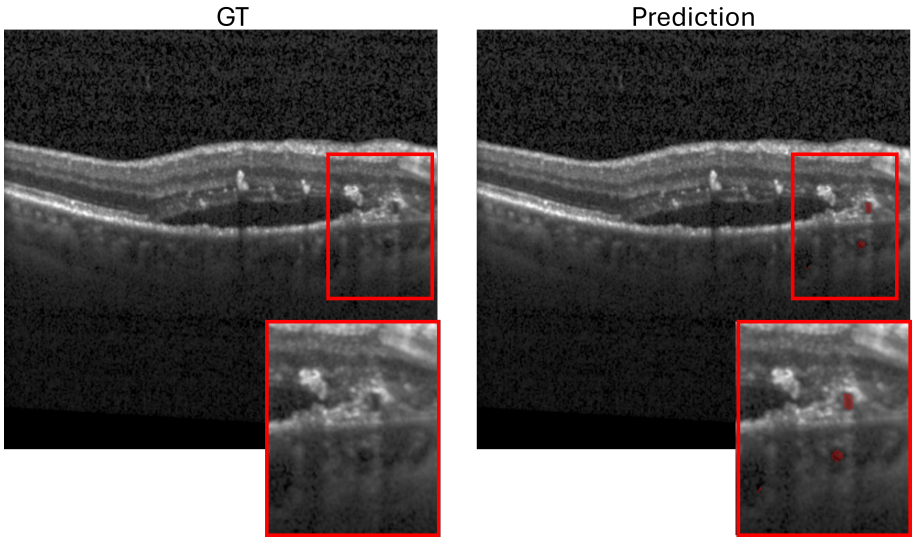


Figure 5.22: Small IRF (red) oversegmentation performed in a Spectralis B-scan. This type of predictions appear commonly across multiple B-scans.

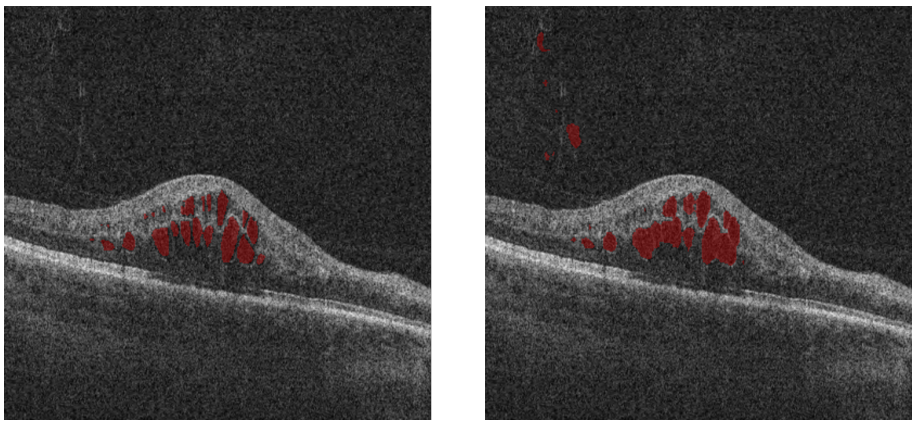


Figure 5.23: IRF (red) segmentation in a Cirrus slice, represented in red, performed by a model trained using BCE (right). The corresponding GT is shown on the left. Although this example presents a decent segmentation, significant oversegmentation is seen on the top left of the image.

5.1.3 Experiment 1 vs Experiment 2: Comparison in the RETOUCH Dataset

In order to compare Experiment 1 with Experiment 2, both their performances were presented in Table 5.15. These experiments are compared by evaluating each experiment's performance on the reserved test fold of the RETOUCH dataset. In the table, the first block corresponds to

Table 5.15: Dice scores for every vendor and fluid in the reserved fold (fold 1), using the best models from Experiment 1 and Experiment 2. In Experiment 1, the model trained on Run 32 was selected, while in Experiment 2 the chosen models were those from Runs 42 (IRF), 52 (SRF), and 59 (PED).

Runs	Slices	VF	Cirrus			Spectralis			Topcon			IRF	SRF	PED	Fluid
			IRF	SRF	PED	IRF	SRF	PED	IRF	SRF	PED				
Run 32	All	1	**0.852**	**0.918**	**0.934**	**0.645**	**0.821**	**0.852**	**0.818**	**0.929**	**0.755**	**0.799**	**0.905**	**0.841**	**0.742**
	Fluid	1	0.654	**0.799**	**0.767**	0.645	**0.687**	**0.703**	0.624	**0.744**	**0.588**	0.641	**0.756**	**0.716**	**0.702**
	No Fluid	1	**0.941**	**0.972**	**0.978**	**0.646**	**0.889**	**0.885**	**0.884**	**0.960**	**0.766**	**0.873**	**0.952**	**0.862**	**0.786**
Runs 42, 52, and 59	All	1	0.697	0.787	0.428	0.628	0.631	0.178	0.810	0.864	0.531	0.733	0.792	0.427	0.544
	Fluid	1	**0.688**	0.663	0.586	**0.660**	0.646	0.636	**0.632**	0.537	0.484	**0.661**	0.627	0.578	0.639
	No Fluid	1	0.701	0.844	0.387	0.520	0.623	0.075	0.870	0.917	0.534	0.766	0.844	0.402	0.440

the performance obtained with the selected multi-class segmentation model, while the second block shows the performance of the combination of models from Experiment 2, with their masks combined using the probability-based merging approach.

By looking at this table, the best segmentation performance becomes evident. The multi-class model consistently outperforms the binary models in across all fluids, vendors, and slice groups.

Regarding the performances per fluid type, the multi-class segmentation model performs better than the binary segmentation models in all fluids when all the slices are considered and when only the slices that do not contain a certain fluid are considered. The binary models only outperform the multi-class model for the segmentation of IRF in slices that contain this fluid. In this case, the binary models attain small improvements over the multi-class model. However, the segmentation of IRF in the slices which contain this fluid is not enough to compensate for the worse performance in fluid detection, when considering the slices without IRF. This is seen as the major flaw of the binary segmentation models, which is tied to the models' weaker capability to understand anatomical structures. This leads to a worse overall performance in IRF even when the segmentation is better. An example of similar performances between these models is seen in the IRF segmentations of Figures 5.19, 5.20, and 5.21.

The performances reached by the binary model in SRF are worse than those obtained with the multi-class model. While the predictions made with the binary model are still satisfactory, they are not as good as those made by the multi-class model, as seen in the first case of the Figures 5.19 and 5.21. This model shows a great fluid detection, with a Dice coefficient greater than 0.85 in all classes, and the best fluid segmentation of all classes.

The biggest performance differences between models are those observed in the PED class. The segmentation of PED in slices that contain this fluid is worse in the binary model than the in multi-class model. Despite the segmentation being worse across all vendors, the performance in the binary model is still acceptable. However, in the fluid detection, the binary model performs poorly, especially when compared to the multi-class model. These results are tied to the models' capabilities in identifying the region in which PED is located, which was seen to be much superior in the multi-class model. Two examples that show this are seen on the left side of Figure 5.19 and in the segmentation at the bottom of Figure 5.17, where the region segmented by PED does not occupy the entirety of the fluid, but only the section closest to its upper boundary.

It is important to compare the results in the RETOUCH dataset with those obtained in the literature to better understand the impact and effect of the chosen methodology. Alsaih et al. [77] present an insightful study on the performance of multiple networks, including the U-Net, in the RETOUCH dataset. This paper was chosen as a benchmark for comparison due to the shared dataset and the similarity with the selected methods.

In this article, the networks were trained and evaluated following a 3-fold approach, with the OCT volumes being equally distributed across the folds while still keeping a fair representation of vendors. The implementations in Experiment 1 and Experiment 2 were compared to two other base U-Net implementations in this paper: one trained with the entire image and another trained with patches. Multiple patch sizes (32×32, 64×64, 96×96, 128×128, 256×256, and 512×512) extracted with different overlapping percentages were tested, but the selected was 128×128 with 80% overlapping as they obtained the best results. All the B-scans were previously resized to 572×572.

For each model, Alsaih et al. [77] trained three networks with different optimizers: one using the stochastic gradient descent while the other two were trained using the Adam. Then, the results from each network were fused through majority voting. The results were also post-processed using a median filter.

These differences bring advantages to the implementations by Alsaih et al. [77], as they enhance the robustness and overall segmentation performance. While these changes provide a performance edge, it is challenging to find other studies with a comparable setup in terms of network architecture, dataset, and evaluation protocol. For these reasons, this paper was considered the most similar and, overall, suitable for comparison with the experiments presented here.

In Table 5.16, the mean of the results in three validation folds of the models trained by Alsaih et al. [77] is shown, as well as the results from "Set 7" and "Sets 10, 12, and 14" together, representing the multiple binary models used in Experiment 2.

Table 5.16: Dice scores for every fluid in "Set 7" and "Set 10, 12, and 14", compared with the results obtained in the two networks proposed by Alsaih et al. [77].

Runs	**IRF**	**SRF**	**PED**
Set 7	0.67	**0.80**	0.72
Sets 10, 12, and 14	0.60	0.75	0.44
U-Net trained with entire image [77]	0.29	0.47	0.18
U-Net with 128×128 patches [77]	**0.68**	0.61	**0.84**

In this table, the first difference which presents a clear advantage in fluid segmentation is the use of patches in training. It becomes clear that the best segmentation performances in all fluids were achieved by the models that used patches for training. This enhances the model's capability of perceiving the image's finer details, which is deemed particularly helpful for IRF and PED.

Despite the significant performance differences between the U-Net trained with the full image and the U-Net trained with patches implemented by Alsaih et al. [77], the variation in performance in SRF is smaller than that seen in IRF and PED. This is because the use of patches for training

particularly favors the segmentation of localized fluids like IRF and PED, but not so much the SRF, since it depends on the model's capability to capture broader anatomical context. Since SRF is often visually similar to the background, the patch-based approaches are more prone to misclassify it.

The use of vertical patches in Experiment 1 and 2 provides the anatomical context necessary for the segmentation of SRF, as reflected in the significantly better performance compared to that reported by Alsaih et al. [77]. In IRF segmentation, the difference in Dice scores between the two approaches is less pronounced. While the smaller patches used by Alsaih et al. [77] allow the model to focus on finer details, such as pixel intensity in fluid regions, the vertical patches from Experiment 1 and 2 offer a broader field of view that captures the structural references helpful for segmentation.

For PED, the difference between experiments is also significant, though smaller than that in SRF. However, the trend seen with this fluid is reversed compared to the one seen in SRF, as the models trained on smaller patches tend to perform better for PED. While models trained with vertical patches are more effective at determining the boundaries of PED fluid, they often lack the attention to the finer details. This is particularly relevant in cases with large PED regions, where the significant deformation of the retina causes parts of the fluid to be positioned further away from key anatomical landmarks like the retinal layers. In such scenarios, models trained on smaller patches, which are attentive to details such as pixel intensity and less dependent on anatomical context, perform better. This contributes to the overall improved performance of models trained with smaller patches in segmenting PED.

Overall, the models trained in Experiments 1 and 2 perform comparably to or better than the U-Nets implemented by Alsaih et al. [77], despite the previously stated advantages employed in this paper. This highlights the effectiveness of the patch sizes explored in these experiments, which is both capable of capturing details fine enough to segment IRF, but also large enough to detect SRF. The absence of ensemble networks used for the same task or refinement techniques further emphasizes the robustness of the proposed approach, making it a competitive and more efficient alternative for multi-class fluid segmentation in retinal OCT images.

In a previous work, the authors also implemented the network developed by Tennakoon et al. [65], using the same dataset. This network is a 2.5D segmentation network that receives three consecutive B-scans and segments the intermediate one. This model was trained on patches of size 256×128, as done in Experiment 1.1, and the results from this model are shown in Table 5.17. The implementation exhibited in this table corresponds to a single model performance, which was trained for 100 epochs using an 80% training and 20% testing split. A fair vendor representation was ensured in the split, but the quantities of fluid were not considered.

In this table, it is seen that the 2.5D implementation performs better than the models in "Set 7" and "Sets 10, 12, and 14" for IRF and SRF, but worse for PED and in overall fluid segmentation.

The main advantage for using this model is the consideration of the surrounding slices as a way to capture the context around the fluid regions. The small transitions, which are noticeable when viewing three consecutive slices, allow the model to learn the limiting regions of the fluids. This

Table 5.17: Dice scores for every vendor and fluid in "Set 7", "Set 10, 12, and 14", compared with the results obtained in the authors' previous implementation of the work of Tennakoon et al. [65].

Runs	IRF	SRF	PED	Fluid
Set 7	0.67	0.80	**0.72**	**0.63**
Sets 10, 12, and 14	0.60	0.75	0.44	-
Tennakoon et al. [65]	**0.71**	**0.85**	0.71	0.62

understanding of surrounding areas, combined with the attention to small changes encouraged by the training on small patches, makes this model capable of both fine and coarse segmentation, as it is able to recognize the limits of fluid regions independently of their size.

This explains the improved segmentation of SRF, as the model better captures the small transitions in the SRF regions while still understanding the changes in the space around it which allow for the identification of this fluid. Similarly, the small patches also promote a better segmentation of IRF, where the understanding of small transitions is required for a correct segmentation.

In PED, the performance is worse, but this is mainly due to the conditions in which this model was trained. Since the data was split randomly, the training volumes contained large quantities of PED, which led to the prediction of significant quantities of this fluid in multiple B-scans. For this reason, the Dice coefficient when considering all the fluid as a single class was also deeply penalized, as seen in the table.

However, it is important to note that the results shown do not come from a cross-validation, and instead from a single split. This means that not only the 2.5D segmentation model was trained on more data than those in Experiments 1 and 2 (approximately 60% of the RETOUCH dataset), but it could also have a split which either benefits or harms the model's performance.

5.1.4 Experiment 1 vs Experiment 2: Comparison in the CHUSJ Dataset

The results obtained in the CHUSJ private dataset were not as good as those obtained in the RETOUCH dataset, on which the model was trained. The main cause for this poor segmentation is the difference between the RETOUCH and CHUSJ datasets regarding image quality and segmentation criteria. In Table 5.18, the Dice coefficients across the different fluids are shown, both for the images in their original size and resized to 496×512. Since the segmentation U-Net is a fully convolutional network and does not contain any shape-dependent layer, the trained models can be applied to CHUSJ B-scans in their original size or after resizing.

In the RETOUCH dataset, the volumes were extracted with consistent dimensions and noise levels in scans from the same vendor. This results in visually similar images, whose differences can be attributed to the patient's characteristics. Therefore, while the differences seen in the CHUSJ dataset are a good test to understand the models' robustness, those trained on RETOUCH did not generalize well to this new dataset. In RETOUCH, the images obtained using the Spectralis device present consistent levels of noise, while in the CHUSJ dataset, the noise levels and image appearances vary significantly depending on the patient, even though all the scans were obtained using

Table 5.18: Dice scores for every vendor and fluid in the CHUSJ dataset, using the best models from Experiment 1 and Experiment 2.

Runs	Resized	Slices	IRF	SRF	PED	Fluid
Run 32	No	All	0.167	0.441	**0.141**	**0.209**
		Fluid	**0.510**	**0.541**	0.011	**0.317**
		No Fluid	0.142	0.380	**0.143**	**0.125**
Run 32	Yes	All	**0.191**	0.455	0.132	0.176
		Fluid	0.475	0.533	0.023	0.300
		No Fluid	**0.170**	0.408	0.134	0.078
Runs 42, 52, and 59	No	All	0.046	0.498	0.001	0.079
		Fluid	0.403	0.460	**0.054**	0.180
		No Fluid	0.019	0.521	0.000	0.000
Runs 42, 52, and 59	Yes	All	0.038	**0.563**	0.001	0.078
		Fluid	0.412	0.468	0.040	0.178
		No Fluid	0.009	**0.620**	0.000	0.000

the same device. This significanly difficults the segmentation by the model, trained on RETOUCH. In Figure 5.24, three B-scans from the CHUSJ dataset with really different characteristics from those of the RETOUCH images are shown. The left B-scan is well oriented, but the choroid region is very well-defined, as opposed to the RETOUCH images where this region is blurry. The middle and right B-scans present an odd orientation and are much noisier than the examples seen in the RETOUCH dataset.

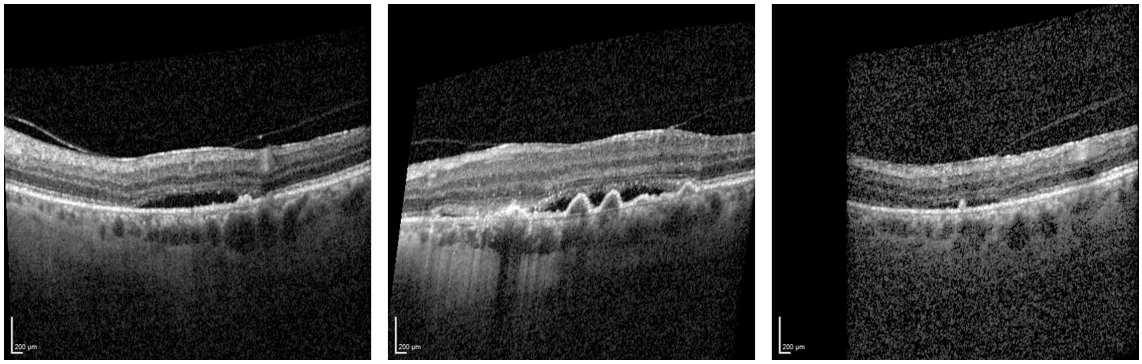


Figure 5.24: Three B-scans from the CHUSJ dataset with characteristics different from those in RETOUCH. The left scan shows definition on the choroid, while the middle and right scans present an odd orientation and significant noise.

The last problem when considering segmentation masks from different sources is the segmentation criteria used. In medical images, the segmentation performed by different professionals may result in different outcomes, as shown in Figure 5.25. Therefore, the criteria followed to segment fluid by the evaluator on the CHUSJ dataset may be different from the criteria followed in the RETOUCH dataset. Furthermore, considering that in RETOUCH, the approach to merge predictions by different evaluators is to consider both correct, the models trained in this data are prone to oversegment in unseen volumes.

An example of different segmentation criteria between the RETOUCH and CHUSJ datasets is seen in Figure 5.26, where visually and anatomically similar regions are segmented only in one dataset. While in manual segmentation the context from the surrounding slices is relevant, the visual similarity between regions is the most important characteristic in the segmentation using CNNs. Therefore, when visually similar regions are not segmented equally or following the same criteria in the training and testing data, the predicted segmentation masks for the testing volumes do not coincide with the GT.

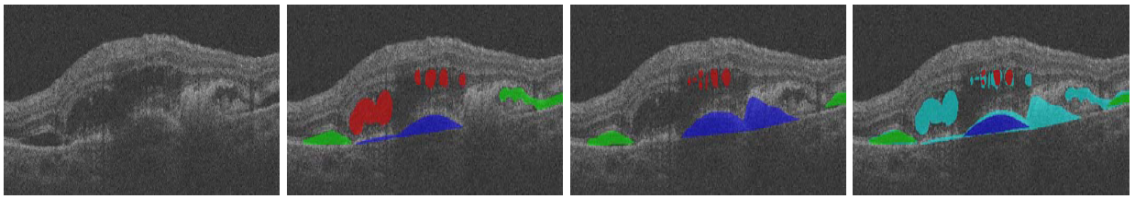


Figure 5.25: Significantly different segmentations (middle images) of the same B-scan (left image) by different evaluators in the RETOUCH dataset. IRF is represented in red, SRF in green, and PED in blue. The regions shown in cyan in the rightmost image represent the regions where the two evaluators diverge [60].

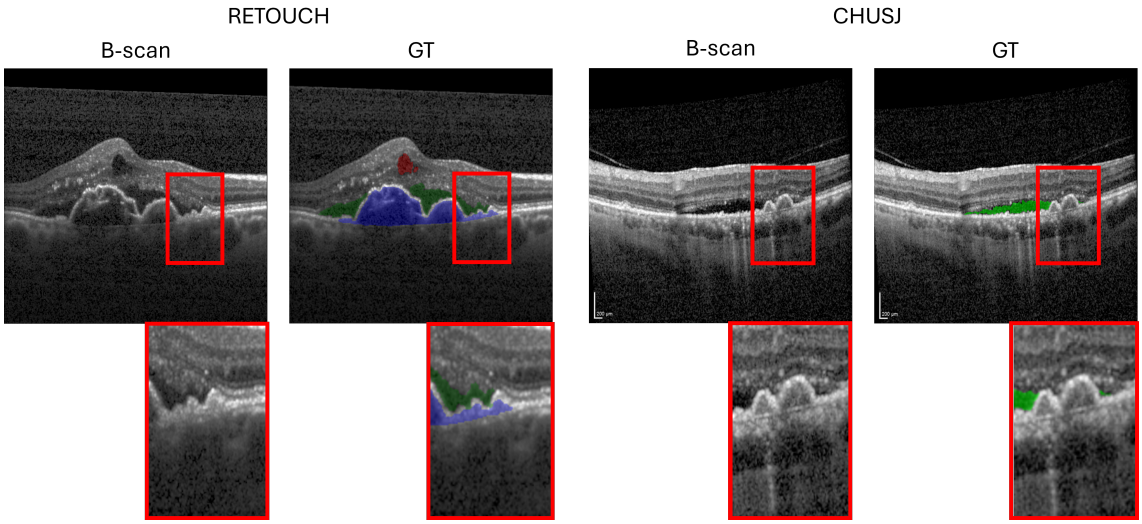


Figure 5.26: Visually similar PED regions are segmented in one dataset but not in the other. These inconsistencies can translate to lead to poorer model performances during testing. IRF is represented in red, SRF in green, and PED in blue.

The worst performing fluid segmented in this dataset is the PED. In the binary models, this fluid was incorrectly detected in almost every slice. Oversegmentation and false prediction of PED had already been observed in previous experiments, especially in the Spectralis volumes during Experiment 2. The segmentation in slices with PED also resulted in small Dice coefficients. While this is largely due to oversegmentations across multiple slices, it is also impacted by the small quantity of PED present across the entire dataset. However, it is important to note that many of the oversegmentations are in regions where the presence of PED could vary according to the applied

segmentation criteria. In Figure 5.27, two slices and their respective GT are shown, highlighting the segmentation of PED in a region where this fluid could arguably be present, along with a satisfying segmentation of SRF.

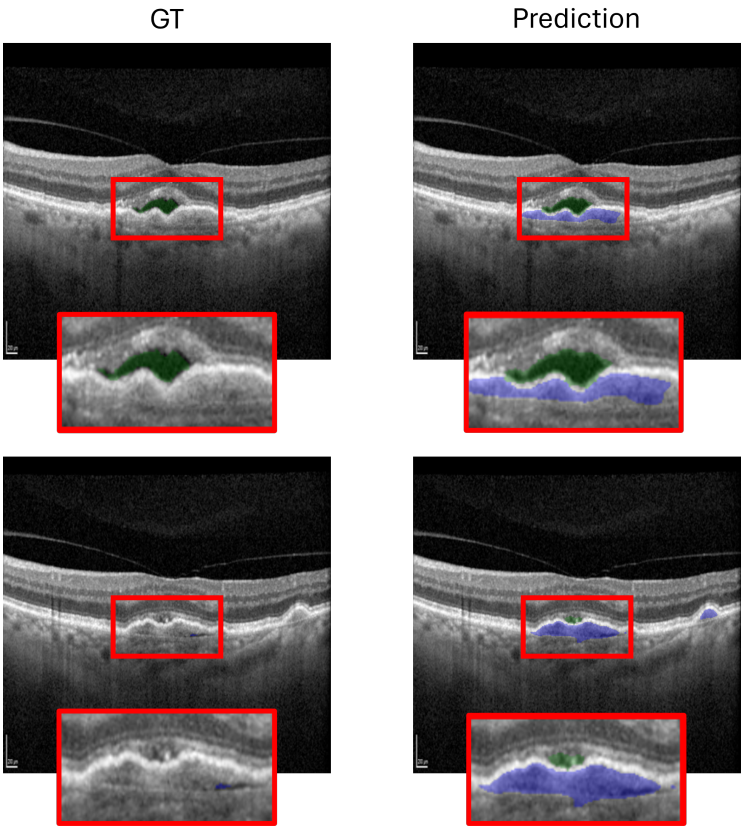


Figure 5.27: SRF (green) and PED (blue) segmentation in two OCT B-scans from CHUSJ. The regions segmented by the model as PED were not considered in the GT, but present characteristics similar to regions segmented as PED in the RETOUCH dataset (used for training).

The segmentation of SRF was the most consistent in this dataset. While the Dice coefficients are lower than those obtained in the RETOUCH dataset, this drop in performance was expected in the CHUSJ dataset, given its diversity of images and the different segmentation criteria compared to those applied in RETOUCH. In Figure 5.28, the segmentation of a B-scan with SRF is shown. While the segmentation by the model correctly identifies the SRF region, a slight oversegmentation occurs. This is seen across multiple B-scans, influenced by the data in which it was trained, resulting in a worse Dice score on the slices with fluid.

Similar to SRF, the IRF segmentation is also adequate in the slices that contain this fluid. Likewise, the model tends to slightly oversegment the fluid regions, as seen in Figure 5.29.

However, the main reason behind the poor Dice coefficients when considering all the slices is the fluid detection. In slices where the fluids are not present, particularly IRF and PED, they are still predicted outside the retina, usually in the choroid. While this is already visible in Figure 5.29, where all the fluids are segmented in the choroid, more examples are shown in Figure 5.30.

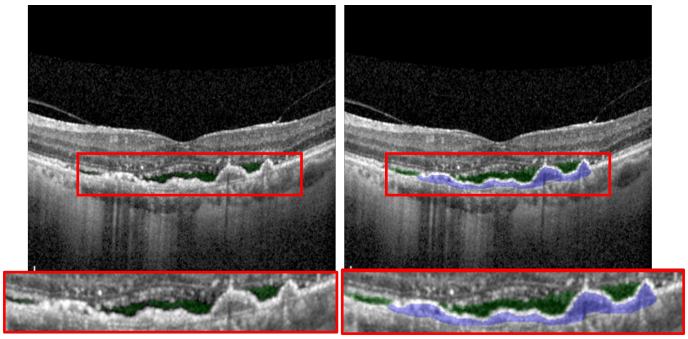


Figure 5.28: SRF (green) and PED (blue) segmentations (right) in a B-scan from CHUSJ, compared to its GT (left). This shows the model's tendency to oversegmentation.

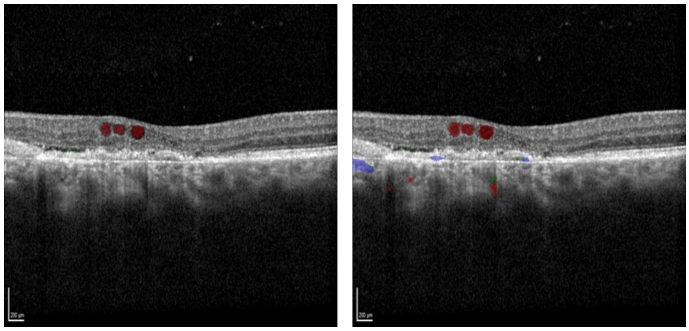


Figure 5.29: Segmentation of IRF regions (right) compared to its respective GT (left). The prediction of fluid regions in the choroid is also noticeable. IRF is represented in red, SRF in green, and PED in blue.

The incorrect segmentations in the choroid are caused by the different image quality in this dataset, as previously explained. When training the segmentation models with the RETOUCH dataset, the Spectralis scans are focused on the retinal layers, blurring the structures outside of it, including the choroid. In the B-scans which compose the CHUSJ dataset, this region is not blurred. This confuses the segmentation model which also interprets the choroid as retina, thus predicting the existence of fluid regions there. The fluid which is the most affected by this segmentation is the IRF, due to the resemblance with the structures and intensities present in this region. Therefore, if the region outside the retina was not considered in the final masks, the Dice coefficients would improve significantly. These segmentation errors are more prominent in the predictions performed by the binary segmentation models as they present weaker capabilities in understanding anatomical references, basing their predictions more on pixels intensities and thus being more easily fooled by the choroid.

The use of resized images or the original ones did not lead to a significant difference in segmentation, as the performances remained almost unchanged when varying the size of the slices. Since these transformations are applied along the horizontal axis, the different resolutions are harder to perceive, as they do not significantly change the perception of the structures in the B-scan.

Lastly, the influence of the noise and different image characteristics present in this dataset, as shown in the examples in Figure 5.24, is more significant outside the retina. In Figure 5.31, the

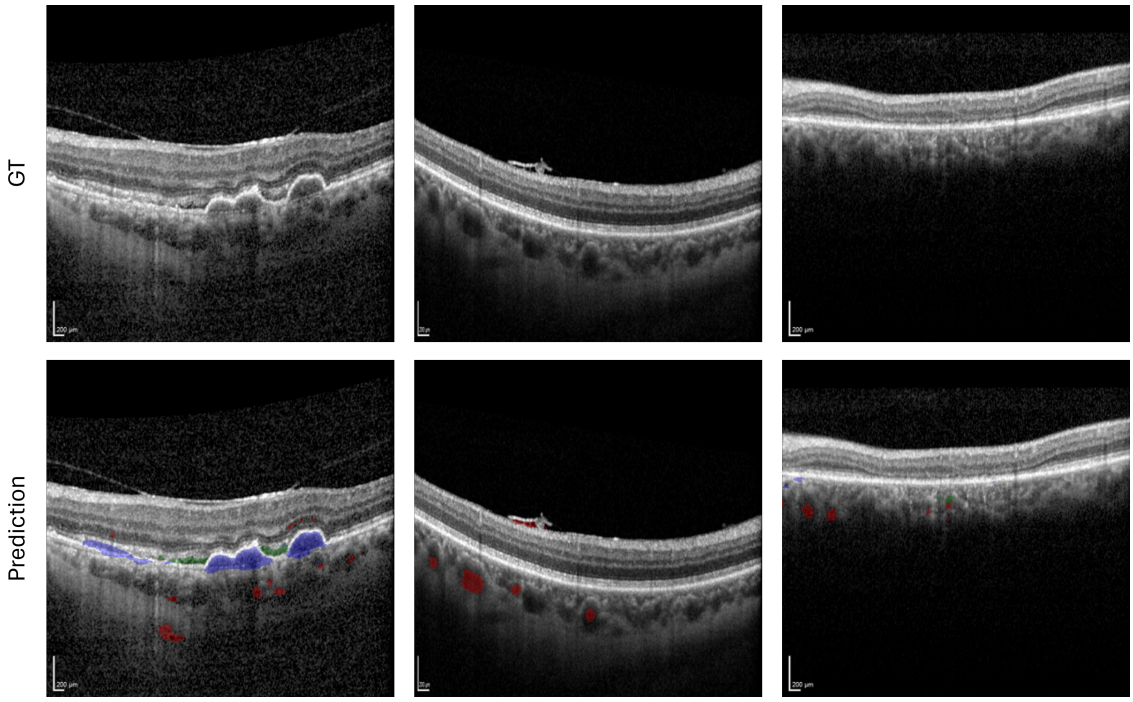


Figure 5.30: Segmentation of multiple fluids in the choroid region of the B-scans. It is a common problem and significantly reduces the Dice coefficients, as reported in Table 5.18. IRF is represented in red, SRF in green, and PED in blue.

segmentation of these noisy B-scans is shown, where a decent segmentation of SRF is observed, with a prediction of PED consistent with the training data. However, it is outside the retina, in the choroid, where the incorrect IRF segmentations occur.

After limiting the output to only the fluids predicted inside the retina, using the method developed by Melo et al. [66], new Dice coefficients were computed (Table 5.19). When compared to the results seen in Table 5.18, the impact of limiting the predicted masks to the retina becomes evident, corroborating the previous claims.

The Dice coefficients in slices with fluid are the least affected by the retinal delimitation. As previously stated, in slices with fluid, the segmentation model often identified the fluid regions and correctly segmented them, with small oversegmentations in the fluid region. However, the model also predicted fluid outside the retina, in the choroid, which reduced the Dice coefficient calculated in these slices. By removing these incorrect segmentations, the Dice coefficient associated with the slices increased, as the resulting predictions led to a more accurate segmentation.

The largest Dice coefficient improvements are seen in B-scans without fluid. By only keeping the segmentations within the retina, most of the oversegmentations that occurred in slices with no fluid were discarded, mainly in the choroid. This significantly improved the Dice coefficient, as even the smallest segmentation in slices with no fluid results in a Dice coefficient of 0.

As the CHUSJ dataset contains a low number of slices with fluid, a significant improvement in the Dice coefficient of the slices without fluid, results in a significant enhancement of the Dice coefficient when all slices are considered. This improvement, combined with the noticeable, but

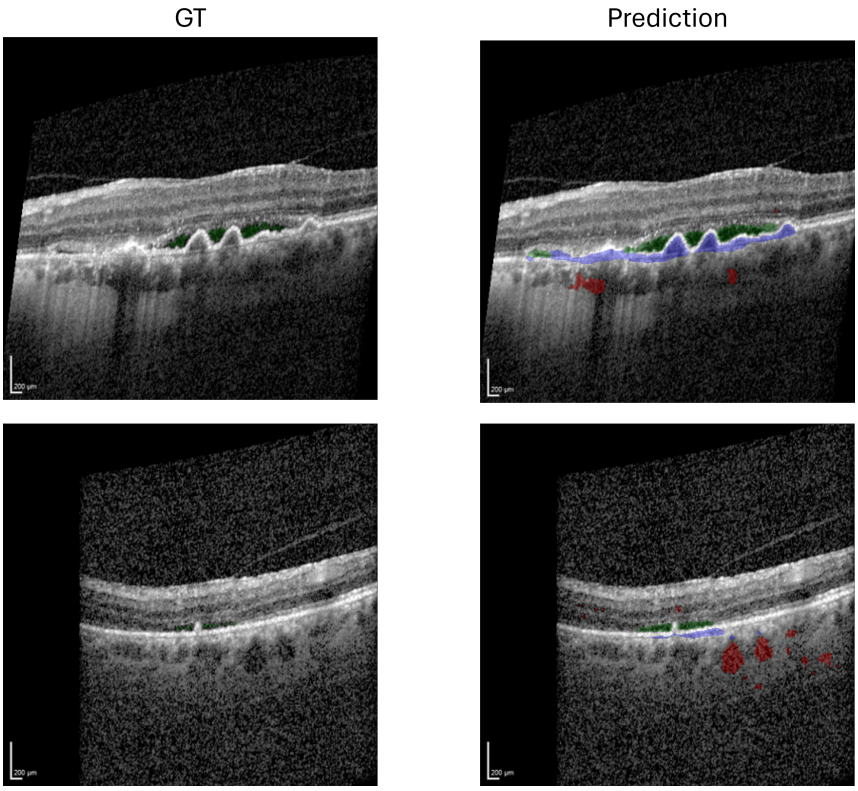


Figure 5.31: Fluid segmentations performed by the multi-class model in noisy B-scans from the CHUSJ dataset. IRF is represented in red, SRF in green, and PED in blue.

Table 5.19: Dice coefficients for every fluid in the CHUSJ dataset, using the best models from Experiment 1 and Experiment 2, after discarding the fluids predicted outside the retina.

Runs	Resized	Slices	IRF	SRF	PED	Fluid
Run 32	No	All	0.522	0.676	0.386	**0.399**
		Fluid	**0.694**	**0.559**	0.011	**0.409**
		No Fluid	0.509	0.746	0.393	**0.391**
Run 32	Yes	All	**0.530**	**0.681**	**0.404**	0.358
		Fluid	0.681	0.551	0.023	0.397
		No Fluid	**0.519**	**0.761**	**0.411**	0.328
Runs 42, 52, and 59	No	All	0.455	0.579	0.230	0.222
		Fluid	0.605	0.466	**0.096**	0.287
		No Fluid	0.443	0.648	0.232	0.172
Runs 42, 52, and 59	Yes	All	0.487	0.593	0.273	0.248
		Fluid	0.683	0.479	0.041	0.285
		No Fluid	0.472	0.662	0.277	0.219

smaller, increase of the Dice coefficient in the slices with fluid, leads to remarkable differences when comparing the results to those obtained without post-processing.

Most of the improvements brought by this post-processing are seen in the segmentation of IRF. This fluid was commonly segmented in the choroid by the segmentation model, more frequently

than any other fluid. Therefore, by filtering the content outside the retina, the Dice coefficient of the segmentation model shows significant enhancements both in slices with and without IRF.

In the segmentation of SRF, a remarkable notable is seen mainly in the slices without this fluid. The SRF segmentation was usually accurately performed within the retina whenever the slices contained this fluid, but some small oversegmentations in the choroid appeared regularly. Despite the predicted masks appearing in this region less frequently than IRF, the Dice coefficient of SRF was also heavily penalized by the oversegmentations. When comparing these results to those before processing, the Dice in slices without fluid improves significantly when the masks are processed. As a consequence, the Dice coefficient when considering all slices follows the same trend. This is due to the number of slices with SRF being much smaller than the number of slices without this fluid in this dataset.

The segmentation of PED is the least affected by the post-processing. It was previously stated that the segmentation of PED did not reach satisfying results in this dataset due to the discrepancies between these masks and the training masks, regarding the segmentation criteria. For that reason, the performance in slices with PED did not change significantly. In contrast, in slices without this fluid, the Dice coefficient improved noticeably, as the processing removed all the oversegmentations performed in the choroid region, a frequent problem in the predicted masks.

Lastly, the addition of this post-processing resulted in general improvements in Dice coefficient and segmentation quality across all fluids and models. Figure 5.32 shows some fluid masks before and after the post-processing, compared to their respective GT. In Figure 5.33, the processed masks predicted in the noisy B-scans exhibited in Figure 5.31 are shown.

5.2 Intermediate Slice Synthesis

This section presents the results obtained in the generation of intermediate B-scans between two consecutive real slices. In this section, the images generated using two different generative models are compared: the GAN and the U-Net.

The comparison between these different methods is mainly qualitative, as the quality and perception of the images are prioritized over the calculated metrics, since the realism of the images and the performance of segmentation models on the generated slices are significantly affected by the appearance of the images.

Nevertheless, the GANs trained on different folds are still compared through PSNR and SSIM, as a way to objectively select the best performing model for the 5.3 Fluid Volume Estimation experiments.

The comparison between these metrics and other articles in the literature is not possible. The only article of intermediate slice synthesis applied in OCT, developed by López-Varela et al. [14], does not evaluate the generated slices using these metrics. The articles in other imaging techniques such as CT and MRI have significantly different levels of noise and the values attained in these metrics in OCT are, therefore, very different.

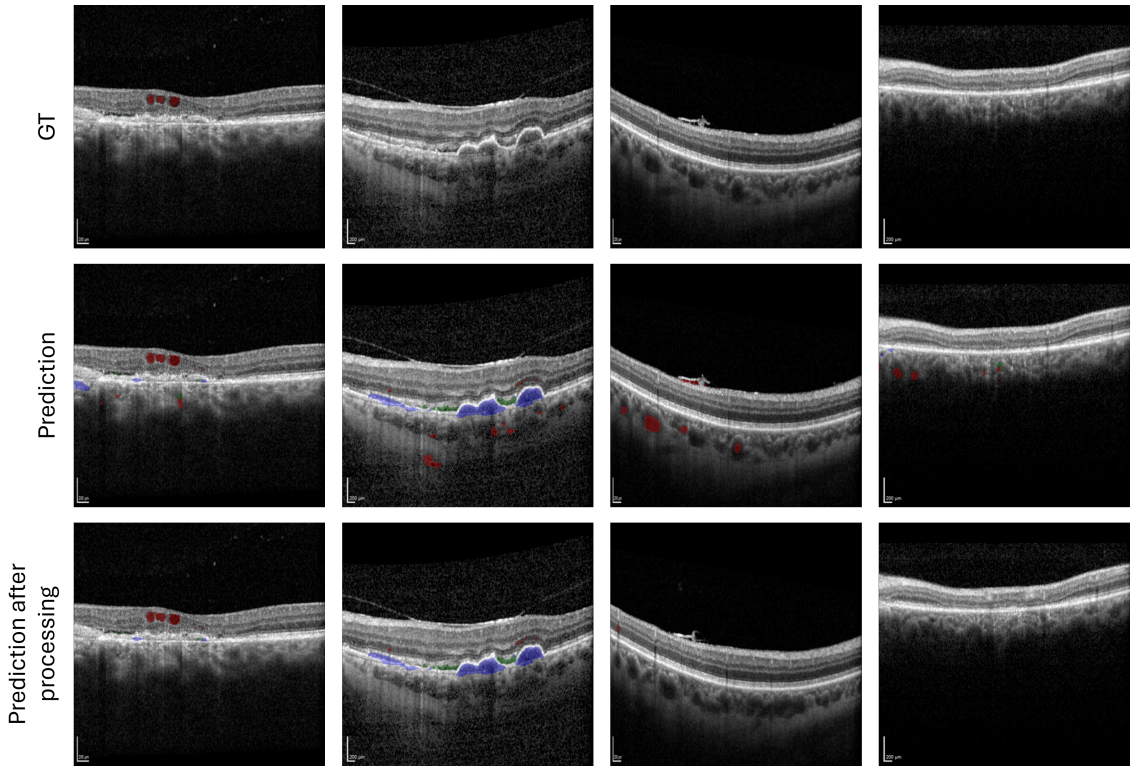


Figure 5.32: Processed fluid segmentations performed by the multi-class model in the examples of Figures 5.29 and 5.30. IRF is represented in red, SRF in green, and PED in blue.

5.2.1 Experiment 3 - Intermediate Slices Synthesis Using a GAN

In Experiment 3, a GAN was trained with the objective of generating the intermediate slice between a pair of two known and consecutive slices. In Table 5.20, the PSNR and SSIM are shown for each fold in which the network was trained.

One of the trends that becomes evident when looking at this table is the difference in results between devices, which is constant independently of the fold in which it was trained.

In every fold, the scans generated with the Spectralis device show better values both in PSNR and SSIM, while those obtained with Cirrus are significantly worse. This is tied to the characteristics of the images obtained with each device. In Spectralis, the inter-slice distance is larger than in Cirrus, but this is compensated by the better B-scan quality, contrast, and reduced levels of

Table 5.20: PSNR and SSIM values in every device, based on the generated slices in each validation fold.

Run	VF	PSNR				PSNR	SSIM				SSIM
		Cirrus	Spectralis	T-1000	T-2000		Cirrus	Spectralis	T-1000	T-2000	
Run 1	2	**20.64**	24.42	21.56	22.51	**21.92**	**0.193**	0.482	0.177	0.250	**0.260**
Run 2	3	20.44	24.58	21.63	22.24	21.69	0.181	0.484	0.188	0.246	0.247
Run 3	4	20.40	**24.93**	**21.95**	**22.56**	21.91	0.175	**0.498**	0.228	0.263	0.257
Run 4	0	20.29	24.75	21.92	22.51	21.72	0.180	0.482	**0.241**	**0.271**	0.258
Set 1	-	20.44	24.67	21.76	22.46	21.81	0.182	0.487	0.208	0.258	0.256

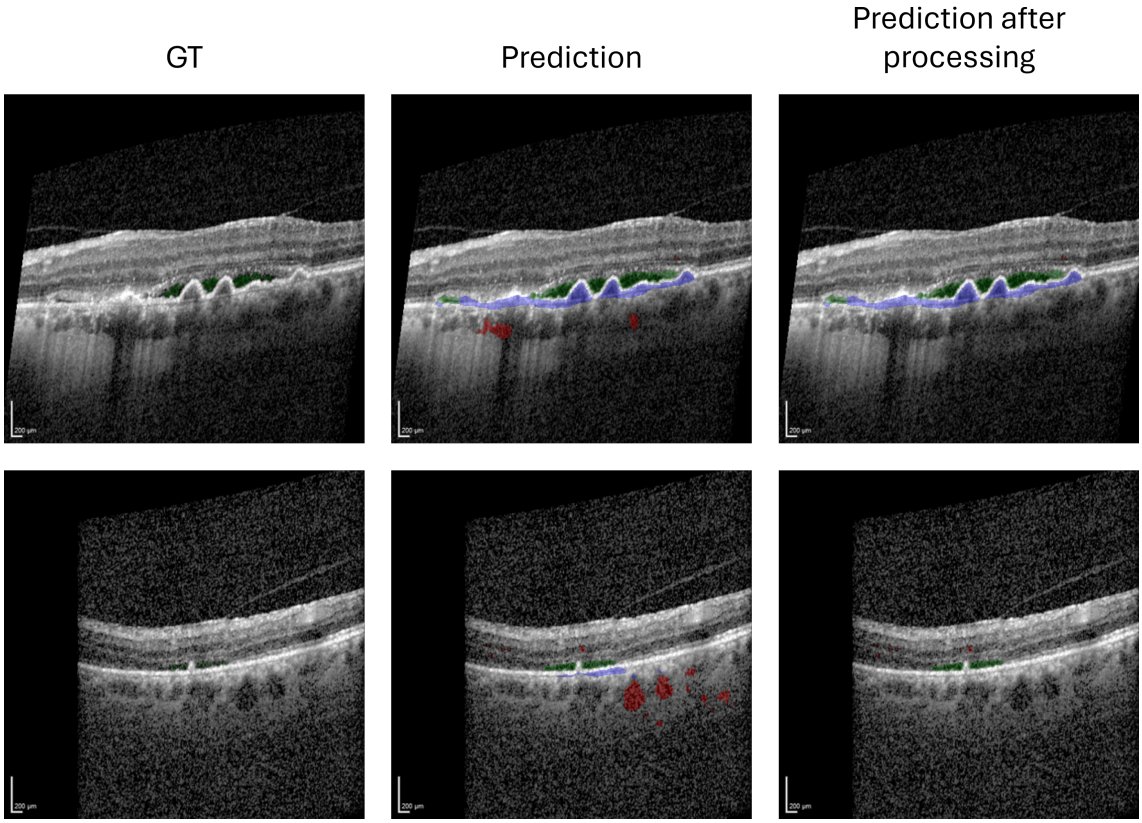


Figure 5.33: Processed fluid segmentations performed by the multi-class model, in the noisy scans shown in Figure 5.31. IRF is represented in red, SRF in green, and PED in blue.

noise. The opposite happens in Cirrus, as the B-scans are noisier and with a decent contrast, but separated by a smaller distance. In Topcon, the distance is similar to the Cirrus, but presents less noise with worse contrast.

As the speckle noise is hard to predict due to its random nature, the performance metrics are worse in the scans with more noise. These metrics compare the similarity between different images at a pixel level and in noisy regions, which occupy most of the B-scan, the differences at a pixel-level between generated and real images are significant, even if the differences are almost imperceptible.

In other imaging techniques like MRI and CT, the noise levels are much less significant than OCT, resulting in better PSNR and SSIM values, that directly translate to image quality.

When comparing the results across different runs, it turns out that the performance changes very slightly when the training data differs. Unlike the segmentation task, where the results in different validation folds were harshly affected by the fluid content present in the OCT volumes that compose them, the generation task appears indifferent to the random distribution of fluid.

In the segmentation task, the model and evaluation focused solely on the fluid, which is a small portion of the image, while in generation the model attempts to accurately generate the whole scan as similar to the target slice as possible. By considering the entirety of the B-scan, less of the performance is affected by the fluid region, which is why the distribution of fluid does

not affect the overall performance as much as it did in segmentation, even if it is the most difficult section to generate.

Despite the significant worse performance in Cirrus when compared to Spectralis, the visual similarity between generated images and real images, at an human level, is not very different. In Figure 5.34, two pairs of real and generated images from these devices are shown.

The generated slices capture the rough idea of the image, especially the retinal layers, but miss on the small details such as noise patterns, and brighter and unexpected structures. Overall, the generated Cirrus scan captures the details present in the real image better than the Spectralis. For example, the structures which appear in the choroid are similar to the original slice in the generated Cirrus slice, while in Spectralis these structures appear blurred. However, due to the highest levels of noise, this does not translate to better PSNR or SSIM.

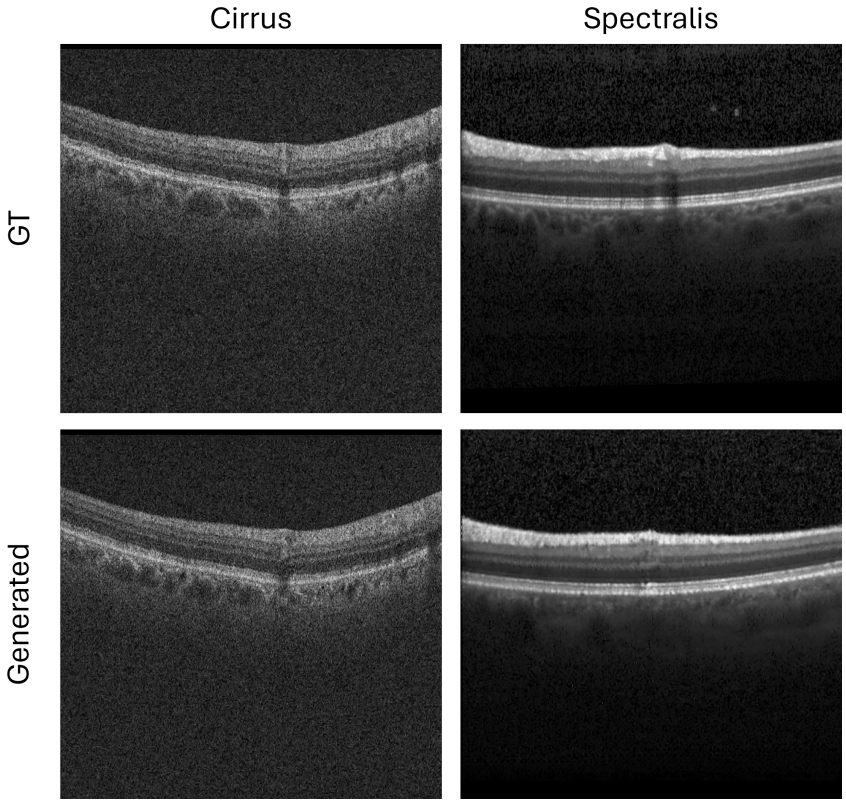


Figure 5.34: Example of two pairs of generated slices, one from Cirrus and one from Spectralis, which capture the retinal layers.

The differences in performance exhibited in this image regarding the representation of details are related with the inter-slice distance of the OCT volumes. In Cirrus, where the inter-slice distance is smaller than in Spectralis, the differences between the surrounding slices and the middle slice are smaller. Therefore, it is easier for the model to predict the content in the middle slice of a Cirrus volume than it is in Spectralis. Figure 5.35 shows the surrounding slices of the B-scans generated in Figure 5.34. It becomes evident the difference in the transitions between the surrounding slices and the intermediate slices. For example, in the choroid region, where the generative model

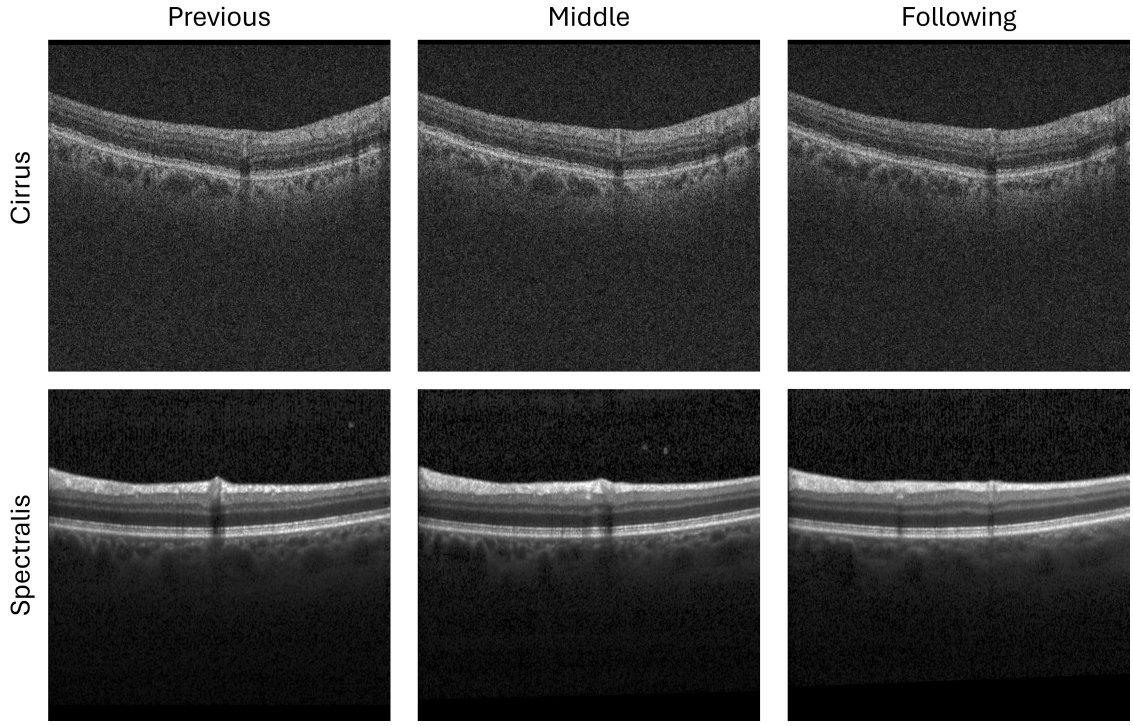


Figure 5.35: Surrounding slices of the B-scans generated in Figure 5.34.

output was blurred, the structures that appear in the previous slice are significantly different in the following slice. In the same region, the Cirrus B-scans present a much smoother transition, which results in an improved and more realistic output.

However, it is of our best interest to generate slices that accurately represent the fluid regions, on top of being visually similar to the original scans, as an accurate representation of the generated fluids, directly translates to a better performance in the segmentation of fluid in theses slices.

An accurate segmentation of fluid in generated B-scans is mainly dependent on the overall quality of the image and its resemblance to the images in which it was trained and the aspect of the generated fluid regions.

Similar to what was seen in other structures such as the choroid and the retinal layers, the quality of the generated fluid regions is dependent on the distance between scans more than the contrast and levels of noise in the image. However, the characteristics of each fluid also influence its generation, mainly because of their progression between consecutive slices. For example, the SRF usually appears in a more homogeneous and continuous shape, making it easier to predict its characteristics given the information of the surrounding slices. Contrasting, the IRF usually appears in small and irregular shapes, commonly separated by small retinal tissue. In consecutive slices, the small IRF regions quickly change its shape and may even merge together, which makes it harder to predict. The PED fluid can appear both as small inconsistent fluid like the IRF, but most commonly as a large and predictable region, similar to SRF.

An example of each generated fluid is seen in Figure 5.36, where the generation of IRF confuses the model while the fake SRF and PED appear similar to its GT. The characteristics of these

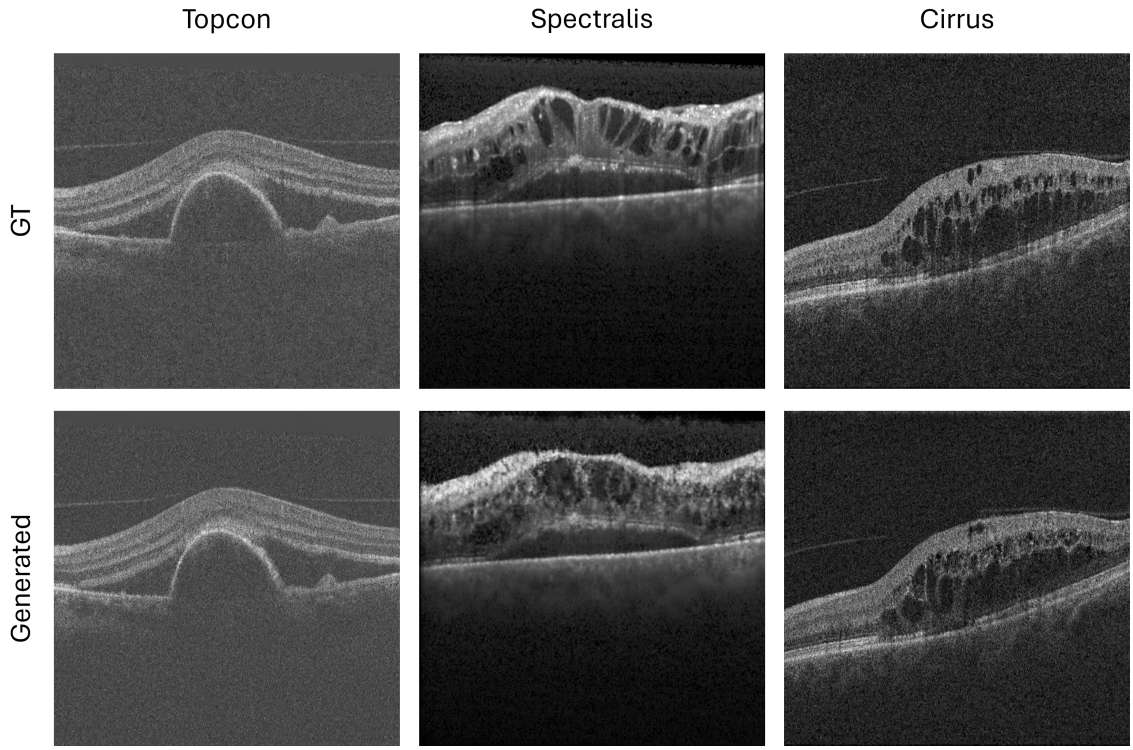


Figure 5.36: Examples of generated B-scans with fluid.

generated fluids are particularly important to understand the consequences that slice generation has in fluid volume estimation.

In this figure, it becomes evident that the generation of SRF and PED, seen in Topcon example, is a much simpler task than the generation of any other fluid. In fact, the overall structure of this region has been very accurately preserved, apart from small visual details.

The generation of IRF, however, appears more troublesome, particularly in Spectralis. In this last device, while in the GT the fluid regions appear individually separated by small retinal tissue, in the generated images the structures surrounding the fluid appear deformed, making it hard to understand which region is fluid and which is retina. It is interesting to note that the SRF shape stays consistent, highlighting that this fluid changes much slower as the slices progress than IRF. In Cirrus, the model preserves the IRF structures accurately, apart from the small and thin sections of retinal tissue which separate the fluid regions, where some merging occurs. This small details occur due to the target slice being located between one slice that contains this small tissues and another that does not. Nevertheless, the characteristics of this fluid are much more similar to the GT than in Spectralis, emphasizing the impact that the inter-slice distance has in slice generation.

The poor generation of IRF is also partially related with the loss used to train the GAN. Since this network was adapted from a video frame interpolation application, the method is not prepared for datasets that contain as much noise as OCT does. In the video use case, it makes sense to use a loss function that primarily evaluates the images at a pixel level, while being complemented by a perceptual component, which is the adversarial loss, that corresponds to the capability of a fake

image to fool the discriminator. However, in OCT, the speckle that corrupts the image makes it considerably more difficult to optimize a network to generate images that are similar at a pixel level, since this noise is impossible to predict. Therefore, as the network attempts to reproduce the noise that occupies most of the B-scans, it does not apply as much attention to the fluid as it should, particularly in IRF that needs it more. Ultimately, this results in visually similar images in the matters of noise and retinal layers, but worse in the generation of fluids.

The model trained on Run 3 was selected for generating intermediate slices in Experiment 6, as it achieved the best PSNR performance in three of the four devices utilized to acquire the B-scans. Additionally, its SSIM performance was comparable to those obtained in other runs.

5.2.2 Experiment 4 - Intermediate Slices Synthesis Using a U-Net

The second experiment in the scope of generating intermediate slices was performed using a U-Net and only a single model was trained. Examples of B-scans generated using this model are shown in Figure 5.37.

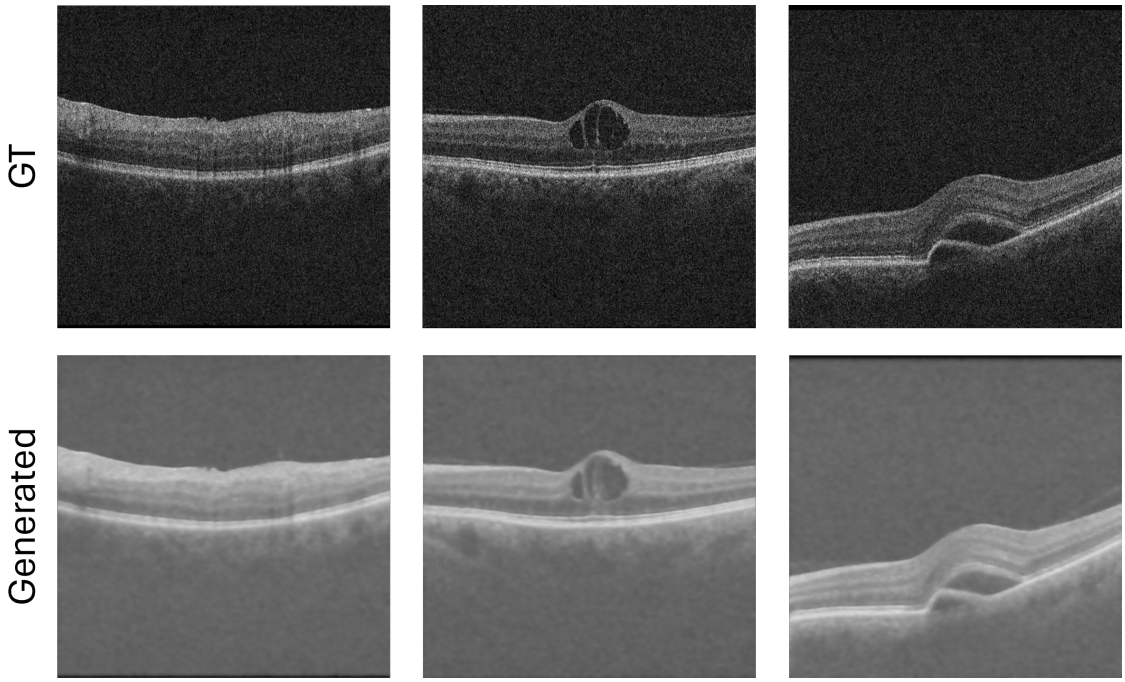


Figure 5.37: OCT B-scans generated using the U-Net. The images on the top row show the original B-scans while those on the bottom row are the corresponding generated slices.

When evaluating the quality of the generated images, a noticeable lack of realism is observed, as these images often resemble a smoothed approximation of the GT. Despite the similarity in architecture between the generator used in the GAN and the U-Net, the second model could not reach a performance comparative with the GAN. The U-Net faces three significant challenges: its loss is not representative, it is trained with large images, and does not have a discriminator.

The main problem comes from the network not being able to produce realistic images by reducing the loss. While the MAE is a good metric to compare two images, it does not perform

well as the loss of a generative model in OCT. The reason MAE is insufficient to motivate the model to properly generate images is because the easiest, fastest, and safest way the model has of reducing the MAE in training is by creating blurred images, especially in noisy regions of the image.

While these blurred images are not as realistic as the GT, the MAE is not as bad as it would be in case the model attempted to create images that are more visually similar to the real images. When dealing with OCT, a large portion of the image contains speckle noise, which is difficult to predict. Therefore, reaching results that are visually similar includes the representation of speckle noise in the intermediate slices, which results in a worse MAE.

It is due to the inability of regularizing a generative model alone that the MAE was combined with other losses in the previous experiment. While this loss is important to represent the direct similarity between two images, it fails to capture the perception of the structures and visual similarity between images. This is why the MAE loss was attributed such small weight in Experiment 3.

When comparing to the work by Nishimoto et al. [43], in which this experiment was inspired, the dataset presented almost no noise in most images. Therefore, the generative task was simpler and the model was able to generalize significantly better, as the MAE was enough to promote an accurate slice interpolation. However, when facing images with more noise, caused by the presence of metal artifacts in the CT, the generation of slices performed worse than the linear interpolation. Therefore, while this model generates good images when the slices have almost no noise present, it lacks the robustness to generate images where noise appears.

The use of a U-Net to generate intermediate B-scans is also harmed by the input it received. While in the GAN, the discriminator and the generator were trained with small patches of 64 $\times$ 64, the U-Net was trained with the entire image. Consequently, this motivates the U-Net to focus on the more general details of the image, rather than on the finer characteristics, as done in the GAN, ultimately contributing to the generation of less realistic images.

Lastly, this model lacks the regularization brought by an external network which is, in the case of a GAN, the discriminator. With the inclusion of a discriminator, a perceptual component is added to the loss. This motivates the model to produce the images that are not only similar to the pixel level, but also visually accurate. Therefore, this results in an image where the textures are more realistic, even if the difference at a pixel level is worse.

Nevertheless, it is seen in Figure 5.37 that the borders of the fluid regions are preserved. This representation is not as difficult as the other components of the B-scan, but is better in the generation of SRF and PED, where the fluid regions are more homogeneous. In IRF, the shape of the fluid is bounded by the deformations caused inside the retina, which makes it harder to generate, leading to the fluid regions appearing significantly different from what is expected. Furthermore, the goal of generating intermediate slices was not only to generate images that are similar in the fluid regions, but also in the overall image, and for this reason, no other generative U-Net model was trained. The accurate delimitation of the generated fluids does not directly translate to a good

segmentation performance in the created slices, since the image's overall different appearance could impact the segmentation model's performance.

5.3 Fluid Volume Estimation

The experiments discussed in this section relate to the estimation of fluid volume using the segmentation masks of each B-scan in an OCT volume. In both experiments, the fluid volumes predicted using the segmentation models are compared to the fluid volumes calculated using the GT masks from the dataset, in the reserved fold.

This fold had not been seen by any segmentation or generative model and the GT fluid masks are available for all OCT scans.

The discussion of these experiments aims to understand the applicability of enhancing the resolution of an OCT volume and how that affects the total fluid volume estimated, among other aspects such as the perception of the generated slices by the segmentation model.

The results are shown in tables where for each considered OCT volume, the fluid volume estimated for each fluid type (IRF, SRF, and PED) is presented in mm^3. At the end of each table, the MAE is shown to resume the differences between the GT fluid volumes and the prediction.

In Experiment 6, the Dice coefficient for the predict segmentation masks in fully generated OCT volumes is shown, allowing the comparison between the segmentation model's performance in real data and generated data, providing an insight on which fluids are affected the most the generation.

Lastly, scatter plots are shown for each comparison between predictions, where the each of the fluids' volumes calculated for each OCT scan are compared with each other, providing an intuitive and visual representation of the differences associated with the used data.

5.3.1 Experiment 5 - Fluid Volume Estimation Using Predicted Masks in Real OCT Volumes

In Experiment 5, the fluid volume was calculated for each type of fluid in the OCT volumes from the reserved fold, using the equation previously described in 3 Materials and Methods. The volume of each fluid, for each OCT volume in the reserved fold, is shown in Table 5.21, in mm^3. In Figure 5.38, the comparison between the values calculated using the GT masks and the masks predicted by the segmentation model is shown, using the values present in this table.

The values shown in this table reveal the similarity between the fluid volumes estimated with the RETOUCH GT masks and the masks predicted by the selected segmentation model. However, there are some outliers and evident trends seen.

For example, in IRF, there are multiple examples where the predicted quantity of fluid is significantly larger than the real quantity, such as in TRAIN002, TRAIN016, TRAIN039, and TRAIN048. These outliers appear mostly on Cirrus and Spectralis. In these vendors, the IRF presents more diverse shapes and deformity of the retina, which makes segmentation significantly harder than in Topcon, where the IRF appears much smaller and without deformation.

Table 5.21: Volume of each fluid calculated for the masks segmented in the RETOUCH (GT) and the masks segmented by the best segmentation model from Experiment 1 and 2 (P). The volumes correspond to the total fluid volume in the OCT scan, in mm^3.

Vendor	**Volume**	**IRF**		**SRF**		**PED**	
		GT	**P**	**GT**	**P**	**GT**	**P**
Cirrus	TRAIN002	0.228	0.373	0.008	0.013	0.000	0.000
Cirrus	TRAIN012	0.000	0.001	0.704	0.673	0.888	0.946
Cirrus	TRAIN014	0.000	0.000	0.517	0.575	0.332	0.252
Cirrus	TRAIN016	0.771	1.129	0.000	0.000	0.000	0.000
Spectralis	TRAIN027	0.937	1.071	0.096	0.132	0.000	0.001
Spectralis	TRAIN033	0.254	0.380	0.314	0.307	0.367	0.471
Spectralis	TRAIN039	0.622	0.839	0.000	0.000	0.000	0.000
Spectralis	TRAIN042	0.002	0.009	1.900	1.875	0.603	0.771
Spectralis	TRAIN048	0.328	0.446	0.000	0.008	0.000	0.000
T-2000	TRAIN062	0.047	0.058	0.000	0.000	0.000	0.000
T-1000	TRAIN063	0.000	0.011	0.747	0.568	0.034	0.110
T-1000	TRAIN064	0.058	0.060	0.000	0.003	0.000	0.075
T-2000	TRAIN067	0.188	0.170	0.000	0.000	0.000	0.000
T-2000	TRAIN068	0.406	0.550	0.004	0.006	0.000	0.008
MAE		0.092		0.025		0.041	

In SRF and PED, the differences between the true and predicted quantity are not so significant, with the largest discrepancies being the SRF and PED in TRAIN063. The shape of these fluids in the OCT volumes present in this fold are more consistent with those seen in training, hence resulting in a more accurate segmentation.

Overall, the volumes predicted in scans with considerable quantities of fluid tend to differ more with its GT. This happens due to the deformation in the retina that is associated with large quantities of fluid, making it harder to predict the segmentation masks.

Nonetheless, the fluid prediction in slices with little to no fluid tends to be accurate, highlighting the good specificity and segmentation of small fluids previously seen in the segmentation model.

Regarding the performances per vendor, there is no significant differences to note, as most of the differences noticeable in these results are related to the different characteristics of the patients in each OCT.

It is interesting to note that in discrepancies between the real volume and the predicted volume, the segmentation model tends to overpredict fluid, which is associated with the oversegmentation trend seen in the predictions performed by this model. This is an important aspect to note, as it is rooted to the segmentation criteria used in the training data, the RETOUCH dataset. This trend further justifies the oversegmentations seen by the model in the CHUSJ dataset.

In the following subsection, the results shown in Table 5.21 will be compared with the estimated fluid volumes obtained using only generated slices and in enhanced volumes, where between every known slice another slice is generated.

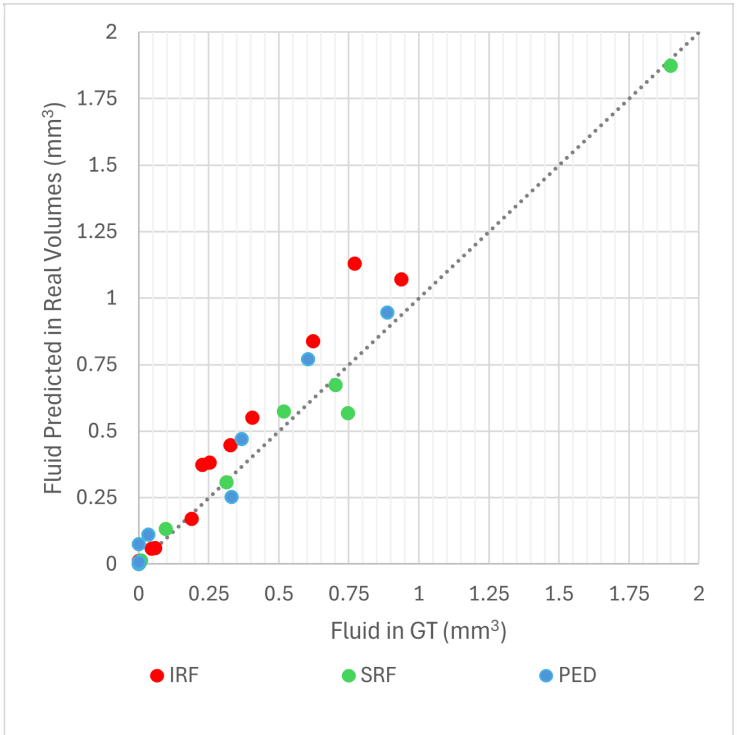


Figure 5.38: Volume estimated for each fluid in OCT volume using the masks predicted by the segmentation model compared with the volumes calculated using the GT masks. The OCT scans considered are the same as in 5.21 and the gray line marks the points in which the fluid volume in the GT is equal to the predicted.

5.3.2 Experiment 6 - Fluid Volume Estimation Using Predicted Masks in Generated and Super-resolved OCT Volumes

The results obtained regarding fluid estimation with generated slices, in the scope of Experiment 6, are shown in Table 5.22. In this table, four different fluid volume estimates are shown for each OCT volume in the reserved fold, separately for each fluid type. The first estimate, "GT", is obtained using the unchanged B-scans from each volume and the fluid masks provided in the RETOUCH dataset. The second estimate, "P", corresponds to the fluid masks predicted by the segmentation model in the unchanged B-scans of the same OCT volumes. Both predictions and fluid estimations were already presented in Experiment 5. In the third estimate, "G", the B-scans from the RETOUCH dataset are substituted with their corresponding generated B-scan, obtained using the selected GAN model. In these generated scans, the segmentation model predicts the fluid masks, from which the estimated volumes presented in this table are obtained. Lastly, in the fourth estimation "E", the original OCT volumes from the RETOUCH dataset are enhanced, as a B-scan is generated between every two consecutive slices. The segmentation model predicts the fluid masks both in the original and generated slices.

In the last three rows of the table, multiple MAE are calculated to compare the results obtained in the multiple columns. The first row, shows the MAE between the estimations in the column

"GT" and the estimations from the other columns. In the second row, the estimations performed using the segmentation model in column "P" are compared to the estimations made in the columns "G" and "E", while the third row compares the results obtained in these last two columns. Below each estimated volume in the columns "P", "G", and "E", except where no fluid is present in the GT, the relative error to the "GT" is shown.

Table 5.22: Volume of each fluid calculated for the masks segmented in the RETOUCH (GT) and for the masks predicted by the best segmentation model in the original RETOUCH images (P), in the generated slices (G), and when the original dataset is enhanced, combining alternating the real images with B-scans generated between them (E). The values correspond to the total fluid volume in the OCT volume, measured in mm^3, with the relative error to the GT shown in parentheses below each volume.

Vendor	**Volume**	**IRF**				**SRF**				**PED**			
		GT	**P**	**G**	**E**	**GT**	**P**	**G**	**E**	**GT**	**P**	**G**	**E**
Cirrus	TRAIN002	0.228	0.373 (64%)	0.356 (56%)	0.372 (64%)	0.008	0.013 (66%)	0.014 (77%)	0.013 (72%)	0.000	0.000 (–)	0.000 (–)	0.000 (–)
Cirrus	TRAIN012	0.000	0.001 (–)	0.001 (–)	0.001 (–)	0.704	0.673 (4%)	0.664 (6%)	0.670 (5%)	0.888	0.946 (7%)	0.939 (6%)	0.943 (6%)
Cirrus	TRAIN014	0.000	0.000 (–)	0.001 (–)	0.003 (–)	0.517	0.575 (11%)	0.561 (8%)	0.572 (11%)	0.332	0.252 (24%)	0.207 (38%)	0.227 (32%)
Cirrus	TRAIN016	0.771	1.129 (46%)	0.957 (24%)	1.050 (36%)	0.000	0.000 (–)	0.001 (–)	0.000 (–)	0.000	0.000 (–)	0.000 (–)	0.000 (–)
Spectralis	TRAIN027	0.937	1.071 (14%)	1.066 (14%)	1.063 (13%)	0.096	0.132 (37%)	0.136 (42%)	0.136 (41%)	0.000	0.001 (–)	0.002 (–)	0.002 (–)
Spectralis	TRAIN033	0.254	0.380 (50%)	0.414 (63%)	0.385 (52%)	0.314	0.307 (2%)	0.298 (5%)	0.320 (2%)	0.367	0.471 (28%)	0.397 (8%)	0.427 (16%)
Spectralis	TRAIN039	0.622	0.839 (35%)	0.726 (17%)	0.779 (25%)	0.000	0.000 (–)	0.002 (–)	0.000 (–)	0.000	0.000 (–)	0.000 (–)	0.000 (–)
Spectralis	TRAIN042	0.002	0.009 (325%)	0.017 (738%)	0.011 (430%)	1.900	1.875 (1%)	1.828 (4%)	1.852 (3%)	0.603	0.771 (28%)	0.822 (36%)	0.781 (29%)
Spectralis	TRAIN048	0.328	0.446 (36%)	0.336 (2%)	0.398 (21%)	0.000	0.008 (–)	0.013 (–)	0.009 (–)	0.000	0.000 (–)	0.001 (–)	0.001 (–)
T-2000	TRAIN062	0.047	0.058 (24%)	0.043 (9%)	0.051 (10%)	0.000	0.000 (–)	0.000 (–)	0.000 (–)	0.000	0.000 (–)	0.000 (–)	0.000 (–)
T-1000	TRAIN063	0.000	0.011 (–)	0.007 (–)	0.010 (–)	0.747	0.568 (24%)	0.507 (32%)	0.546 (27%)	0.034	0.110 (224%)	0.111 (227%)	0.108 (216%)
T-1000	TRAIN064	0.058	0.060 (4%)	0.061 (5%)	0.063 (9%)	0.000	0.003 (–)	0.025 (–)	0.008 (–)	0.000	0.075 (–)	0.215 (–)	0.114 (–)
T-2000	TRAIN067	0.188	0.170 (10%)	0.137 (27%)	0.160 (15%)	0.000	0.000 (–)	0.000 (–)	0.000 (–)	0.000	0.000 (–)	0.000 (–)	0.000 (–)
T-2000	TRAIN068	0.406	0.550 (35%)	0.543 (34%)	0.548 (35%)	0.004	0.006 (32%)	0.006 (37%)	0.006 (45%)	0.000	0.008 (–)	0.010 (–)	0.008 (–)
MAE compared to GT		-	0.092	0.067	0.079	-	0.025	0.036	0.029	-	0.041	0.052	0.043
MAE compared to P		-	-	0.037	0.016	-	-	0.012	0.005	-	-	0.023	0.009
MAE compared to G		-	-	-	0.022	-	-	-	0.009	-	-	-	0.014

In Figures 5.39, 5.40, and 5.41, the estimated volumes shown in Table 5.22 are plotted and compared with each other, providing a visual insight in the relationships between the estimated fluids, making the analysis easier. Figure 5.39 compares the volumes estimated using the GT masks with the volumes estimated in the other columns, "P", "G", and "E". Similarly, Figure 5.40 confronts the fluid volume estimations made using the fluid masks predicted by the segmentation model to the estimations made in columns "G" and "E". Finally, Figure 5.41 confronts the fluid volumes predicted in the generated dataset to the fluid volumes predicted in the enhanced dataset.

Considering the information exposed in the previous table and figures, many conclusions can be drawn regarding fluid volume estimation. For example, the differences between columns in

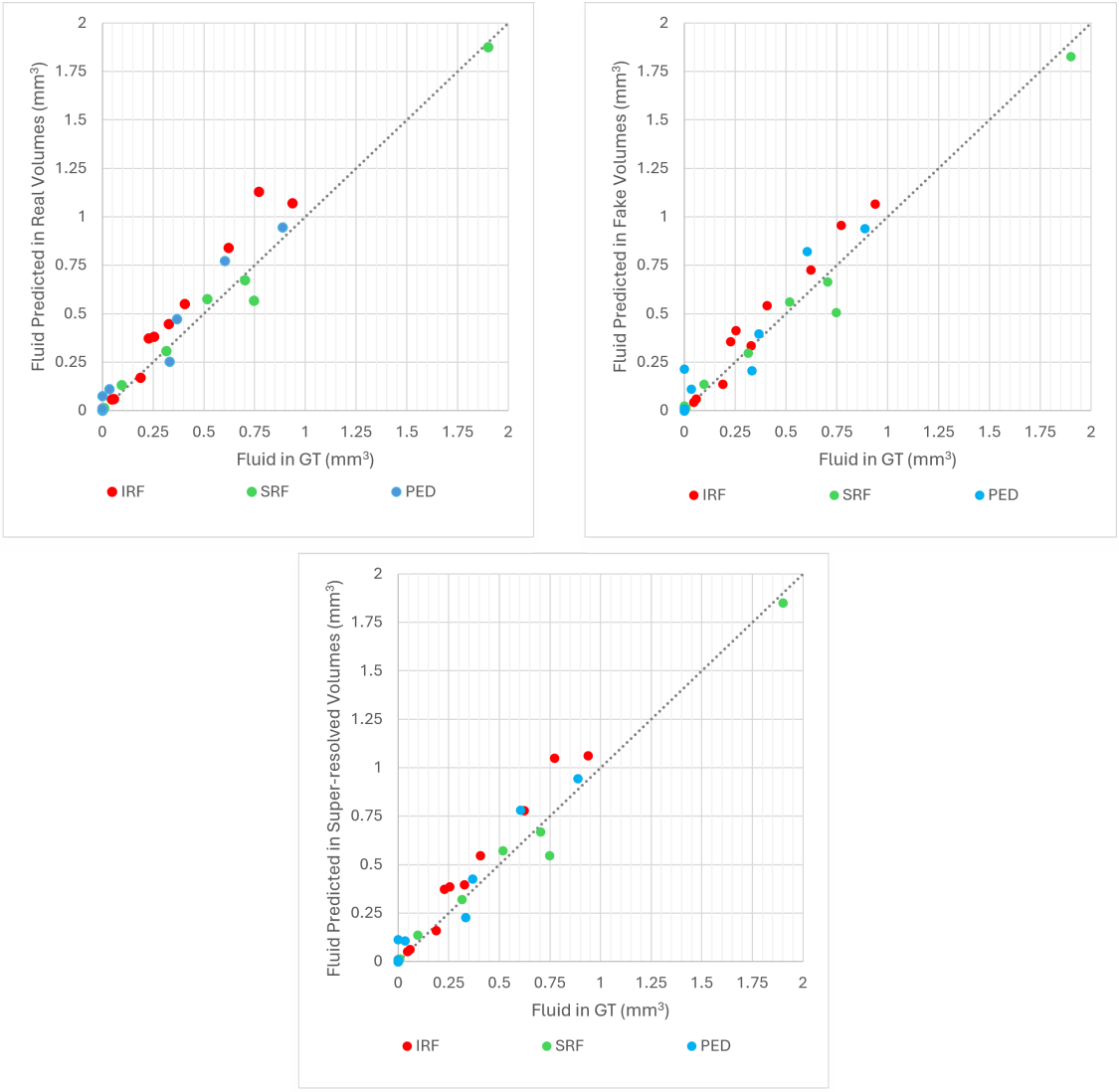


Figure 5.39: Volume estimated for each fluid in the OCT volumes using the GT masks, compared with the volumes estimated in columns "P", "G", and "E".

the estimated fluid volumes, are always larger in IRF. The generation of this fluid was seen in Experiment 3, to be the most difficult, due to the finer details that change in the transitions between consecutive B-scans. Combined with the worse quality in the generated slices, the segmentation of this fluid was also seen to be the most difficult fluid to segment in the volumes from this fold, as shown in Table 5.15, where the IRF Dice coefficient is lower than in any other fluid. Both aspects significantly influence the estimation of this fluid in the unseen data.

The SRF presents the smallest differences between columns, as the MAE in this fluid are smaller than in any other fluid. The fluid estimation of SRF benefits from the accurate representation of this fluid in the generated slices, due to its uniform, homogeneous, and predictable characteristics, as seen in Experiment 3. Contrary to IRF, the segmentation model performs exceptionally well in the segmentation of SRF, when compared to the other fluids. Therefore, an

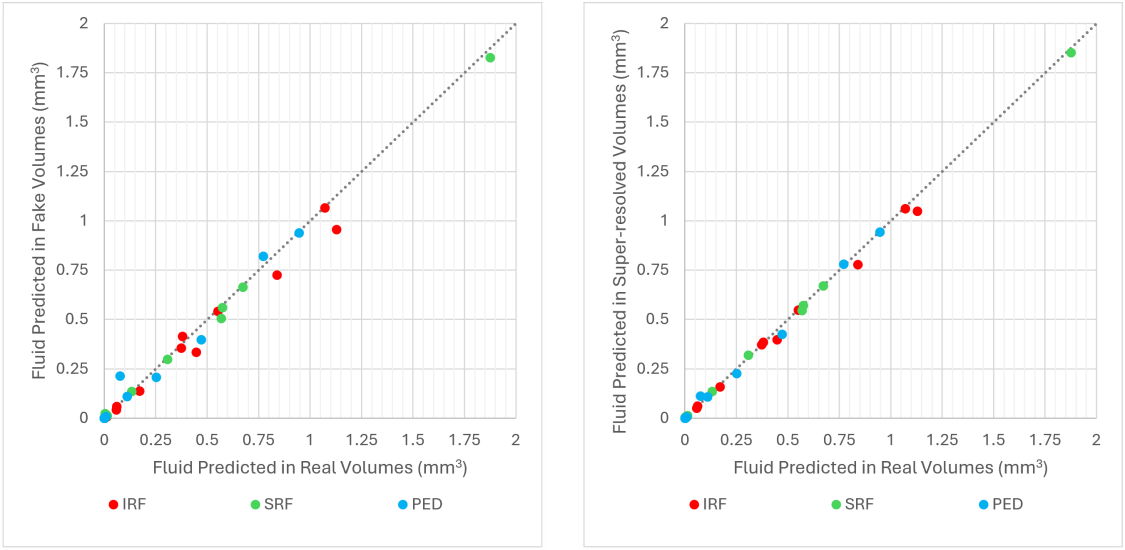


Figure 5.40: Volume estimated for each fluid in the OCT volumes using the masks predicted by the segmentation model compared with the volumes calculated using the masks from columns "G" and "E".

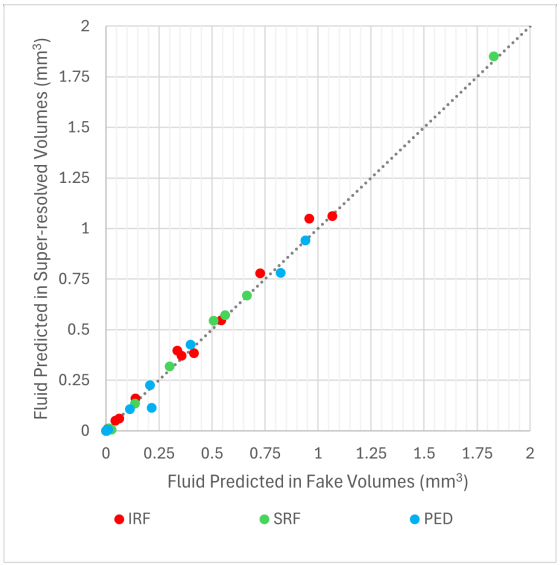


Figure 5.41: Volume estimated for each fluid in the OCT volumes using the masks predicted by the segmentation model in the generated dataset, compared with the volumes calculated in the enhanced dataset.

accurate representation of the SRF masks is obtained, leading to an accurate fluid estimation.

The estimation of the PED volume appears as an intermediate performance between the other two fluids. The segmentation of this fluid is better than IRF, but worse than SRF, while its representation in generated slices is accurate when presented in large regions, but not so good when appearing in small regions. This translates to a smaller deviation between columns when comparing to IRF, but not as good as SRF.

The differences between the columns “GT” and “P” are inherent to the capability of the segmentation model. This model’s capabilities also limit the accuracy of the estimated volumes in the generated and enhanced datasets, as the masks predicted in the generated slices are obtained using this model. Therefore, to understand the effect that generating slices has in total fluid volume estimation, it is important to observe the differences between the “P”, “G”, and “E” columns.

The differences between these columns are much smaller than the differences calculated when comparing these results to the GT. This indicates that the fluid estimation is more limited by the segmentation model, than it is by the generated slices. This is particularly true when comparing column “P” to column “E”, as illustrated in Figure 5.40. In this figure, most of the points appear close to the gray line, which indicates that the fluid volumes predicted when adding the generated slices are similar to the fluid volumes estimated when predicting the segmentation masks in the original RETOUCH dataset.

In fact, the largest relative errors shown in this table occur mainly in cases with smaller fluid volumes, where even small absolute differences translate into large relative errors. This explains why, for example, the IRF relative errors in TRAIN042 are so high despite the corresponding absolute difference being minimal. Moreover, the relative errors emphasize the close agreement between the predicted values in columns “P”, “G”, and “E”, which consistently show similar values. This consistency supports the previous observation that most of the fluid estimation errors arise from the prediction of fluid masks, rather than from generating intermediate slices.

To isolate the impact of the generated slices in the fluid estimation, it is more important to look at the differences between the fluids estimated in columns “P” and “G”, as well as Figures 5.40 and 5.41. By observing this data, it becomes evident the similarity between performances in these datasets. In the SRF fluid estimation, the values are particularly similar, as well as in most cases of PED. However, it is in IRF where the largest differences appear, where the predictions of IRF by the segmentation model are of less quantity in the generated dataset.

As seen in Experiment 3, IRF is the most difficult fluid to generate and attempts to recreate this fluid in intermediate slice, originating structures that do not resemble the IRF fluid. As a result, the segmentation models do not perceive this generated regions as IRF and do not segment them. This trend also seen in Figure 5.40 in the comparison between the original dataset and the enhanced dataset, where the largest differences appear in the segmentation of IRF, as the segmentation model predicts less fluid in the generated slices. However, since these B-scans are also mixed with original B-scans to form the OCT volume, the resulting differences are less significant.

It is interesting to note that of the three biggest differences in IRF between the columns “P” and “E”, TRAIN016, TRAIN039, and TRAIN048, which are highlighted in Figure 5.40 as the furthest points from the gray line, are in OCT volumes obtained with the Spectralis device. This trend is related with the larger inter-slice distance in Spectralis than in any other device, which makes it harder to create accurate IRF regions, which is amplified by this fluid’s characteristics.

In Table 5.23, the Dice coefficients is calculated for the predicted masks in the generated dataset. These predicted fluid masks are compared to the real GT masks of the equivalent slices. The information shown in this table helps the understanding of which fluids’ segmentation and

Table 5.23: Comparison in Dice scores for every vendor and fluid in the reserved fold (fold 1), using the best segmentation model to segment the original B-scans and their generated equivalent.

OCT Volumes	Slices	VF	Cirrus			Spectralis			Topcon			IRF	SRF	PED	Fluid
			IRF	SRF	PED	IRF	SRF	PED	IRF	SRF	PED				
Original	All	1	0.852	0.918	0.934	0.645	0.821	0.852	0.818	0.929	0.755	0.799	0.905	0.841	0.742
	Fluid	1	0.654	0.799	0.767	0.645	0.687	0.703	0.624	0.744	0.588	0.641	0.756	0.716	0.702
	No Fluid	1	0.941	0.972	0.978	0.646	0.889	0.885	0.884	0.960	0.766	0.873	0.952	0.862	0.786
Generated	All	1	0.740	0.896	0.903	0.307	0.707	0.850	0.763	0.870	0.726	0.671	0.850	0.816	0.627
	Fluid	1	0.431	0.762	0.710	0.315	0.580	0.627	0.457	0.670	0.538	0.407	0.692	0.656	0.542
	No Fluid	1	0.879	0.957	0.953	0.299	0.772	0.900	0.867	0.903	0.738	0.793	0.901	0.843	0.722

which vendors are the most affected by generating intermediate slices and how these slices are understood by the model.

As expected, the segmentation model performs better in the original dataset, where the structures present in the B-scans are the most similar to those seen in training. Nevertheless, in some metrics, the differences between the generated slices and the original slices is larger than in others.

Regarding the segmentation of slices without fluid, the segmentation model behaves almost similarly to what is seen in the original data in most fluids and vendors. However, there are some exceptions, particularly in IRF and slices from Spectralis. Of all fluids, it is in IRF where the detection of fluid changes the most. This is due to the quality of the generated scans in IRF regions: since this area is harder to predict, the model outputs fluid regions that visually do not resemble IRF, leading the segmentation model to not predict this region as fluid. This particularly affects Spectralis, where the distance between slices is larger, making it harder to interpolate an intermediate slice. For this same reasons, the detection of SRF is affected, particularly due to the poor generation of small SRF regions.

However, it is in the segmentation of slices with fluid where the performances significantly decrease. The segmentation of SRF and PED is not very affected by the generated slices in Cirrus and Topcon when compared to the other fluids. As seen in Experiment 3, the overall shape and characteristics of these fluids are easier to interpolate than IRF, as their structure is constant and homogeneous, with smooth transitions between consecutive slices, particularly in these vendors where the interslice distance is smaller.

Again, the Spectralis generated slices are the ones that result in the largest decrement of segmentation performance in IRF and SRF. As stated previously, the reason behind this poor segmentation is tied to the poor generation of slices when compared to other vendors, which is related to the distance between consecutive slices. The reasons that cause the poor segmentation of the slices with IRF and SRF are the same reasons that lead to the oversegmentation of these fluids in B-scans that do not contain the fluids.

One of the most noticeable outliers of the IRF volume estimation is the Spectralis volume TRAIN039. In Figure 5.42, the segmentation of IRF, as well as two generated slices, are shown.

It is visible in this figure that the second generated slice does not clearly represent the IRF region, as it appears slightly blurred and a large portion of the information does not resemble any structure present in the training images. For this reason, the segmentation model predicts a small area of fluid as IRF in this slice. The predicted area is much smaller than the area of IRF in the

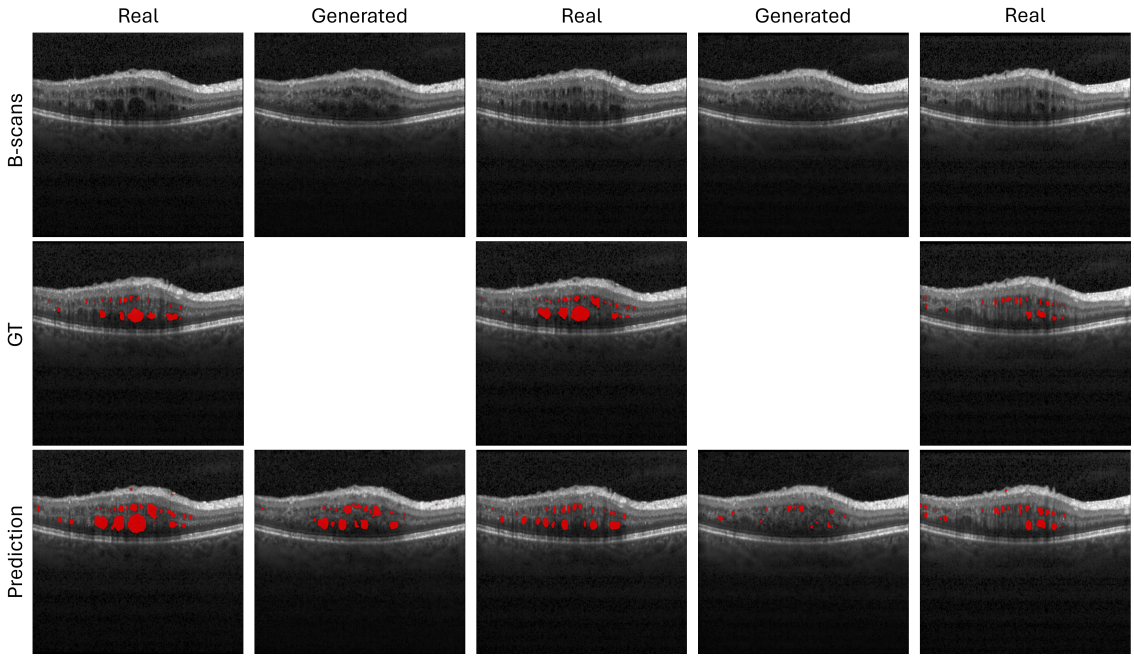


Figure 5.42: Example of IRF (red) segmentation in the enhanced volume TRAIN039.

surrounding slices. This indicates that the area segmented is not coherent with the information in the surrounding slices, leading to an incorrect predicted fluid volume. The same occurs in the first generated slice of the shown example, but only a smaller portion of the image is unintelligible.

These two different situations further highlight that the rough transitions particularly affect the slices with small fluid regions, as the model is not capable of predicting which of these regions will appear in the intermediate slice.

In the presented example, a single example of incorrect segmentation due to a poorly generated slice is shown, but multiple examples such as this appear in the predictions made in the mentioned enhanced volume. Accumulating this errors from multiple slices, results in a small total fluid volume than in the predictions made in the original dataset. This explains the decrease in total IRF segmented in the enhanced Spectralis volume when compared to the predictions in the original volume, as seen in Table 5.22.

Figure 5.43 shows the segmentation of large SRF and PED volumes in the enhanced Cirrus TRAIN012 volume.

In the shown generated images, the overall structure of the SRF and PED fluids is easier to accurately represent in the intermediate slices. Not only the transitions are smoother due to the smaller inter-slice distance in the Cirrus volumes, but also because of their homogeneous and regular shapes. Small artifacts that are seen in the GT slices are not represented in the intermediate slice, as they are unpredictable.

As the generated fluid regions show the appearance that is characteristic of the true fluid regions in the original scans that compose the RETOUCH dataset, its segmentation is visually similar to the segmentation performed in the surrounding slices, which resemble the GT masks of the

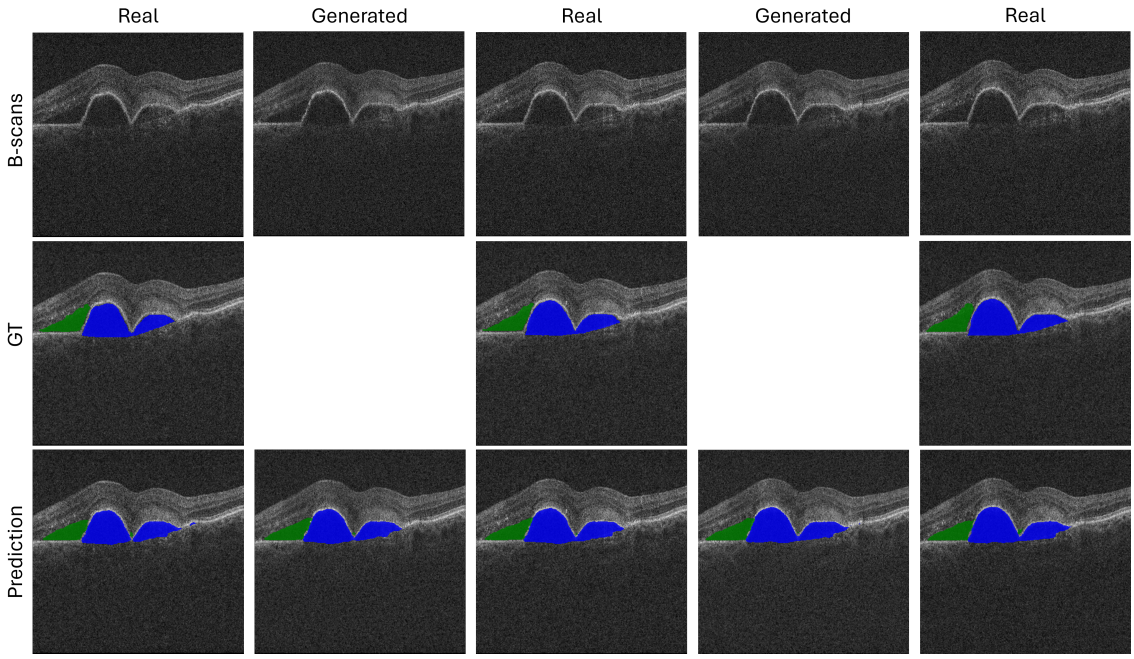


Figure 5.43: Example of SRF (green) and PED (blue) segmentation in the enhanced volume TRAIN012.

surrounding slices.

This example shows an accurate segmentation and generation of an enhanced OCT volume. This is a sample of segmentation and generation in the TRAIN012 volume, but most slices are segmented and generated as accurately as the shown example. This results in predicted fluid volumes that are similar to those estimated with the GT masks and with the predicted masks in the unchanged OCT volume.

Chapter 6

Conclusion

Throughout this dissertation, two main tasks were tackled in retinal OCT: fluid segmentation and intermediate slice generation. Theses tasks were explored with the goal of improving the automatic retinal fluid characterization, which includes the segmentation, identification, and quantification of IRF, SRF, and PED. This final chapter revisits the key findings and insights obtained from the experiments performed, presenting the strengths and limitations of the proposed methods.

In the first experiments, multiple approaches to multi-class segmentation were studied. The segmentation experiments started with a baseline U-Net, used in the multi-class segmentation of fluids. While still using this network, the input was changed, testing different patch sizes and extraction procedures, which provided an insight on how each affects the network performance.

When using smaller patches for training, extracted predominantly from the retina, the U-Net failed to understand the anatomical boundaries and relationships that define the retina, limited by the area captured in those patches. Therefore, the model would often segment outside the retina, and different fluids would be predicted in similar regions.

The use of larger patches, which occupied the majority of each B-scan, significantly improved the model's performance, but did not solve all the problems seen in smaller patches. While the model better understood the limits imposed by the retinal layers and learned how to leverage these landmarks in segmentation, the segmentation lacked detail in smaller regions. This is a common issue in models trained with the full image or larger patches: the model learns the broader structures better, such as the relationships between layers and fluids, but fails to capture the finer details, often merging small fluid regions.

During the extraction of these larger patches, the retinal layers would sometimes be split into two separate patches, which also affected the perception of the retinal layers, leading to segmentations outside the retina. Therefore, the models trained with larger patches lacked both the capability of accurately segmenting the fluid and a consistent understanding of the influence of retinal layers in segmentation.

As a way to find a compromise between the large patches, which lack detail, and the small patches, which fail to capture anatomical references, a different patch size was used. These patches were larger vertically than horizontally, capturing the whole height of the image, but small enough

along the horizontal direction so that the model can focus on finer details. Additionally, all images were resized to the same dimensions, ensuring that the same structures appeared consistently, regardless of the OCT device used for obtaining the B-scan.

With these implementations, the results significantly improved when compared to the previous patch sizes. The model understood the relationships between the anatomical landmarks in the retina and the different fluid regions, while still performing detailed segmentations in complex areas, with consistent performances across multiple fluids.

Still, new patch extraction methods were tested. Instead of dividing each image into four disjoint vertical patches, seven and thirteen overlapping patches were extracted. When training the model under these two conditions, the model learned significantly faster, reaching its maximum potential earlier than when trained with four patches. Since the number of images seen in each epoch increased, training progressed faster. While this did not translate to a performance increase when training with thirteen patches, the results obtained using seven patches were better.

Different rotations used in the transformation of the training patches were tested, with the goal of mitigating incorrect segmentations. In these cases, the fluids would appear in anatomically implausible positions. The model was tested with no rotation, 5° rotation, and 10° maximum rotation. The model's performance of the model significantly decreased without rotation, highlighting the importance of this transformation in OCT, where volumes often appear with varying retinal inclinations. With a maximum 5° rotation, the performance was slightly worse than with 10°, but it effectively mitigated the problematic segmentations. For this reason, the best model using 5° rotation was selected to perform the final inference on unseen data, including the reserved test fold from the RETOUCH dataset and the CHUSJ dataset.

In Experiment 2, a binary segmentation model based on U-Net was trained to segment each of the three fluids. The motivation behind this experiment was that by simplifying the task, the resulting segmentation might improve. However, this did not occur, as the classification of fluid in its different classes by a single model encouraged better learning of retinal anatomy and its relationship with the fluids. When the model task was changed from multi-class to binary segmentation, the model failed to understand this relationship as deeply as the multi-class segmentation model did. Instead, the model relied more on the pixel intensity to make its predictions. As a result, the model frequently oversegmented fluids, often outside the retina.

In this experiment, two different data splits were used for training each model. The first split was the one used for multi-class segmentation, while the second was specifically designed for each fluid being segmented.

The performances changed significantly depending on the split used for training. While the average Dice coefficients showed only small improvements, the performances presented less deviation across folds, since the data was distributed more uniformly. The models selected to segment unseen data were those that obtained the best segmentation performances, and all of them were trained using fluid-specific splits.

The first models were trained using the same loss as the multi-class models, which was a combination of Dice loss and cross-entropy. However, these binary segmentation models were

also implemented using just the cross-entropy. The results obtained in segmentation when using this loss were significantly worse, as the model kept predicting fluid in slices where it was not present.

The performances in the unseen RETOUCH volumes were satisfactory, as the results obtained with the multi-class segmentation model were better than the performance on validation, while those obtained with the binary segmentation models performed similarly to the validation, in general. These results highlight the models' robustness, since they had not been trained or validated on this data.

When comparing these two approaches with those in the literature and past works that were also trained and evaluated on RETOUCH, the results were satisfying. Among other disadvantages, the models implemented in these experiments are simpler than any of those to which they were compared. Yet, they achieved similar or comparable results with less training time and complexity.

In the CHUSJ dataset, the performance was worse than in the RETOUCH dataset. While the segmentation of the fluid regions was reasonable and all the fluids were correctly identified, the different characteristics of the dataset lead to a poor overall performance. The different image qualities, which highlighted the choroid region, led to frequent segmentation of IRF in this region. Meanwhile, the different criteria in the segmentation of PED resulted in multiple instances of oversegmentation.

When post-processing the predicted masks by limiting the fluid to the retinal region in the CHUSJ dataset, the model's performance improved significantly. This post-processing removed all the problematic oversegmentations in the choroid, increasing the Dice coefficient across all fluids. While IRF and SRF saw significant improvements in this metric in the slices with fluid, PED did not improve much. The segmentation of this fluid was primarily affected by the differing segmentation criteria used for image annotation in the RETOUCH and CHUSJ datasets.

Overall, the fluid segmentation experiments were pivotal for understanding of how the input size and the patch extraction method influence the segmentation of fluid in OCT, particularly through the observation of the models' learning process.

The first experiment in intermediate slice generation evaluated the use of a GAN for generating a single B-scan between two consecutive real B-scans. The results from the GAN were visually very similar to the real slices, as the generated retinal layers, speckle noise, and some fluid regions were really similar to the examples it had learned from.

At a pixel level, the metrics used in the comparison between images revealed significant differences. However, these values were highly influenced by the presence of speckle noise in the images, which is difficult to predict due to its random nature. Therefore, an analysis between generated images must always be complemented with a perceptual loss assessment and visual inspection, and visually, the images were really similar.

However, the generation of these intermediate slices is highly dependent on the characteristics of the inputs. For example, the intermediate slices in OCT volumes obtained using the Cirrus device were easier to generate, as the distance between two consecutive slices was significantly smaller than in Spectralis. In the scans obtained with this last vendor, the distance between slices

is larger, which resulted in rougher transitions between consecutive scans. This led to a poorer quality in the generated images, as the information was harder to predict.

The larger distance between slices seen in Spectralis is compensated by the lower noise levels present in the scans obtained with this device when compared to others. This leads to better metrics that compare the images at a pixel level, such as PSNR and SSIM, even if it does not equate to more realistic images.

The characteristics of each fluid also affect the quality of these regions in generated slices. For example, the generation of SRF, which is a fluid that usually appears as an homogeneous and well defined area that occupies a large portion of the retina, is easier to generate between slices, as its visual appearance does not change quickly in consecutive scans, and the modifications that occur in this transition are easy to predict.

The generation of PED regions in intermediate slices, follows a similar pattern to that of SRF, as most of the PED regions are well-defined and of considerable size. Nevertheless, there are some cases of small PED where generation is complicated, as the boundaries of the fluid tend to be more irregular.

Generating IRF in an intermediate slice is the most difficult part of slice generation. This fluid usually appears in small regions with irregular borders, which are separated by thin retinal tissue structures. Between two consecutive B-scans, even in the OCT volumes obtained with the devices with the highest inter-slice resolution, these structures rapidly change their shape and visual appearance. This makes it hard to predict the borders, size, and even if the regions are merged in the intermediate slice. This issue particularly affected the Spectralis volumes, as the larger inter-slice distance made the transitions between scans rougher and the prediction of intermediate slices more difficult.

Visually, the generated IRF regions did not look natural, and some were confusing, not exhibiting the characteristics that define this fluid. Moreover, the GAN did not improve on the generation of IRF, as the network's loss components mainly targeted the differences between generated and real images at a pixel level. While this worked in the dataset from which the GAN was adapted, it does not work as well in OCT, where the images are corrupted with high levels of speckle noise. In these conditions, the model optimizes to generate the noise which is, at an human level, almost imperceptible, while not improving as much the harder regions to generate that are actually understood by the human eye, such as the IRF regions.

The results from Experiment 4 reveal the necessity of using a more complex approach to the generation task, in OCT. While the methods used in this experiment were enough to produce satisfying results in the original article, since the CT dataset used was less noisy, they are not robust enough to be applied in OCT.

The capability of the U-Net to generate images is fundamentally constrained by the loss that regularizes the training. When using the MAE loss, the network can only compare the generated slices with their GT at a pixel level, without considering the general perception of the image. As a result, the U-Net produces blurry images, particularly in the regions with noise and uncertainty.

In these regions, since predicting the exact noise pattern is impossible, the network learns to average the possible variations, smoothing the final output. This minimizes the MAE, at the cost of sharpness and realism.

Since the speckle noise corrupts every region of the B-scans, the entirety of the output is blurred. This issue is mitigated by the addition of an adversarial loss in the GAN used in Experiment 3. The adversarial loss, which corresponds to the ability to fool the discriminator network, enhances the perception of the image and promotes the representation of noise in the generated images, making them more similar to the real outputs. This highlights the need for a more complex and robust approach to the generation of OCT images, where images are significantly affected by noise.

The use of the entire image instead of smaller patches for training, as done in Experiment 3, also contributes to the model generating images with less detail. The use of smaller patches motivates the network to focus on local structures more than the overall scan, resulting in images richer in detail.

In Experiment 5, where the volumes of each fluid in the OCT scans that compose the reserved fold were calculated, conclusions were drawn that supported the previous statements.

While the calculated volumes in the predicted masks were similar to those in the GT without presenting any extreme outlier, the model exhibited consistent oversegmentation. In most OCT volumes and particularly in IRF, the model segmented more fluid than there was in the GT. This oversegmentation is highly related to the RETOUCH dataset, in which it was trained, which encourages oversegmentation, as seen in the fluid segmentation experiments.

However, the oversegmentation in IRF is not only attributed to the dataset, but also to the model's capabilities. While the shape and intensity of the SRF and PED in the training data is similar to those seen in the reserved fold as these characteristics are more consistent in these fluids, IRF presents diverse shapes and surroundings across the OCT B-scans, making it harder to generalize.

For the same reason, the fluid estimation is also more accurate in Topcon. In the scans obtained with devices from this vendor, the patients present less retinal deformation associated with the presence of fluid. Therefore, the model generalizes better, as the fluids seen in these scans tend to be more similar to the majority of those seen during training.

In the last experiment, the fluid volumes were also estimated for OCT volumes generated or enhanced by the GAN from Experiment 3. The limitations that were faced by the segmentation model in Experiment 5, were also observed in this experiment. The segmentation model did not perform as well in IRF as it did in other fluids, especially in the generated scans. However, the limiting factor in this experiment was not only the segmentation model, but mainly the generated slices that contained IRF. The generative model was not capable of representing the IRF regions as accurately as it represented the other fluid types, which was caused by the irregular and less predictive transitions between slices seen in IRF. This, combined with the segmentation model, which performed slightly worse in this fluid, resulted in some outliers regarding the IRF volume estimation in the generated slices.

Apart from these occurrences, which mainly occurred in Spectralis due to its larger inter-slice distance, the estimated fluid volumes in the enhanced OCT volumes were very similar to those in original slices. In fact, the differences between the fluid volume estimated in the enhanced dataset and the original dataset with the GT masks, were mainly associated with the performance of the segmentation model, rather than the outcomes from the generative model.

Therefore, the conclusions drawn in these experiments demonstrate the strong potential of intermediate slice synthesis in OCT for generating anatomically coherent and structurally plausible images that support other tasks such as retinal fluid segmentation and, consecutively, retinal fluid volume estimation. Nevertheless, the methods applied can still be improved, such as the current lack of fine detail both in segmentation and inter-slice generation, which especially harms the representation of IRF. With further refinements, which can include the use of more advanced networks and better loss functions, this approach represents a promising solution for improving the confidence in the estimation of fluid volumes.

With the work developed throughout this dissertation, we aimed to study the applicability of state-of-the-art approaches for the segmentation, generation, and volume estimation of the retinal fluids in retinal OCT images. By methodically performing experiments that allowed us to draw the conclusions shown, highlighting the strengths and weaknesses of this methodology, we aim to provide the support needed for studies that seek to improve the results presented here. In the following subsections, the main limitations of this study are summarized and future adjustments to the applied methods are recommended, which we believe will improve the performances obtained, promoting a faster progress in the characterization of retinal fluids in OCT.

6.1 Limitations

Despite the promising results obtained in both segmentation and generative tasks, several limitations were associated with the methodology selected for this work, as choices were made in accordance with the available time and hardware for training the models.

In the segmentation task, the experiments performed utilized the baseline U-Net architecture, which was selected for its reliability and widespread adoption. As seen in the literature review, other methods have been developed to tackle the segmentation problems previously shown, such as incorrect segmentation of complex structures and detection of fluid beyond the retina. Recent studies approached these problems by including enhancements such as attention mechanisms, transformer-based modules, and multi-scale feature integration. However, due to the time constraints, the difficult access to these developed solutions, and the limited computational resources, these more complex approaches were not considered.

Similarly, three-dimensional convolutional networks were not considered for the segmentation of the retinal fluids. Despite the accessibility to three-dimensional networks such as the 3D U-Net, the increased computational resources required, including memory and training time, and the necessity for larger datasets made their application unfeasible.

The same limitations applied to the generative models. When training a GAN, there are two networks being trained simultaneously, which increases the duration of training. This, combined with the increase in the available data, resulted in longer training times for the GANs compared to the segmentation models, which constrained the selection of the model used for generating intermediate slices. Despite the trend to incorporate different modules and even different networks in the generative framework as seen in the literature, the limitations in time, computational resources, and accessibility to these frameworks led to the implementation of a simpler model.

Since the calculation of the fluid volumes is inherently dependent on the quality of the segmentation masks, in Experiment 5, and also the quality of the generated slices, in Experiment 6, the obtained results in fluid volume estimation are also conditioned by the selection of the segmentation and generative models.

The segmentation model also exhibited poor generalization on the unseen CHUSJ dataset, which can be largely attributed to the characteristics of the training data. Since the model was trained exclusively on the RETOUCH dataset, its segmentation performance was optimized for images following specific acquisition protocols and visual characteristics.

In RETOUCH, a benchmark dataset in the segmentation of retinal fluid, the images obtained with the same device follow the same imaging protocol and, therefore, present similar visual characteristics between each other, except for the patient-specific details. Consequently, when presented with images acquired using different protocols, such as those in the CHUSJ dataset, the model struggles to adapt to features that differ from those it was trained on, such as varying noise levels and differences in choroidal structure appearance. This led to multiple incorrect fluid segmentations.

The generation of slices using the GAN also lacked attention to finer details. The networks were driven to optimize the accurate representation of speckle noise more than the accurate representation of retinal structures, including retinal fluids.

The network was motivated to generate similar images at a pixel level because its loss was mainly composed of parts that compare the real and generated images at this level. In fact, only the adversarial loss function was a perceptual loss. Ultimately, this resulted in generated images where speckle noise was accurately represented, but the finer retinal details were poorly perceived by the observer. This led to a poor generation of IRF, the fluid that appears in smaller and more irregular regions, which requires, therefore, more attention to be accurately synthesized.

The segmentation of these regions was affected by the low quality of generated images, as the segmentation model also struggled to interpret the characteristics depicted in the generated scans.

Ultimately, this affects the fluid estimation results, since this estimation is only as good as the quality of the segmentation masks. Therefore, to improve fluid volume estimation in enhanced datasets, it is pivotal that both the segmentation and generative models produce realistic outputs, and this can only be achieved if they are more attentive to finer details.

6.2 Future Research

Building upon the limitations presented in the last section, there are several improvements that could be considered for future work, with the goal of enhancing fluid segmentation, slice generation, and, consequently, fluid volume estimation.

The experiments performed in this dissertation used state-of-the-art methods combined with innovative input strategies, producing good outputs in the rough delimitation and classification of fluid. However, three main issues were detected: oversegmentation outside the retina, poor attention to detail in smaller fluids, and weak generalization to different datasets, which leave room for further improvement.

The segmentation beyond the retina has been addressed in multiple ways in the literature. One of the most common approaches is limiting the input to only contain the information within it. Some studies go further and instead of just limiting the input, they provide a relative distance map, which helps not only with oversegmentation, but also with misclassification of pixels. For these reasons, the use of a relative distance map would significantly enhance the model's robustness and generalization, improving the overall Dice scores in segmentation. The use of retinal delimitation in the CHUSJ dataset as a post-processing method significantly improved the segmentation results, further motivating research in this area.

However, this does not improve segmentation details in small structures, which is a much more complex problem. The use of a single multi-class segmentation model makes it harder to tackle this, since it needs both global and local features to perform the segmentation. For example, when the U-Net was trained with small patches, the model did not understand the influence of the larger structures in the segmentation. The opposite happened when using larger patches, as the model did not segment the finer details correctly.

To tackle this problem, a single U-Net is not enough, especially considering that some fluids are better segmented with networks of different depth, based on their varying shapes and sizes. Therefore, other simple and state-of-the-art networks could be explored, such as the U-Net++ or the DeepLabV3+. These networks combine both shallow and deep features together, ensuring the rough segmentation of fluid and the attention to small details. Instead of changing the model completely, additional attention mechanisms modules could also be added to the U-Net, such as self-attention.

Lastly, generalization to external datasets remains a challenge. In the CHUSJ dataset, the model would frequently predict fluid outside the retina, which was significantly mitigated by using retinal layer delimitation. However, variations in speckle noise, which depend on the scan acquisition protocols, impacted the segmentation quality. To improve robustness under these conditions, future works could explore data augmentation with Gaussian noise or by training the models with data from different sources to increase variability in the training data.

Nevertheless, an improvement in segmentation models would lead to a more accurate fluid estimation. The suggested changes, which also enhance the segmentation of IRF, would be particularly useful in the estimation of this fluid, which does not generalize as well as SRF and PED,

due to its characteristic diversity.

In the generation of intermediate slices, the GAN can be improved both in terms of loss and architecture, to improve its weak attention to details and shift the focus from noise representation to more accurate fluid depiction.

To better perceive the wrongfully generated fluid regions that do not resemble anything seen in the other training images, the model's loss should incorporate stronger perceptual loss functions. One widely adopted perceptual loss is the LPIPS (Learned Perceptual Image Patch Similarity), which compares the perceptual similarity between the real and the generated images, correlating well with human visual perception, guiding the model to create B-scans that are more natural in detail.

To enhance the focus on smaller details by the generative network, some changes can be made to the generator, which is a simple U-shaped model. Similar to the suggestions for the segmentation model, attention blocks such as self-attention can be introduced into network, promoting the focus on relevant spatial features, thus improving the details in the generated slice.

Similarly to what is seen in the literature, the generator could also adopt a multi-scale architecture, where images of different resolutions are output and compared, promoting both an accurate global structure, and the needed attention to detail.

Lastly, it would be interesting to study the results produced when changing the patch dimensions from small patches to the whole image, and to understand how this affects the outcomes of the generative model. The proposed suggestions could also be adapted to this larger patch size.

By improving the details in the generated B-scans through these suggestions, the results from the fluid volume estimation in enhanced and generated volumes would therefore improve as well, reaching values closer to those seen on the masks predicted in the unchanged OCT volumes. This would significantly improve the confidence in the calculated fluid volumes, especially for the Spectralis device, where the distance between slices is larger and, therefore, so is the uncertainty in the predicted volumes.

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. 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.

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] B. Niu et al., "Single Image Super-Resolution via a Holistic Attention Network," in *Computer Vision – ECCV 2020*, Horst, B. Thomas, F. J.-M. V. Andrea, and Bischof, Eds., Springer International Publishing, 2020, pp. 191–207, ISBN: 978-3-030-58610-2. DOI: 10.1007/978-3-030-58610-2_12.

[54] T. Ding, L. Liang, Z. Zhu, and I. Zharkov, "CDFI: Compression-Driven Network Design for Frame Interpolation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2021, pp. 8001–8011. DOI: 10.48550/arXiv.2103.10559.

[55] H. Lee, T. Kim, T.-y. Chung, D. Pak, Y. Ban, and S. Lee, "AdaCoF: Adaptive Collaboration of Flows for Video Frame Interpolation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020. DOI: 10.48550/arXiv.1907.10244.

[56] 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.

[57] 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.

[58] 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.

[59] 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.

[60] 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.

[61] 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.

[62] 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.

[63] 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.

[64] 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.

[65] 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.

[66] T. Melo, Â. Carneiro, A. Campilho, and A. M. Mendonça, "Retinal layer and fluid segmentation in optical coherence tomography images using a hierarchical framework," *Journal of Medical Imaging*, vol. 10, p. 14 006, 1 2023. DOI: 10.1117/1.JMI.10.1.014006.

[67] 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.

[68] 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.

[69] 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.

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

[71] 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.

[72] 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.

[73] 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.

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

[75] J. Åkesson, J. Töger, and E. Heiberg, "Random effects during training: Implications for deep learning-based medical image segmentation," *Computers in Biology and Medicine*, vol. 180, p. 108 944, 2024, ISSN: 0010-4825. DOI: 10.1016/j.compbiomed.2024.108944.

[76] M. G. Altarabichi, S. Nowaczyk, S. Pashami, P. S. Mashhadi, and J. Handl, "Rolling the dice for better deep learning performance: A study of randomness techniques in deep neural networks," *Information Sciences*, vol. 667, p. 120 500, 2024, ISSN: 0020-0255. DOI: 10.1016/j.ins.2024.120500.

[77] K. Alsaih, M. Z. Yusoff, T. B. Tang, I. Faye, and F. Mériaudeau, "Deep learning architectures analysis for age-related macular degeneration segmentation on optical coherence tomography scans," *Computer Methods and Programs in Biomedicine*, vol. 195, p. 105 566, 2020, ISSN: 0169-2607. DOI: 10.1016/j.cmpb.2020.105566.

Appendix A

Impact of Random Initialization of the U-Net in Segmentation Performance

Table A.1: Dice scores for every vendor and fluid from multiple runs performed in the same conditions. "Set 1" includes Runs 1 to 5, all validated on fold 2. Meanwhile, "Set 2" includes Runs 6 to 14, validated on fold 3. All runs in each set were trained under the same conditions for 100 epochs. The random transformations applied were horizontal flipping and a maximum rotation of 10°. This provides insight into how the inherent randomness of the U-Net affects the segmentation performance.

Runs	**VF**	**Cirrus**			**Spectralis**			**Topcon**			**IRF**	**SRF**	**PED**	**Fluid**
		IRF	**SRF**	**PED**	**IRF**	**SRF**	**PED**	**IRF**	**SRF**	**PED**				
Run 1	2	0.555	0.773	0.634	0.849	0.857	0.839	0.783	0.869	0.765	0.686	0.821	0.716	0.663
Run 2	2	0.490	0.813	0.698	0.786	0.848	0.856	0.772	0.903	0.834	0.639	0.850	0.773	0.692
Run 3	2	0.349	0.774	0.600	0.544	0.799	0.759	0.674	0.894	0.772	0.494	0.819	0.687	0.561
Run 4	2	0.542	0.776	0.677	0.831	0.826	0.822	0.852	0.873	0.856	0.699	0.818	0.764	0.685
Run 5	2	0.611	0.817	0.626	0.832	0.836	0.834	0.850	0.913	0.822	0.732	0.853	0.730	0.685
Set 1	2	0.509	0.791	0.647	0.768	0.833	0.822	0.786	0.890	0.810	0.650	0.832	0.734	0.657
Run 6	3	0.661	0.746	0.495	0.632	0.842	0.736	0.702	0.875	0.728	0.671	0.810	0.622	0.617
Run 7	3	0.688	0.622	0.596	0.665	0.707	0.728	0.846	0.851	0.747	0.742	0.721	0.675	0.634
Run 8	3	0.794	0.862	0.632	0.663	0.887	0.778	0.874	0.833	0.780	0.801	0.856	0.712	0.719
Run 9	3	0.622	0.822	0.682	0.677	0.846	0.758	0.736	0.853	0.741	0.673	0.838	0.717	0.685
Run 10	3	0.649	0.718	0.671	0.653	0.872	0.760	0.828	0.824	0.792	0.715	0.784	0.731	0.700
Run 11	3	0.607	0.802	0.614	0.669	0.841	0.760	0.837	0.864	0.787	0.702	0.831	0.703	0.697
Run 12	3	0.565	0.788	0.671	0.550	0.868	0.755	0.784	0.831	0.773	0.643	0.818	0.723	0.648
Run 13	3	0.618	0.699	0.406	0.471	0.724	0.630	0.752	0.878	0.645	0.641	0.769	0.533	0.531
Run 14	3	0.757	0.815	0.701	0.646	0.894	0.773	0.881	0.848	0.836	0.783	0.841	0.763	0.738
Set 2	3	0.662	0.764	0.608	0.625	0.831	0.742	0.804	0.851	0.759	0.708	0.808	0.687	0.663