Fast, Accurate and Steerable Segmentation of Drusen in Optical Coherence Tomography

KUMULATIVE DISSERTATION

zur Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Shekoufeh GORGI ZADEH

aus Genaveh, Iran

Bonn, September 2020

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn

1. Gutachter: Prof. Dr. Thomas Schultz
2. Gutachter: Prof. Dr. Reinhard Klein

Tag der mündlichen Prüfung: 01. September 2020
Erscheinungsjahr: 2020

Abstract

Shekoufeh GORGI ZADEH

Fast, Accurate and Steerable Segmentation of Drusen in Optical Coherence Tomography

Age-related macular degeneration (AMD) is known to be the leading cause of irreversible blindness in developed countries. One of the early appearing biomarkers of AMD are drusen. Drusen are extracellular deposits that develop between the retinal pigment epithelium (RPE) layer and Bruch's membrane (BM). Drusen size, number, and location are among the most important biomarkers for staging AMD. In addition, assessment of these biomarkers is essential for testing new treatments and identifying AMD risk factors. Optical coherence tomography (OCT), on the other hand, is a 3D imaging technique in which the layer structure and the above-mentioned biomarkers can be seen. Particularly in epidemiological studies that may contain thousands of images, manual drusen quantification in OCT is infeasible. Thus it is necessary to use automated segmentation algorithms.

Many drusen segmentation algorithms first segment the RPE layer of the retina, then they segment the drusen. These algorithms use bilateral or median filters to reduce the speckle noise of OCT images, prior to their main segmentation pipeline. In this thesis we first propose a novel multi-scale anisotropic fourth-order diffusion (MAFOD) filter that is well suited for stable localization of ridges and valleys. It allows to smooth along ridges at multiple scales while sharpening them in the perpendicular direction. Compared to other existing diffusion filters as well as bilateral filter, MAFOD better restores the center line of elongated structures. This makes it a suitable filter for many different applications, including the preprocessing of OCT images. In particular for common drusen segmentation algorithms that would first segment the RPE layer of the retina, which is a ridge structure.

Our motivation for the next work was to create a baseline for our newly developed segmentation approach. We evaluated one of the state-of-the-art drusen segmentation algorithms, which was proposed by Chen et al. [*Medical Image Analysis* 17.8 (2013), pp. 1058–1072], on a new set of data. Through the evaluation, we found a substantially inferior performance than the reported results in the original paper. We identified multiple factors that might explain this, including lower axial resolution, greater diversity of drusen load, and simultaneous presence of other pathologies in our data-set, compared to the data-set that was described in the original paper. This motivated us to refine the state-of-the-art algorithm even further by adding additional steps, including using MAFOD filter for preprocessing the data. Even though we did not achieve the accuracy reported in the original publication, our refined algorithm significantly improved the performance of the state-of-the-art algorithm. In addition, our results highlight that the current way of evaluating algorithms does not allow to reliably infer, how good the results are going to be on an independent data-set, and that there is a need to do more work on the proper replication and validation of algorithms in the field of medical image analysis.

At the time this work was started, all the available drusen segmentation algorithms were using traditional segmentation methods, such as image thresholding or active contours. It was expected that using CNNs could lead to better segmentation results. Therefore we presented the first CNN-based drusen segmentation pipeline. In particular, we evaluated three different choices of integrating a CNN into the segmentation pipeline, and found that all outperformed the state-of-the-art method. Among the three proposed pipelines for drusen segmentation, the one with a CNN trained for segmenting RPE and BM layers combined with shortest path finding and polynomial fitting was the most successful.

Despite providing high segmentation accuracy, still, some of CNN's results required further correction. To address this, our final major contribution to this topic was designing an interactive visual system for retinal layer and drusen segmentation. The interactive system allows the user to apply a pretrained CNN on a set of data and then lets them correct the results. To speed up the correction, the system guides the user to those images where the segmentation is more likely to have failed. To do so, we derived two uncertainty measures from the CNN, and visualize them. In addition, we designed intelligent tools that take user-specified constraints as well as the 3D context information into account to propose improved segmentations. Compared to state-of-the-art correction tools, through a small user study, we observed a time reduction of 53% for layer segmentation correction and 73% for drusen segmentation correction. We expect that this system can be used for both segmentation correction and rapid creation of larger training sets for CNNs. We hope that our system with its uncertainty visualization and intelligent tools can inspire similar semi-automated tools for other CNN-based segmentation tasks.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor Prof. Thomas Schultz, for his guidance, encouragement and advice throughout working on this dissertation and the related research papers. I will not forget the gracious supports he provided throughout my time in his team. This dissertation would not have been possible without his invaluable insights and expertise, and I could not have imagined having a better supervisor and mentor for my research.

Besides my supervisor, I would like to thank my colleagues for their friendship, interesting discussions, and for all the fun we have had during the last years.

My wonderful parents, sister and brother have been encouraging and supporting me spiritually throughout writing this thesis. I am greatly thankful to all of them for their unlimited kindness, love and support.

Last but not the least, I would like to express my gratitude to all my past and present teachers.

Contents

List of Figures

List of Tables

List of Abbreviations

ADAD	**A**bsolute **D**rusen **A**rea **D**ifference
AMD	**A**ge related **M**acular **D**egeneration
AREDS	**A**ge-**R**elated **E**ye **D**isease **S**tudy
A-scan	**A**xial scan
BM	**B**ruch’s **M**embrane
CED	**C**reaseness **E**nhancing **D**iffusion
CFP	**C**olor **F**undus **P**hotography
CNN	**C**onvolutional **N**eural **N**etwork
CSP	**C**onstrained **S**hortest **P**ath
FCN	**F**ully **C**onvolutional **N**etwork
FED	**F**ast **E**xplicit **D**iffusion
FPE	**F**alse **P**ositive **E**limination
GA	**G**eographic **A**trophy
IFOD	**I**sotropic **F**ourth **O**rder **D**iffusion
LS	**L**ocal **S**moothing
MAFOD	**M**ultiscale **A**nisotropic **F**ourth **O**rder **D**iffusion
MLSEC	**M**ultilocal **L**evel-**S**et **E**xtrinsic **C**urvature
MODIAMD	**Mo**lecular **Di**agnostics of **A**ge related **M**acular **D**egeneration
OCT	**O**ptical **C**oherence **T**omography
OR	**O**verlap **R**atio
RNFL	**R**etinal **N**erver **F**iber **L**ayer
RPE	**R**etinal **P**igment **E**pithelium
SDE	**S**emi-automated **D**rusen **E**xtraction
SD-OCT	**S**pectral **D**omain **O**ptical **C**oherence **T**omography
SNR	**S**ignal to **N**oise **R**atio
SVM	**S**upport **V**ector **M**achine
SVP	**S**ummed-**V**oxel **P**rojection
TD-OCT	**T**imel **D**omain **O**ptical **C**oherence **T**omography
VED	**V**esselness **E**nhancing **D**iffusion

Chapter 1

Introduction

1.1 Motivation

Understanding and preventing neurodegenerative diseases is one of the main challenges in medicine. Age related macular degeneration (AMD) is one example, that is characterized by progressive degeneration of central regions of the retina. This has made AMD the reason for about 8% of blindness worldwide, and the leading cause of blindness in developed countries [196]. Particularly AMD affects people older than 60 years and is likely to become more prevalent due to population ageing. Figure 1.1 shows the prevalence of AMD across 10 European countries.

In addition, anatomically, the retina layer of the eye is known to be an extension of the brain [106]. Therefore understanding neurodegenerative changes in the retina can help to better understand, or diagnose brain's neurodegenerative disorders like Alzheimer's disease at their earlier stages [76].

One key feature of AMD is the formation of drusen around or in the macula, which is at the central region of the retina. Drusen are extracellular deposits that lift the retinal pigment epithelium (RPE) layer, as it is illustrated in Figure 1.2. Although different laser treatments for drusen have been proposed [41, 56, 136], still none has been proven to be broadly effective.

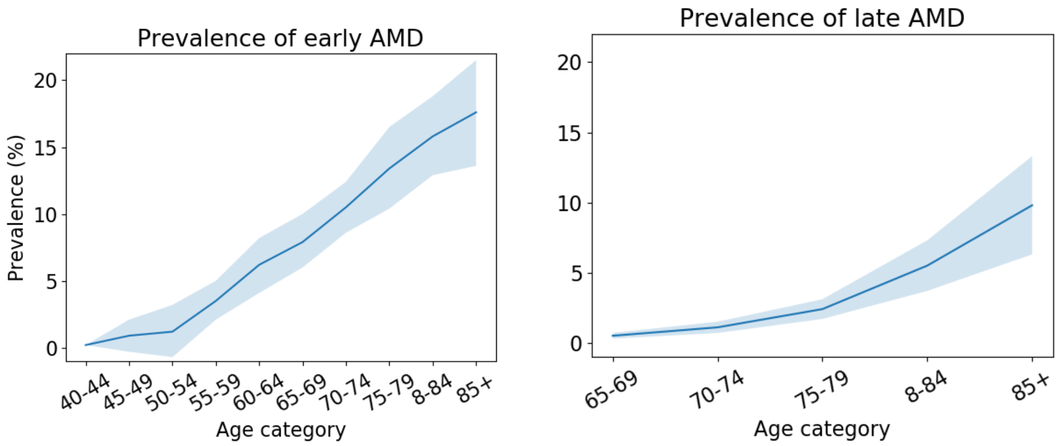


FIGURE 1.1: Prevalence of early and late AMD in Europe from 1990 to 2013. Early and late AMD are different stages of this disease. The plots show the 95% confidence interval of AMD prevalence across different age categories. This study was done by Colijn et al. [23] in 2017, using the European Eye Epidemiology (E3) consortium. The data used in this study was collected from 14 population-based cohorts with a total of 42, 080 individuals from 10 different European countries.

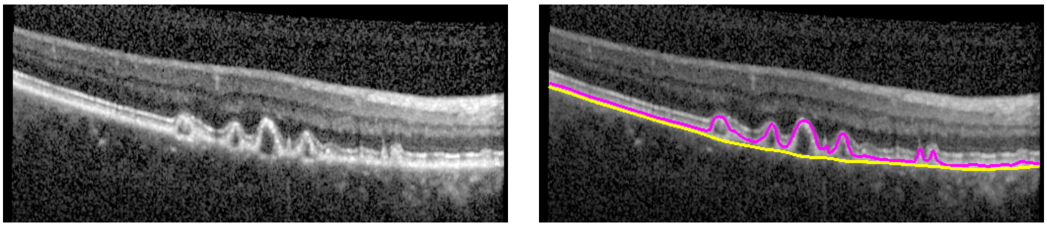


FIGURE 1.2: The left image is an example of Drusen formed on the retinal layer. On the right, Retinal pigment epithelium (RPE) layer (magenta), and Bruch's membrane (BM) layer (yellow) are marked over the 2D slice of optical coherence tomography.

One of the non-invasive techniques for retinal imaging is optical coherence tomography (OCT). It can be utilized to render a 3D scan by taking series of 2D images. OCT can be used to assess drusen size, number, or extent [192]. For clinical trials, in which new treatments are evaluated, estimating drusen load before and after the treatment is essential. However, this requires drusen size assessment in hundreds of images, which is heavily time consuming if done manually. In addition, AMD is known to be caused by a mixture of environmental and genetic risk factors such as smoking, age or family history [84]. In order to identify correlation between these risk factors and AMD progression, assessment of drusen size, number and location becomes necessary in thousands of images. For this reason designing a semi- or fully-automated drusen segmentation technique becomes essential.

1.2 Contributions

The research on drusen segmentation can be divided into different categories based on the underlying image modality or based on the proposed algorithm, i.e., whether it is semi-automated or fully-automated. Usually drusen segmentation algorithms first segment the RPE layer. Then by fitting a low degree polynomial onto RPE layer, they estimate an ideal drusen-free RPE. With this, the area between the two layers is considered as drusen [18, 31, 55]. From the different fully-automated algorithms that work on OCT scans, we identified the method by Chen et al. [18] as the state-of-the-art, based on their reported high accuracy.

In order to create a baseline, we replicated the algorithm by Chen et al. [18]. We evaluated their algorithm on our data-set that was more heterogeneous, and had a greater diversity of drusen load, and different image quality compared to the data-set that was described in the original paper. Consequently we observed a segmentation accuracy that was significantly lower than the reported results by Chen et al. [18]. This motivated us to improve their segmentation pipeline by adding local histogram equalization and several additional regularization steps.

In addition we propose a novel multi-scale anisotropic fourth-order diffusion (MAFOD) filter, that is suitable for enhancing ridge- and valley-like structures. It allows to selectively smooth along ridges or valleys at multiple scales, while sharpening them in the perpendicular direction. In this thesis, we show that compared to other existing filters, MAFOD is better suited for restoring the center-line of elongated structures. This makes it a suitable filter for many different applications, including the preprocessing of OCT images. In particular, for those segmentation algorithms that would first segment RPE layer, because RPE is a ridge structure in OCT

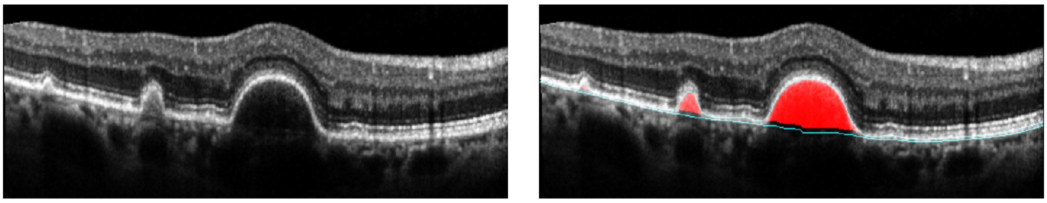


FIGURE 1.3: Drusen segmentation (red) using our proposed pipeline. In this example the best performing pipeline, i.e., the one with a CNN that segments RPE and BM layers (cyan) first, was used.

images. By replacing the bilateral filter, which was proposed in Chen et al. [18] algorithm with MAFOD, we could improve our refined drusen segmentation algorithm further.

The fact that recently convolutional neural networks (CNNs) have substantially improved medical image segmentation [5], and that no one had used CNNs for drusen segmentation, motivated us to use a CNN for this task. In particular, we evaluated three different ways of integrating a CNN into the segmentation pipeline. In our first pipeline we trained a network to directly segment drusen. In the second pipeline, we trained the network to first segment RPE and bruch's membrane (BM) layers. From the RPE and BM layers, drusen are segmented by first rectifying the RPE layer, using the BM layer, and then fitting a polynomial on the RPE layer to estimate a drusen-free RPE. In the end the area between the drusen-free RPE and RPE layer was considered as drusen. In the third pipeline, we trained the network to segment the RPE/BM complex. We treated the upper and lower boundary of the complex as RPE and BM layers for further drusen segmentation. All three pipelines outperformed the state-of-the-art method by Chen et al. [18], and the best performance was achieved when the CNN was used for segmenting the RPE and BM layers first (see Figure 1.3).

Medical images that come from different imaging devices can have different characteristics such as contrast level, making it hard for a CNN trained on a specific data-set to perform equally well on a data-set from a different device. Instead of retraining the CNN with the new data, as part of our collaboration with Wu et al. [198], we preprocessed each image by matching their histogram to the average histogram of the data-set that was used to train the CNN. Figure 1.4 illustrates how this preprocessing step improved the segmentation.

Despite the high accuracy of CNNs for image segmentation, they can sometimes fail to correctly segment the object of interest. In particular, for sensitive applications this will require a human rater to check/correct the final results. This motivated us to introduce a system that allows to use a pretrained CNN for automatic RPE and BM layer segmentation. Then, this system uses the internal states of the CNN to provide two uncertainty measures, which numerically estimate CNN's uncertainty for the segmentation. The uncertainty measures are visualized to guide the user to those image slices that are more likely to have an erroneous segmentation. In addition, our interactive system offers multiple intelligent tools, that utilize the inner states of the CNN, and the context of each 2D image slice in the 3D volume, for helping the user to correct the segmentation in a timely manner. Compared to the correction time required when users had no guidance and could only use other conventional tools, we observed a speed-up by a factor of two for retinal layers segmentation correction, and by more than a factor of three for drusen segmentation correction.

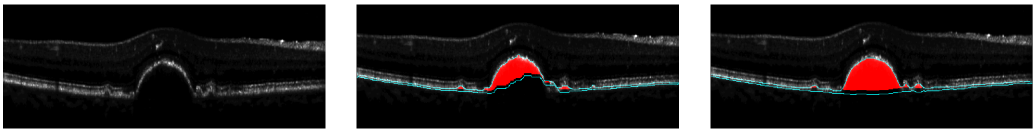


FIGURE 1.4: Segmentation results on the input (left) from a different device type, before (middle) and after (right) the histogram matching step.

1.3 Relationship to prior and concurrent works

As mentioned earlier, most of the drusen segmentation algorithms use an enhancing filter on the input image prior to the main segmentation pipeline. For instance Chen et al. [18] use a bilateral filter, or Iwama et al. [68] use a median filter to denoise B-scans before binarizing them. Given the elongated shape of the retina in B-scans, we show that our proposed fourth-order multiscale anisotropic diffusion filter is better suited when we are interested in extracting the ridges. Our filter not only preserves the ridge location, but it can also enhance them. Another anisotropic fourth-order diffusion filter was proposed by Hajiaboli et al. [59]. However, their filter was particularly designed to enhance edges.

For evaluation of the state-of-the-art algorithm for drusen segmentation, amongst other fully automated algorithms [31, 55, 68] that use classical image processing approaches, we picked the method by Chen et al. [18]. This decision was made according to the review by Wintergerst et al. [192] on algorithms for the automated analysis of AMD biomarkers in OCT, and the reported accuracy in Chen et al.'s paper. However, after applying their algorithm on a completely new data-set, we observed a significantly lower accuracy, which motivated us to improve their segmentation pipeline (see Chapter 6).

In the recent years deep learning has become popular for retinal image analysis [161]. In particular, convolutional neural networks (CNNs) have improved the segmentation of AMD biomarkers [74, 98, 158]. Our work in Chapter 7 utilizes a CNN for drusen segmentation. In 2019 Asgari et al. [7] used a separate decoder per target class, i.e., RPE, BM, and RPE/BM complex in the U-Net. They show an improvement in segmentation when these modifications are used, compared to multiple baselines, including the standard U-Net for direct drusen segmentation, and the standard U-Net trained for RPE and BM layer segmentation. In contrast to our work, they assume outer boundary of RPE and BM layers overlap in non-pathological regions. Even though they use a U-Net trained to segment RPE and BM layers as one of their baselines, a comparison to our complete segmentation pipeline (including shortest path finding and false positive elimination step) is not reported in their work. In another work, Asgari et al. [8] use the standard U-Net architecture and extend it with the spatial pyramid pooling components, allowing it to utilize a larger context for segmentation.

Despite the great improvement in medical image segmentation accuracy using deep learning, the results are still not perfect and these networks often need large and comprehensive training data-sets. For these reasons semi-automated algorithms are still widely used [124]. Particularly for drusen segmentation, Farsiu et al. [40] provided a GUI that allowed drusen number area and volume estimation in OCT scans. Later on Jain et al. [72] developed a similar user-interactive method to estimate area covered by drusen in OCT scans. In both cases an active contour-based RPE segmentation method was used. Other semi-automated algorithms focus only on segmentation of retinal layers [117, 139]. In addition we present an interactive system, that uses an uncertainty score obtained from the trained convolutional neural network, to guide the user to those poorly segmented regions by the neural network. The user is provided with a set of intelligent tools for correcting these failures, speeding up the correction process.

1.4 Outline and publications

Chapter 2 provides background knowledge on age related macular degeneration (AMD), optical coherence tomography, drusen, and drusen segmentation algorithms. Chapter 3 familiarizes the reader with basic concepts of convolutional neural networks that are necessary to discuss our main fully- and semi-automated drusen segmentation pipelines. Chapter 4 provides mathematical background and an overview of diffusion filters.

The main research contribution of this dissertation is based on four publications that are presented in Chapters:

- Chapter 5: Gorgi Zadeh et al. "Multi-scale Anisotropic Fourth-Order Diffusion Improves Ridge and Valley Localization" *Journal of Mathematical Imaging and Vision* 59.2 (2017): 257-269. URL: `https://doi.org/10.1007/s10851-017-0729-1` [204].

- Chapter 6: Wintergerst and Gorgi Zadeh et al. "Replication and Refinement of an Algorithm for Automated Drusen Segmentation on Optical Coherence Tomography" *Scientific Reports* 10.1 (2020): 1-7. URL: `https://doi.org/10.1038/s41598-020-63924-6` [191].

- Chapter 7: Gorgi Zadeh et al. "Cnns enable accurate and fast segmentation of drusen in optical coherence tomography" *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support* Springer, Cham, (2017): 65-73. URL: `https://doi.org/10.1007/978-3-319-67558-9_8` [51].

- Chapter 8: Gorgi Zadeh et al. "Intelligent interaction and uncertainty visualization for efficient drusen and retinal layer segmentation in Optical Coherence Tomography" *Computers & Graphics* 83 (2019): 51-61. URL: `https://doi.org/10.1016/j.cag.2019.07.001` [205].

In addition, we published:

- Gorgi Zadeh et al. "Uncertainty-Guided Semi-Automated Editing of CNN-based Retinal Layer Segmentations in Optical Coherence Tomography." *Visual Computing for Biology and Medicine* (2018): 107-115. URL: `https://doi.org/10.2312/vcbm.20181235` [50].

This work got extended into [205] later on.

We also contributed to the following research studies, in which our drusen segmentation pipeline was used, but which are not part of this dissertation:

- Wu et al. "Secondary and Exploratory Outcomes of the Subthreshold Nanosecond Laser Intervention Randomized Trial in Age-Related Macular Degeneration: A LEAD Study Report." *Ophthalmology Retina* 3.12 (2019): 1026-1034. URL: `https://doi.org/10.1016/j.oret.2019.07.008` [198].

  This was a 36-month trial study to test the effect of subthreshold nanosecond laser (SNL) treatment on slowing down AMD progression from early to late stage. Our pipeline [51] was used to automatically segment drusen. These segmentations were used to investigate the influence of the laser treatment on the changes in drusen volume.

- Pondorfer et al. “Association of Visual Function Measures with Drusen Volume in early stages of Age-Related Macular Degeneration”. *Investigative Ophthalmology & Visual Science* 61.3 (2020), 55-55. URL: `https://doi.org/10.1167/iovs.61.3.55` [132].

  In this work, association between measurements from visual function tests and drusen volume in AMD is investigated. In this study our drusen segmentation pipeline [51] was used to estimate drusen volume in 100 OCT scans.

Chapter 9 concludes all the proposed methods and discusses future possibilities to extend these works.

Chapter 2

OCT image analysis for AMD

2.1 Optical Coherence Tomography (OCT)

Optical Coherence Tomography (OCT) is a non-invasive diagnostic technique which works similar to ultrasound, except it uses light waves instead of sound waves. OCT imaging was first introduced by Huang et al. [65] and has gained wide popularity in ophthalmology, as it can be utilized to render an in-vivo cross-sectional image of biological systems such as the retina.

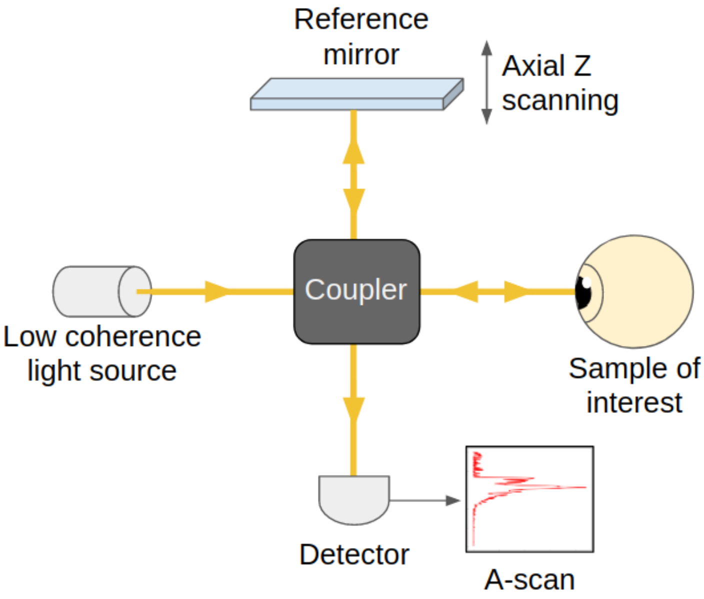


FIGURE 2.1: A simple sketch of a basic OCT.

OCT measures the echo time delay and the magnitude of the back-scattered light beam to image within tissues. The spatial resolution achieved by OCT can be in $1 - 15\mu$m scale, which is of one to two orders of magnitude finer than the spatial resolution achieved by conventional ultrasound [133]. OCT is a highly sensitive interferometric technique. As Figure 2.1 illustrates, a very basic OCT consists of a low-coherence light source, whose light beam is split into two beams through a coupler. One beam is directed to a reference arm, which consists of a reference mirror, and the other beam is directed onto a sample arm, which consists the sample of interest. The back-scattered light beams from the reference mirror and the sample are again recombined at the coupler, creating interference patterns. These patterns are then recorded by a detector. When both light beams travel the same distance, they create a certain interference pattern. Therefore by changing the position of the

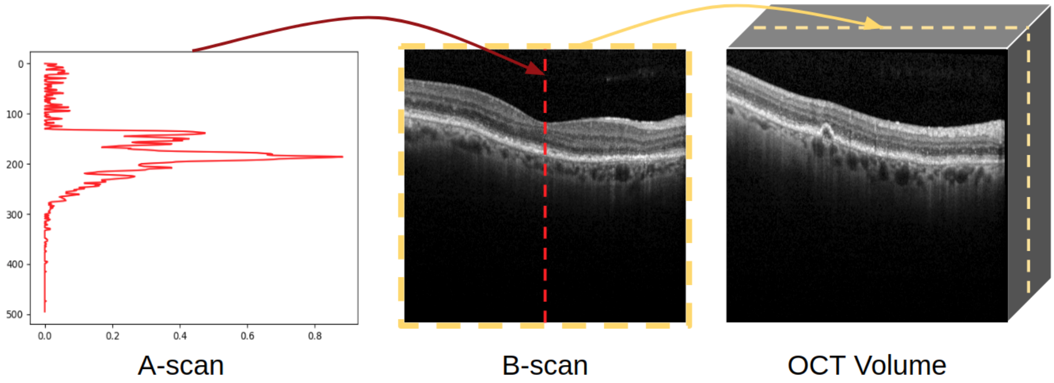


FIGURE 2.2: A three dimensional OCT scan is generated by taking a series of B-scans in XY plane. Each B-scan is generated by taking a series of A-scans in transverse positions in X direction. The red vertical line over the B-scan shows an A-scan which corresponds to back-scattered light intensity along the axial direction.

reference mirror and monitoring the recorded interference pattern at the detector, one can measure the magnitude of the back-scattered beam knowing the axial depth within the sample from which the light is reflected [133]. In a time domain OCT (TD-OCT) the reference arm is quickly scanned over a distance that corresponds to the intended scanning depth range [119].

Figure 2.2 illustrates an example use of OCT for retinal imaging. The depth profile of a sample structure is reconstructed by measuring the intensity of reflected or back-scattered light beam versus depth in an Axial scan (A-scan). The two-dimensional cross-sectional image (B-scan) of the sample structure is generated by measuring a series of A-scans at different transverse positions. Finally the complete three-dimensional image is reconstructed by taking a series of B-scans. In addition to popularity in ophthalmology for retinal imaging, OCT is also used in other medical research areas such as endoscopic imaging of gastro-intestinal and cardiovascular systems.

Recent spectral domain OCT (SD-OCT), also known as Fourier domain OCT systems has increased the scan rate, resolution and speed. In SD-OCT a spectrometer is used to measure the cross-spectral density at the detection arm. This replaces the mechanical scanning of the reference arm technique, which is used in TD-OCT [119]. This allows to quickly and continuously image a 6mm area of the retina, which decreases the chance of missing pathologies in scans compared to TD-OCT systems [194].

2.2 Retinal layers

For diagnosing and studying many ocular diseases, anatomical and pathological analysis of retinal layers and their thickness is crucial [65]. Figure 2.3 shows the international nomenclature for the classification of retinal and choroidal layers on SD-OCT images of a normal eye [174]. In this dissertation Nerve Fiber Layer known as RNFL, and RPE (Retinal Pigment Epithelium)/Bruch's membrane complex are particularly interesting.

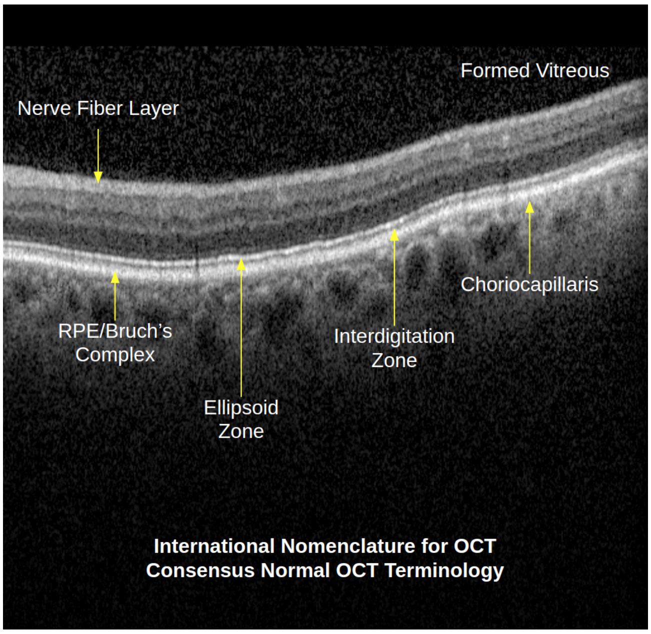


FIGURE 2.3: SD-OCT image, that shows the international nomenclature for the classification of retinal and choroidal layers. The image is labeled according to an article by Staurenghi et al. [174].

2.3 Age related Macular Degeneration (AMD)

One common ocular disease, which is the major cause for irreversible vision loss world-wide, i.e., more than 20% of the ageing population might have it [100], is age related macular degeneration (AMD). This disease mostly develops at the central retina and is chronic and progressive, which at its later stages causes severe visual loss [100]. AMD is a complex trait disease with multiple genetics, lifestyle and medical risk factors [37]. Lim et al. [100] list some of the risk factors, with *age* being the major one. Some other major risk factors include, cigarette smoking, obesity, nutritional factors such as low dietary intake of particular vitamins, cardiovascular diseases, genetic markers and so on.

2.3.1 Wet and dry AMD

There are two different known late AMD stages: wet AMD and dry AMD.

Wet AMD, also known as neovascular or exudative AMD, happens when blood vessels grow abnormally in the choriocapillaris (see Figure 2.3) through the BM, that eventually causes leaking fluid, lipids, and blood, leading to fibrous scarring below the macula. At earlier stages of wet AMD, the choroidal vessels start as capillaries, which are seen to develop into veins later on. Hemorrhagic or serous RPE and retinal detachment might be observed in this form of AMD.

Wet AMD is the minority type. About 85 to 90% of patients with AMD, suffer from the dry form of the disease. The patient who develop wet AMD are on average 70.5 years old, whereas those who develop dry AMD are younger with 56.8 years old on average [168]. In late dry AMD, also known as nonexudative AMD, geographic atrophy (GA), which is the atrophy of RPE, choriocapillaris, and photoreceptors, occurs [100]. Figure 2.4 shows a sample B-scan of a patient who has developed GA. In severe cases a secondary degeneration of photoreceptors that lies on top of the degenerating RPE cells happens [206].

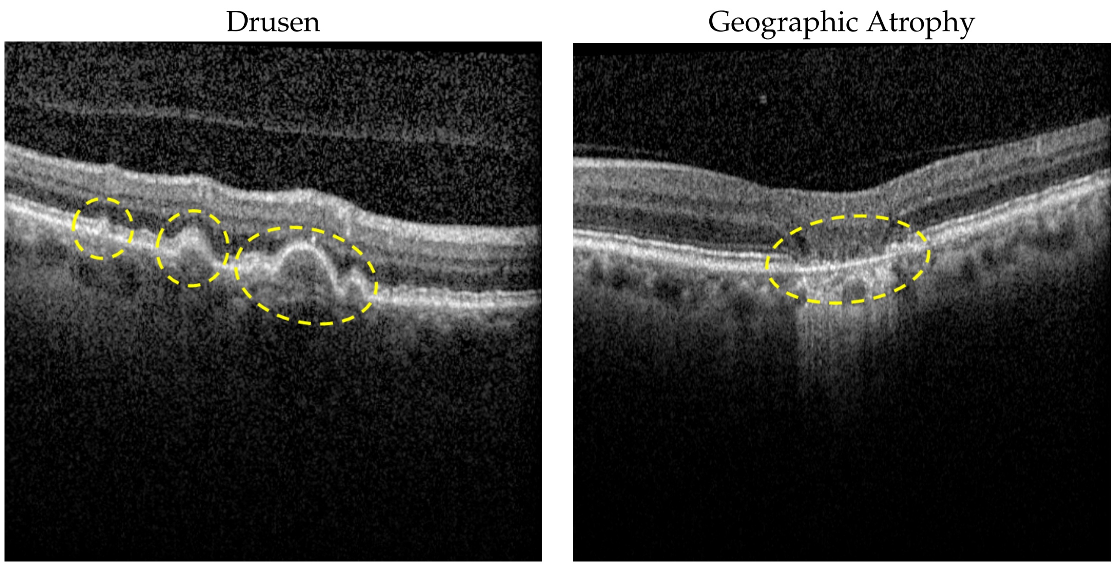


FIGURE 2.4: Two B-scan samples that represent drusen (left), that have lifted the RPE layer up, and geographic atrophy (GA) (right).

2.3.2 Drusen

One of the early appearing signs of AMD are soft drusen [69], that are focal deposition of acellular detritus between RPE and BM layers (see Figure 2.4), and appear as small yellow deposits in the macula. They can lead to RPE degeneration and secondary BM thickening [11]. Many studies have shown a positive correlation between estimated drusen maximum size and total drusen area with the risk of progression into advanced AMD in color fundus photography (CFP) [27]. Green et al. [53] lists four different types of drusen; 1) soft drusen 2) hard drusen 3) crystalline drusen and 4) cuticular or basal laminar drusen. However not all drusen types are associated with AMD. Hard drusen are smaller than 50 microns in diameter and are common in young people. This druse type is not a sign of AMD and is not noted to increase in number with age. Similar to hard drusen, basal laminar drusen are small in size and appear as white deposits and are observed in normal aging eyes [171]. Unlike the other two drusen types, soft drusen are usually larger than 50 microns in diameter and are known to be an early sign of AMD. Finally crystalline drusen are the soft drusen that are predisposed to geographic atrophy [153]. The Age-Related Eye Disease Study (AREDS) breaks the signs of AMD development into the following four categories [100]:

1. None or a few small drusen (with diameter $< 63\mu m$).
2. Multiple small drusen or few intermediate drusen ($63 - 124\mu m$ in diameter), or retinal pigment epithelium abnormalities.
3. Extensive intermediate drusen, or at least one large drusen ($\geq 125\mu m$ in diameter, or geographic atrophy not involving the fovea.
4. GA involving the fovea or any of the features of neovascular age-related macular degeneration.

Reticular drusen are yet another form of yellowish subretinal lesions that are arranged in a network. Through a population-based study, Klein et al. [86] found an association between long-term cumulative incidence of reticular drusen, also known as pseudo drusen, and high development risk of late AMD.

In order to assess AMD progression, and understand the disease better through epidemiological studies, it becomes necessary to use semi- or fully automated algorithms, as manual grading of drusen in volumetric data is unfeasible. The SD-OCT data-set, which was used in this dissertation is collected from patients that mostly fall into categories 2 and 3, i.e., the intermediate AMD.

The following section briefly summarizes some of the most important and state of the art semi- and fully-automated algorithms, that are developed for drusen segmentation and quantification task.

2.4 Drusen segmentation

There are many algorithms developed for the segmentation of drusen from color fundus photographs (CFPs). CFPs are useful for assessing the en-face, which is the 2D view in axial direction, of the macula. Duanggate and Uyyanonvara [33] provide a review of automated drusen segmentation algorithms for this modality. It includes texture-based approaches, e.g. [99, 126], histogram-based approaches, e.g. [138], morphological approaches, e.g. [155], multi-level approaches, e.g. [12] and fuzzy logic approaches [154]. In general the common challenge for drusen segmentation in CFPs is the varying background of drusen, making it difficult to reliably localize drusen boundary [18]. Moreover CFPs are 2D images, therefore they are not suitable for assessing drusen volume. This makes high resolution OCTs an attractive alternative modality for drusen segmentation.

In their survey, Wintergerst et al. [192] list various semi- or fully-automated algorithms for analysis of AMD biomarkers in OCTs. Out of those, seven algorithms are mentioned for drusen segmentation. Some of these algorithms only focus on estimating the area covered by drusen or drusen volume, and some other on estimating drusen number or maximum drusen diameter. Most of drusen segmentation algorithms segment RPE and sometimes BM layers prior to segmenting drusen. In this chapter these algorithms, in addition to other recent ones, are reviewed.

2.4.1 Semi-automated algorithms

Farsiu et al. [40] proposes a semi-automated algorithm for drusen number, area and volume estimation. They provide a graphical user interface (GUI) to fine-tune a pipeline of image thresholding, shortest path finding, and active contours to segment the RPE layer of the retina. Then the GUI can be used to fit a second or fourth order polynomial to the estimated RPE curve. The estimated polynomial is considered as healthy/normal (drusen-free) RPE. Finally the area between RPE and the normal RPE is marked as drusen.

Jain et al. [72] uses the semi-automated active contour-based RPE segmentation similar to Farsiu et al.'s method, to estimate the area covered by drusen from SD-OCT images. They also develop another user-interactive method that works only on color fundus photography images. It allows users to select a region of interest (excluding other reflective structures), to help developing a background model. Using the background model, drusen are segmented. They compare drusen burden assessment from these two different modalities and show larger drusen in eyes

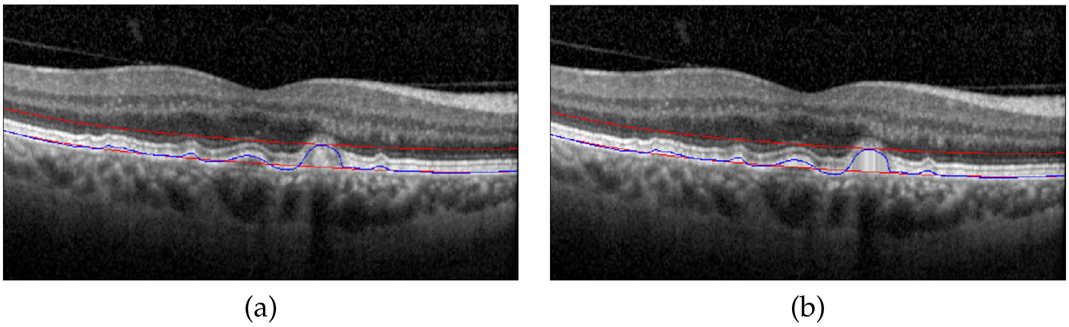


FIGURE 2.5: (a) Red curves show the selected sub-volume for generating en-face projection. Top curve passes through the tallest druse in the whole volume. (b) Intensity of pixels between RPE and drusen-free RPE are replaced with their maximum in each column.

with hyperpigmentation were better segmented in SD-OCT. There are other semi-automated tools, that instead of drusen, mainly focus on segmentation of retinal layers [117, 139].

2.4.2 Fully automated algorithms

In the work by Gregori et al. [55], the authors automatically estimate drusen volume and area, using a proprietary RPE segmentation algorithm, commercially available on the Cirrus HD-OCT software. The same polynomial fitting strategy, as described in Section 2.4.1, is used to estimate a drusen-free RPE. By computing the distance between RPE and ideal RPE, a drusen height map is estimated. To reduce the number of falsely detected drusen due to noise, the authors propose to use a height threshold. Diniz et al. [31] uses these elevation maps to identify drusen clusters and estimate drusen count as well as the size of each individual druse.

Iwama et al. [68], first use a median filter to denoise B-scans and then binarize them. Unlike the above-mentioned algorithms, here drusen are computed between RPE layer and Bruch's membrane (instead of the estimated drusen-free RPE). Finally they assess drusen area and their maximal diameter in the en-face projection of the retina.

The method by Chen et al. [18] is considered as one of the state-of-the-art for drusen segmentation, according to their reported accuracy. First they use a bilateral filter to denoise the B-scans to better estimate RNFL and RPE layers. In the second step, they automatically remove the RNFL complex by considering a band around the bottom boundary of background (the vitreous of the eye), which they already extract by a simple image thresholding. Due to high reflectivity of RPE layer, another thresholding is used to segment RPE. Small regions, with area less than 150 pixels in their data-set, are removed from the binary maps. Finally RPE curve is extracted by finding the center line of the binary RPE component. In the next step they use a third order polynomial to fit on the RPE layer in order to estimate the drusen-free RPE. The area between RPE and drusen-free RPE is then marked as drusen. The pipeline proposed by Chen et al. [18] has a false positive elimination (FPE) step, that discards any drusen that appear only in one B-scan. In addition, by using selective summed-voxel projection (SVP), they generate an en-face projection of the volume. To generate the en-face projection image, a sub-volume of the complete OCT volume is selected so that the upper boundary of the sub-volume goes through the peak of the tallest druse in the whole volume, in parallel to the drusen-free RPE.

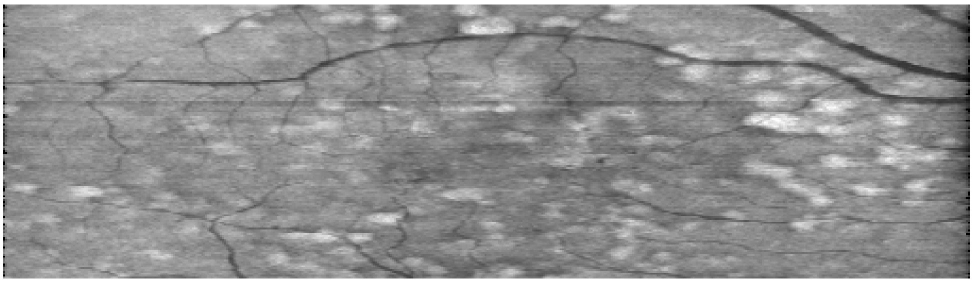


FIGURE 2.6: Selective SVP (en-face projection) of an OCT volume with 145 B-scans of size 496 × 512. The resulting en-face is of size 145 × 512. Brighter regions indicate drusen.

The drusen-free RPE is considered as the lower boundary of the sub-volume. To make drusen brightness be more in-line with their height, the intensity of pixels in each column between RPE and ideal RPE is replaced with their maximum intensity value as illustrated in Figure 2.5. Finally pixels in the sub-volume are summed up together column-wise to create the projection image. Figure 2.6 shows an example of the en-face projection.

If the average intensity within a druse area compared to the average intensity of its boundary is less than 4, or if the ratio of width and height of druse on the en-face projection is less than 6, it is considered as false positive and is removed. In the end, drusen segmentation is smoothed with a Gaussian filtering. In the work by de Sisternes et al. [166], the algorithm by Chen et al. [18] was successfully used for longitudinal study, where predicting AMD progression was the main focus.

2.4.3 Machine learning approaches

The algorithms discussed so far are mostly categorized as the classical image processing methods, that mainly rely on a set of mathematical functions applied to OCT images. These algorithms can be categorized either as rule-based, which are deterministic models that require no training, or as the traditional machine learning approaches. Usually the traditional machine learning approaches require less data to develop a set of features and a statistical model for the analysis. In contrast, supervised deep learning approaches are able to learn their own representation/features when enough annotated images are available. This makes them very powerful, but often harder to interpret. Thus each of these approaches have their own specific strength and weakness points that make them suitable for specific research questions [161].

Ren et al. [142] use image patches with and without drusen from CFP, and a support vector machine (SVM) to create a probabilistic segmentation of drusen. Despite reaching a high accuracy, as CFP images do not contain any information on geometry of the RPE, they are not enough for drusen volume estimation [55].

In the recent years deep learning has become popular for retinal segmentation tasks [161]. Many of these algorithms use convolutional neural networks (CNNs) to segment different biomarkers of AMD such as intraretinal fluid [158], subretinal fluid, pigment epithelial detachment, and subretinal hyperreflective material [98], or geographic atrophy [74] in OCT. Particularly very recently CNNs are used for segmentation of drusen. Chapter 7 presents (to our knowledge) the first research work that uses deep learning for drusen segmentation.

After our approach was published, other groups also introduced CNN-based systems for drusen segmentation. De Fauw et al. [28] used deep learning for diagnosis and referral in retinal diseases. In their pipeline they use 3D U-Nets to create tissue segmentation maps for 15 different classes including drusen. In the next step these maps are used to train a classification network to create diagnosis probabilities and referral suggestions.

In 2019 Asgari et al. [7] used separate decoders for different target class, i.e., RPE, BM, and RPE/BM complex in the U-Net. Their work is under the assumption that RPE and BM layers are overlapped in non-pathological regions, which is in contrast to our work. To increase regularization effect, they added extra connections between these class-specific decoders. They show an improvement in segmentation using these modifications compared to training a U-Net for directly segmenting drusen, and a U-Net for segmenting RPE and BM layers. From their experiments,it is not clear if the complete pipeline (including shortest path finding and false positive elimination steps) of our work in Chapter 7 has been used as the baseline. In order to use a larger context for the segmentation, in another work, Asgari et al. use spatial pyramid pooling in the U-Net [8]. This extension can particularly help with segmenting drusen in advanced AMD, as they cover a larger area.

Chapter 3

Convolutional neural networks for image segmentation

The segmentation pipeline that we propose in Chapter 7 uses a convolutional neural network (CNN) for layer and drusen segmentation. In addition Chapter 8 utilizes the prediction maps from a CNN to guide users to highly likely erroneous segmentations. For this reason in this chapter a background on CNNs for image segmentation is presented.

Convolutional Neural Networks (CNNs/ConvNets) [94] are most commonly used for analyzing data with grid-like topology such as visual imagery. The convolutional operators allow CNNs to have translation invariance property for classification, and translation equivariance property for segmentation tasks. These networks have become very popular during the recent years amongst many researchers for significantly improving the state-of-the-art results in semantic image segmentation [46]. CNNs replace the classical hand-crafted feature selection with an automatic mechanism.

Image segmentation tasks can be divided into two sub-categories; semantic segmentation and instance segmentation. In semantic segmentation different instances of the same category are not differentiated. It aims to assign each pixel in an input image as belonging to a semantic class label. Pixels that belong to no class are regarded as background. Researchers have already designed CNNs for semantic segmentation [17, 107, 123, 146]. On the contrary, in instance segmentation [16, 26, 105], the aim is to assign each pixel to differentiate between multiple instances of the same object class. In this dissertation, we are interested in semantic segmentation of drusen.

The field of image segmentation using CNNs is so broad that we do not exhaustively survey it here. There are many reviews published on different CNN architectures used for image segmentation [46, 102, 209]. Our goal in this chapter is to discuss the major building blocks of a CNN and provide necessary details for better understanding Chapters 7 and 8.

3.1 Artificial neural networks

To have a universal function that maps an input image into a per pixel label, amounts to a highly complex function. Artificial neural networks allow to have a general learnable function, which learns this mapping through training. An artificial neural network is a connected network of artificial neurons.

3.2 Artificial neurons

The following schematic describes a simplified biological neuron.

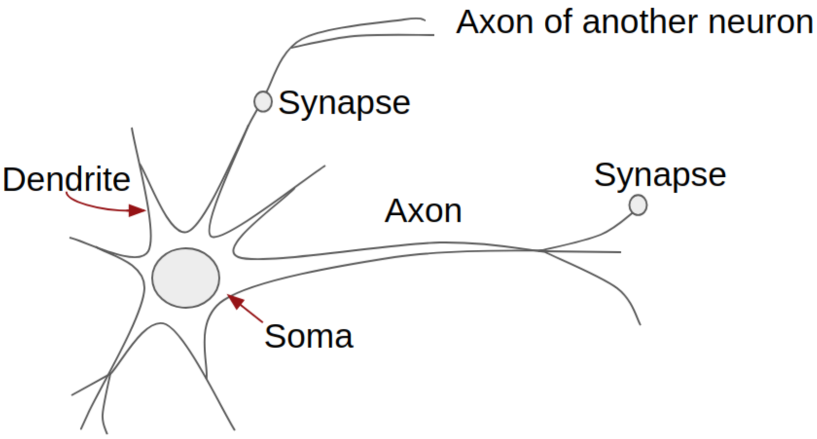


FIGURE 3.1: A simple sketch of a biological neuron.

Artificial neuron structure is inspired by biological neurons, however rigidly following the constraints of biological networks is neither productive nor necessary, as they can lead to a need for very complex and time consuming simulations for artificial intelligence applications. Therefore the inspiration is taken from biology and is adapted to the available computing infrastructures. Schematic in Figure 3.2 shows the commonly used artificial neuron and its correspondence to a biological neuron. The input connection corresponds to dendrites, the summation and activation function corresponds to neuron's soma, output connections correspond to axons, and weights are simplified version of synapses [90]. To put it mathematically, an artificial neuron is defined as follows [2]

$$y_k = f\left(\sum_{i=0}^{n} w_{ki} x_i\right), \tag{3.1}$$

where f is called the transfer (activation) function, that transfers the activation signal into one output y_k of the kth neuron; w_k are the weights of the kth neuron, and x is the input to the neuron as illustrated in Figure 3.2. Usually x_0 is assigned to 1, which makes $w_{k0} = b_k$ to be the bias.

3.2.1 Activation function

The activation function in a neural network determines how the neuron works, i.e., it sets the condition for a neuron to fire (pass the information to the neighbouring neurons) [2]. Simplest activation function is binary

$$f(x) = \begin{cases} 1 & \sum_{i=0}^{n} w_i x_i > 0 \\ 0 & o.w. \end{cases} . \tag{3.2}$$

In this case, a neuron works like a linear classifier, where weights w define the classification hyperplane. However binary activation function is not used in practice as it is not differentiable and does not allow multi-value outputs. Different activation functions work better for different types of problems. One of the most commonly used activation functions is logistic Sigmoid, because its output is a value between

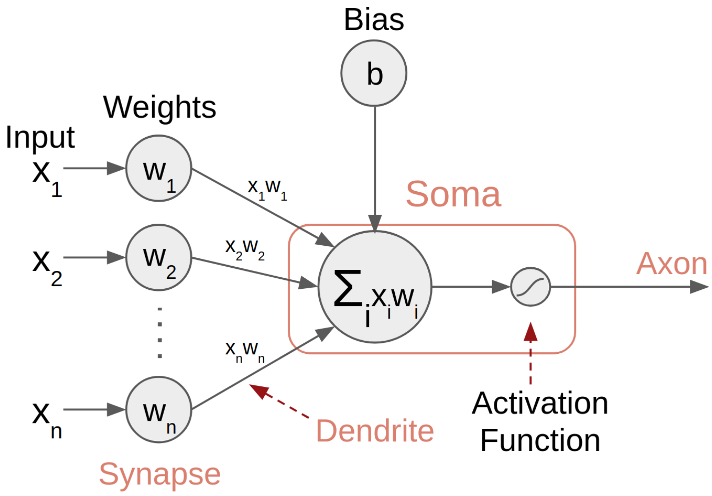


FIGURE 3.2: Artificial neuron schematic. Here the bias $w_0 = b$ is shown separately.

0 and 1. Thus the output can be interpreted as the probability of the neuron to fire [71]. Logistic sigmoid is defined by

$$f(x) = \frac{1}{1 + \mathrm{e}^{-(\sum w_i x_i)}}. \tag{3.3}$$

For very high or very low values the gradient vanishes when logistic sigmoid is used. This causes the network not to learn anymore. In addition, this activation function is known to have a low convergence rate.

Another common activation function, which is used in this dissertation is the Rectified Linear Unit (ReLU). It is defined as

$$f(x) = \max\left(\sum w_i x_i, 0\right), \tag{3.4}$$

and has a faster convergence rate than the sigmoid function [25].

Assuming the output of the last layer of the network is $s = \sum w_i h_i$, where vector h is the input to the last layer, usually the *softmax* function is used as the activation function to represent the probability distribution over M different classes at the end of a classifier network. Vector $p_i \in \mathbb{R}^M$ is computed as

$$p_i = \frac{e^{s_i}}{\sum_j e^{s_j}}. \tag{3.5}$$

As presented in [49], softmax function maps the output scores s into an interpretable vector p with

$$\sum_{l=1}^{M} p_{i,l} = 1.$$

Some other commonly used activation functions are Leaky ReLU [199] and Tanh [113].

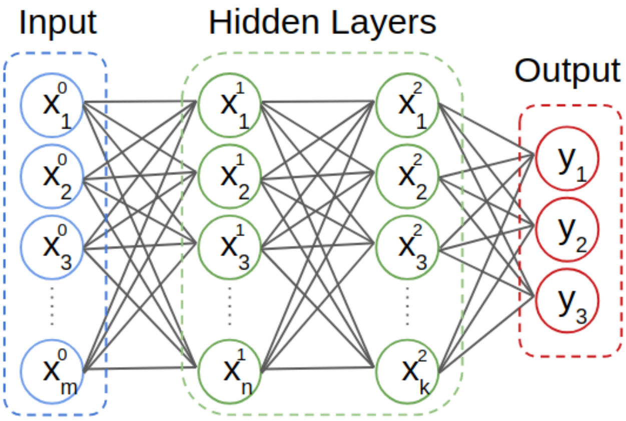


FIGURE 3.3: Graph of a feed-forward, fully connected neural network with two hidden layers..

3.2.2 Fully connected layers

A neural network is a collection of neurons ordered in layers. A layer is called fully connected if the neurons are connected to all neurons in the adjacent layer. Any layer between the input and output is called hidden, as their value is not observed directly. A network that has no loops is called feed-forward neural network [49]. Figure 3.3 shows a feed-forward classifier network, that maps input x into categories $\{y_1, y_2, y_3\}$, with two hidden layers – first one having n neurons, and the second k neurons. Each connection has its own weight w. Number of layers and neurons play the major role for determining the learning capacity of a network.

A single layer network works as a linear classifier as it uses a linear combination of inputs to make a decision. It is possible to add as many hidden layers as desired, although stacking further hidden layers without using non-linear activation functions in between still works as a linear classifier.

3.3 Deep neural networks

Stacking multiples hidden layers on top of each other creates deep neural networks. In general deep learning is part of the machine learning field, with the key advantage that it allows learning features that represent the data for the problem at hand. In other words, it allows computers to understand the world in terms of a hierarchy of features. The computer learns more complex concepts through a combination of simpler ones in the hierarchy [49]. Deep learning is not only popular for image segmentation, but also for object detection [141], image captioning [79], image super-resolution [95], or even generating art [45].

3.4 Convolutional neural networks

When working with images, using fully connected layers is wasteful and can immediately lead to a huge number of parameters. In CNNs instead of making full connections, neurons in a layer are only connected to a small region of the previous layer. Mathematically speaking, CNNs use convolution operation instead of general matrix multiplication. In addition convolution operator allows the CNN to be translation equivariant, which means the translation operated on the input image is detectable in the output feature set [21]. A typical CNN has three major building blocks:

1. Convolution layer
2. Activation layer (commonly ReLU)
3. Pooling layer

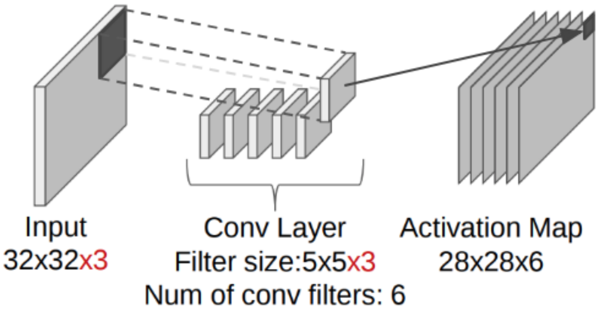


FIGURE 3.4: Example of a convolution layer with 6 convolution kernels of size 5 × 5. Input has 3 channels of size 32 × 32. Each kernel is expanded to fit input channel-size. Each convolution kernel results in an activation map.

3.4.1 Convolution layer

In a discrete 2D setting, convolution operation is defined as

$$s(i,j) = (x * W)(i,j) = \sum_{m}^{M}\sum_{n}^{N} x(i-m, j-n)W(m,n), \tag{3.6}$$

where W is called the convolution kernel/filter of size $M \times N$, and s is referred to as feature/activation map. In CNNs, weights are spatially arranged as convolution kernels. As Figure 3.4 and Equation 3.6 show, for each pixel (i,j) in a feature map, convolution of the corresponding kernel and input is computed. Each convolution filter results in a separate activation map. The final activation volume will be the input volume to the next CNN's layer.

Usually in shallower layers more primary features (e.g. edges) are detected, and deeper layers can learn to capture more complex features by combining simpler ones from previous layer. In addition, the receptive field of neurons, which is the region of the input volume that a neuron is affected by, increases in deeper layers. This means neurons in deeper layers can be indirectly connected to a large part of input image [49].

Strided convolution

By setting a stride for the convolution one can decide on the step size of the kernel when it traverses over the input image. If the stride is larger than 1, input image is down-sampled [172]. This down-sampling can decrease the computational cost and make feature representation more manageable.

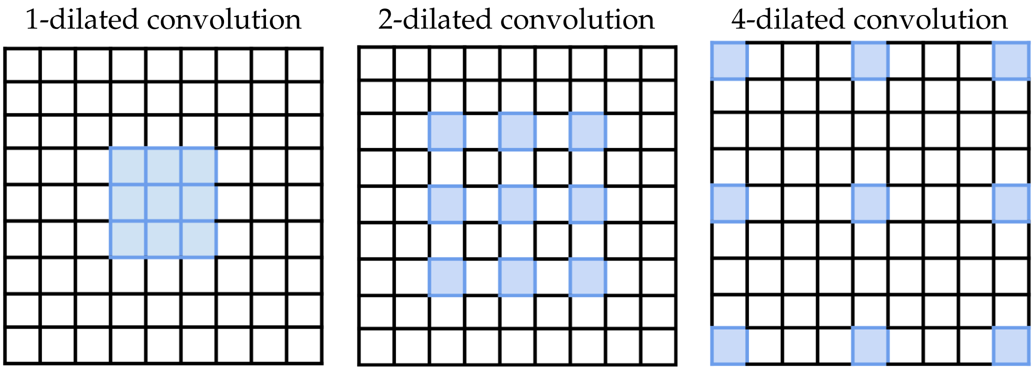


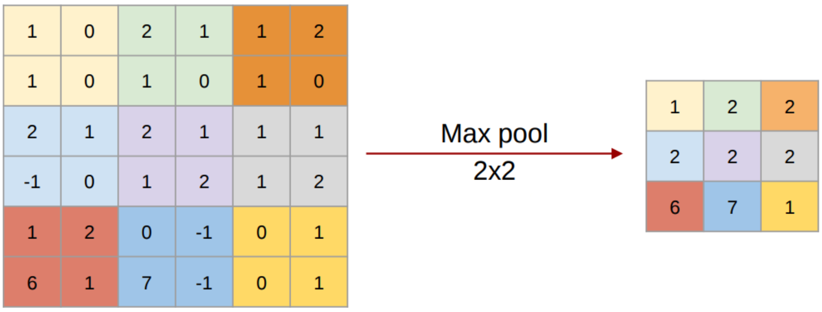
FIGURE 3.5: Dilated convolution with a 3 × 3 convolution kernel. Blue squares indicate the pixels of the input image, used in the convolution. Larger dilation increases the receptive field of the CNN.

Dilated convolution

Yu and Koltun [203] proposed using dilated convolutions for improving semantic image segmentation. Figure 3.5 schematically shows the extent of a dilated convolution with different dilation values. Dilation convolutions allow exponential expansion of the receptive field without loss of resolution or coverage [203]. This type of convolution layer is specially useful when one desires to reach a high coverage over the input image without making the network too deep.

3.4.2 Pooling Layer

Another down-sampling technique is pooling layer, which replaces a neighborhood with its summary statistics. For instance maximum pooling [211], minimum pooling, average pooling, or L^2 norm pooling, replace a rectangular neighborhood with its maximum, minimum, average or L^2 norm value. Using summary statistics of sub-regions means that if the input is translated by a small amount, the output of the pooling do not change, i.e., pooling helps to reduce the effect of small translations of input [49].



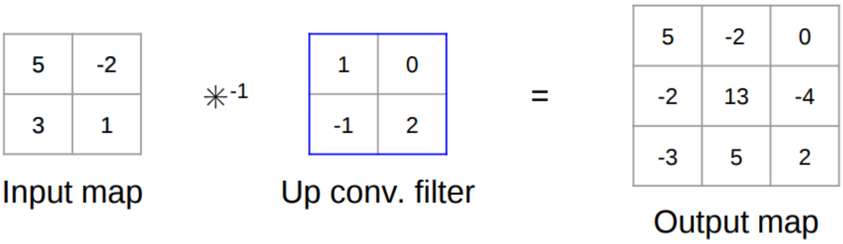
1	0	2	1	1	2
1	0	1	0	1	0
2	1	2	1	1	1
-1	0	1	2	1	2
1	2	0	-1	0	1
6	1	7	-1	0	1

1	2	2
2	2	2
6	7	1

FIGURE 3.6: Example of 2 × 2 maximum pooling with stride (steps) size 2.

3.4.3 Up Convolution

Some CNNs also contain up convolution layer, also known as deconvolution layer. This layer is used to learn an up-sampling step, as illustrated in Figure 3.8. It works as a reverse of convolution. With the up-convolution, the elements of input work as up convolution filter multipliers as they are being copied to the output. Values are summed up in points where they overlap. The following example clarifies this:



$$\begin{bmatrix} 5 & -2 \\ 3 & 1 \end{bmatrix} *^{-1} \begin{bmatrix} 1 & 0 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} 5 & -2 & 0 \\ -2 & 13 & -4 \\ -3 & 5 & 2 \end{bmatrix}$$

FIGURE 3.7: Elements of input work as up convolution filter multipliers. In this example $*^{-1}$ is the up-convolution operation.

3.5 Learning the weights

Learning is the process of finding proper weights w to optimize the neural network with respect to an ideal function. This is usually done by setting the weights to some initial random values. Then the weights are iteratively updated so that the mapping from the input training data matches the ground truth. The matching quality between the prediction from the network and the true labels is realized via a loss function.

3.5.1 Loss Function

Loss (cost) function quantifies how close a given neural network is to an ideal, with respect to a training data-set [49]. For classification tasks, hinge loss and cross entropy loss are the most commonly used loss functions, where the first is well suited for hard classification problems and the second is best when probabilities are of greater interest. For a multi-class classification, the cross entropy loss is defined as

$$L_i = -\sum_{l=1}^{M} y_{i,l} \log p_{i,l}, \tag{3.7}$$

where M is the number of classes, $y_{i,l}$ is a binary value, which is 1 when the class label l is the correct classification class for observation i, and $p_{i,l}$ is the probability of observation i to belong to class l.

3.5.2 Weight update

Neural networks are trained using gradient descent algorithm. In other words, in order to reduce loss L, each weight parameter is updated in the direction of the negative gradient of loss w.r.t. the same parameter.

Backpropagation of gradients

Given a random initialization for weights w, gradient descent iteratively minimizes the loss function as follows:

$$W_{t+1} = W_t - \epsilon_t \nabla_W L \left(x^{(i:i+n)}, W_t \right), \tag{3.8}$$

where t is the iteration step; $\nabla_W L$ is the gradient of loss with respect to current weights W_t; ϵ is the *learning rate*; and x is a batch of N vectors, and $x^{(i:i+n)}$ is a mini-batch of n vectors that have been sampled from it. In this notation, in each iteration, L is computed over n input subjects. Using the same mini-batches in each epoch can lead to a bias in gradient estimation [49], thus in this notation the mini-batches are randomly drawn i.i.d. from the available data. More details on choosing n is provided in the next subsection.

Each iteration consists of a forward and a backward pass. Through the forward pass, the values are propagated from the input x through the hidden layers to a predicted output, where loss (L) is computed. Gradient of loss ($\nabla_W L$) is computed through back-propagating [150] the gradients to the beginning of the network. Weights are updated to minimize the loss. In order to compute the gradient of each node of the network during back-propagation the *chain rule* is applied [49].

For training deep networks, it is common to start with larger learning rates and then gradually decrease it with respect to iteration number t. There are more complex learning rate decays such as exponential decay or $\frac{1}{t}$ decay. In Chapter 7 a fixed number of epochs are used to decrease the learning rate.

Gradient descent optimization algorithms

There are different ways of updating the weight parameters, each well suited for different type of optimization tasks. If the entire training set is used in one single batch ($n = N$, where N is the size of training set) in the optimization algorithm, the optimization is called *batch gradient descent* [49]. However for large data-sets using the complete training set is challenging and very slow. In addition it is not easy to incorporate new data into training. Instead, one can use one training subject ($n = 1$) for weight update in each iteration. This optimization algorithm is called *stochastic gradient descent (SGD)*, which addresses both of the potential problems of batch gradient descent. However the downside of this approach is that updating the weights per training subject makes the steps taken towards the minima noisy. This can lead to a slower convergence to the minima of the loss function. As a solution the *mini-batch gradient descent* falls somewhere between the two methods and takes more than one and less than the complete training set in each iteration [49].

As learning with SGD can be slow sometimes, the *momentum* algorithm [131] can be used to speed up the learning. With hyperparameter momentum $\mu \in [0, 1)$ and velocity v, the weight update rule is defined as

$$\begin{aligned} v_{t+1} &= \mu \times v_t - \epsilon \nabla_W L(x, W_t) \\ W_{t+1} &= W_t + v_{t+1}. \end{aligned} \tag{3.9}$$

Velocity v is gradually increased over the training iterations if the gradient is consistent. It makes the optimization steps not only depend on the current sample, but also heavily on the previously seen samples. In this thesis SGD with momentum ($\mu = 0.99$) is used, which is the default in U-Net architecture [146]. Adaptive Moment Estimation (Adam) [83] is another optimizer that uses the first and second moments of the gradients to compute an adaptive learning rate for each parameter. This optimizer scales the learning rate with respect to the exponential moving average of the gradients. The authors show that this optimizer works well compared to other adaptive learning method algorithms. There are many research has been done in developing a fast optimizer. Other optimizers include Nesterov accelerated gradient (NAG) [120], Nesterov-accelerated Adaptive Moment Estimation (Nadam) [32] and Adadelta [207]. Sebastian Ruder [149] reviews and compares these algorithms.

3.5.3 Weight initialization

When SGD is applied to loss functions, there is no guarantee for convergence and the result is sensitive to the initial values of the parameters [49]. For instance initializing all the weights with 0 will lead to exactly the same weight update everywhere which is not desirable. Another way is to initialize the weights with small random numbers picked using a zero-centered Gaussian distribution with 0.01 standard deviation. This works fine for shallow networks, however it could cause early death of neurons if used in deeper settings.

Xavier et al.[48] pick w_is randomly from a Gaussian distribution with $\sqrt{\frac{1}{N}}$ as the standard deviation, where N is the number of input connections to each neuron. Since at each layer the weighted sum of inputs is computed, picking w_is with variance $\frac{1}{N}$ better preserves the variance of the input, which helps to avoid saturation of neurons in deeper layers. Saturation phenomenon happens when the neuron outputs values near the asymptotic end of its activation function. For instance with sigmoid activation function, which has non-zero mean, at the asymptotic ends the gradient in the backward flow becomes almost zero. This prevents lower layers from learning useful features.

He et al. [63] showed that, when ReLU activation function is used, weight initialization via unit Gaussian with standard deviation of size $\sqrt{\frac{2}{N}}$ works better. In this thesis the approach by He el al. is used for weight initialization. Biases are initialized with zero.

3.6 Regularization

3.6.1 Network size

Typically deep learning algorithms are applied to complex domains such as images, text or audio. For each application it is not easy to find a model of the right size or number of parameters. By increasing the number of layers and neurons in a network, the capacity for learning complex functions increases. However the high capacity of the network can lead to overfitting problem. An overfitted network performs very well on training data, but not on new inputs. To avoid overfitting problem, regularization can be used. Here three of the regularization techniques are presented.

L_2 regularization of loss

One of the simplest ways for regularization is to use parameter norm as penalty, i.e., the loss function is penalized using the sum of all squared weights,

$$L = \frac{1}{N}\sum_i L_i + \lambda \left\| W \right\|_2^2 . \tag{3.10}$$

Parameter λ is used to adjust the intensity of the regularization. L^2 regularization is also known as ridge regression or Tikhonov regularization [49].

Drop out

With drop out technique, during training neurons (including their input and output connections) are randomly dropped. This prevents the neurons from co-adapting [173].

Data Augmentation

A good classifier should be invariant to different transformations of input. Therefore one can extended the training set by producing augmented data via transforming input x for the same label y. For example images can be randomly cropped, translated, mirrored, or warped to greatly improve generalization [49]. This approach is specially well suited for image segmentation, which can be regarded as a pixel-wise classification problem. However in this case the labels must be adapted by undergoing the same transformations input.

3.6.2 Batch Normalization

The gradient-based updating of weights is under the assumption that other layers do not change while updating each parameter. However, in practice, all layers are updated simultaneously. Changing the parameters in a layer can result in a different output distribution, which can lead to unexpected results. Particularly in deeper networks smaller changes in shallower layers can escalate as they propagate through the network, and consequently slow down the training process. To tackle this issue, Ioffe and Szegedy [67] proposed batch normalization, which normalizes the data through the network. Usually the batch normalization step is done before the activation functions, by subtracting the batch mean from the output of the previous layer and dividing it by the batch standard deviation. Using batch normalization within the network can help in different ways, such as faster learning rate, higher accuracy, and less sensitivity to weight initialization [67].

3.7 Image Segmentation Using CNNs

In 2015, Long et al. [107] proposed replacing the fully connected layers in the existing well-known classification architectures with convolution layers creating fully convolutional networks (FCNs). This enabled to create spatial maps instead of classification scores for imagary data. However these likelihood maps had a far lower resolution than the input due to pooling layers. Therefore, Long et al. [107] upsampled the likelihood maps into the input dimension via a final layer with deconvolutional filters. Their model not only allowed inputs with arbitrary sizes, but also improved segmentation accuracy substantially [44].

Noh et al. [123] proposed the deconvolution network by using a deeper deconvolution path, with multiple layers of unpooling and deconvolution to learn the up-sampling. The unpooling layer reverses the pooling operation by using switch variables, that are used to remember the original location of the activations while pooling. Noh et al. [123] use a convolution network (path), which corresponds to a feature extractor, and a deconvolution network (path), which uses the extracted features to produce segmentation maps.

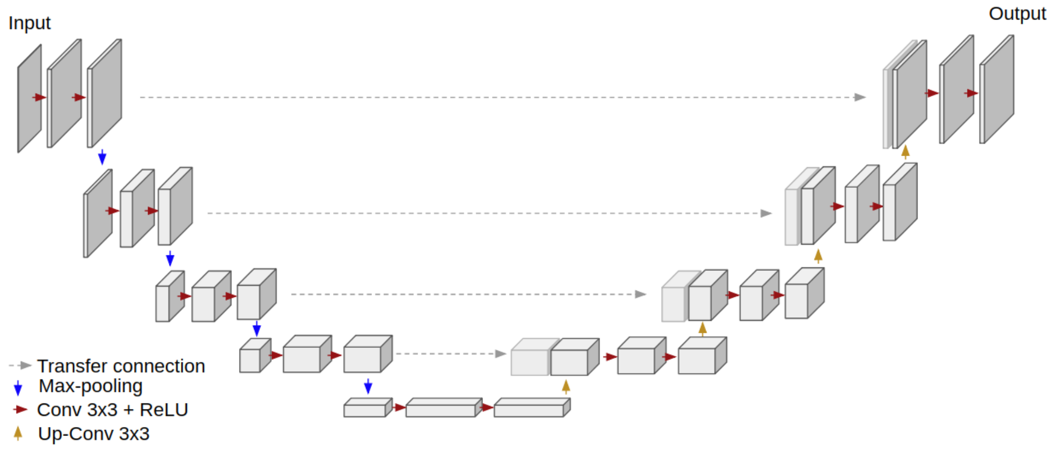


FIGURE 3.8: U-Net architecture.

Later on Ronneberger et al. [146] built the U-Net architecture upon the idea of FCNs and deconvolution network. In their architecture authors used skip-connection between down-sampling and up-sampling paths, which allowed to improve the segmentation even further. Next section provides more details on this architecture. In 2016, Çiçek et al. [20] extended the idea of U-Net for volumetric input data.

Further improvements on the U-Net architecture were proposed later on. For example Milletari et al. [118] used U-Net layout with ResNet-like residual blocks and a Dice loss layer to minimize the segmentation error. In 2018, Roy et al. [148] showed that employing concurrent spatial and channel-wise squeeze and excitation blocks can further improve segmentation performance. Zhou et al. [212] was one of the many to used the idea of the deeply-supervised nets [97] to improve the U-Net. With the deeply-supervised approach, supervision through loss function is provided not only at the output layer, but also at the hidden layers. Another common extension to U-Net is the idea of spatial pyramid pooling. For instance Asgari et al. [8] shows that U-Net with spatial pyramid pooling layers better segments drusen in OCT scans. Garcia-Garcia et al. [44] extensively compares many different segmentation algorithms with respect to their efficiency, accuracy multi-modality and etc.

3.7.1 U-Net Architecture

In this thesis the original U-Net architecture [146], which was the state of the art for medical image segmentation at the time, was picked for semantic segmentation in OCT data. U-Net consists of two major paths, i.e., the contracting and the expanding paths, with multiple transfer connections between them as illustrated in Figure 3.8. Each level of the contracting path consists of two back to back 3×3 convolution layers, followed by ReLU activation functions. Then through 2×2 maximum pooling layers, the resolution of feature maps are decreased. In the original U-Net architecture, this pattern is repeated for four levels until the final bottle neck level is reached. At this stage due to convolution and pooling operations the resolution of the segmentation is highly reduced, thus the expanding path of the network is designed to upsample the prediction. The expanding path consists of four levels, each with two 3×3 convolution layers followed by ReLU, and up convolutions to do the upsampling job. In each level, the last feature map from the corresponding level in contracting path is cropped and concatenated to the first feature map in expanding path. This helps restoring small details in the segmentation and gives the network a symmetric U-shape architecture. Since through the network, image boundary information is lost, Ronneberger et al. [146] propose to pad input images by mirroring them at their boundaries. They also proposed an overlap-tile strategy to allow segmentation for arbitrarily large input images.

The expanding path makes U-Net a suitable architecture for fine-grained image segmentation, which is essential for various medical image analysis problems.

Chapter 4

Ridge and valley enhancement using partial differential equations

Many drusen segmentation algorithms that work on OCT images, first segment the RPE layer from retina [18, 40, 51, 72]. For instance Chen et al. [18] finds the RPE layer by finding the center-line of an initial estimation of RPE. To find this initial estimation, first the noise of B-scan is reduced by applying bilateral filter. Then histogram statistics are used to find a threshold for binarizing the B-scan. However, specially for tubular structure detection and segmentation [35, 36, 88, 89, 93] or finding the approximate core-line of elongated, anisotropic shapes such as RPE layer, ridge-enhancing filters can be a more effective preprocessing step. In this chapter a brief background on ridges and valleys, and partial differential equations that can help enhancing them is provided.

4.1 Ridges and Valleys

So far different definitions for ridges and valleys has been proposed and there are discussions about finding out which definition is correct [87]. If the intensity value at each pixel is interpreted as height, then an image can be viewed as a heightfield or a landscape. With this in mind, one of the intuitive definitions for ridges/valleys is the watershed definition, that defines a watershed ridge as a line structure that separates the rain water to flow into two different valleys of a land [189]. With this definition, if we assume a 2D image u as a scalar field, ridges can be extracted from the gradient vector field $g_{i,j}$ and hessian tensor field $H_{i,j}$ of u at pixel position (i, j). Valleys are the counterpart of a ridge, defined as ridges of inverted image $-u$. Ridges and valley together are referred to as creases. One approach for estimating gradients and hessians is to first reconstruct a continuous heightfield from the discrete samples of image u, and then analytically compute the gradients and the hessians densely. However this approach is quite costly. Instead, one can compute these values discretely, and then interpolate between the precomputed gradients and hessians if needed. In a discrete setting $g_{i,j}$ and $H_{i,j}$ are defined as

$$g_{i,j} = \begin{bmatrix} \frac{u_{i+1,j}-u_{i-1,j}}{2h_x} \\ \frac{u_{i,j+1}-u_{i,j-1}}{2h_y} \end{bmatrix} , \tag{4.1}$$

$$H_{i,j} = \begin{bmatrix} D_{xx}u & D_{xy}u \\ D_{yx}u & D_{yy}u \end{bmatrix} , \tag{4.2}$$

$$\text{where} \quad D_{xx}u = \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h_x^2} , \tag{4.3}$$

$$D_{yy}u_{i,j} = \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j-1}}{h_y^2} , \tag{4.4}$$

$$\text{and } D_{xy}u_{i,j} = D_{yx}u_{i,j} = \frac{u_{i-1,j-1} + u_{i+1,j+1} - u_{i-1,j+1} - u_{i+1,j-1}}{4h_x^2 h_y^2} , \tag{4.5}$$

where h_x and h_y are the pixel size in x and y directions. By extracting the topological skeleton of the gradient vector field g, ridges are extracted [189]. The topological skeleton of a 2D vector field is a set of *critical points* and *separatrix* stream lines. Depending on the eigenvalues of the underlying Hessian matrix $H_{x,y}$, a critical point can fall into 6 different categories; 1) Repelling node (also known as source or maximum), where both eigenvalues are positive and real, 2) Attracting node (also known as sink or minimum), where both eigenvalues are negative and real, 3) Saddle point, where eigenvalues have different signs and are real, 4) Center, where both eigenvalues are imaginary and equal but with different signs, 5) Repelling focus, where the real part of both eigenvalues are positive and equal, and their imaginary parts are equal with different signs, 6) Attracting focus, which is similar to repelling focus except that the real parts are negative. Since Hessian matrix of a smooth function is symmetric and real, thus the gradient vector field of image u can only have the first three types of critical points. A separatrix stream line separates different regions with different asymptotically flow behaviour. One way to extract separatrices is to follow the eigenvectors of the Hessian at saddle points. More details on topological structures of a gradient vector field can be found in the Ph.D. thesis of Tino Weinkauf [189].

One major drawback of the watershed ridge and valley definition is that one cannot determine if a point is on a ridge or valley line (separatrix), only by looking at its local neighborhood. This can be achieved only through a global process. Thus Eberly et al. [34] proposed an alternative definition, called height ridge definition, that has similar results to the watershed ridge definition in practice, and also allows to locally check if a point is on a ridge or valley line.

Eberly et al. [34] define ridges in 2D gray-scale images as points of image u, that are a local maximum in at least one direction. Eigenvalues of the Hessian matrix at each point (i, j) corresponds to the two principal curvatures of the surface at that point. When the surface is smooth, the hessian matrix is symmetric, therefore it has real eigenvalues λ_1, λ_2, and orthogonal eigenvectors ν_1,ν_2. In definition by Eberly et al. [34], ridges are:

$$\begin{cases} \text{0-dimensional ridge} & |g| = 0 \text{ and } \lambda_{1,2} < 0 \\ \text{1-dimensional ridge} & g \cdot \nu_2 = 0 \text{ and } \lambda_2 < 0 \end{cases} , \tag{4.6}$$

where $\lambda_1 \geq \lambda_2$ are eigenvalues of H and $\nu_{1,2}$ are the corresponding eigenvectors. Mathematically speaking, ridge lines are where the projection of gradient vector on the eigenvector with strongest negative eigenvalue (convexity) vanishes. Given these definitions, next section describes how ridges can be extracted from scalar fields.

4.2 Ridge extraction

Peikert and Sadlo [127] suggested using raw features in order to extract ridges as defined in Eq. 4.6. Raw feature points of a 2D image is defined as the zero contour of scalar field d:

$$d = \det(g|Hg), \tag{4.7}$$

where det(.) is the determinant, and $g|Hg$ is a matrix whose first column is gradient vector g and its second column is the vector resulted from Hg multiplication. The extraction of ridges in this thesis was inspired by the method proposed in [164], that uses *marching square algorithm* [163].

In order to be able to find raw feature points even within pixels, one can linearly interpolate g and H on pixel edges. For instance for the pixel edge between pixel (i,j) and $(i+1,j)$, $f_g(\alpha)$ is the the function that computes g at position α as

$$f_g(\alpha) = \begin{pmatrix} [g_{i.j}]_x + \alpha\left([g_{i+1,j}]_x - [g_{i,j}]_x\right) \\ [g_{i.j}]_y + \alpha\left([g_{i+1,j}]_y - [g_{i,j}]_y\right) \end{pmatrix}, \tag{4.8}$$

and $f_H(\alpha)$ linearly estimates the hessian on the same edge as

$$f_H(\alpha) = \begin{pmatrix} [H_{i,j}]_{xx} + \alpha\left([H_{i+1,j}]_{xx} - [H_{i,j}]_{xx}\right) & [H_{i.j}]_{xy} + \alpha\left([H_{i+1,j}]_{xy} - [H_{i,j}]_{xy}\right) \\ [H_{i,j}]_{yx} + \alpha\left([H_{i+1,j}]_{yx} - [H_{i,j}]_{yx}\right) & [H_{i,j}]_{yy} + \alpha\left([H_{i+1,j}]_{yy} - [H_{i,j}]_{yy}\right) \end{pmatrix}.$$

It may happen that a ridge ends somewhere inside a pixel (not on edges). These ending points are called the degenerate points of tensor field H and can be extracted by solving the following system

$$\begin{aligned} &(1-\alpha)(1-\beta)H_{i,j} + \alpha(1-\beta)H_{i+1,j} \\ &+(1-\alpha)\beta H_{i,j+1} + \alpha\beta H_{i+1,j+1} = \begin{pmatrix} c & 0 \\ 0 & c \end{pmatrix}, \end{aligned} \tag{4.9}$$

here c is constant. This equation can have up to two solutions for $0 \leq \alpha \leq 1$ and $0 \leq \beta \leq 1$.

Given Equations 4.8, 4.2, and 4.9, with marching square algorithm, for every pixel, roots of d (Equation 4.7) are computed, to extract raw feature points. The extracted raw feature points are connected to each other if their connection minimizes the average of d in the middle of connection lines, as well as minimizing the number of intersections between connection lines. Finally the degenerate points are connected to raw feature points, satisfying the same minimization conditions.

4.3 Scale space

In the previous section we saw that the extraction of ridges depends on estimation of curvature of the image function using Hessians and image gradient, where both are estimated in a small neighborhood. On the other hand, in real applications, ridges can appear at different scales. For example, in an angiography image, vessels can have different cross-sectional thickness. For this reason, for noisy images or for ridges in higher scales, an extra step is necessary to know the right scale of the structures of interest. Figure 4.1 shows that how neglecting the scale of ridges can result in very poor ridge extraction. *Scale space* framework can be helpful in this regard. A scale space framework consists of the original image and simplified versions of the original, where the level of simplification is increased by increasing the scale.

Gaussian scale space is one of the well known scale spaces, that is constructed by convolving original image u with a Gaussian kernel

$$G_\sigma = \frac{1}{2\pi\sigma^2} \mathrm{e}^{-\frac{(x^2+y^2)}{2\sigma^2}} ,$$

where σ indicates the scale of the resulting image. Larger σ increases the width of the Gaussian kernel, therefore the convolution will result in a smoother image.

4.4 Scale selection using vesselness measure

In addition to the scale space representation of the image, a method is needed to automatically estimate the scale of the underlying ridge at each point.

For automatic scale selection, Frangi et al. [42] introduced the *vesselness measure* that can as well be used for background suppression. The vesselness measure uses the eigenvalue and eigenvector information of the Hessian H_σ at scale σ

$$H_\sigma = \sigma^2 \begin{bmatrix} \frac{\partial^2 u_\sigma}{\partial x^2} & \frac{\partial^2 u_\sigma}{\partial x \partial y} \\ \frac{\partial^2 u_\sigma}{\partial y \partial x} & \frac{\partial^2 u_\sigma}{\partial y^2} \end{bmatrix} , \tag{4.10}$$

where $u_\sigma = u * G_\sigma$ and σ^2 is multiplied to the Hessian matrices to compensate for loss of intensity contrast. Otherwise, the magnitude of derivatives for larger σ would be reduced, making it hard to compare the values at multiple scales [101]. Given λ_1 and λ_2 to be the eigenvalues of H_σ with $|\lambda_1| \leq |\lambda_2|$, an ideal ridge is where

$$\lambda_1 \approx 0 \quad \text{and} \quad |\lambda_1| \ll |\lambda_2| . \tag{4.11}$$

On the other hand, if $\lambda_1 = \lambda_2$, the underlying structure is a perfect blob. In order to avoid them, Frangi et al. [42] use

$$\mathcal{R}_\mathcal{B} = \frac{\lambda_1}{\lambda_2} , \tag{4.12}$$

with $\mathcal{R}_\mathcal{B} \in [0, 1]$. For suppressing background pixels, Frangi et al. [42] use the fact that the magnitude of derivatives over tubular structures are larger compared to background. This is measured by $\mathcal{S}$ as

$$\mathcal{S} = \|H_\sigma\|_\mathcal{F} = \sqrt{\lambda_1^2 + \lambda_2^2} .$$

Using $\mathcal{S}$ and $\mathcal{R}_\mathcal{B}$, vesselness measure is defined as

$$\mathcal{V}_\sigma u = \begin{cases} 0 & \text{if} \quad \lambda_2 > 0 \\ \left(\mathrm{e}^{-\frac{\mathcal{R}_\mathcal{B}^2}{2\beta^2}}\right)\left(1 - \mathrm{e}^{-\frac{\mathcal{S}^2}{2c^2}}\right) & \text{otherwise} \end{cases} , \tag{4.13}$$

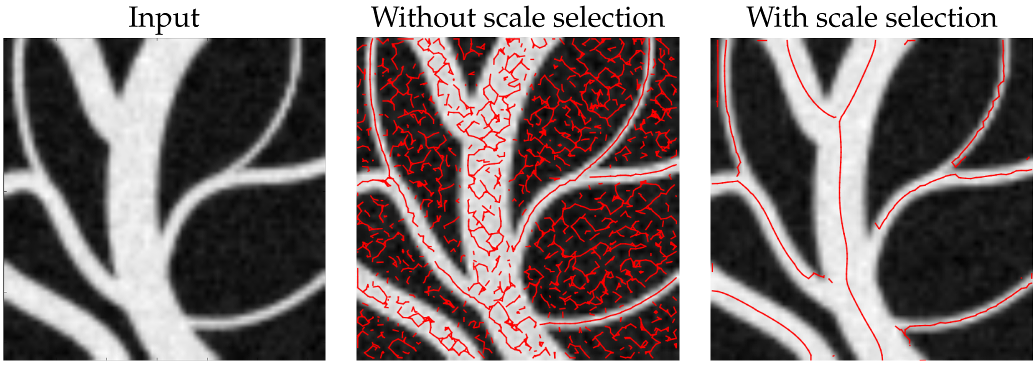


FIGURE 4.1: On the left, the input contains ridges at different scales. In the middle ridges (in red) of input are extracted. On the right the ridges are extracted using Gaussian scale space and vesselness measure for scale estimation.

for bright ridges over a dark background. In order to tune $\mathcal{V}_\sigma u$ to be more sensitive toward tubular shapes or toward suppression of background pixels, Frangi et al. [42] suggests to use $\beta = 0.5$ and $c = \frac{1}{2}(\max(\mathcal{S}))$ since they seem to work fine for most cases.

To estimate the scale of the underlying ridge for each pixel, the scale at which the vesselness measure is maximal is selected

$$\mathcal{V}u = \max_{\sigma=\sigma_{\min},\ldots,\sigma_{\max}} \mathcal{V}_\sigma u. \tag{4.14}$$

Scale selection is done in a range of expected scales $\{\sigma_{\min}, \ldots, \sigma_{\max}\}$ in the image. In order to suppress background, any pixel that $\mathcal{V}u < \theta$ is set to 0. The value of θ can be set empirically.

Another approach for automatic scale selection is proposed by Lindeberg [101], where he uses the normalized *ridge strength*. As in this work the focus is only on enhancing tubular structures, the scale selection using the *vesselness* measure [42] is more suited.

Given the scale of the underlying ridge at each pixel, the Hessian matrix and gradient vector can be computed at the proper scale before the extraction process, which leads to a more plausible result as shown in Figure 4.1.

So far we showed why ridge extraction is important in image analysis, and how Gaussian scale space and vesselness measure can aid the extraction process. Even though Gaussian smoothing helps to remove the noise and to locate large-scale ridges, it blindly smoothes out all underlying structures leading to dislocated ridges and destroyed bifurcations at coarser scales [193]. One way to deal with this problem is to trace ridges back from coarse to fine scale to find their correct location [193]. Alternatively, instead of Gaussian scale space, nonlinear diffusion equations can be used to create a scale space that not only is able to preserve the important underlying structures, but also can help to enhance them.

4.5 Diffusion (heat) equation for image processing

Partial differential equations (PDEs) describe the relation between an unknown multivariable function and its partial derivatives. They can be used to model natural phenomena such as flow of heat through a conductor material [22]. One of the well-known and frequently used PDEs in the context of image processing is the *diffusion (heat) equation*, that considers pixel intensities of image u as a heat distribution. Pixels with higher intensity are considered hot and those with lower intensity as colder. With the diffusion equation the flow of the intensity (heat) over artificial time t is modeled as following

$$\partial_t u = \operatorname{div}(D.\nabla u), \tag{4.15}$$

where D is called the *diffusion tensor*, and div(.) is the divergence operator. This equation shows that the intensity flow happens between regions with intensity difference (∇u), with a speed proportional to D.

4.5.1 Diffusion tensor D

The diffusion process highly depends on how D, that is a positive definite matrix, is defined:

- For a constant diffusion everywhere, one can define $D = g$, with g being a scalar constant. This is called *homogeneous diffusion*. If D is defined as a function of location x, i.e., $D = D(x)$, it is called *inhomogeneous*.
- For a diffusion that depends on the evolving signal u, i.e., $D = D(u)$. This is called *nonlinear diffusion*, presented by Perona and Malik [128].
- For a diffusion that varies in different directions, D must be defined as a tensor, and the diffusion is called *anisotropic*[185].

4.6 Nonlinear diffusion filtering

Nonlinear diffusion filtering can be used to smooth out an image, yet preserve or even enhance its relevant structures. This is achieved by defining the diffusion function to depend on the evolving image and location of diffusion.

4.6.1 Perona and Malik model

In 1990 Perona and Malik [128] introduced the following diffusivity function, that changes the amount of diffusion depending on the underlying gradient magnitude (serving as edge indicator):

$$g(|\nabla u|^2) = \frac{1}{1 + \frac{|\nabla u|^2}{\lambda^2}} \qquad (\lambda > 0), \tag{4.16}$$

where $|\nabla u| = \sqrt{\partial_x u^2 + \partial_y u^2}$. In this equation λ is called the *contrast parameter*. With this formulation, the diffusion converges to 0 over strong edges, but increases over weaker ones. In their diffusion model, Perona and Malik [128] show that, wherever $|\nabla u| > \lambda$, *backward diffusion* occurs. As its name suggests, backward diffusion reverses the smoothing effect, meaning that over time t, edges become even sharper. On the other hand if $|\nabla u| \leq \lambda$, *forward diffusion* smoothes out the underlying edges. This dual diffusivity behaviour depends on the chosen diffusivity function. Didas et al. [30] list some of the well known diffusivity functions and their behaviour.

Choosing the best value for contrast parameter depends on the features of u. For instance in [110] authors select different λ at each pixel with respect to the geometry of the underlying structure. In this work, a constant λ is used all over the image.

4.7 Numerical solver

Some PDEs have a closed form solution (analytical solution), but in most cases such as the diffusion equation with Perona and Malik [128] diffusivity function requires a numerical solution. In addition, PDEs may have more than one unique solution. In such cases boundary (initial) conditions are used to get a unique solution.

In order to find the solution of a PDE at time T, i.e, $u(x; t = T)$, one can start at the initial condition $u(x; t = t_0) = u_0$, then take small steps τ toward future and accumulate the changes numerically $u(x; t = t_0 + k\tau)$ until $t = T$ is reached.

4.7.1 Discretization

The first step of numerically solving a PDE is to discretize it. Given the diffusion equation for 2D signal u

$$\partial_t u = \partial_x(g(|\nabla u|^2)\partial_x u) + \partial_y(g(|\nabla u|^2)\partial_y u) \,, \tag{4.17}$$

one can discretize it using any of the following schems: *Explicit Euler forward discretization*, *Semi-Implicit discretization*, or *Implicit discretization* [188]. The explicit discretization is the simplest one. In matrix-vector notation it can be written as

$$\frac{u^{k+1} - u^k}{\tau} = A(u^k)u^k.$$

Here k is the step number and τ is the step size. Elements of matrix $A(u^k)$ are computed using the diffusivity function g, and a discrete approximation of $|\nabla u| = \sqrt{\partial_x^2 + \partial_y^2}$ with $\partial_x = \frac{u_{i+1,j} - u_{i-1,j}}{h_x}$, where h_x is the pixel size in x-direction, and (i, j) are pixel indices. Derivative in y-direction ∂_y can be discretized similar to ∂_x. This numerical scheme is called ℓ_2 stable if $\|u^{k+1}\|_2 \le \|u^k\|_2$, meaning that through diffusion process ℓ_2 norm of input image u must not increase. In order to guarantee ℓ_2 stability for the explicit diffusion scheme, time step τ must be restricted to very small numbers, which makes this approach very slow. If matrix A is replaced with an arbitrary symmetric, positive semi-definite matrix P, the explicit scheme is stable if the eigenvalues of matrix $I - \tau P$ are in range $[-1, 1]$, meaning that $\tau \le \frac{2}{\rho(P)}$, where $\rho(P)$ is the largest modulus of P's eigenvalues that can be estimated with Gershgorin's theorem. In contrast to the explicit scheme, the semi-implicit scheme

$$\frac{u^{k+1} - u^k}{\tau} = A(u^k)u^{k+1}, \tag{4.18}$$

and fully implicit scheme

$$\frac{u^{k+1} - u^k}{\tau} = A(u^{k+1})u^{k+1}, \tag{4.19}$$

guarantee absolute stability regardless of size of τ. However for larger τ the accuracy decreases. Plus the fully-implicit scheme leads to nonlinear system of equations that are way more complicated to solve [188].

4.7.2 Fast explicit diffusion scheme

Weickert et al. [187] proposed the Fast Explicit Diffusion (FED) that is well-suited to solve parabolic PDEs. Unlike the explicit scheme that requires a fixed step size τ, the fast scheme applies cycles of varying time step sizes. In these cycles, up to 50% of the steps can violate the stability condition, and yet at the end of the cycle approximate a stable filter. Consequently the FED can speed-up the diffusion by several orders of magnitude compared to the explicit scheme. Further details on the FED scheme are presented in Chapter 5.

4.7.3 Regularization

To numerically solve the diffusion equation, the derivative of image u must be computed. However the differentiation is not well-posed, i.e., small changes in the original image, e.g. due to noise, leads to arbitrary changes in its derivatives that do not continuously depend on the initial changes. In addition, the Perona and Malik model can enhance or preserve strong noisy oscillations as important image structures. Therefore it is important to regularize the original image [180]. One way to do the regularization is to use Gaussian smoothing, i.e., $u_\sigma = u * G_\sigma$, where G_σ is a Gaussian with standard deviation σ.

4.7.4 Boundary Condition

As images are signals on a finite domain, the diffusion process at the image borders must be carefully defined. Here Neumann boundary condition is used, which sets the values outside of the boundary so that

$$\partial_n u \,|_{\partial\Omega} = \langle \nabla u, n \rangle \,|_{\partial\Omega} = 0, \tag{4.20}$$

where $\Omega \in [0, M] \times [0, N]$ is the domain of image u, and n is the normal vector of the boundary. With this boundary condition, no intensity will leave or enter the image, making sure the average intensity remains constant during the diffusion process [30].

4.8 Anisotropic nonlinear diffusion filtering

So far the diffusion equations introduced here were all isotropic, meaning that the diffusion process is always parallel to ∇u. However, isotropic diffusion fails to properly enhance or remove noise over the structures of interest, as diffusivity $g \to 0$ in those area. To solve this, Graham [52] proposed the idea of using anisotropic diffusion in image processing that allows to have different diffusivity values in different directions. To steer the diffusion, later on Joachim Weickert used the structure tensors, that can identify the orientation of 1-D structures by considering how the gradient changes in a small neighborhood [70], as follows:

$$S_{\rho,\sigma} := G_\rho * (\nabla u_\sigma \otimes \nabla u_\sigma) := G_\rho * (\nabla u_\sigma . \nabla u_\sigma^T) \,. \tag{4.21}$$

Since the structure tensor is computed from the tensor product of the local gradient vector, its first eigenvector is parallel to ∇u_σ with eigenvalue $\mu_1 = |\nabla u_\sigma|^2$, and its second eigenvector is perpendicular to ∇u_σ with $\mu_2 = 0$ as the second eigenvalue (when $\rho = 0$). With this, it is possible to define two different diffusion functions λ_1 and λ_2, one for describing diffusivity along ∇u_σ and one for the perpendicular direction. These two functions can be put together in a diffusion tensor as

$$D = V \Lambda V^T, \tag{4.22}$$

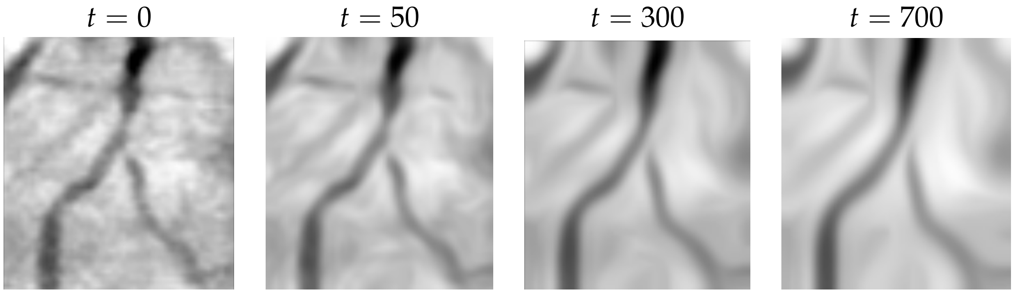


FIGURE 4.2: Coherence-enhancing filter at different time points with $\sigma = 1$, $\rho = 3$, $\lambda = 3.5$ and $\tau = 0.03$. The input image is the same as the one in [15].

where V is a square matrix whose columns are the eigenvectors of S and Λ is a diagonal matrix whose diagonal elements are λ_1 and λ_2. Finally the anisotropic diffusion equation can be written as

$$\partial_t u = \operatorname{div}\left(D\left(\nabla u_\sigma\right)\nabla u\right). \tag{4.23}$$

4.8.1 Coherence-enhancing anisotropic diffusion

In order to enhance creases, Weickert proposed the coherence enhancing anisotropic diffusion[186], where the eigenvalues of the coherence-enhancing tensor is defined as

$$\begin{cases} \lambda_1 := \alpha \\ \lambda_2 := \alpha + (1-\alpha)\exp^{\left(\frac{-1}{(\mu_1-\mu_2)^2}\right)} \end{cases}.$$

In this definition $\alpha \in (0,1)$, and μ_1, μ_2 are the eigenvalues of the structure tensor S. With this definition, over elongated 1-D structures where $\mu_1 \geq \mu_2$, diffusion along the structure (in $\perp\ \nabla u_\sigma$ direction) is increased compared to the across direction, while over the blob-like structures with $\mu_1 \approx \mu_2$, diffusion becomes isotropic.

As mentioned earlier, convolving the image with a Gaussian kernel such as in [87, 101, 109, 116] tends to destroy crease bifurcations or merge them [169]. Using the anisotropic second order diffusion filter in Equation 4.23 improves image coherence, nevertheless it still destroys crease junctions and deforms nonlinear structures as illustrated in Fig. 4.2.

4.8.2 Creaseness enhancing diffusion filter

In order to preserve crease junctions and nonlinear structures, Sóle et al. [169] proposed the *creaseness* enhancing diffusion (CED) filter [1]. In order to estimate surface curvature they normalize the regularized Hessian matrices and define a multilocal normalized Hessian by convolving them with a Gaussian kernel G_ρ,

$$H_{\rho,\sigma} = G_\rho * \frac{1}{\sqrt{1+\|\nabla u_\sigma\|^2}} \begin{bmatrix} \partial_{xx}u_\sigma & \partial_{xy}u_\sigma \\ \partial_{yx}u_\sigma & \partial_{yy}u_\sigma \end{bmatrix}.$$

[1] In the literature sometimes CED filter is used as an abbreviation for coherence enhancing diffusion, introduced by Joachim Weickert [186]. In this thesis CED corresponds to creaseness enhancing diffusion filter.

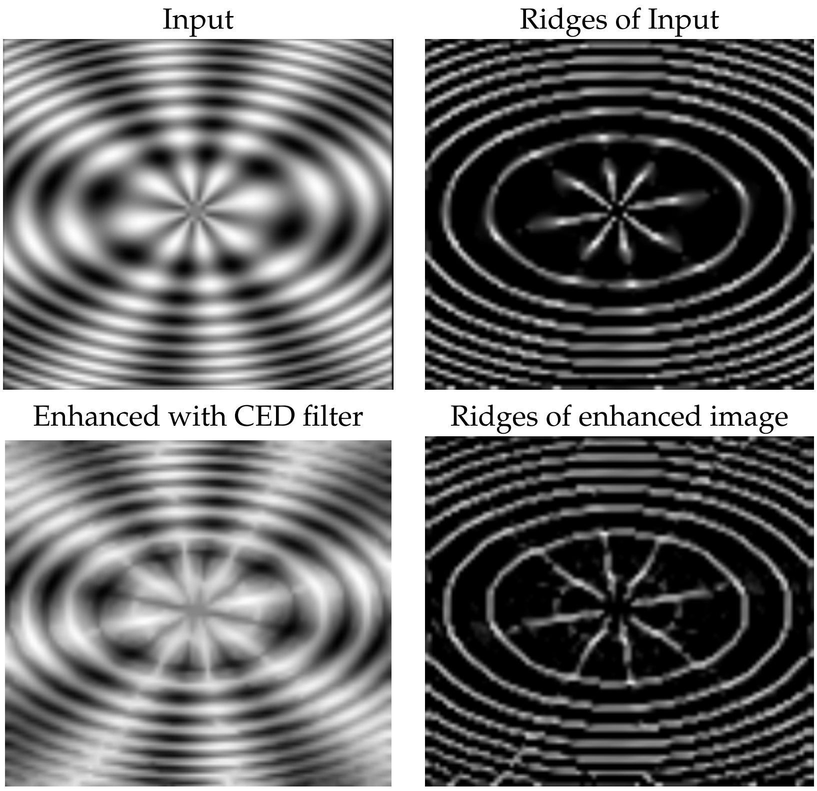


FIGURE 4.3: CED filter for enhancing ridges. In this experiment, $\alpha = 1, \beta = 0.5, \sigma = 1$ and $\rho = 0.5$.

Then Sóle et al. [169] define a creaseness measure as

$$\mu = \frac{\widetilde{k}_1 - \widetilde{k}_2}{\widetilde{k}_1 + \widetilde{k}_2},$$

where $\widetilde{k}_1 = \max(|k_1|, |k_2|)$ and $\widetilde{k}_2 = \min(|k_1|, |k_2|)$ and $k_{1,2}$ are the eigenvalues of $H_{\rho,\sigma}$. In order to be able to selectively enhance either ridges or valleys, they define ridgeness μ_r and valleyness μ_v measures, considering the sign of strongest eigenvalue (k_1)

$$\mu_r = \begin{cases} \frac{\widetilde{k}_1 - \widetilde{k}_2}{\widetilde{k}_1 + \widetilde{k}_2} & \text{if } k_1 < 0 \\ 0 & \text{if } k_1 \geq 0 \end{cases} \quad \text{and} \quad \mu_v = \begin{cases} 0 & \text{if } k_1 \leq 0 \\ \frac{\widetilde{k}_1 - \widetilde{k}_2}{\widetilde{k}_1 + \widetilde{k}_2} & \text{if } k_1 > 0 \end{cases}.$$

Finally μ_r and μ_v are used to define the eigenvalues of the *creaseness diffusion tensor*, i.e.,

$$\begin{cases} \lambda_1 := \epsilon \\ \lambda_2 := \alpha\mu_r + \beta\mu_v \end{cases},$$

where $\alpha, \beta \in [0, 1]$ and are there to tune the amount of diffusion in presence of ridges or valleys. Along the crease, diffusion is set to λ_2 and in the perpendicular direction, to $\epsilon \in (0, 1)$. Using the eigenvectors of $H_{\sigma,\rho}$, creaseness diffusion tensor D is computed using Equation 4.22. Figure 4.3 shows how CED filter can be used to enhance mostly ridges. To roughly extract ridges for this visualization, the Multilocal level-set extrinsic curvature (MLSEC) [108] was used. With this extraction method (unlike the previous method in Section 4.2), maximum resolution for ridges is one pixel.

Even though CED filter improves crease enhancement at junctions and over non-linear structures, it has two drawbacks:

- Image contrast is not taken into account to steer the filter, leading to unwanted enhancement of undesired background structures.
- The filter is only able to enhance creases that are at a certain scale (constant value of parameter σ).

4.8.3 Vesselness enhancing diffusion filter

In order to overcome the issues of CED filter, Cañero et al. [15] introduced the vesselness enhancing diffusion (VED) filter that uses the vesselness measure (see Section 4.4) to detect and suppress background enhancement and to enhance creases at multiple scales. Cañero et al. [15] define their vesselness diffusion tensor using the following eigenvalues

$$\begin{cases} \lambda_1 := & \epsilon \\ \lambda_2 := & \mathcal{V} \cdot \frac{\widetilde{k}_1 - \widetilde{k}_2}{\widetilde{k}_1 + \widetilde{k}_2} \end{cases},$$

where $\mathcal{V}$ is computed by Equation 4.14, and $\widetilde{k}_1, \widetilde{k}_2$ are computed from eigenvalues of the multilocal normalized Hessian, which is computed at the automatically selected scales.

4.9 Fourth order diffusion equations

All the diffusion filters discussed so far were second order filters. When these filters are used to enhance features, they lead to *staircase artifact* as the solution for these filters are piecewise constant [202]. Figure 4.4 shows the staircase artifact. One way to overcome this problem is to increase the order of derivatives in the diffusion equation. Lysaker et al. [112] show that using a fourth order diffusion equation not only removes the noise faster, but also leads to piecewise linear solutions, that is desirable for preserving ramp edges. However for sixth order diffusion equation, Lysaker et al. observered ripples appearing near step-like edges. Thus they concluded the fourth order diffusion filter is best suited for image enhancement as it does not suffer from staircase artifact and does not lead to unwanted ripples. The fourth order diffusion equation proposed in [112] is defined as follows:

$$\begin{aligned} \partial_t u = & - \partial_{xx}(D(\|H(u)\|_F^2)u_{xx}) - \partial_{yx}(D(\|H(u)\|_F^2)u_{xy}) \\ & - \partial_{xy}(D(\|H(u)\|_F^2)u_{yx}) - \partial_{yy}(D(\|H(u)\|_F^2)u_{yy}) \end{aligned} \tag{4.24}$$

where $\|H(u)\|_F$ is the Frobenius norm of matrix $H(u)$.

4.9.1 Crease enhancement

In 2009 Didas et al. [30] showed that depending on the underlying diffusivity function, the fourth order diffusion Equation 4.24 can be used to enhance image curvature. For example when Perona and Malik diffusivity function is used, in regions with $|\partial_{xx}u| > \sqrt{3}\lambda$, second and fourth order backward diffusion occurs, leading to edge and curvature enhancement. Over regions with $|\partial_{xx}u| < \lambda$ forward diffusion smooths the signal, and over the remaining parts second order forward diffusion and fourth order backward diffusion occur.

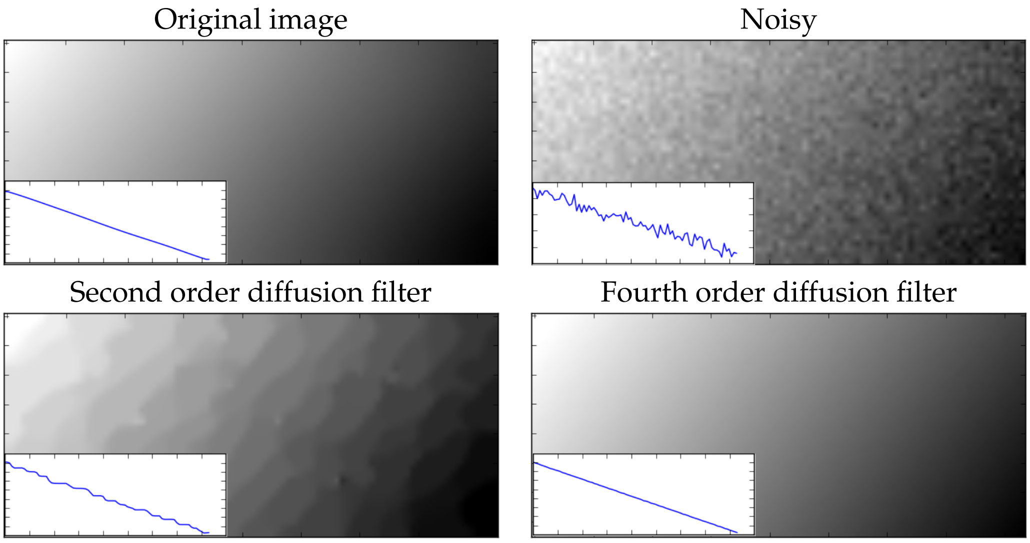


FIGURE 4.4: Unlike second order diffusion filter with Perona and Malik diffusivity function, fourth order diffusion filter with the same diffusivity, restores ramp edges without causing staircase artifact.

To numerically solve Equation 4.24, it can be discretized as

$$\begin{aligned} u^{k+1} = u^k - \tau \Big(& D_{xx} \left(g(\left\| H^k_{i,j} \right\|_F) D_{xx} u^k_{i,j} \right) + D_{xy} \left(g(\left\| H^k_{i,j} \right\|_F) D_{yx} u^k_{i,j} \right) \\ & + D_{yx} \left(g(\left\| H^k_{i,j} \right\|_F) D_{xy} u^k_{i,j} \right) + D_{yy} \left(g(\left\| H^k_{i,j} \right\|_F) D_{yy} u^k_{i,j} \right) \Big) \ , \end{aligned}$$

where D_{xx} , D_{xy}, D_{yx} and D_{yy} are defined in Section 4.1. Since creases are elongated structures, it is desirable to enhance them in cross-sectional direction, and at the same time smooth along them. However Equation 4.24 is isotropic, leading to the same diffusion for all directions. It is shown that those images filtered with Equation 4.24 usually have speckle noise [59]. Plus, for getting a good feature preservation effect, the filter has a low convergence rate [112, 201].

4.10 Anisotropic fourth order diffusion equation

Hajiaboli [59] proposed the following non-linear anisotropic fourth order diffusion filter

$$u_t = -\nabla^2 \left(g\left(\|\nabla u\|\right)^2 u_{\eta\eta} + g\left(\|\nabla u\|\right) u_{\xi\xi} \right) \ . \qquad (4.25)$$

where η and ξ are two different directions, defined as:

$$\eta = \frac{1}{\sqrt{u_x^2 + u_y^2}} \begin{bmatrix} u_x \\ u_y \end{bmatrix} \ ,$$

$$\xi = \frac{1}{\sqrt{u_x^2 + u_y^2}} \begin{bmatrix} -u_y \\ u_x \end{bmatrix} \ ,$$

and the second order derivative of image u, in these directions can be computed using:

$$\begin{aligned} u_{\eta\eta} &= \frac{u_{xx}u_x^2 + 2u_x u_y u_{xy} + u_{yy}u_y^2}{u_x^2 + u_y^2}, \\ u_{\xi\xi} &= \frac{u_{xx}u_y^2 - 2u_x u_y u_{xy} + u_{yy}u_x^2}{u_x^2 + u_y^2}. \end{aligned}$$

Given that $g(\|\nabla u\|) \in (0,1]$, thus $g(\|\nabla u\|)^2 \leq g(\|\nabla u\|)$, which leads to less or equal diffusion in η direction compared to the diffusion in ξ direction. Consequently their filter allows to choose different diffusivities along and across the 1-D feature directions. Plus they show, compared to the fourth-order diffusion model introduced by Lysaker et al. [112], their filter has a better convergence rate, and results in less or no speckle noise [59]. In Chapter 5 we introduce a diffusion model that generalizes Hajiaboli's filter. Our proposed filter is particularly designed to enhance ridges or valleys at multiple scales.

Chapter 5

Multi-scale anisotropic fourth-order diffusion improves ridge and valley localization[1]

5.1 Abstract

Ridge and valley enhancing filters are widely used in applications such as vessel detection in medical image computing. When images are degraded by noise or include vessels at different scales, such filters are an essential step for meaningful and stable vessel localization. In this work, we propose a novel multi-scale anisotropic fourth-order diffusion equation that allows us to smooth along vessels, while sharpening them in the orthogonal direction. The proposed filter uses a fourth order diffusion tensor whose eigentensors and eigenvalues are determined from the local Hessian matrix, at a scale that is automatically selected for each pixel. We discuss efficient implementation using a Fast Explicit Diffusion scheme and demonstrate results on synthetic images and vessels in fundus images. Compared to previous isotropic and anisotropic fourth-order filters, as well as established second-order vessel enhancing filters, our newly proposed one better restores the centerlines in all cases.

5.2 Introduction

In image analysis, ridges and valleys are curves along which the image is brighter or darker, respectively, than the local background [35]. Collectively, ridges and valleys are referred to as creases. Reliable detection and localization of creases in noisy images is an important and well-studied problem in medical image analysis, one very common application being the detection of blood vessels [42].

Often, ridges and valleys occur at multiple scales, i.e., their cross-sectional radius varies throughout the image. For example, the stem of a vessel tree is thicker than its branches. Gaussian scale spaces are a classic strategy for extracting creases at different scales [101]. However, the fact that Gaussian filters do not offer any specific mechanisms for preserving creases gave rise to image filters such as coherence enhancing diffusion [185], crease enhancement diffusion (CED) [169], and vesselness enhancement diffusion (VED) [15]. They are based on second order anisotropic diffusion equations with a diffusion tensor that, in the presence of crease lines, smoothes only along, but not across them. In addition, the VED filter includes a multi-scale analysis that automatically adapts it to the local scale of creases.

[1] The content of this chapter has been previously published: Gorgi Zadeh et al. "Multi-scale Anisotropic Fourth-Order Diffusion Improves Ridge and Valley Localization" *Journal of Mathematical Imaging and Vision* 59.2 (2017): 257-269. URL: https://doi.org/10.1007/s10851-017-0729-1 [204].

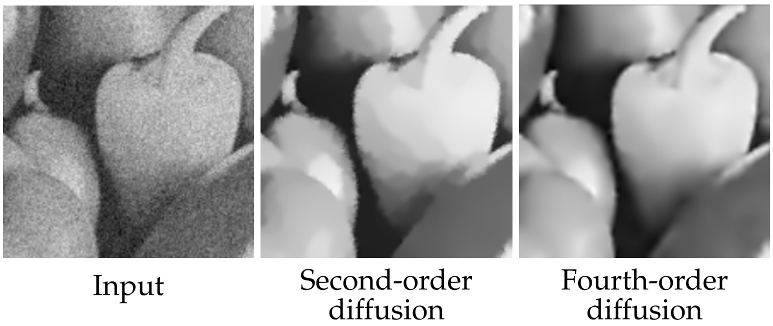


FIGURE 5.1: On smoothly shaded surfaces, second-order Perona-Malik diffusion creates a staircasing artifact that is avoided by fourth-order diffusion.

In this work, we argue that using fourth-order instead of second-order diffusion to enhance creases allows for a more accurate localization of their centerlines. We propose a novel fourth-order filter that introduces a fourth-order diffusion tensor to specifically enhance ridges, valleys, or both, in a scale-adaptive manner. Increased accuracy of the final segmentation is demonstrated on simulated and real-world medical images.

5.3 Related Work

Diffusion-based image filters treat image intensities as an initial heat distribution $u_{t=0}$, and solve the heat equation $\partial_t u = \operatorname{div}(g\,\nabla_{\mathbf{x}} u)$ for larger values of an artificial time parameter t, corresponding to increasingly smoothed versions of the image. If the diffusivity function g is constant, the diffusion is linear and uniformly smoothes image u. If $g = 1$, the solution at time t can be obtained as the convolution $u * G_\sigma$ with a Gaussian kernel G_σ with standard deviation $\sigma = \sqrt{2t}$ [185].

Since linear diffusion fails to preserve important image structures, Perona and Malik [128] introduced the idea of using *nonlinear* diffusion equations. By making the scalar diffusivity g a function of the spatial gradient magnitude $\|\nabla_{\mathbf{x}} u\|$, they reduce the amount of smoothing near image edges, and thus preserve edges. One such diffusivity function is

$$g\left(\|\nabla_{\mathbf{x}} u\|^2\right) = \frac{1}{1 + \frac{\|\nabla_{\mathbf{x}} u\|^2}{\lambda^2}}\,, \tag{5.1}$$

where λ is called the contrast parameter, and determines the minimum strength of edges that should be preserved [128].

Perona-Malik diffusion turns smoothly shaded surfaces into piecewise constant profiles, an effect that is often referred to as a staircasing artifact, and that can be seen in the central "Pepper" image in Figure 5.1. To avoid this effect, higher-order diffusion replaces the two first-order spatial derivatives in the heat equation with second-order derivatives. More recently, higher-order PDEs were also generalized to implicit surfaces [54] and image colorization [129].

In a one-dimensional setting, discrete variants of higher order data regularization can be traced back to a 1922 article by Whittaker [190]. A first approach for higher order regularization in image processing involving the absolute value of all second order partial derivatives has been proposed by Scherzer [157]. The resulting method

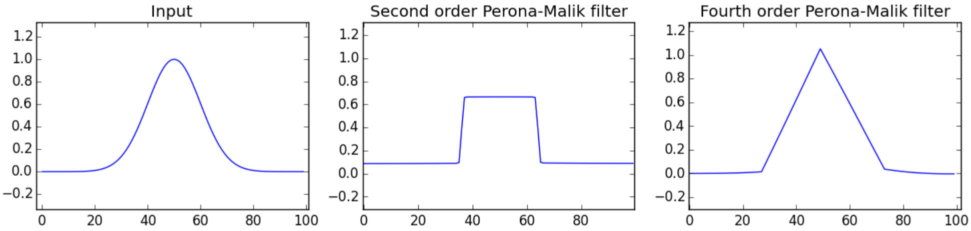


FIGURE 5.2: Second-order diffusion filter and fourth-order diffusion filter applied on a 1-D input signal. Both filters use the Perona-Malik diffusivity function.

has the drawback of not being rotationally invariant. An extension of classical regularization by choosing $(\Delta u)^2$ as argument of the penalizer in the smoothness term has been proposed by You and Kaveh [201]. Their method introduces speckle artifacts around edges that require some post-processing. Both problems can be solved by using the squared Frobenius norm of the Hessian matrix $\|H(u)\|_F^2$ as argument of the penalizer. This has been proposed by Lysaker et al. [112].

Two very similar higher order methods based on evolution equations without underlying variational formulation have been introduced by Tumblin and Turk [181] and Wei [184]. They use fourth-order evalution equations of the form

$$\partial_t u \;=\; -\mathrm{div}\Big(g(m)\nabla\Delta u\Big), \tag{5.2}$$

where m is the gradient norm [181] or the Frobenius norm of the Hessian [184].

Didas et al. [30] have generalized higher order regularization methods and the corresponding partial differential equations to arbitrary derivative orders. They have shown that, when combined with specific diffusivity functions, *fourth-order* equations can enhance image curvature analogous to how, by careful use of forward and backward diffusion, second-order equations can enhance, rather than just preserve, image edges [185].

Figure 5.2 shows a simple one-dimensional example with the Perona-Malik diffusivity function (5.1). The edge enhancement of nonlinear second-order diffusion, which leads to a piecewise constant result, and the curvature enhancement of nonlinear fourth-order diffusion, which leads to a piecewise linear result, are clearly visible. Obviously, localizing the maximum will be much easier and more reliable in case of the sharp peak created by fourth-order diffusion than in the extended plateau that results from second-order diffusion.

It is this curvature-enhancing property of fourth-order diffusion that we exploit in our novel filter. We combine it with the idea of *anisotropic* diffusion, which was introduced to image processing by Weickert [185] to address another limitation of the Perona-Malik model. Namely, a consequence of preserving edges by locally reducing the amount of smoothing is that the neighborhoods of edges remain noisy. Anisotropic diffusion $\partial_t u = \mathrm{div}(D\,\nabla_{\mathbf{x}} u)$ replaces the scalar diffusivity g by a second-order diffusion tensor D, which makes it possible to reduce smoothing orthogonal to, but not along image features, and therefore to denoise edges more effectively than isotropic nonlinear diffusion, while still avoiding to destroy them.

We generalize this approach to a novel anisotropic fourth-order diffusion equation that smoothes along the crease, while creating a sharp peak in the orthogonal direction, to clearly indicate its center. We are aware of only one previous formulation of anisotropic fourth order diffusion, proposed by Hajiaboli [59]. However, it has been designed to preserve edges, rather than enhance creases. Consequently, it is not well-suited for our purposes, as we will demonstrate in the results. Moreover, it differs from our approach in that it does not make use of a fourth-order diffusion tensor, and includes no mechanism for scale selection.

Despite the long history of research in this area, improved filtering and detection of ridges continues to be an active topic in medical image analysis. Our work on improving localization through fourth-order diffusion complements recent advances. For example, the SCIRD ridge detector by Annunziata et al. [4], or the vesselness measure by Jerman et al. [73] that gives better responses for vessels of varying contrasts, could replace the vessel segmentation by Frangi et al. [42] that we use as a prefiltering step. Several recent works [43, 61, 156, 177] have addressed diffusion in crossings and bifurcations, and could be combined with our work to improve the performance of our filter in such cases.

5.4 Method

5.4.1 Anisotropic Fourth-order Diffusion

Building on work of Lysaker et al. [112], Didas et al. [30] formulate nonlinear fourth-order diffusion as

$$\begin{aligned}\partial_t u = &- \partial_{xx}(g(\|H(u)\|_F^2)u_{xx}) - \partial_{yx}(g(\|H(u)\|_F^2)u_{xy}) \\ &- \partial_{xy}(g(\|H(u)\|_F^2)u_{yx}) - \partial_{yy}(g(\|H(u)\|_F^2)u_{yy}),\end{aligned} \tag{5.3}$$

where $\|H(u)\|_F^2$ is the Frobenius norm of the Hessian matrix of image u, and $u_{xy} = \partial_{xy} u$. We propose the following novel *anisotropic fourth-order* diffusion model, which combines the ideas of higher-order diffusion with that of making diffusivity a function of both spatial location and direction:

$$\begin{aligned}\partial_t u = &- \partial_{xx}\left[\mathcal{D}(H_\rho(u_\sigma)) : H(u)\right]_{xx} \\ &- \partial_{yx}\left[\mathcal{D}(H_\rho(u_\sigma)) : H(u)\right]_{xy} \\ &- \partial_{xy}\left[\mathcal{D}(H_\rho(u_\sigma)) : H(u)\right]_{yx} \\ &- \partial_{yy}\left[\mathcal{D}(H_\rho(u_\sigma)) : H(u)\right]_{yy}.\end{aligned} \tag{5.4}$$

Equation (5.4) introduces a general linear map $\mathcal{D}$ from the Hessian matrix H to a transformed matrix. Linear maps from matrices to matrices are naturally written as fourth-order tensors, and we use the “double dot product” $\mathcal{D} : H$ as a shorthand for applying the map $\mathcal{D}$ to the Hessian matrix H. This results in a transformed matrix T, and we use square brackets $[T]_{ij}$ to denote its (i, j)th component. Formally,

$$\begin{aligned}[T]_{ij} &= \left[\mathcal{D}(H_\rho(u_\sigma)) : H(u)\right]_{ij} \\ &= \sum_{k=1}^{2}\sum_{l=1}^{2}\left[\mathcal{D}(H_\rho(u_\sigma))\right]_{ijkl}[H(u)]_{kl}.\end{aligned} \tag{5.5}$$

In this notation, we can define second-order eigentensors E of $\mathcal{D}$ corresponding to eigenvalue μ by the equation $\mathcal{D} : E = \mu E$. An alternative notation, which will be used for the numerical implementation in Section 5.4.4, writes the Hessian and transformed matrices as vectors. This turns $\mathcal{D}$ into a matrix whose eigenvectors are nothing but the vectorized eigentensors as defined above. Similar to others [10, 81], we find the fourth-order tensor and "double dot" notation more appealing for reasoning at a higher level, because it allows us to preserve the natural structure of the involved matrices.

Using our square bracket notation, an equivalent way of writing one of the terms from Equation (5.3), $\partial_{ji}(g(\|H(u)\|_F^2)u_{ij})$, is $\partial_{ji}\left[g(\|H(u)\|_F^2)H(u)\right]_{ij}$. Thus, the difference between the model from Equation (5.3) and our new one in Equation (5.4) is to replace the isotropic scaling of Hessian matrices using a scalar diffusivity g, with a general linear transformation $\mathcal{D}$, which acts on the second-order Hessian in analogy to how the established second-order diffusion tensor acts on gradients in second-order anisotropic diffusion. Due to this analogy, we call $\mathcal{D}$ a fourth-order diffusion tensor.

In our filter, $\mathcal{D}$ is a function of the local normalized Hessians, which are defined as

$$H_\rho(u_\sigma) = G_\rho * \left(\frac{1}{\sqrt{1 + \|\nabla u_\sigma\|}} H\left(u_\sigma\right) \right), \tag{5.6}$$

where regularized derivatives are obtained by convolution with a Gaussian kernel, $u_\sigma := u * G_\sigma$. Its width σ should reflect the scale of the crease, as will be discussed in Section 5.4.3. Since scale selection might introduce spatial discontinuities in the chosen σ, the normalized Hessians are made differentiable by integrating them over a neighborhood, for which we use a Gaussian width $\rho = 0.5$ in our experiments. As shown in [62], and used for vesselness enhancement diffusion in [15], the inverse gradient magnitude factor is used to make the eigenvalues of $H_\rho(u_\sigma)$ match the surface curvature values.

We emphasize that, unlike in a previous generalization of structure tensors to higher order [165], the reason for going to higher tensor order in Equation (5.4) is not to preserve information at crossings; this is a separate issue that was recently addressed by others [61], and that we plan to tackle in our own future work. In our present work, our goal is to smooth along ridges and valleys, while sharpening them in the orthogonal direction. This sharpening requires the curvature-enhancing properties of fourth-order diffusion, and a fourth-order diffusion tensor is a natural consequence of making fourth-order diffusion anisotropic.

5.4.2 Fourth-order Diffusion Tensor $\mathcal{D}$

We now need to construct our fourth-order diffusion tensor $\mathcal{D}$ so that it will smooth along creases, while enhancing them in the perpendicular direction. Similar to Weickert's diffusion tensors [185], we will construct $\mathcal{D}$ in terms of its eigentensors E_i and corresponding eigenvalues μ_i, as defined above.

Didas et al. [30] have shown that fourth-order diffusion with the Perona-Malik diffusivity [128] allows for adaptive smoothing or sharpening of image curvature, depending on a contrast parameter λ. In particular, in the 1-D case, only forward diffusion (i.e., smoothing) happens in regions with $|\partial_{xx}u| < \lambda$, while only backward diffusion (i.e., curvature enhancement) occurs where $|\partial_{xx}u| > \sqrt{3}\lambda$. We wish to exploit this to enhance creases whose curvature is strong enough to begin with, while smoothing out less significant image features.

This is achieved by deriving the eigenvalues μ_i of $\mathcal{D}$ from the eigenvalues ν_1, ν_2 of the normalized Hessian $H_\rho(u_\sigma)$ using the Perona-Malik diffusivity [128], i.e.,

$$\mu_i = \frac{1}{1 + \nu_i^2/\lambda^2}, \text{ for } i \in \{1, 2\}. \tag{5.7}$$

If the user wishes to specifically enhance either ridges or valleys, the sign of ν_i could be taken into account. For instance, a ridge-like behaviour in the ith direction is characterized by $\nu_i < 0$. Therefore, we can decide to smooth out valleys by setting $\mu_i = 1$ wherever $\nu_i \geq 0$, and enhance ridges wherever $\nu_i < 0$ by defining μ_i as before. Enhancing only valleys can be done in full analogy. In our experiments on synthetic data, we found that, in terms of the ℓ_2 difference between the ground truth and the filtered image, better results were obtained when enhancing both ridges and valleys. This is the setting used in all our experiments.

The ridge and valley directions can be found from the eigenvectors e_1, e_2 of the normalized Hessian matrix $H_\rho(u_\sigma)$, and are reflected in the eigentensors E_i of $\mathcal{D}$ by setting

$$\begin{array}{llll} E_1 = & e_1 \otimes e_1 & E_3 = & \frac{1}{\sqrt{2}}(e_1 \otimes e_2 + e_2 \otimes e_1) \\ E_2 = & e_2 \otimes e_2 & E_4 = & \frac{1}{\sqrt{2}}(e_1 \otimes e_2 - e_2 \otimes e_1) \end{array} . \tag{5.8}$$

The E_i are orthonormal with respect to the tensor dot product $A : B := \operatorname{tr}(B^{\mathrm{T}} A)$. By definition, E_4 is antisymmetric. Since Hessians of smooth functions are symmetric, the value of μ_4 does not play a role, and is simply set to zero. We define μ_3 as the average of μ_1 and μ_2.

5.4.3 Scale Selection

In the previous sections, crease orientation was estimated using the eigenvectors of the regularized and normalized Hessian in Equation (5.6). As in previous approaches such as vesselness enhancement diffusion (VED) [15], this involves a regularization parameter σ that should be adapted to the local radius of the crease. Setting this parameter is referred to as scale selection.

The vesselness measure $\mathcal{V}_\sigma$ by Frangi et al. [42] is maximal at the scale σ that matches the corresponding vessel size, and has been widely used for detecting the local radius of vessel like structures. $\mathcal{V}_\sigma$ is obtained from sorted and scale-normalized eigenvalues $|\tilde{\nu}_1| \leq |\tilde{\nu}_2|$, computed as $\tilde{\nu}_i := \sigma^2 \nu_i$ from eigenvalues ν_i of the Hessian $H(u_\sigma)$ at a given scale σ. The factor σ^2 compensates for the loss of contrast at larger scales [101].

A vesselness measure $\mathcal{V}_\sigma$ should be low in background regions where overall curvature and thus $\mathcal{S} = \sqrt{\tilde{\nu}_1^2 + \tilde{\nu}_2^2}$ are low overall. Moreover, it should detect tubular structures, where $|\tilde{\nu}_1| \ll |\tilde{\nu}_2|$, as opposed to blobs, in which $\mathcal{R}_\mathcal{B} = \frac{\tilde{\nu}_1}{\tilde{\nu}_2}$ would be large. For ridges ($\tilde{\nu}_2 < 0$), Frangi et al. achieve this by combining $\mathcal{S}$ and $\mathcal{R}_\mathcal{B}$ according to

$$\mathcal{V}_\sigma u = \begin{cases} 0 & \text{if} \quad \tilde{\nu}_2 > 0 \\ \left(\mathrm{e}^{-\frac{\mathcal{R}_\mathcal{B}^2}{2\beta^2}}\right)\left(1 - \mathrm{e}^{-\frac{\mathcal{S}^2}{2c^2}}\right) & \text{otherwise} \end{cases} , \tag{5.9}$$

where the β and c parameters tune $\mathcal{V}_\sigma u$ to be more specific with respect to suppression of blob shapes or background structures, respectively. We use $\beta = 0.5$ and $c = \frac{1}{2}(\max(\mathcal{S}))$, as recommended in [42].

The scale for each pixel is selected as the σ for which the maximum $\mathcal{V}u = \max_{\sigma=\sigma_{\min},\ldots,\sigma_{\max}} \mathcal{V}_\sigma u$ is attained, where $\{\sigma_{\min}, \ldots, \sigma_{\max}\}$ are the range of expected scales in the image. For pixels that are part of the background, $\mathcal{V}u$ is low, and it can be thresholded by parameter $\theta \in [0, 1]$ for vessel segmentation. This segmentation indicates the extent of vessels, and is used for our scale-image postprocessing, as described below.

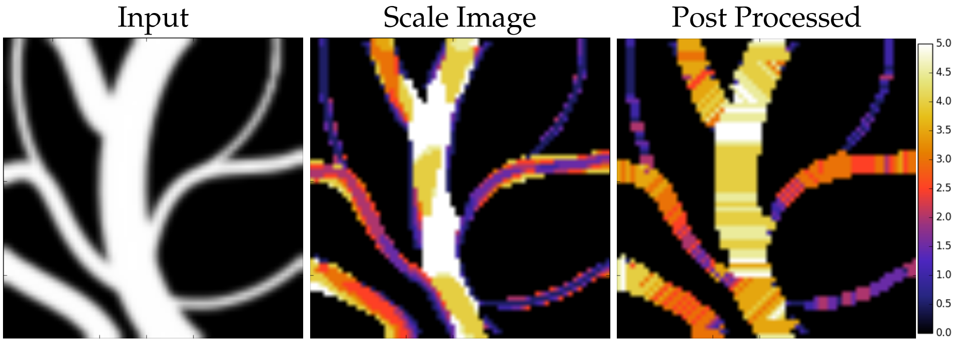


FIGURE 5.3: The color of each pixel in the scale image illustrates the scale of the underlying vessel. Artifacts near the boundaries are removed in a post processing step.

It has been observed previously [29] that vesselness often fails to correctly estimate the scale of the vessel along its boundary. This can happen in two cases: When the ridge has a step-like shape, the curvature near the corner points will be much larger at the finest scale than at all other scales, leading to an underestimation of the real scale near the boundary. On the other hand, the cross-sectional intensity profile of vessels may have inflection points near its edges, where $\tilde{v}_2$ changes its sign. In this case, some points near the boundary will have zero vesselness at the finest scale, but the sign of $\tilde{v}_2$ will flip, and therefore vesselness becomes non-zero, at coarser scales, leading to an overestimation of scale. Figure 5.3 shows both scale underestimation or overestimation at vessel boundaries.

While such effects are less problematic for the VED filter, which uses the same vesselness measure for scale selection, it can lead to serious artifacts in our filter, where misestimating the scales at boundaries can cause the curvature-enhancing diffusion to enhance the boundary of large-scale ridges more than their center.

We avoid such boundary effects by introducing a novel postprocessing of the computed scales. For each pixel on a vessel, the vessel cross-section containing that pixel is extracted by following the eigenvector direction that corresponds to the strongest eigenvalue of the Hessian matrix computed at the scale suggested by the vesselness measures at each point. Then, all pixels are assigned the scale closest to the average of all pixels that lie on the same cross-section. This removes the problem of scale over- or underestimation on the boundaries. Figure 5.3 shows the scale image before and after being post processed.

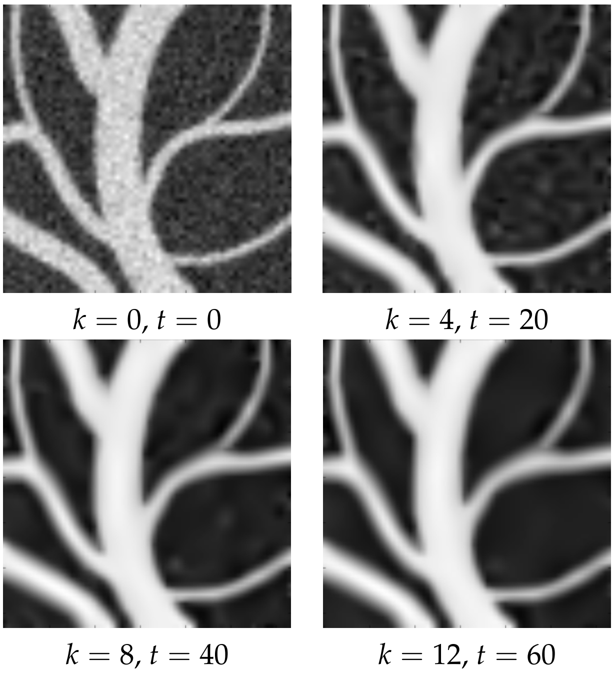
5.4.4 Stability

In order to solve Equation (5.4), we discretize it with standard finite differences, and use an explicit numerical scheme. In matrix-vector notation, this can be written as

$$u^{k+1} = u^k - \tau P u^k = (I - \tau P) u^k \, , \tag{5.10}$$

where $u^k \in \mathbb{R}^m$ is the vectorized image at iteration k, and the exact form of matrix $P \in \mathbb{R}^{m \times m}$ will be discussed later. We call a numerical scheme ℓ_2 stable if

$$\left\| u^{k+1} \right\|_2 \leq \left\| u^k \right\|_2 \, , \tag{5.11}$$



$k = 0, t = 0$ $k = 4, t = 20$

$k = 8, t = 40$ $k = 12, t = 60$

FIGURE 5.4: MAFOD filter applied on the noisy synthesized branch image at different FED cycles k. Value t represents the overall diffusion time at each cycle k.

i.e., the ℓ_2 norm of the image is guaranteed not to increase from iteration k to $k+1$. It follows from Equation (5.10) that

$$\left\| u^{k+1} \right\|_2 \leq \| I - \tau P \|_2 \cdot \left\| u^k \right\|_2 , \tag{5.12}$$

where $\|P\|_2$ denotes the ℓ_2 norm of P, i.e., $\|P\|_2 := \sqrt{\rho(P^T P)}$, where $\rho(P^T P)$ computes the largest modulus of eigenvalues of the symmetric matrix $P^T P$.

Consequently, the condition in Equation (5.12) is satisfied if

$$\| I - \tau P \|_2 \leq 1 \, . \tag{5.13}$$

Since P is positive semi-definite, the eigenvalues of $I - \tau P$ are within the interval $[1 - \tau \|P\|_2 , 1]$. Thus, Equation (5.13) is satisfied if $1 - \tau \|P\|_2 \geq -1$. This results in the following constraint on the permissible time step size τ:

$$\tau \leq \frac{2}{\|P\|_2} \, . \tag{5.14}$$

This clarifies that the restriction on the time step size only depends on $\|P\|_2$. To compute it, we will now write down the system matrix P for our discretization of fourth-order anisotropic diffusion filtering.

Let L_{xx}, L_{xy}, L_{yx}, L_{yy} be matrices approximating the corresponding derivatives. For "natural" boundary condition, it is important only to approximate the derivatives at pixels i where the whole stencil fits in the image domain, i.e., where enough data is available. Let us combine these four matrices pixelwise into one big matrix L

such that

$$Lu \approx \begin{pmatrix} \vdots \\ [L_{xx}u]_i \\ [L_{xy}u]_i \\ [L_{yx}u]_i \\ [L_{yy}u]_i \\ \vdots \end{pmatrix}, \tag{5.15}$$

i.e., the approximations of the four derivatives will be next to each other for every pixel i.

The 4×4 matrix form of the fourth-order diffusion tensor in pixel i, acting on

$$\left([L_{xx}u]_i \;\; [L_{xy}u]_i \;\; [L_{yx}u]_i \;\; [L_{yy}u]_i \right)^T,$$

can be written as $D_i = EME^T$, where E is an orthogonal matrix containing the vectorized E_1, E_2, E_3, E_4 from Equation (5.8) as its columns and M is a diagonal matrix with the eigenvalues μ_1, μ_2, μ_3, μ_4 on its diagonal. Due to the choice of the Perona-Malik diffusivity in our model, $\|D_i\|_2 \leq 1$.

If we arrange all per-pixel matrices D_i in one big matrix D with a 4×4 block-diagonal structure,

$$D = \begin{pmatrix} D_1 & & \cdots & & 0 \\ & D_2 & & & \\ \vdots & & D_3 & & \vdots \\ & & & \ddots & \\ 0 & & \cdots & & D_m \end{pmatrix}, \tag{5.16}$$

it is clear that $\|D\|_2 \leq 1$, and the whole scheme reads as

$$u^{k+1} = u^k - \tau L^T D L u^k. \tag{5.17}$$

Substituting into Equation (5.14) yields

$$\tau \leq \frac{2}{\|L^T D L\|_2}, \tag{5.18}$$

meaning that, in order to find a stable step size τ, we have to bound

$$\left\| L^T D L \right\|_2 \leq \|L\|_2^2 \leq \sum_{i,j \in \{x,y\}} \|L_{i,j}\|_2^2, \tag{5.19}$$

whose value will depend on the exact second-order finite difference stencils. We will use the same discretization as Hajiaboli [59], i.e.,

$$\begin{aligned} u_{xx} &\approx \frac{(u_{i-1,j} - 2u_{i,j} + u_{i+1,j})}{(\Delta x)^2} \\ u_{yy} &\approx \frac{(u_{i,j-1} - 2u_{i,j} + u_{i,j+1})}{(\Delta y)^2} \\ u_{xy} &\approx \frac{(u_{i-1,j-1} + u_{i+1,j+1} - u_{i-1,j+1} - u_{i+1,j-1})}{4\Delta x \Delta y} \\ u_{yx} &= u_{xy}\,, \end{aligned}$$

where Δx and Δy are the pixel edge lengths in x and y directions, respectively. It is easy to verify using Gershgorin's theorem that this results in

$$\tau \leq \frac{2}{16\left(\Delta x\right)^2 + 16\left(\Delta y\right)^2 + 2\left(\Delta x \Delta y\right)}, \tag{5.20}$$

i.e., for $\Delta x = \Delta y = 1$, $\tau \leq 1/17$.

5.4.5 Implementation Using Fast Explicit Diffusion

Since the time step size τ derived in the previous section is rather small, solving the discretized version of Equation (5.4) numerically using a simple explicit Euler scheme requires significant computational effort. The recently proposed Fast Explicit Diffusion (FED) provides a considerable speedup by varying time steps in cycles, in a way that up to half the time steps within a cycle can violate the stability criterion, but the cycle as a whole still remains stable [187]. Consequently, a much smaller number of iterations is required to reach the desired stopping time.

The FED scheme is defined as follows:

$$\begin{aligned} u^{k+1,0} &= u^k, \\ u^{k+1,i+1} &= (I - \tau_i P(u^k_\sigma))u^{k+1,i} \qquad i = 0, \ldots, n-1, \\ u^{k+1} &= u^{k+1,n} \end{aligned} \tag{5.21}$$

where index k is the cycle iterator, i is the inner cycle iterator, and n is the number of sub-steps in each cycle. In order to ensure stability, $P(u^k_\sigma)$ must be constant during each cycle. For computing τ_i, first the number of sub-steps in each cycle must be computed using

$$n = \left\lceil -0.5 + 0.5\sqrt{1 + \frac{12T}{M\tau_{\max}}} \right\rceil, \tag{5.22}$$

where T is the diffusion stopping time, M is the number of FED cycles and $\tau_{\max}$ is the step size limit that ensures stability. In our experiments we set $\tau_{max} = 0.05$ according to the limit computed in Section 5.4.4. As it is shown in [187], n determines τ_i:

$$\tau_i = \frac{3T}{2M\left(n^2 + n\right)\cos^2\left(\pi \cdot \frac{2i+1}{4n+2}\right)}, \ (i = 0, \ldots, n-1). \tag{5.23}$$

In order to decrease balancing error within each cycle, τ_i's order should be rearranged. In our experiments we have used the *κ-cycles* method for τ_i reordering [187].

The fast explicit diffusion framework can be combined with our discretization in a straightforward manner, and has led to a speedup of around two orders of magnitude in some of our experiments. For both τ_i computation and reordering we have used the provided source code by Weickert et al. [187]. Figure 5.4 shows our filter applied on a synthesized image using the FED scheme with different cycle iterators k, corresponding to different stopping times.

5.4.6 Ridge and Valley Extraction

After enhancing ridges and valleys with our filter, we extract a polygonal representation of them using a 2D counterpart of an established 3D algorithm [164]. Our algorithm is based on the idea of marching squares [163] and involves the zero contour of the scalar field $d = \det(g|Hg)$, where $(g|Hg)$ indicates a matrix whose first

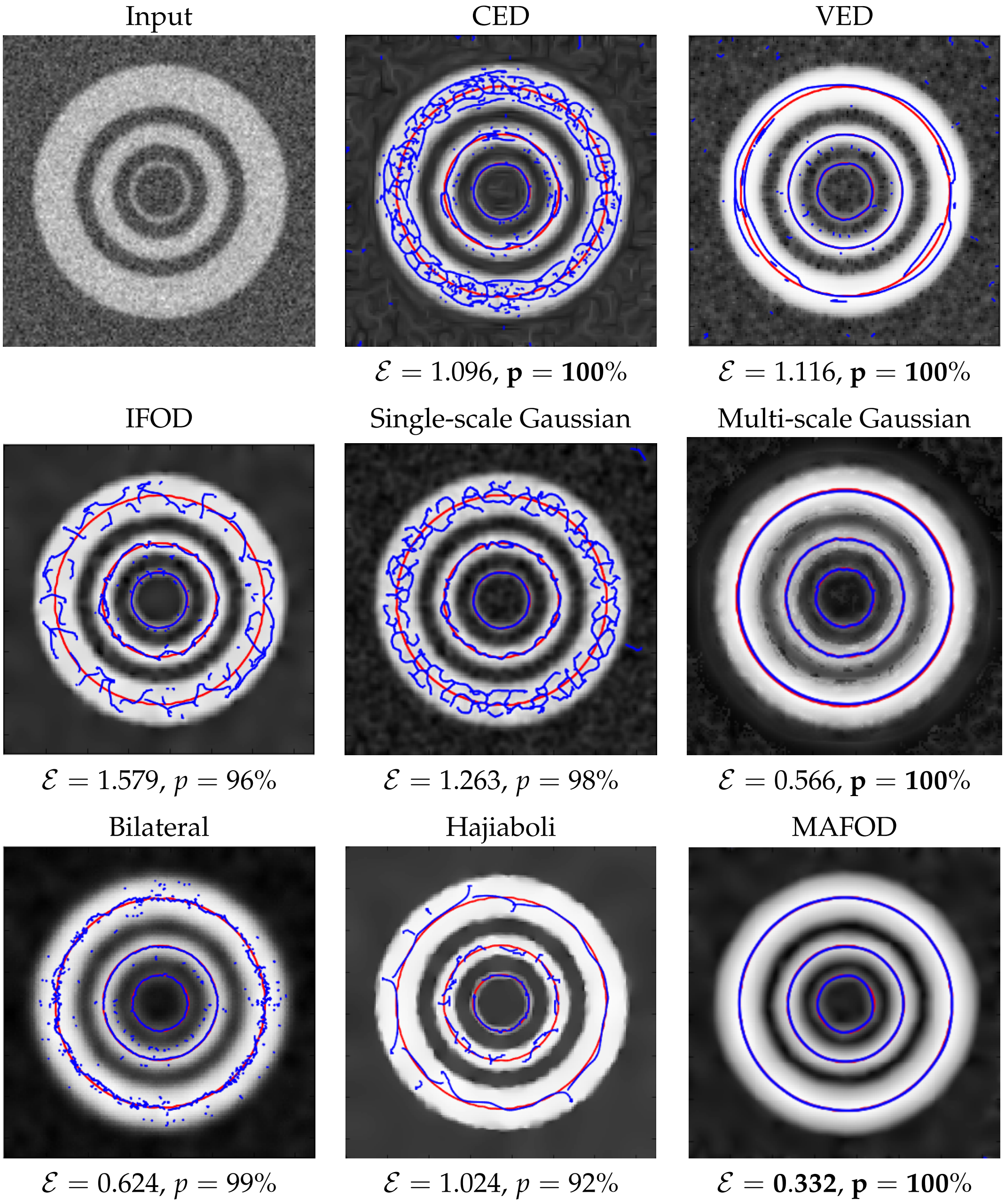
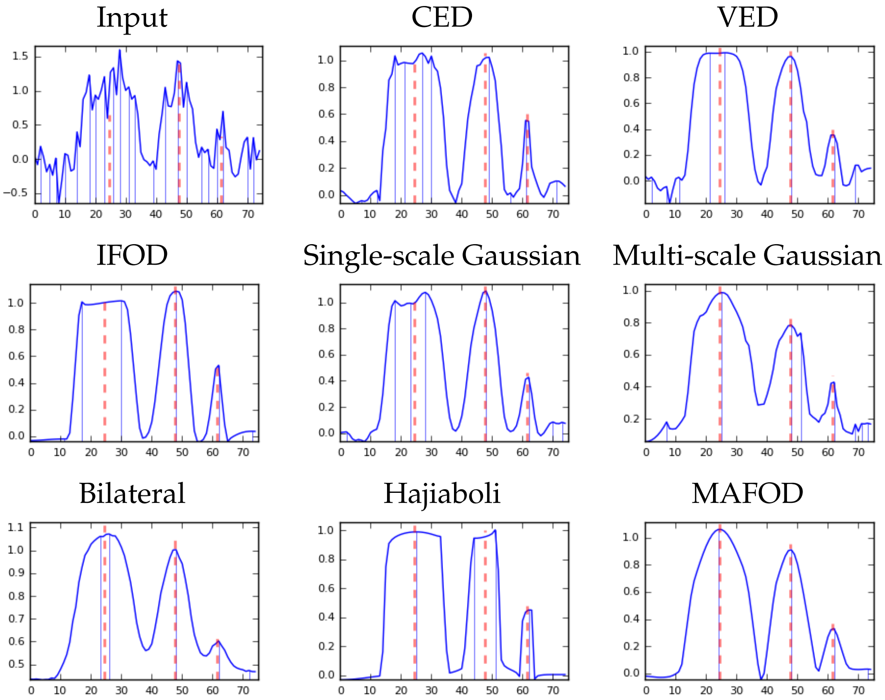


FIGURE 5.5: Red curves show the ground truth ridge location, while blue curves show the location reconstructed from the filtered noisy image. Our MAFOD filter restores ridge locations from the noisy image with ridges of different scales better than other filters.

FIGURE 5.6: Each plot shows the intensities on a left to center line of the 2D images in Figure 5.5. The lines are taken from the middle of each image. The red dashed lines show the true position of the ridge points, and the blue lines show the position of local maxima over the intensity line scan of each filtered image.



column is the local gradient vector g and the second column Hg is the result of multiplying the gradient vector to the Hessian matrix; det$(.)$ is the matrix determinant. The zero level set of d is a superset of the creases [127].

Our overall approach of ridge extraction involves two different notions of scale: The first one refers to the selection of scales at which derivatives are taken, as discussed in Section 5.4.3; the second one to the stopping time t of our filter. To clarify their respective roles, we compare our approach to the seminal work by Lindeberg [101] on ridge extraction in Gaussian scale space.

In Lindeberg's approach, ridge curves in 2D images sweep out surfaces in three-dimensional scale space, and curves on these surfaces are found along which a measure of ridge strength is locally maximal with respect to diffusion time t. An example of such a measure is

$$R(u_\sigma) = t^{4\gamma}\left(u_{xx} + u_{yy}\right)^2 \left(\left(u_{xx} - u_{yy}\right)^2 + 4u_{xy}^2\right). \tag{5.24}$$

In this approach, stopping time t and the scale σ of derivatives are related by $t = \sigma^2/2$, and can thus be considered as one single parameter, whose value is determined automatically. The exponent γ in the normalization factor that is used to compensate for the loss of contrast at later diffusion times is treated as a tunable parameter. In our experiments, we set it to $\gamma = \frac{3}{4}$, as proposed in [101].

Decoupling the t and σ parameters is a price that we pay in our method in order to preserve and enhance creases, for which Gaussian scale space does not have any mechanism. Our current implementation selects the derivative scale σ automatically, as discussed in Section 5.4.3, but does not have an objective criterion for setting the stopping time t, unless ground truth is available. In practice, we found it relatively simple to tune this parameter based on viewing the corresponding images, especially given that, after image noise has been removed, results are relatively stable (cf. Figure 5.4). Future work might investigate automated selection of this parameter.

Another difference between our approach and Lindeberg's is that his crease extraction algorithm operates on the full scale space, while ours, similar to previous work by Barakat et al. [9], works on a single, pre-filtered image. Both approaches have relative benefits and drawbacks: Scale space crease extraction is challenging to implement, and requires much more time and memory, especially when dealing with the four-dimensional scale space resulting from three-dimensional input images [82]. On the other hand, it might, in rare cases, indicate spatially intersecting creases at different scales, which our current approach is not able to reproduce.

5.5 Experimental Results

We compare our multi-scale anisotropic fourth-order diffusion (MAFOD) to crease enhancement diffusion (CED) [169], vesselness enhancement diffusion (VED) [15], isotropic fourth-order diffusion (IFOD) [112], the anisotropic fourth-order diffusion by Hajiaboli [59], bilateral, and a multi-scale Gaussian filter. Since it was already shown in [169] that the coherence enhancing diffusion filter [185] tends to more strongly deform non linear structures compared to the CED filter, it is not included in the comparison.

The multi-scale Gaussian filter is defined to approximate Lindeberg's scale selection, as described in Section 5.4.6. From a range of stopping times between $t = 1$ and $t = 30$, it first selects an optimal scale for each pixel, by finding the t that maximizes $R(u_\sigma)$ from Equation (5.24). Then, the intensity of each pixel in the output image is obtained by convolving the input image with a Gaussian at the locally optimal scale $\sigma = \sqrt{2t}$ that is then normalized between $[0, 1]$. The normalization is necessary to compensate for the intensity range shrinkage after Gaussian blurring.

The crease extraction algorithm from Section 5.4.6 results in a set of polygonal chains. For each crease line segment in the ground truth, a corresponding segment in the reconstruction is selected by picking the one with minimum Hausdorff distance [66] in a neighborhood around the ground truth line segment. This neighborhood is set to six pixels for the experiments on synthetic data, and to ten pixels for real data. The average Euclidean distance $\mathcal{E}$ between the ground truth and the corresponding reconstruction is then used to quantify the accuracy of vessel locations in the filtered image. In addition to $\mathcal{E}$, we show the percentage p of ground truth for which a corresponding ridge was detected from the filtered images while computing $\mathcal{E}$.

In the experiments on synthesized images, image evolution of all filters, except for multi-scale Gaussian and bilateral filters, was stopped when the ℓ_2 difference between the filtered image and the noise-free ground truth was minimized. ℓ_2 difference was chosen over $\mathcal{E}$ as a stopping criterion due to its much lower computational cost.

5.5.1 Confirming Theoretical Properties

Our filter has been designed to improve localization accuracy while accounting for creases at multiple scales and being rotationally invariant. Results on a simple simulated image with three concentric ridges of different radii, which is contaminated

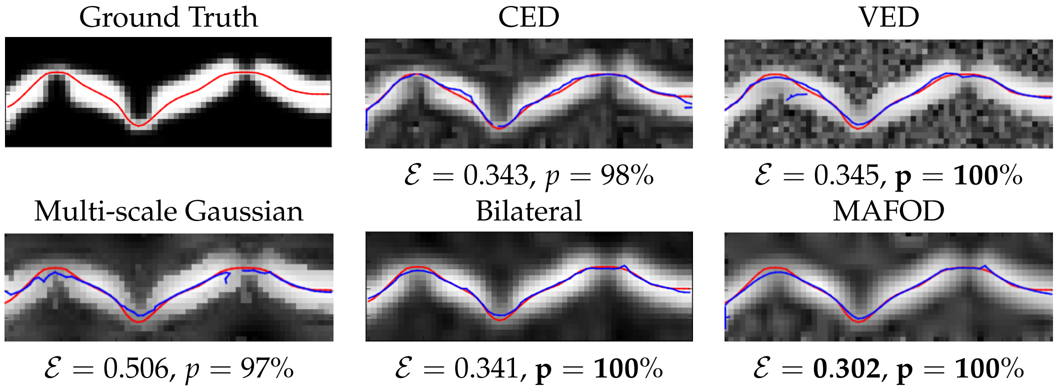


FIGURE 5.7: In a simulated occluded vessel, restored (blue) curves again best match the (red) ground truth locations in case of our MAFOD filter. In addition, MAFOD better preserves the occlusions than VED.

with zero-mean Gaussian noise with a signal to noise ratio SNR $= 6.81$, verify that these design goals are met.

Both Figure 5.5 and Figure 5.6 show that our MAFOD filter restores ridge locations most accurately as assessed both by visual inspection and Euclidean distance $\mathcal{E}$. MAFOD outperforms CED, IFOD, Hajiaboli and bilateral filtering since it accounts for different scales. On the other hand, the curvature enhancement of our filter, which is not part of multiscale VED or Gaussian filters, clearly makes it easier for the ridge extraction algorithm to localize the centerline, especially in the largest ridge. IFOD does perform curvature enhancement but, due to its isotropic nature, it is not effectively guided to act specifically across the ridge. As it is obvious on the largest circle, the multi-scale Gaussian filter leads to ridge displacement. The result of the anisotropic fourth-order filter by Hajiaboli clearly illustrates the fact that it was designed to preserve edges, not to enhance creases.

For the MAFOD filter, scales σ and vesselness threshold θ are the same as for VED, $\sigma = \{0.5, 1.0, ..., 8.5, 9.0\}$, $\theta = 0.2$. Other parameters are $\lambda = 0.005$ for MAFOD, IFOD, and $\sigma = 1.0$ for IFOD; for Hajiaboli, $\lambda = 0.01$; for CED, $\sigma = 2.0$ and it is set to enhance both ridges and valleys; for single-scale Gaussian smoothing, $\sigma = 1.25$. For the MAFOD filter, FED stopping time is set to 500, and the number of cycles is set to 10000. For other fourth-order equations $\tau = 0.03$ and for the second-order diffusion equations such as the VED and CED filters, $\tau = 0.2$; for the bilateral filter $\sigma_{spatial} = 3.0$ and $\sigma_{range} = 1.0$.

5.5.2 Simulated Vessel Occlusion

Figure 5.7 shows a second image, simulating an occluded vessel, and corrupted with Gaussian noise with SNR $= 6.40$. Our MAFOD filter leads to the most accurate localization in terms of Euclidean error $\mathcal{E}$. In particular, we observed that VED widens the occlusions. They are better preserved by our filter, which we set to enhance both ridges and valleys.

Again, an amount of smoothing that minimized ℓ_2 error was used for all filters except for multi-scale Gaussian and bilateral filters. The parameters for VED and MAFOD are $\sigma = \{0.5, 1.0, 1.5, 2.0, 2.5, 3.0\}$, and $\theta = 0.35$; $\lambda = 0.017$, stopping time is 20 and the number of cycles is set to 1000 for MAFOD; for CED, $\sigma = 1.0$; for

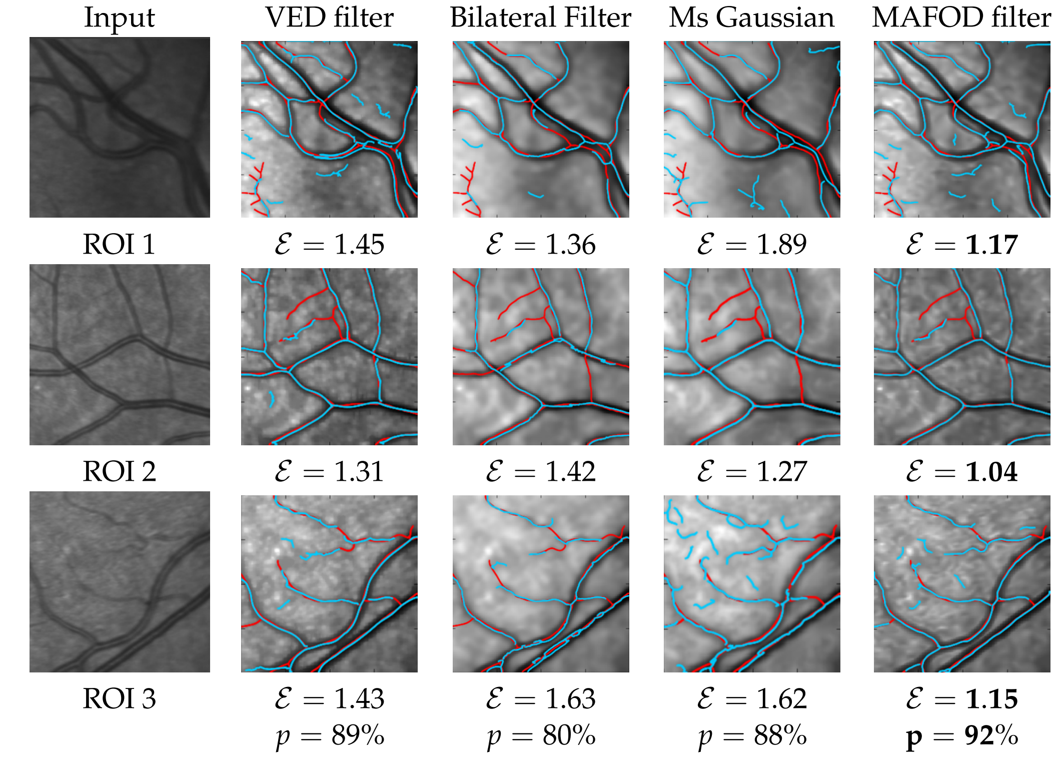


FIGURE 5.8: In three ROIs of a fundus image, reconstructed vessel locations (blue) best match a manually marked ground truth (red) when our MAFOD filter is used.

the bilateral filter, $\sigma_{\text{spatial}} = 1.5$ and $\sigma_{\text{range}} = 1.0$. For the numerical solver, we set $\tau = 0.05$ for the fourth-order equation and for second-order equations, $\tau = 0.2$.

5.5.3 Real Vessel Tree

To demonstrate our filter on a real-world example, we applied it to several ROIs from an infrared fundus image, on which one of our co-authors (MWMW), who is an ophthalmologist, manually marked the exact vessel locations to provide a ground truth for comparison, without being shown the filtered images. Results in Figure 5.8 show that our MAFOD filter outperforms VED, multi-scale Gaussian and bilateral filters in restoring vessel locations. In ROI 3, at some point the two thickest vessels run close to each other. By looking at the filtered image with the VED, the two vessels are erroneously connected to each other in that area, even though they are not connected in the corresponding extracted valley curves. Our MAFOD filter correctly avoided connecting the vessels to each other.

Even though vessels generally appear dark (i.e., as valleys) in these images, the larger ones exhibit a thin ridge at their center, due to a reflex in the infrared image. This leads to an incorrect double response in single-scale filters as shown for the bilateral filter. CED and IFOD filters suffer from similar problems (results not shown).

For each filter separately, we carefully tuned the parameters for optimum results. Specially θ, λ and the stopping time are the parameters that need more careful tuning compared to others. For the MAFOD filter, we set $\sigma = \{0.2, 0.3, 0.5, 1.0, ..., 6.5, 7.0\}$, $\lambda = 0.005$, $\theta = 0.13$, and used a FED scheme with stopping time 12 and cycle number 2. For the VED filter, an explicit Euler scheme is used with 600 iterations and $\tau =$

Filter	VED	Bilateral	Ms Gaussian	MAFOD
Time (*sec*)	1251	0.16	2.24	6.51

TABLE 5.1: The average filtering time for a single ROI of size 200 × 200 pixels in Figure 5.8.

0.2, and the same parameters for scale selection as for MAFOD; for the bilateral filter, $\sigma_{\text{range}} = 0.3$ and $\sigma_{\text{spatial}} = 3.0$; for the multi-scale Gaussian filter an additional Gaussian smoothing with kernel size $\sigma = 2$ is applied to the filtered image to blur out discontinuities from scale selection and thus achieve an even better result. The computational effort of all filters is reported in Table 5.1.

5.6 Conclusion

We have proposed a new multi-scale fourth order anisotropic diffusion (MAFOD) filter to enhance ridges and valleys in images. It uses a fourth order diffusion tensor which smoothes along creases, but sharpens them in the perpendicular direction, and optionally enables enhancing either ridges or valleys only. Our results indicate that the curvature enhancing properties of fourth-order diffusion allow our filter to better restore the exact crease locations than traditional methods. In addition, we found that our filter better preserves vessel occlusions.

In the future, we would like to extend our 2-D filter to 3-D images, and to better handle crossings and bifurcations [61].

Chapter 6

Replication and Refinement of an Algorithm for Automated Drusen Segmentation on Optical Coherence Tomography[1]

6.1 Abstract

Here, we investigate the extent to which re-implementing a previously published algorithm for OCT-based drusen quantification permits replicating the reported accuracy on an independent dataset. We refined that algorithm so that its accuracy is increased. Following a systematic literature search, an algorithm was selected based on its reported excellent results. Several steps were added to improve its accuracy. The replicated and refined algorithms were evaluated on an independent dataset with the same metrics as in the original publication. Accuracy of the refined algorithm (overlap ratio 36–52%) was significantly greater than the replicated one (overlap ratio 24–39%). In particular, separation of the retinal pigment epithelium and the ellipsoid zone could be improved by the refinement. However, accuracy was still lower than reported previously on different data (overlap ratio 67–76%). This is the first replication study of an algorithm for OCT image analysis. Its results indicate that current standards for algorithm validation do not provide a reliable estimate of algorithm performance on images that differ with respect to patient selection and image quality. In order to contribute to an improved reproducibility in this field, we publish both our replication and the refinement, as well as an exemplary dataset.

Key Words: automated image analysis, optical coherence tomography, age-related macular degeneration, algorithm replicability, algorithm reproducibility

6.2 Introduction

To increase our understanding of risk factors for age-related macular degeneration (AMD), the leading cause of irreversible blindness in the developed world, large, prospective epidemiological studies on AMD retinal biomarkers are warranted [24,

[1] The content of this chapter has been previously published: Wintergerst and Gorgi Zadeh et al. "Replication and Refinement of an Algorithm for Automated Drusen Segmentation on Optical Coherence Tomography" *Scientific Reports* 10.1 (2020): 1-7. URL: https://doi.org/10.1038/s41598-020-63924-6 [191].

85, 111, 162]. Increasing amounts of optical coherence tomography (OCT) data are being generated and need to be assessed. However, as the manual grading of enormous data volumes is unfeasible automated algorithms for OCT image analysis are needed [1, 159]. To date a variety of different algorithms on quantitative OCT image analysis for AMD biomarkers including drusen, geographic atrophy, pigment epithelial detachment and intra- and subretinal fluid have been published [78, 192]. However, algorithm quality and performance differ substantially making comparison between algorithms challenging [147, 192].

When selecting an algorithm from the literature, one for which high accuracy has been reported would be preferable. However, this does not guarantee comparable performance on one's own data. In particular, accuracy depends not only on the algorithm itself, but also on the characteristics of the dataset. Frequently, privacy and legal reasons make it impossible to openly share the data on which algorithms have been evaluated. At the same time, only very few authors make their software implementations available. Even though proper validation of an algorithm should involve testing it on data that was not available during its development, and characterizing conditions under which it can be expected to work well, the reasons above make this difficult within the field of medical image analysis.

The current best practice for validating medical image analysis algorithms is to organize so-called "challenges", in which different teams apply their algorithms to a common reference image data set. In regards to OCT image analysis both the "Retinal OCT Classification Challenge (ROCC)" [144] and the MICCAI "Retinal OCT Fluid Challenge (RETOUCH)" [143] are available and have been used for some of the respective algorithms published. Challenges are an important tool for algorithm validation, but their open nature implies incomplete coverage, as promising approaches might be missing because their authors chose not to participate. Moreover, it has been found that reproducibility and interpretation of their results is limited by the fact that participants are given the opportunity to adapt their methods to the data at hand, but often have to do so based on insufficient information. Challenges have also been shown to rank algorithms in a way that depends not only on their quality, but also substantially on contestable choices in the ranking scheme [115]. Therefore, we believe that replication studies, which received almost no attention in the literature so far, should complement challenges as another tool for algorithm validation.

We conducted such a replication study. Specifically, we re-implemented an algorithm for drusen quantification on OCT proposed by Chen et al. [18], which we identified as an established algorithm that was reported to achieve strong results in a previous systematic literature review [192]. Following assessment of its performance, we refined this algorithm to better meet the characteristics of our data. Our results highlight the severe limitations that incomplete algorithm validation poses for practical use.

6.3 Methods

6.3.1 Image data acquisition

The patient sample is a random subsample from the "Molecular Diagnostics of Age-related Macular Degeneration" (MODIAMD) study from the University of Bonn, Germany (Federal Ministry of Education and Research funding number 13N10349) [175] . Briefly, inclusion criteria were age $>$ 50 years and retinal alterations classified as Age-Related Eye Disease Study (AREDS) category 3 or 4. Exclusion criteria for the MODIAMD study were any other ophthalmic disease potentially comprising the assessment of the retina as was concomitant injection-therapy for AMD. All study subjects consented to participate in the study. The tenets of the Declaration of Helsinki were followed and this study was approved by the ethics committee of the University of Bonn, Germany (ethics committee number: 175/10 and 408/15). SD-OCT raster scans were acquired using the Spectralis HRA + OCT (Heidelberg Engineering, Heidelberg, Germany) with a field size of 20° x 15° centered on the fovea and an OCT image resolution of 512 x 496. A representative subset of 81 volume scans each with 145 B-scans consisting of at least 15 averaged frames and with an approximate inter-B-scan distance of 30 μm was used for this study. These scans were selected from a larger set of 682 OCT volume scans from 98 patients, keeping about one scan per patient in order to cover a wide range of image quality and drusen phenotypes. Those volumes that had very few B-scans or insignificant drusen load were discarded from the final subset. Geographic atrophy was defined according the cRORA criteria [152].

6.3.2 Replication of Chen et al. algorithm

As described in the work of Chen et al. [18] several techniques are used to denoise the image, remove the retinal nerve fiber layer (RNFL) and compute the centerline of the retinal pigment epithelium (RPE) (Figure 6.4 and 6.5). In their algorithm, Chen et al. first denoise the input, using a bilateral filter with an anisotropic window to account for the stretch of B-scans in the horizontal direction. Then B-scans are binarized using a threshold $0.3 \times t$ to detect and remove the RNFL layer, and with threshold t in order to detect ellipsoid zone and the RPE. The center-line of the estimated layer is considered as the final estimation of the RPE. In order to detect drusen, a 3rd degree polynomial is fit on the RPE layer, estimating a drusen-free RPE. The area between the drusen-free RPE and RPE layer are considered as drusen. After detecting drusen per B-scan, the en face OCT image was used for a false-positive- elimination-step as proposed by Chen et al. [18] (Figure 6.6). The details of this algorithm, as well as the data-specific tuning needed to adjust the algorithm for the MODIAMD data-set, is provided in the supplementary materials.

We observed that, in Chen's approach, inclusion of parts of the ellipsoid zone along the RPE can lead to jumps in the estimated RPE layer, as shown in Figure 6.7 (false positives). Therefore, our work suggests further refinements for a more robust determination of the RPE layer.

6.3.3 Refinement of the algorithm

Our refined algorithm reduces the above described false positives by better separating ellipsoid zone and RPE. In particular we found that shadows can cause the brightness of the RPE to vary substantially within B-scans, as it can be seen in Figure 6.7. In these cases, it is less suitable to use a global threshold for RPE segmentation. Therefore, we propose three refinement steps for thresholding, where the segmentation mask of each step is used as a guide to remove irrelevant components of the mask that is computed at the next step. We also replace the bilateral filter, which is used for denoising by Chen et al., with the multi-scale anisotropic fourth-order diffusion (MAFOD) filter [204]. This filter was developed specifically to enhance ridges, which helps better localizing the center-line of the estimated RPE layer (Figure 6.8).

In the first refinement step (Figure 6.9), we perform the thresholding with respect to both denoised input B-scan and a local histogram equalization of the input

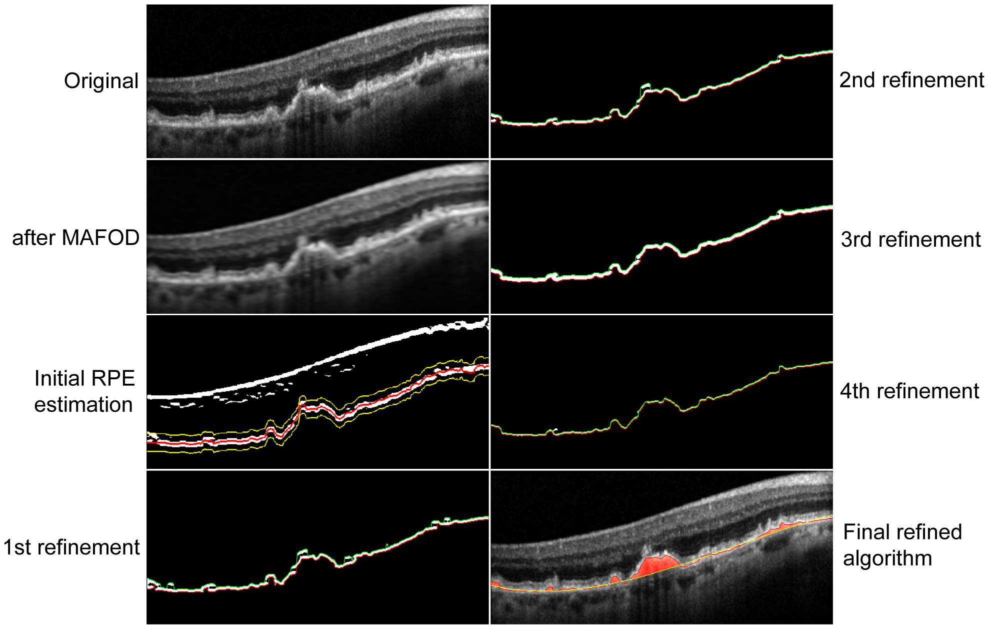


FIGURE 6.1: The input B-scan (left, first row) is filtered with MAFOD filter (left, second row). The retinal pigment epithelium (RPE) is estimated, twofold enlarged and threshold-positive pixels outside a 20-pixel band around the RPE centerline (left, third row; red and yellow indicate the centerline and the 20-pixel band, respectively) are removed (left, last row; green and red indicate the upper and lower boundary of the RPE, respectively). The second, third and fourth refinement steps are applied (right, first, second and third row; green and red indicate the upper and lower boundary of the RPE, respectively) to achieve the final refined algorithm (right, last row). See supplement for details.

B-scan. In the second refinement step (Figure 6.10), a threshold with respect to the local histogram equalization of the denoised image, and the denoised image itself is selected to binarize the B-scan. In the third refinement step (Figure 6.11), a Gaussian blur is applied on the denoised image. Then the same steps as in the second refinement step are applied on the smoother version of the denoised B-scan. Finally, in the fourth refinement step, we implemented an improved method for estimating the lower and upper boundaries of the RPE from the segmentation mask by boundary tracing and polynomial fitting (Figure 6.12). We take the center line between the upper and lower boundary as the final refined estimation of the RPE layer. Figure 6.1 illustrates the different refinements step-by-step on one B-scan from the original image till the final refined algorithm. More details on the refined algorithm can be found in the supplementary materials.

6.4 Results

In exact analogy to Chen et al. [18], algorithm performance was quantitatively evaluated by the absolute drusen area difference (ADAD) and overlap-ratio in relation to the ground truth. Similar to Chen et al. we evaluated the algorithm's performance for a dataset where only B-scans with drusen were included and for a subset where only the B-scan with most drusen load in each volume scan was included [18]. Original results from Chen et al. are displayed in Table 6.1 and our results

	ADAD [μm]	ADAD [%]	OR ± SD
B-scans with drusen ('4/340 dataset')	10.29 ± 8.9	15.70 ± 15.50	76.33 ± 11.29
B-scans with largest drusen load per volume ('143/143 dataset')	19.97 ± 14.68	23.77 ± 13.8	67.18 ± 9.14

TABLE 6.1: Original results from Chen et al.[18]. Image resolution of the used dataset: 512 × 1024 and 128 B-Scans per volume scan; OR = overlap ratio; SD = standard deviation

of the replicated algorithm are reported in Table 6.2. Performance of the replicated algorithm was inferior to the reported original algorithm and performance of the refined algorithm was superior to the replicated one.

Even though we present the same error metrics as used by Chen et al. in order to facilitate a direct comparison, we noticed two limitations in the way they are defined. First, restricting the evaluation to B-scans in which drusen are present might conceal some of the false positives that occur in drusen-free B-scans. However, these are practically relevant, since we want to employ the algorithm fully automatically, without having to flag drusen present B-scans manually. Second, overlap ratio is a relative error metric, as is ADAD when expressed in percent. Therefore, computing based on B-scans can result in inflated estimates in the presence of B-scans with low drusen load, since even small absolute segmentation errors will correspond to a large relative error. We also note that based on the information given in the publication by Chen et al. [18] it is not completely clear how overlap ratio was aggregated, it is merely stated to be "similar as for the ADAD metrics".

For these reasons, Table 6.2 presents an additional evaluation, referred to as "volumetric computation", which is based on the full OCT volumes. Given a three-dimensional ground truth drusen mask M_i for the ith OCT volume, and a corresponding algorithmic estimate $\widehat{M_i}$, our volumetric computation error measures can be expressed as

$$\mathrm{ADAD_i} = \left|\mathrm{Area}\left(M_i\right) - \mathrm{Area}(\hat{M}_i)\right|$$

$$\text{Overlap Ratio}_i = \frac{\mathrm{Area}\left(M_i \cap \hat{M}_i\right)}{\mathrm{Area}\left(M_i \cup \hat{M}_i\right)}$$

where $\mathrm{Area}(M_i)$ denotes the overall drusen area, summed over all B-scans of the ith volume. For error measure $E \in \{\mathrm{ADAD}, \mathrm{OR}\}$, mean and standard deviation are computed according to their established definitions:

$$\mu_E = \frac{1}{N}\sum_{i=1}^{N} E_i, \quad \sigma_E = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(E_i - \mu_E\right)^2}$$

These alternative error metrics differ from the ones used by Chen et al. in two main ways: First, they also account for false positives in drusen-free B-scans (there were 1,934 drusen-free B-scans in our dataset). Second, mean and standard deviation are taken over volumetric OCT scans rather than B-scans to avoid an inflated effect of B-scans with low drusen load on relative error measures. In non-volumetric measures as proposed in Chen et al. paper, per B-scan ADAD is divided by the number of A-scans with drusen present in that B-scan. For a fair comparison between the

	Replicated Chen et al. [18]		
	ADAD ± SD (μm)	**ADAD ± SD (%)**	**OR ± SD (%)**
B-scans with drusen	17.60 ± 36.70	100.59 ± 304.80	24.52 ± 20.56
B-scans with largest drusen load per volume	19.94 ± 13.54	42.70 ± 22.70	39.24 ± 22.06
Volumetric Computation	11.96 ± 12.11	46.37 ± 75.36	29.35 ± 17.32
	Refined algorithm		
	ADAD ± SD (μm)	**ADAD ± SD (%)**	**OR ± SD (%)**
B-scans with drusen	13.28 ± 29.40	73.54 ± 217.80	35.88 ± 25.25
B-scans with largest drusen load per volume	15.64 ± 11.05	36.86 ± 24.34	51.90 ± 23.70
Volumetric Computation	8.31 ± 6.87	30.05 ± 29.76	42.20 ± 20.47

TABLE 6.2: Comparison of the algorithms to the ground truth . Image resolution of the used dataset: 512 x 496 and 145 B-Scans per volume scan. OR = overlap ratio; SD = standard deviation.

volumetric ADAD measure to its non-volumetric alternatives, we divide ADAD by the sum of the A-scans with drusen for all B-scans in the volume.

Separation of the RPE and the ellipsoid zone, which was a major source of segmentation errors in the replicated algorithm, could be improved by the refined algorithm. Our refinement of this algorithm was able to detect the course of the RPE-centerline more reliably (Figure 6.2). This suggests that our refined algorithms allows for a more robust determination of the RPE layer than the replicated algorithm. When comparing the replicated and refined Chen et al. algorithm there were only two OCT volume scans where the replicated algorithm outperformed the refined one. It should be noted that algorithm performance was low for both algorithms in these two OCT volume scans (OR of 13% and 11% for the refined algorithm, and 14% and 15% for the replicated algorithm, respectively). Both of the OCT volume scans were from eyes with subretinal drusenoid deposits.

The mean drusen load of the complete dataset was $153,166,119$ μm^3 in the ground truth, $119,321,550$ μm^3 in the replicated algorithm, $125,837,159$ μm^3 in the refined algorithm. Hence, both algorithms underestimate drusen load, but our refined one less severely. To investigate the role of the false-positive elimination step for this underestimation, we compared total drusen calculation with and without false-positive elimination. We found that drusen load was underestimated even without false positive elimination (replicated algorithm: $132,385,230$ μm^3, refined algorithm: $132,555,984$ μm^3).

In order to estimate the effect of the drusen load we stratified our results for small ($0-26,465,894 \mu m^3$), medium ($26,465,894$ – $92,630,630 \mu m^3$) and large ($> 92,630,630$ μm^3) drusen load per volume scan (Figure 6.3). Overlap ratio increased for all datasets and algorithms with increasing drusen load.

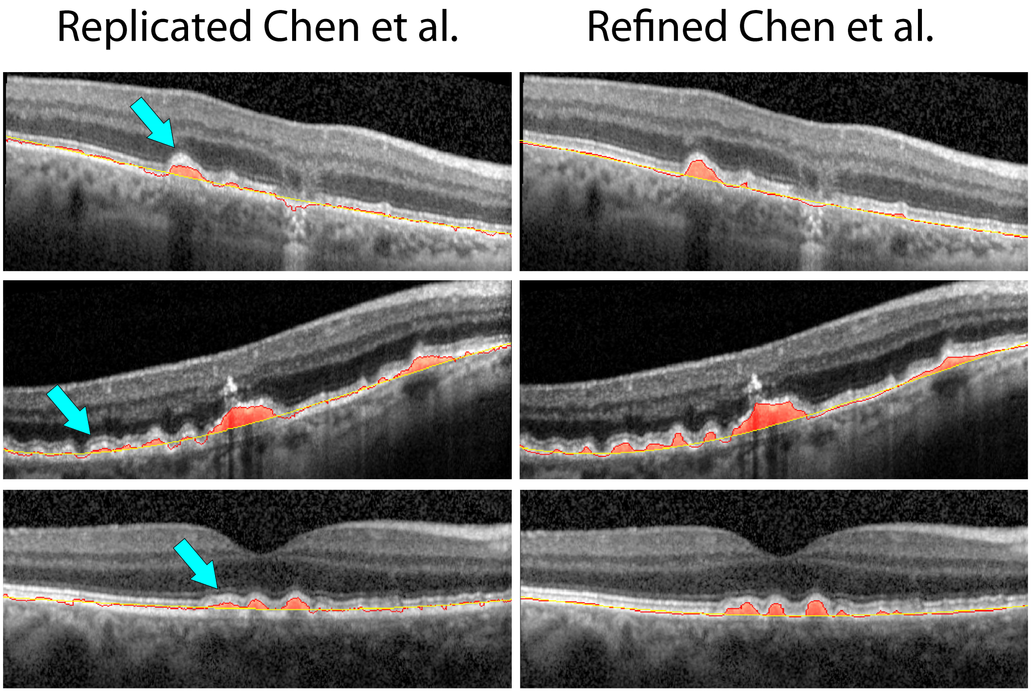


FIGURE 6.2: Comparison of replicated and refined drusen segmentation. Results by rebuilt Chen et al. method (left), and from refined method (right) marked over the input B-scan.

To investigate the effect of geographic atrophy on algorithm performance, we compared results in B-scans with and without geographic atrophy, which was present in 17 of our OCT volume scans (21%). As presence of geographic atrophy is correlated with greater drusen load, and since we found algorithm accuracy depending on drusen load, we performed a multiple regression analysis for both algorithms with OR from our volumetric computation as the dependent variable and the binned drusen load categories "small", "medium", and "large" and presence of geographic atrophy as independent variables. The regression analysis showed a significant association with drusen load categories but not with geographic atrophy (see Table 6.3).

6.4.1 Statistical analysis

We used the Wilcoxon signed-rank test to evaluate the segmentation improvement using our proposed refined algorithm, with respect to ADAD measure in μm. The Shapiro-Wilk test on the ADAD values showed a non-normal distribution with the p-values of 6.9x10^{-12} for ADADs computed per OCT volume, p-value=0 for B-scans with at least one druse, and p-value=1.1x10^{-8} for B-scans with largest drusen load per volume. This motivated us to use a non-parametric version of the paired Student's t-test, i.e., the Wilcoxon signed-rank test. With this test, the p-values for the paired samples using the replicated and refined Chen et al. algorithms are; 2.4x10^{-6} for ADADs computed over the complete OCT volumes, p-value=0 for B-scans with at least one druse, and p-value=1.2x10^{-5} for B-scans with largest drusen load per volume. These numbers show a significant segmentation improvement with respect to ADAD measure, when refined Chen et al. algorithm is used.

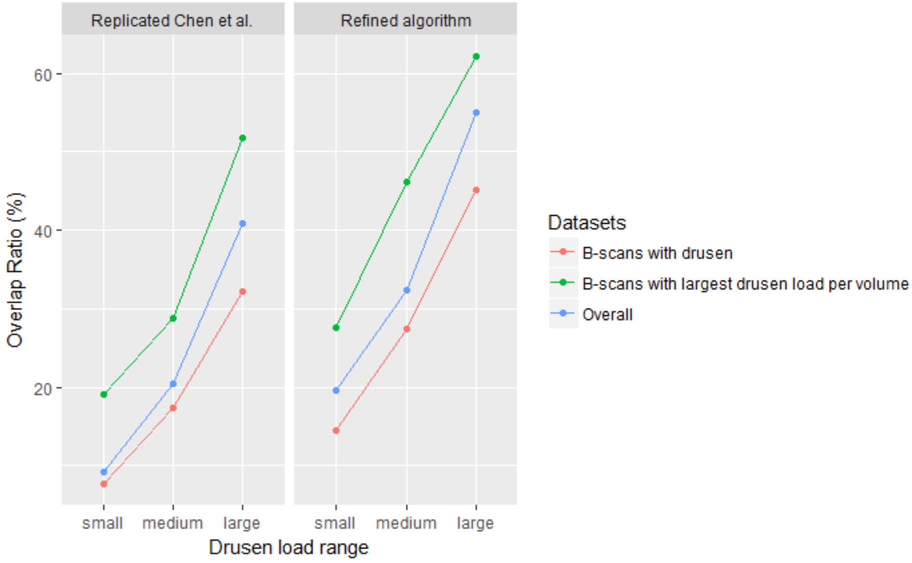


FIGURE 6.3: Algorithm performance stratified for drusen load.

6.4.2 Risk of bias evaluation

Risk of bias for the new drusen segmentation algorithm was evaluated according to the standardized protocol as published previously [192]. The patient sample was recruited out of the MODIAMD data with inclusion criteria as stated above. The algorithm development and testing were performed in separated data subsets, the reference standard was a manually corrected segmentation of the RPE layer and the reference standard was objectively compared with the index test (via ADAD and OR). There were two ophthalmologists as readers for the annotations (ST and MW) and there were no repetitive measurements by these readers. All included patients received the same reference standard. Parameters increasing risk of bias are that subjects were not randomly recruited and there were no repetitive measurements done.

6.5 Discussion

Re-building an algorithm based on the details provided in the publication we found its performance to be inferior to the reported results. This was likely due to a more heterogeneous dataset including a more real-life patient selection and a greater diversity in drusen load and differences in image quality. Following this, several refinements considerably improved the overall performance of the algorithm in our image dataset. These findings highlight that automated algorithms have to be used with caution, particularly when little or no independent evaluation and validation are available.

We identified multiple factors that might explain the discrepancy between our results and those reported by Chen et al., the most important being the patient cohort. Different from Chen et al., our study population included individuals with geographic atrophy, making it more heterogeneous and closer to a real-world situation. In addition, we conjectured that differences in drusen load might explain differences in performance. Against this background we performed a multiple regression analysis which showed no association of algorithm performance with presence of geographic atrophy, however strong association with drusen load. Therefore, we hypothesize differences in drusen load might be an important factor for the observed difference in algorithm performance. However, this remains speculative as the absolute drusen load was not published by Chen et al.. Another factor concerns image quality. The reduced axial resolution of 496 pixels compared to the input data of the Chen et al. algorithm which had an axial resolution of 1024 pixels is likely to reduce accuracy of the separation of RPE and ellipsoid zone and reduce performance somewhat. Although algorithm accuracy might further improve by exclusion of poor or reduced image quality , e.g. due to lens opacities, we did not apply any ancillary exclusion criteria additional to the MODIAMD exclusion criteria[175], as our intention was to investigate algorithm performance in a dataset which is as close to 'real-life' conditions as possible.

These differences in patient and image characteristics are unavoidable, since the data used by Chen et al. is not available to others. Characteristics of OCT image data can differ greatly, e.g. due to differences between devices or in patient samples. Even when using the same device, differences in image acquisition are possible based on varying number of repetitive frames and consecutive image averaging, altered resolution or field-of-view settings, etc.. Against this background, it is even more important to provide highly specific details on both the algorithms as well as the image data used in their creation and evaluation. It also highlights the importance of testing algorithms on different datasets to achieve a full validation.

As an aside we proposed alternative or additional error metrics which might bear advantages over the so far used error metrics. However, none of our main results depend on the choice of error metric and in particular the improvement in algorithm performance following our refinements is independent of this.

As the refined Chen et al. algorithm still underestimated overall drusen load, one possible approach to reduce underestimation of drusen load might be the improvement of the polynomial fitting step. Both algorithms (as many others) use polynomial fitting for ideal RPE estimation. However in presence of large drusen or many small drusen, the estimated ideal RPE is 'lifted up', hence, leading to drusen underestimation. There were only two OCT volume scans where the replicated algorithm outperformed the refined one. Overall performance was low for both OCT volume scans and both were from eyes with subretinal drusenoid deposits, rendering presence of subretinal drusenoid deposits a potential limitation of both the replicated and the refined algorithm. A possible next step would be to statistically compare algorithm performance in eyes with and without subretinal drusenoid deposits. In case this assumption is confirmed, further refinements might be introduced to specifically deal with subretinal drusenoid deposits. Our results showed a lower ADAD in the volumetric computation metric compared to the subsets "B-scans with drusen" and "B-scans with largest drusen load per volume". This observation can be explained by the fact that both algorithms underestimate overall drusen load. The volumetric computation metric adds the contribution of false positives in the drusen-free B-scans, which reduces this underestimation, and therefore leads to a smaller difference between estimated and actual drusen load. The fact that the volumetric computation metric increases OR compared to averaging over all B-scans with drusen, but not to B-scans with largest drusen load, can be explained by OR being a relative error measure. Therefore, its volumetric computation variant reduces the effect of B-scans with low drusen load, where even small absolute errors can cause a drastic reduction of OR.

Herein, we focussed on an established segmentation algorithm based on traditional image processing techniques such as filtering and thresholding. Recently, convolutional neural networks (CNNs) and deep learning have become more popular also for automated image analysis in ophthalmology [51, 161]. We expect that issues of reproducibility and transferability will become even more relevant when using deep learning. Reproducing a learning based method involves not just re-implementation, but also re-training of the algorithm. Consequently, the used training datasets and their evaluation need to be highly transparent and reproducible. Moreover, CNNs' results can be easily skewed by specific targeted manipulation of the input data, not recognizable by humans [179]. It is even possible to generate images completely unrecognizable to humans, which deep learning algorithms believe to be recognizable objects with >99% certainty [122]. This highlights some of the advantages of conventional, human-designed algorithms, where it is easier to rationalize the effects of factors such as reduced image resolution, and to ameliorate them with refinements such as those proposed in our current work.

To our knowledge, within the field of ophthalmic image analysis, we present the first study on re-implementing an algorithm based on the details provided in its publication, drawing attention to the important issues of algorithm reliability and replicability. Our results highlight that more details, both concerning the algorithm and the data that it is applied to, might be relevant for a proper replication than is typically given in a publication.

Further strengths of our study are the employment of a more real-life patient selection as dataset and a detailed step-to-step explanation of the algorithm's refinement. Furthermore, we also made a comprehensive evaluation of algorithm performance, a direct comparison of the replicated and the refined algorithm in the same dataset and a subgroup analysis for drusen load. A limitation of our study is the reduced axial resolution as compared to the input data of the original algorithm. However, this reflects a more realistic real-life dataset.

In conclusion, we replicated a reportedly well-performing algorithm for OCT-based drusen quantification and found algorithm performance to be inferior to the reported results for various reasons. Several refinements considerably improved algorithm performance in our sample but still did not achieve published results. Replication of a published algorithm based on the details provided in a publication is challenging and better standards to ensure algorithm reproducibility, reliability and validity should be established as an increasingly large part of day to day clinical medicine is informed by automated image analysis algorithms. An important step towards this goal is to make program code publicly available. The Python code underlying our current manuscript can be found at https://github.com/MedVisBonn/DrusenSegmentation-ModifiedChen.

6.6 Acknowledgement

This research was supported by the Else Kröner-Fresenius Foundation / German Scholars Organization (EKFS/GSO 16) to RF, the BONFOR GEROK Program, Faculty of Medicine, University of Bonn, (Grant No O-137.0028) to MW, the GEROK Program, Faculty of Medicine, University of Bonn, (Grant No O-137.0026) to ST and the German Ministry of Education and Research (BMBF), FKZ 13N10349.

6.7 Contributors

MWMW, VW, RPF and TS conceived the study, SGZ and VW performed the study, SGZ, VW, MWMW and TS analyzed the data, ST and SS provided image material, TS, RPF and FGH provided scientific oversight, MWMW, VW and TS did the literature review, MWMW, VW, SGZ, RPF and TS drafted the manuscript and all authors critically reviewed the manuscript.

6.8 Competing financial and non-financial interests' statement

MWMW: Heine Optotechnik GmbH (research material, consultant), Digisight Technologies (research material, travel reimbursements), D-EYE (research material), Carl Zeiss Meditec (research material), CenterVue (research material), Heidelberg Engineering (research material), Optos (research material); SGZ: nothing to disclose; VW: nothing to disclose; ST: Bayer (honoraria / travel reimbursement), Carl Zeiss Meditec (financial support), CenterVue (financial support), Heidelberg Engineering (financial support, honoraria / travel reimbursement), Novartis (honoraria / travel reimbursement), Optos (financial support); SS: Acucela (financial support), Alcon/Novartis (consultant, financial support, honoraria / travel reimbursement), Allergan (consultant, financial support, honoraria / travel reimbursement), Bayer (financial support, honoraria / travel reimbursement), Bioeq/Formycon (financial support, consultant), Carl Zeiss MedicTec (financial support, honoraria / travel reimbursement), CenterVue (financial support), Galimedix (consultant), Genentech/Roche (financial support, honoraria / travel reimbursement), Heidelberg Engineering (financial support), Katairo (financial support), Optos (financial support); FGH: Heidelberg Engineering (financial support, consultant), Novartis (financial support, honoraria / travel reimbursement), Bayer (financial support, honoraria / travel reimbursement), Genetech (financial support, honoraria / travel reimbursement), Acucela (financial support, honoraria / travel reimbursement), Boehringer Ingelheim (honoraria / travel reimbursement), Alcon (financial support, honoraria / travel reimbursement), Allergan (financial support, honoraria / travel reimbursement), Optos (financial support, honoraria / travel reimbursement), Carl Zeiss MediTec (financial support); RPF: Heidelberg Engineering (financial support), Optos (financial support), Carl Zeis Meditec (financial support), CenterVue (financial support), Novartis (financial support, consultant), Bayer (financial support, consultant), Ellex (consultant), Santhera (consultant), RetinaImplant (consultant), Opthea (consultant); TS: nothing to disclose;

6.9 Role of the funding source

The funders had no role in study design, data collection, data analysis, data interpretation, or writing of the report.

6.10 Supplementary material and methods

6.10.1 Replication of Chen et al. algorithm

In the algorithm proposed by Chen et al. [18], an anisotropic bilateral filter is applied on the input B-scan, with a window size of 7x19 pixels to account for the stretch in the horizontal direction of B-scans. In our work, we adapted the window size to 5x15 pixels to match the horizontal stretch in our dataset. In order to detect the

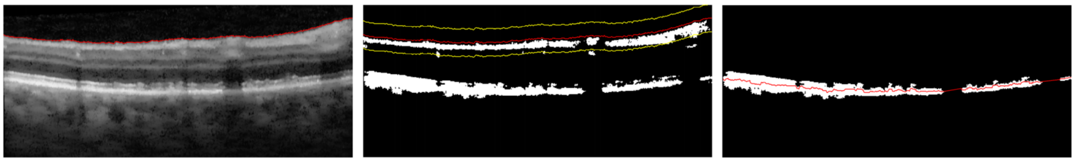


FIGURE 6.4: Retinal nerve fiber layer removal and retinal pigment epithelium estimation. Estimation of the retinal nerve fiber layer (RNFL) (left), thresholding and narrow band around the estimated RNFL (middle) and estimation of the retinal pigment epithelium centerline (right).

retinal pigment epithelium (RPE) and ellipsoid zone, a threshold *t*, which is computed using the cumulative histogram, is applied on the denoised B-scan. The left and middle section of Figure 6.4 show the B-scan before and after thresholding at *t*. Since the RNFL layer is bright as well, it will also be detected by threshold *t*. In order to remove it, the denoised image is thresholded at 0.3**t*. After this, regions of size less than 50 pixels are removed from the thresholded image. The RNFL is then estimated using the first pixel that belongs to the foreground pixels in each column.

For estimating the RPE layer, after thresholding the denoised image at *t*, regions smaller than 50 pixels and all foreground pixels in a neighborhood of 20 pixels within the RNFL layer are removed from the mask. For further clean up, the lower boundary of the RPE region is estimated using the last foreground pixel in each column. Any foreground pixel that falls outside of a 20-pixel neighborhood of the lower boundary is removed. Finally, the RPE layer is estimated by fitting a 3rd degree polynomial to the points at the center of the RPE region (right section in Figure 6.4). Drusen were defined as the area between the RPE fitted curve as the upper and the ideal RPE estimation as the lower boundary with a minimum drusen height of 2 pixels (Figure 6.5). After detecting drusen per B-scan, the en face OCT image was used for a false-positive- elimination-step as proposed by Chen et al. [18] (Figure 6.6.

However, Figure 6.7 (false positives) shows how inclusion of parts of the ellipsoid zone along the RPE can lead to jumps in the estimated RPE layer.

6.10.2 Refinement of Chen et al. Algorithm

Our proposed refinement of Chen et al. algorithm consists of four refinement steps and replacement of the bilateral filter with the multi-scale anisotropic fourth-order diffusion (MAFOD) filter [204].

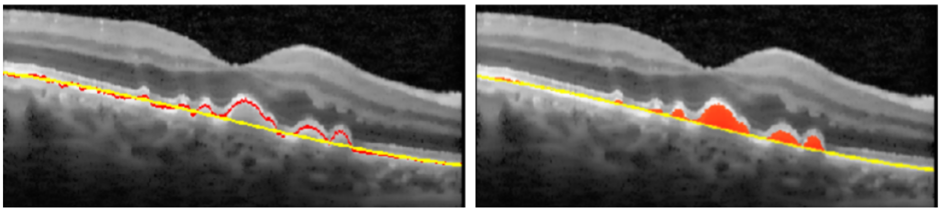


FIGURE 6.5: Drusen detection. Cropped image of an exemplary B-scan with the retinal pigment epithelium centerline in red and fitted curve in yellow (left) and the resulting drusen segmentation with red domes indicating the detected drusen (right).

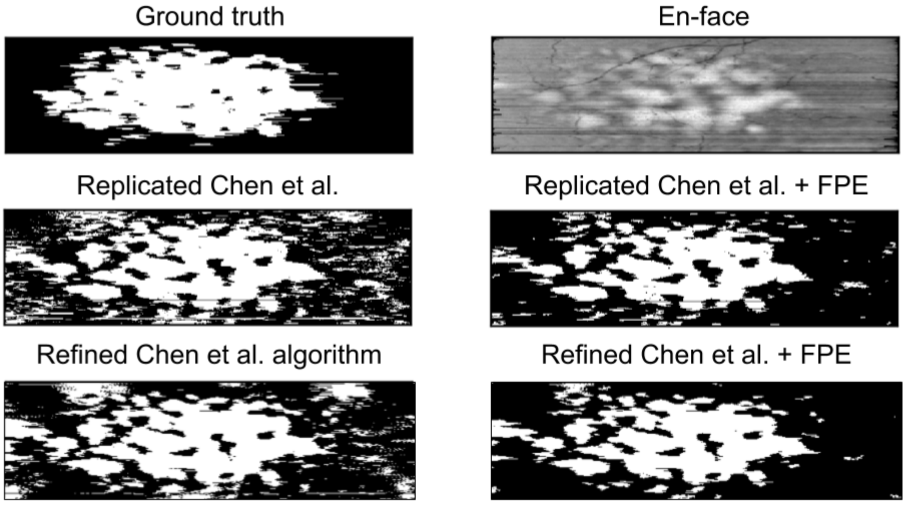

Ground truth
En-face
Replicated Chen et al.
Replicated Chen et al. + FPE
Refined Chen et al. algorithm
Refined Chen et al. + FPE

FIGURE 6.6: False Positive Elimination (FPE). Segmented drusen are displayed as white against black background with and without FPE. The en-face image is based on the OCT volume scan.

The MAFOD filter was developed specifically to enhance ridge structures, which appear as bright curves on darker background, e.g., the RPE layer in OCT images. The MAFOD filter automatically detects the scale of the underlying ridge. It then uses the eigenvalues and eigenvectors of the local Hessian matrix, computed at that scale to create a fourth order diffusion tensor, which is utilized to create an anisotropic diffusion effect. The enhancement results in a more accurate estimation of RPE (see Figure 6.8). Since within each A scan, shadows similarly affect the brightness of RPE and ellipsoid zone, we address this issue by performing a local histogram equalization within A scans before applying the thresholding. Moreover, in order to reduce speckle noise, Chen et al. work with a filtered version of the OCT scan. Even though we found this noise reduction to be desirable, it also involves a certain amount of blurring that makes it more difficult to separate RPE from ellipsoid zone. We balance the advantages and drawbacks of filtering by incrementally refining the segmentation by Chen et al. in three steps, which are described below, and are based on the original B-scan, as well as two differently filtered versions. As a fourth refinement, we propose an improved method for extracting the RPE middle axis from the segmentation, in a way that filters out some of the remaining false positives.

In the first refinement step, after initial estimation of RPE using Chen's method, we use a local histogram equalization with a window of height 80 pixels and width of 1 pixel. Since the goal is to find the RPE in this equalized image, a thresholding with respect to both denoised and locally equalized image is used to detect the brighter regions that are more probable to be part of the RPE. The pixels with values

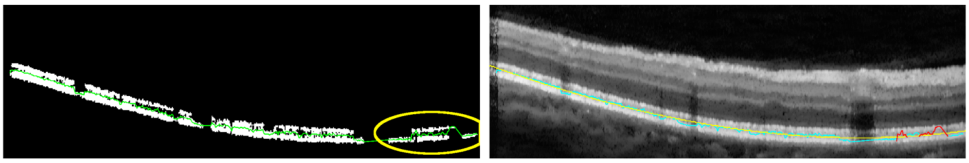


FIGURE 6.7: Limitation of the replicated algorithm: false positive drusen. Threshold image and centerline (left) and resulting false positive drusen detection (right).

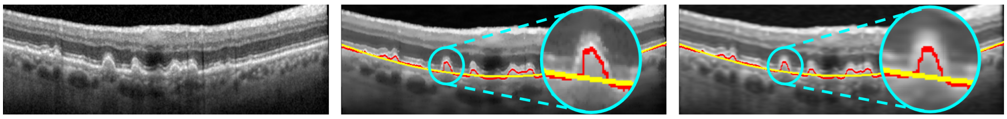


FIGURE 6.8: Replacing bilateral filter with MAFOD filter. Input B-scan (left), filtered with bilateral filter (middle), and filtered with MAFOD filter (right). MAFOD filter preserves or enhances centerline's intensity value with respect to its neighboring pixels, which results in a more accurate binary estimation of the RPE layer, and consequently a more natural shape of the final layer segmentation.

less than 20% of the intensity range in the denoised image, and pixels with values less than 90% of the intensity range in the locally equalized image are set to be background (not part of the RPE). The rest are set as foreground pixels. Using the knowledge that the RPE is thin, far away binary components are set to background. Thus the initial RPE layer estimation that was computed using Chen's method is used to set any pixel outside a 20-pixel neighborhood of the estimated RPE as background. After this step, components smaller than 50 pixels are removed. Then, in order to fill gaps and holes in the estimated area, a dilation of size 2 is performed (see Figure 6.9). The newly estimated RPE gives us a more accurate boundary of the RPE area that we will deploy in the further refinement steps.

In the second refinement step as shown in Figure 6.10, a window of size 10 x 1 pixels is used for histogram equalization on the denoised image. Again, a threshold with respect to the denoised image and the locally equalized image is used to binarize the B-scan, i.e., all pixels that have an intensity less than 20% of the intensity range in the denoised image, and pixels with value less than 80% of the intensity range in the equalized image itself, are set to background. In addition, pixels that are outside the estimated RPE area from the first refinement step are removed. Finally, a dilation followed by erosion filter is used to fill in the small holes on the estimated RPE.

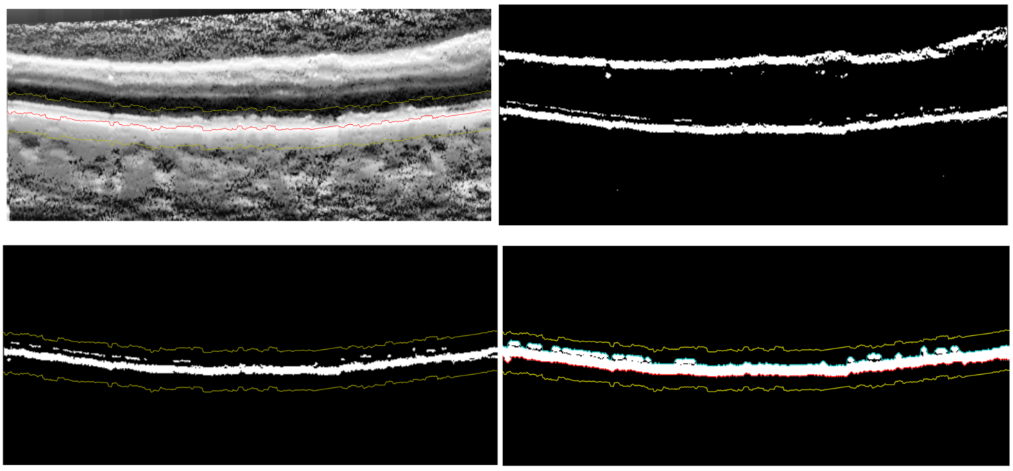


FIGURE 6.9: First refinement of the retinal pigment epithelium determination. Estimated centerline and local histogram equalization with window size: 80 × 1. (upper left), thresholding with threshold t = 0.9 (upper right), removed RNFL and other bright pixels outside the centerline band (lower left) and double dilation and computation of upper and lower lines (lower right).

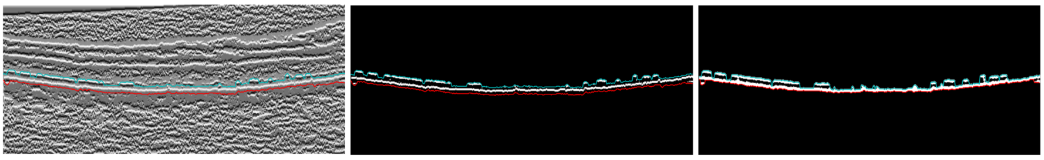


FIGURE 6.10: Second refinement of the retinal pigment epithelium determination. Local histogram equalization with a window size of 10 × 1 (left), thresholding with t = 0.8 (middle) and an update of the upper and lower lines (right) is shown.

In the third refinement step, the same procedure is repeated on a Gaussian blurred version of the denoised image (see Figure 6.11). We empirically found σ=2 works well on our data-set. The remaining steps are similar to the second refinement step, except that the boundary of the mask computed from the second refinement step are used for removing the non RPE pixels, and only one dilation is performed for filling the gaps.

Drusen detection is based on the middle axis of the RPE. To extract it, Chen et al. simply took the center of the highest and lowest pixels from the RPE segmentation mask in each column of the image, and linearly interpolate in case of gaps. To reduce the impact of small erroneous components that might remain in the final segmentation mask, in the fourth refinement step, we implemented an improved method for estimating the lower and upper boundaries of the RPE from the segmentation mask (see Figure 6.12).

We will explain the procedure in detail for the upper boundary. The lower boundary is found in complete analogy. For finding the upper boundary, our procedure is based on considering the connected components of the segmentation mask, and finding two paths through them, from left to right and from right to left. We only

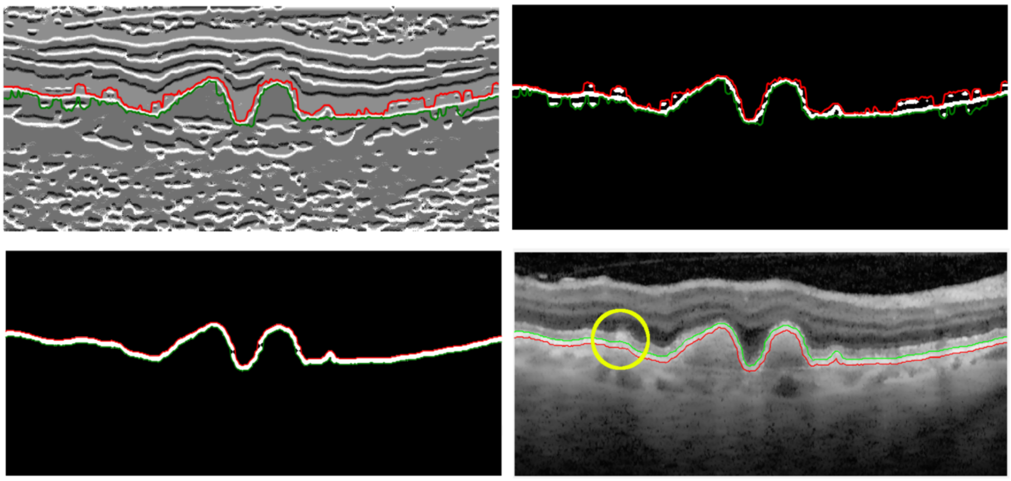


FIGURE 6.11: Third refinement of the retinal pigment epithelium determination. The resulting image after Gaussian blur and local histogram equalization with a window size of 10 ×1 and red and green lines indicating the upper and lower boundaries estimated from the previous refinement (upper left), thresholding (upper right), the resulting regions produced by shortest path computation (lower left) and an under estimated druse indicated by a yellow circle (lower right).

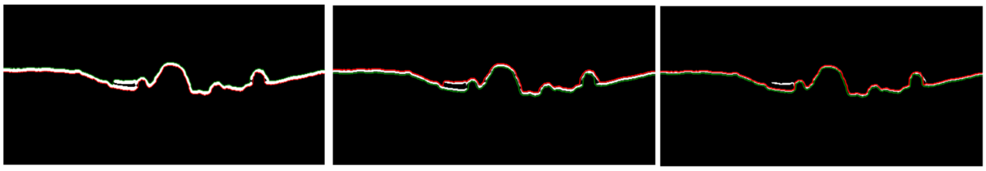


FIGURE 6.12: Fourth refinement: Improved estimation of the upper retinal pigment epithelial boundary. Result from previous refinement (left), Gaussian, local histogram equalization and thresholding without dilation as in the left image (middle) and the improved upper line estimates (right).

	Replicated Chen et al. [18]		
	coefficient	**p-value**	**95% CI**
Intercept	9.23	.025	1.19 till 17.28
Binned 'medium' relative to 'small' drusen load	11.18	.019	1.90 till 20.46
Binned 'large' relative to 'small' drusen load	31.57	< .0001	22.58 till 40.57
Presence of geographic atrophy	-0.035	.992	-6.99 till 6.92
	Refined algorithm		
	coefficient	**p-value**	**95% CI**
Intercept	20.00	< .0001	10.07 till 29.96
Binned 'medium' relative to 'small' drusen load	13.04	.026	1.57 till 24.51
Binned 'large' relative to 'small' drusen load	35.86	< .0001	24.75 till 46.98
Presence of geographic atrophy	-3.80	.381	-12.41 till 4.80

TABLE 6.3: Multiple regression analysis for drusen load and presence of geographic atrophy as independent variables and overlap ratio from volumetric computation as dependent variable. CI = confidence interval.

preserve components that are either larger than 200 pixels, or have been used in the paths in both directions. Other components are considered as outliers, and are discarded. From left to right path, we start with the topmost pixel of the leftmost component. From there, always the highest pixel from the current component is picked and is heuristically followed over the upper boundary of the same component until the end is reached. If we reach a column that is not covered by the current component, we move to the topmost component in that column and skip empty columns. After the left-to-right path reaches the image boundary, the last position is used as an initialization point for the search in the inverse direction using the same technique. In the end, we only preserve components that are either very large, or have been used in the paths in both directions. The others are discarded as outliers. As the final step, a third degree polynomial is fit to the upper and the lower boundary, separately. The center-line between the fitted curves is considered as the RPE layer. The python code for both replicated and refinement of Chen et al.'s algorithm can be found at `https://github.com/MedVisBonn/DrusenSegmentation-ModifiedChen`.

Chapter 7

CNNs Enable Accurate and Fast Segmentation of Drusen in Optical Coherence Tomography[1]

7.1 Abstract

Optical coherence tomography (OCT) is used to diagnose and track progression of age-related macular degeneration (AMD). Drusen, which appear as bumps between Bruch's membrane (BM) and the retinal pigment epithelium (RPE) layer, are among the most important biomarkers for staging AMD. In this work, we develop and compare three automated methods for Drusen segmentation based on the U-Net convolutional neural network architecture. By cross-validating on more than $50,000$ annotated images, we demonstrate that all three approaches achieve much better accuracy than a current state-of-the-art method. Highest accuracy is achieved when the CNN is trained to segment the BM and RPE, and the drusen are detected by combining shortest path finding with polynomial fitting in a post-process.

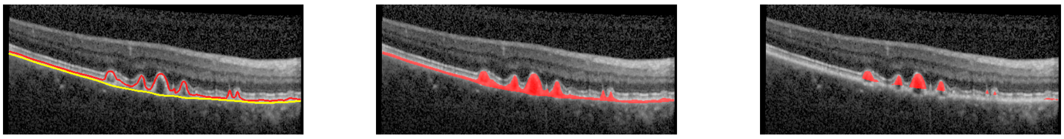
7.2 Introduction

Age-related macular degeneration (AMD) is the most common cause of irreversible vision loss for people over the age of 50 in the developed countries [69]. Drusen, i.e., focal deposits of acellular debris between the retinal pigment epithelium layer and Bruch's membrane, are usually the first clinical sign of AMD. Their size, number, and location can serve as biomarkers for disease progression.

Optical Coherence Tomography (OCT) is a fast and non-invasive way of obtaining three-dimensional images of the retina, and is increasingly used to monitor the onset and progression of AMD [1]. We would like to use OCT in large epidemiological studies, which requires detecting and quantifying drusen in tens of thousands of eyes, and is infeasible by manual analysis.

Even though considerable progress has been made on (semi-)automated quantitative OCT image analysis, in recent years [170], current approaches still do not achieve sufficient accuracy, or require too much interaction, to be practical for use

[1] The content of this chapter has been previously published: Gorgi Zadeh et al. "Cnns enable accurate and fast segmentation of drusen in optical coherence tomography" *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support* Springer, Cham, (2017): 65-73 URL: https://doi.org/10.1007/978-3-319-67558-9_8 [51].



Layer segmentation map Complex segmentation map Drusen segmentation map

FIGURE 7.1: Different label images overlaid on the corresponding B-scan. The layer segmentation map has different classes for RPE (red) layer and BM (yellow) layer.

in large-scale studies. The fact that convolutional neural networks (CNNs) have recently achieved excellent results in related biomedical image analysis tasks [146] motivates us to develop and compare three CNN-based approaches to drusen segmentation. In an evaluation on more than 50,000 semiautomatically annotated B-scans (i.e., two-dimensional cross-sectional images acquired by OCT), we demonstrate that all three approaches outperform a current state-of-the-art technique for drusen segmentation. Highest accuracy is achieved by combining CNNs with application-specific post-processing.

7.3 Related Work

Among the many approaches for drusen segmentation [18, 40, 68, 72], we identified the method by Chen et al. [18] as a state-of-the-art reference, based on the reported accuracy and its successful use in a longitudinal study [166]. In Section 7.5.2, we will show that our CNN-based results compare very favorably to this method.

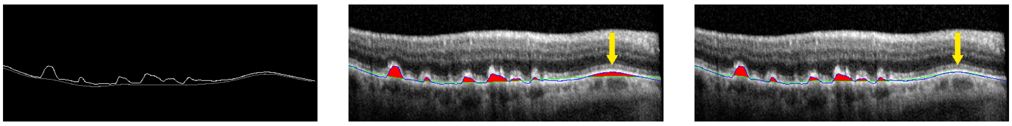
Even though we are not aware of any prior work that would have applied deep learning to the segmentation of drusen or retinal layers, two very recent works by Lee et al. [96] and Zheng et al. [210] have used convolutional neural networks to classify OCT images as either healthy or having AMD. The advantage of a segmentation, as it is achieved in our work, is that it can be used to derive intuitive measures such as size and number of drusen, which can be entered into progression models [160, 166], or correlated with genetic or lifestyle variables.

7.4 Method

7.4.1 Data Preparation

In each OCT scan, a three-dimensional volume is covered by a varying number of noncontiguous two-dimensional slice images, so-called B-scans. Our data set is generated as part of the MODIAMD (Molecular Diagnostics of Age-related Macular Degeneration) study, an observational cohort study on intermediate AMD, and it consists of 52,377 such B-scans (512×496 pixels) that belong to 682 OCT scans from 98 different subjects, each having a different number of followup scans taken. Each eye is scanned with a density of either 19 or 145 B-scans that cover approximately $5-6$ mm. Due to this varying density, we segment B-scans independently; accounting for 3D context is left for future work.

The annotations provided with the data are segmentation maps of the RPE and BM, as shown in Figure 7.1 (left). For each B-scan, a medical expert performed a careful manual correction of an initial segmentation that was created with a proprietary algorithm, spending about two minutes per B-scan on average.



BM and RPE layers Without rectification With rectification

FIGURE 7.2: Rectification helps to avoid false drusen detections. The blue curve is the RPE layer, the green curve is the estimated normal RPE and the red areas show the drusen.

Since a gold standard segmentation of the drusen themselves was not available, we generated reference drusen maps from the RPE and BM curves. To this end, we rectified the images by vertically shifting each column so that the BM forms a straight horizontal line. We then estimated the normal RPE layer by fitting a third degree polynomial to the segmented RPE in the rectified image, which was then warped back into the original image coordinates, where drusen were detected as areas in which the segmented RPE is elevated by more than two pixels above the normal RPE, cf. Figure 7.1 (right). An experienced rater confirmed that this procedure led to plausible drusen segmentation masks. Figure 7.2 illustrates how rectification helps to avoids false positive detections.

7.4.2 Network Architecture and Training

Because of its good performance in related segmentation tasks, we chose the U-Net architecture [146] as the basis of our methods. It consists of two symmetric paths, a contracting one that is used to capture image context, and an expanding one to recover the original resolution.

Since background is a dominant class in all our maps, it is crucial to assign spatially varying pixel weights. Similar to [146], we define weights according to

$$w(x) = w_c(x) + w_0 G_\sigma(x), \tag{7.1}$$

where w_c is defined as the overall number of pixels divided by those from class c, in order to account for class frequencies. In addition, training is focused on the exact boundaries in the segmentation masks by centering two-dimensional Gaussian weights $G_\sigma(x)$ with standard deviation σ on them. We keep the maximum weight in case weights from neighboring Gaussians overlap.

Since the number of foreground and boundary pixels differs greatly between our three approaches, we had to adapt the values of w_0 and σ accordingly. We will mention the exact values that were used in each experiment. They were found empirically in a small pilot experiment that involved only a small fraction of all data (3 out of 682 scans for training, one for testing).

Stochastic gradient descent from the caffe framework [75] is used for training. The momentum is set to 0.99. The initial learning rate is 10^{-3}, and is reduced at every $200,000$ steps by a factor of 0.1. We stopped the training when the validation accuracy plateaued, i.e., at $603,000$ iteration. We used the U-Net implementation provided by the authors, including their augmentation layer, and used the same overlap-tile strategy with 2 tiles per image. We added batch normalization layers after ReLu layers to avoid internal covariate shift [67].

7.4.3 Three Approaches to Drusen Segmentation

The most straightforward way of applying the U-Net to our problem is **direct segmentation,** i.e., training on the drusen maps derived in Section 7.4.1. We used weights according to Equation (7.1) with parameters $w_0 = 150$ and $\sigma = 30$. As will be shown in Section 7.5.2, this simple strategy already achieves much better accuracy than a state-of-the-art method. Our main technical contribution is to explore two more complex approaches, which improve accuracy further by combining the CNN with application-specific post-processing.

In our **layer based approach,** we train the U-Net to segment the RPE and BM layers, and estimate the drusen from the results. In this case, the network is set up to output four class probabilities for each pixel, i.e., background, RPE layer, BM layer, and a class that captures cases in which the two layers overlap. We used weights according to Equation (7.1) with parameters $w_0 = 100$ and $\sigma = 15$.

We recover continuous curves that represent the RPE layer and the BM via a shortest path algorithm. To this end, we convert the combined probability p^{c+o} of class c (i.e., RPE or BM) and the overlap class o, to a cost χ^{c+o} so that, in each image column, the pixel with highest probability can be traversed at zero cost. This is achieved by defining

$$\chi^{c+o} = -\log\left(\frac{p_{ij}^{c+o}}{\max(p_{:j}^{c+o})}\right) \tag{7.2}$$

where $\max(p_{:j}^{c+o})$ is the maximum combined probability in column j. Dijkstra's algorithm is used to find continuous curves that connect the left and right image boundaries with the lowest accumulated cost. To obtain smoother curves, we favor non-diagonal moves over diagonal ones by multiplying the step size to the local path cost.

Based on these layer estimates, we proceed to detect drusen via rectification, polynomial fitting and thresholding, in the same way as described in Section 7.4.1.

In our **RPE+drusen complex based approach,** the U-Net segments the area between RPE layer and BM, which has been termed RPE+drusen complex in [19], and which is shown in the center image of Figure 7.1. We simply perform a hard segmentation into RPE+drusen complex or background. We used weights according to Equation (7.1) with parameters $w_0 = 30$ and $\sigma = 15$.

In almost all cases, the largest component of the segmentation connects the left and right image boundaries, and its upper and lower boundaries can be used as RPE layer and BM, respectively. In the case of a discontinuity, which is usually due to a layer atrophy, the respective boundaries of all components are kept as candidates, and a shortest path is found that connects the left and right image boundaries by adding the lowest possible number of additional pixels. From the resulting curves, drusen are detected in the same way as in the layer based approach.

7.5 Experiments and Results

7.5.1 Cross-Validation Setup

We used five-fold cross-validation to evaluate the methods on all available data. To ensure unbiased results, all scans from the same subject were placed in the same fold. For each of the three approaches described in Section 7.4.3, this resulted in five runs with $\sim$ 70 subjects ($\sim$ 37,500 images) for training, 8 subjects ($\sim$ 4,000 images) for validation, and $\sim$ 20 subjects ($\sim$ 10,500 images) for testing. The U-Net was

Measure	Chen et al.	Direct	Layer based	Complex based
ADAD	92.33 ± 460.95	28.45 ± 74.08	**7.19 ± 36.04**	11.14 ± 52.92
OR(drusen)	20.48% ± 26.87	41.35% ± 31.3	**55.88% ± 33.85**	47.42% ± 33.45
OR(complex)	-	-	**82.6% ± 7.26**	76.74% ± 6.9
ADAD(FPE)	55.67 ± 277.97	15.08 ± 44.76	**4.92 ± 29.69**	9.58 ± 50.87
OR(FPE,drusen)	24.20% ± 31.52	47.90% ± 33.23	**64.24% ± 36.09**	58.66% ± 35.63

TABLE 7.1: All CNN-based methods achieve much better results than Chen et al. [18], the previous state of the art. Lowest absolute drusen area difference (ADAD), and highest overlap ratio (OR), have been achieved by our layer based approach. The bottom two rows show the results of after an additional false positive elimination (FPE) step.

trained end-to-end with random initial weights that were sampled from a Gaussian distribution with a standard deviation of size $\sqrt{2/N}$, with N being the number of input nodes of each neuron. On a Titan X GPU, each of these 15 experiments took about three days.

7.5.2 Quantitative Evaluation

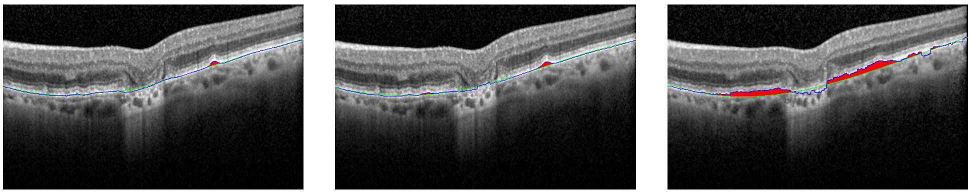
We evaluated the drusen segmentation with two quality measures that were previously established for this application [18]. The overall area covered by drusen is an important summary of drusen load, and the absolute drusen area difference (ADAD) measures the error in estimating it. Formally, $\text{ADAD}(Y^k, X^k) = |\text{Area}(Y^k) - \text{Area}(X^k)|$, where Y and X are the ground truth and the predicted segmentation, and k is the B-scan index. Following [18], the mean and standard deviation of ADAD are computed only over the columns that contain a druse.

Since the ADAD does not capture the accuracy of spatially localizing the drusen, it is complemented by the overlapping ratio (OR), which is defined as $OR(Y^k, X^k) = \frac{X^k \cap Y^k}{X^k \cup Y^k}$, i.e., the ratio of pixels with true positive drusen detections over the sum of true positives, false positives, and false negatives. Again, mean and standard deviation are computed.

In addition to the final drusen segmentation, our layer based and RPE+drusen complex based approaches produce a segmentation of the RPE layer and BM. We compare these two approaches with respect to the accuracy of this intermediate result by also computing the OR for the area between these two layers.

Table 7.1 compares the errors (ADAD) and accuracies (OR) of our three approaches over all 52, 377 images. As a baseline, it also includes results from our reimplementation of the state-of-the-art method by Chen et al. [18], which performs bilateral filtering, detection and removal of the retinal nerve fiber layer (i.e., the bright layer that can be seen at the top in Figure 7.2), thresholding based segmentation of the RPE, detection of drusen by comparing the RPE segmentation to the result of a polynomial fit, and finally false positive elimination using an *en face* projection image. The top three rows of the table are the evaluation results without the false positive elimination (FPE) step, and the two bottom rows are the results after using the FPE step with similar parameters as in [18].[2] Clearly, the FPE step has a positive overall effect on all methods.

[2] The selective *en face* projection relies on an estimate of the RPE layer, which the direct drusen segmentation does not provide. Thus, only those steps of the FPE that do not rely on the *en face* could be applied in case of the direct segmentation.



Ground truth Layer based approach Chen et al.

FIGURE 7.3: This visual comparison indicates that drusen segmentation using our approach is more robust to additional pathology such as GA than the method by Chen et al. [18].

We note that our results from the Chen et al. method are significantly worse than those reported in [18]. We believe that the main reason for this is the much lower axial resolution of our B-scans (496 pixels) compared to theirs (1024 pixels). As a result, layers that they could easily distinguish via thresholding often got merged in our data. Table 7.1 clearly shows that, at the image resolution available to us, all three CNN-based approaches were able to segment drusen much more accurately than the previous state of the art, both with or without the FPE step. Further improvements are obtained by our customized layer and complex based approaches, which use higher-level knowledge, e.g., that drusen can be recognized as deviations from a smooth normal RPE layer, which runs in parallel to the BM. The fact that the layer based approach achieved highest accuracy might, in part, be due to the fact that it makes use of continuous CNN-derived class probabilities.

Without the FPE step, all methods take less than a second per B-scan: 0.31 second for Chen's method; 0.34 second for the direct drusen segmentation; 0.57 second for the complex based and 0.81 second for the layer based approach.

7.5.3 Robustness to Additional Pathology

In more advanced stages of AMD, it is important that methods for automated drusen detection are robust to the presence of additional types of pathology, such as geographic atrophy (GA), which can be seen from the relative hyperreflectance in the choroidal layer, at the center of all images in Figure 7.3. Our training data included correctly annotated images with such pathology, so the CNN should have learned how to deal with it.

We computed the same measures as in Table 7.1 for a subset of 49 OCT scans that contained GA. The results were similar, with or without the FPE step. For the layer based approach, results were ADAD $= 2.84 \pm 8.28$, OR $= 49.87\% \pm 28.08$, and for the complex based approach, ADAD $= 5.66 \pm 19.74$ and $40.94\% \pm 24.76$. The fact that these numbers are similar to those in the overall dataset illustrates the robustness of these approaches in the presence of GA.

7.5.4 3D Visualization of Results

As an additional visual check of the segmentation quality, Figure 7.4 shows summaries of the full 3D OCT data in an eye for which a high-resolution scan was available. Chen et al. [18] propose a selective volume projection that reduces such data to a 2D *en face* image in which the drusen stand out as bright spots. Figure 7.4 (left) shows such an image with the ground truth and CNN-derived drusen segmentations overlaid as colored curves. It can be seen that they strongly overlap.

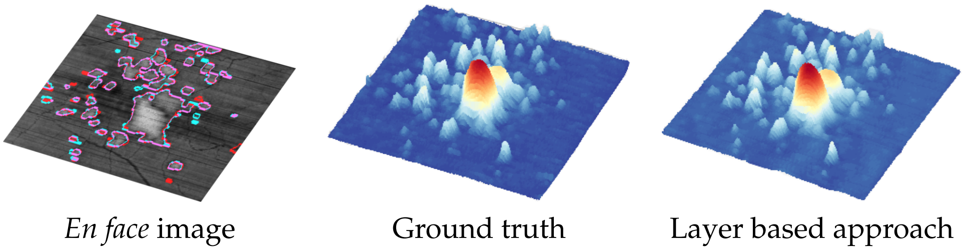


FIGURE 7.4: On the *en face* image, red curves delineate the ground truth drusen segmentations, blue curves our segmentation, purple curves regions where the two overlap. Surface renderings illustrate that the RPE layer has been reliably detected.

The center and right subfigures visualize the ground truth and segmented RPE layers in the same eye as 3D surfaces, which have been rectified based on the BM for better visualization. A high degree of continuity between adjacent B-scans is observed despite the fact that we process each B-scan independently.

7.6 Conclusion

We present the first three CNN-based approaches for a fully automated segmentation of drusen in OCT images, which is an important task for diagnosing age-related macular degeneration and modeling disease progression. On the 52, 377 annotated images available to us, all three approaches produced much better results than a state-of-the-art method. Best results were achieved by combining the U-Net architecture with application-specific post-processing.

After training, segmentations are obtained fast enough for use in epidemiological studies, and they have been shown to be robust to the additional presence of geographic atrophy. In future work, we would like to investigate how to best account for three-dimensional context that has a highly variable resolution.

Chapter 8

Intelligent Interaction and Uncertainty Visualization for Efficient Drusen and Retinal Layer Segmentation in Optical Coherence Tomography[1]

8.1 Abstract

Convolutional neural networks (CNNs) represent the state of the art for fully automated medical image segmentation. However, few works have combined CNNs with interactive user feedback in order to verify and, where necessary, correct their results. We present an interactive visual system that achieves this for the specific use case of segmenting drusen, which serve as a biomarker of age related macular degeneration, from Optical Coherence Tomography. Our main idea is to exploit the probabilistic nature of CNN-based segmentation. First, we derive two uncertainty measures from it. We demonstrate that they indicate cases in which automated segmentation is likely to have failed, and that visualizing them makes manual verification and correction more efficient. Second, based on the probabilistic information, we design intelligent tools for segmentation correction, which automatically propose the most likely alternative segmentation in agreement with user-specified constraints. In a small user study, uncertainty visualization and intelligent interaction reduced the time required to correct retinal layer segmentation by around 53% and, for drusen segmentation, even by 73%. In the future, we plan to use our system not only for efficient segmentation correction, but also for rapid creation of larger training sets.

8.2 Introduction

For most automated image segmentation tasks in computer vision [57] and medical image analysis [102], current state-of-the-art techniques are based on convolutional neural networks (CNNs[2]). In this work, we describe the design, implementation,

[1] The content of this chapter has been previously published: Gorgi Zadeh et al. "Intelligent interaction and uncertainty visualization for efficient drusen and retinal layer segmentation in Optical Coherence Tomography" *Computers & Graphics* 83 (2019): 51-61. URL: `https://doi.org/10.1016/j.cag.2019.07.001` [205].

[2] Table 8.1 lists all acronyms used in this article.

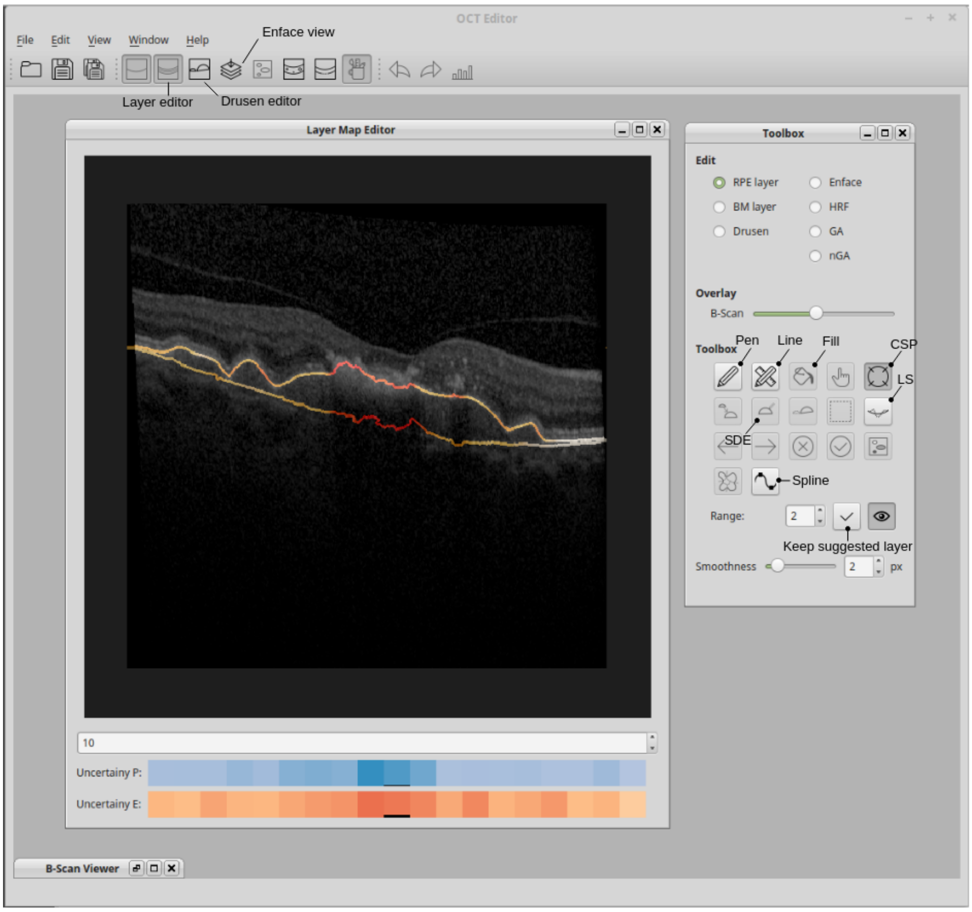


FIGURE 8.1: This view of our system illustrates two of our main contributions: The layer segmentation editor (larger subwindow) uses the proposed uncertainty visualization (shades of red) both on the RPE (top) and BM (bottom) layers. The heatmap below visualizes probability (P) and entropy (E) based uncertainty measures, aggregated over two-dimensional B-scans; it can also be used for navigation. The toolbox on the right includes our intelligent interaction tools, including constrained shortest path (CSP), local smoothing (LS), and semi-automated drusen extraction (SDE). For comparison, our system also implements traditional tools, such as pen, line, flood fill, or spline based editing.

and evaluation of a system for fast and efficient proof-reading and editing of CNN-based segmentations. Few such systems currently exist, but we believe that they are relevant for two main reasons:

First, even though CNNs substantially advanced the previous state of the art, they can rarely be relied upon "blindly" in cases where crucial diagnostic or therapeutic decisions should be taken based on them. In particular, the accuracy of automated segmentations can be highly influenced by the presence of other pathologies, especially if they have not been part of the training data. Even when deriving reliable image-based biomarkers for statistical evaluation in scientific or clinical studies, it is important to ensure that the underlying segmentations are correct.

Second, for achieving high accuracy and robustness, a diverse and large labeled data set is required. Especially for high-resolution three-dimensional images, creating such training data fully manually can be a prohibitive effort. To reduce it, one might consider an iterative approach in which an initial CNN is trained with a

AMD	Age Related Macular Degenration
BM	Bruch's Membrane
CSP	Constrained Shortest Path
CNN	Convolutional Neural Network
LS	Local Smoothing
OCT	Optical Coherence Tomography
RPE	Retinal Pigment Epithelium
SDE	Semi-automated Drusen Extraction

TABLE 8.1: List of all acronyms used in this paper.

dataset of limited size, and is applied to a new, still unlabeled set of data. The resulting segmentations should be corrected by a human rater, and can then be used to further refine the CNN. This cycle can be repeated for a few times, until a well-trained CNN is achieved. This strategy should be more efficient than annotating all images from scratch if suitable tools for correcting CNN-based segmentations are available.

The use case of our system is segmentation of Optical Coherence Tomography (OCT), which provides three-dimensional images of the macula as a sequence of 2D cross-sectional scans, called B-scans. Figure 8.1 shows the 10th B-scan of an OCT volume with 19 B-scans overall. The retinal layers are visible in OCT, and several biomarkers can be derived from their segmentation [192]. Among them, we focus on drusen, which evolve as deposits of extracellular debris under the retinal pigment epithelium (RPE) layer, the upper of the two layers in Figure 8.1. In a normal eye, the RPE should run parallel to the Bruch's membrane (BM), shown as the lower of the two layers. This work focuses on segmenting the RPE and BM layers, and from them drusen, as abnormal deviations between the two. For this, we provide a semi-automated system that implements the following novel features:

1. We derive uncertainty measures from the probabilistic CNN output and visualize them to quickly guide the user towards remaining errors in a CNN-based segmentation of RPE and BM. This is described in Section 8.5.

2. We introduce intelligent interaction tools that allow for efficient correction of layer segmentation results, again based on the probabilistic CNN output. In particular, our constrained shortest path tool forces the layer segmentation to pass through a point clicked by the user, and automatically provides suggestions for improved segmentations in neighboring 2D slices (B-scans). These tools are described in Section 8.6.

3. We introduce a semi-automated tool for segmenting drusen based on layer segmentations of RPE and BM. This is described in Section 8.7.

The remainder of our paper reviews similar systems (Section 8.3), provides the background that is required to make our paper self-contained (Section 8.4), and reports experiments that were done to evaluate our system (Section 8.8). In particular, measuring the time that was required by test users to correct segmentations with uncertainty visualization and intelligent interaction, compared to finishing the same task with standard tools, indicates a speedup of more than a factor of two.

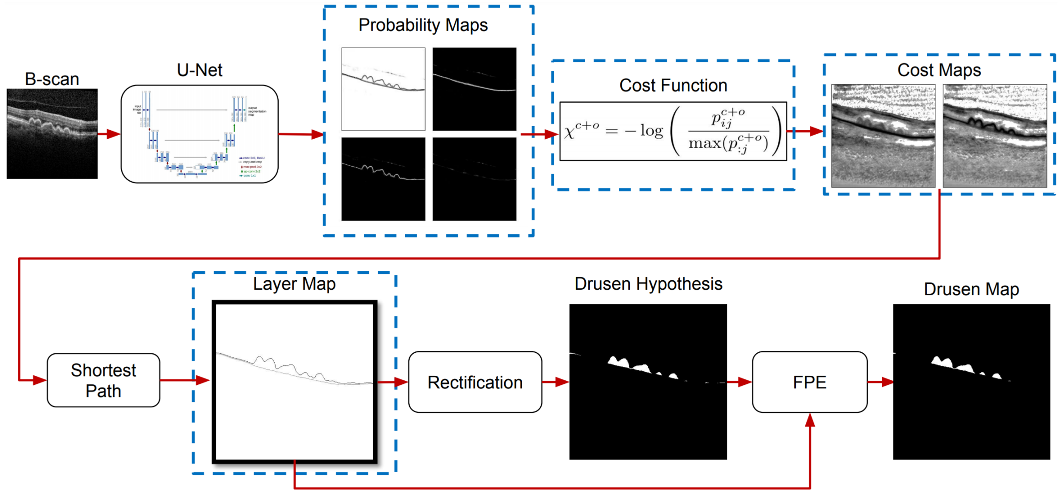


FIGURE 8.2: In this overview of the original segmentation pipeline, the parts that had to be modified for our guided semi-automated editing framework are highlighted with dashed blue boxes. The individual stages are explained in Section 8.4.

8.3 Related Work

This work describes a significantly extended version of a system that we presented previously at a workshop [50]. It now includes multiple novel methods: In particular, we exploit the 3D nature of retinal layers to transfer the user's feedback into adjacent B-scans, and to make automated suggestions for more plausible segmentations. Moreover, we added a novel overview visualization of the uncertainty, as well as a semi-automated tool for drusen segmentation, a task that was not supported at all in the earlier version. The user study reported in Section 8.8.1 has been extended accordingly, and has been made more realistic by including another basic tool, spline-based layer editing. It confirms the usefulness of all newly introduced methods.

Semi-automated methods for medical image segmentation have a long history [124], and several recent works specifically addressed segmentation correction [58, 183]. According to the overview given by Heckel et al. [64], they follow three main strategies: Making parameter tuning more convenient, integrating additional information about the desired result into an automated segmentation ("refinement"), or correcting segmentations manually, independently from the algorithm that first generated them. Our intelligent methods belong to the second and third of these categories: Our constrained shortest path tool (Section 8.6.1) allows the user to specify points that should lie on the final layer segmentation. This is a classic and widely used type of interaction [121], which we adapt to work with our specific segmentation method, and extend to also affect neighboring B-scans. Suggesting improved segmentations of Bruch's membrane (Section 8.6.2) uses some internal information from the segmentation algorithm, while our local smoothing method (Section 8.6.3) and Semi-automated Drusen Extraction (Section 8.7.1) work independently from the underlying segmentation method.

Most existing software for visualizing and analyzing data from Optical Coherence Tomography is commercial, and limited information is available on the underlying algorithms. Stratus OCT (Carl Zeiss Meditec, Inc. Dublin, CA) can be used for measuring retinal nerve fiber layer thickness [14]. Other software includes

Spectralis HRA+OCT (Heidelberg Engineering, Inc., Heidelberg, Germany), Spectral OCT/SLO (Opko/OTI, Inc., Miami, FL), SOCT Copernicus (Reichert/Optopol Technology, Inc., Depew, NY), RTVue-100 (Optovue Corp., Fremont, CA), and Cirrus HD-OCT (Carl Zeiss Meditec, Inc.). They can all be used for macular thickness measuring and are compared with each other in [195]. To the best of our knowledge, none of these systems currently offer CNN-based segmentations or specific mechanisms for highlighting potential segmentation errors.

An alternative open-source system for semi-automated OCT segmentation is OCTSEG [117]. However, its segmentation is based on edge detection and minimization of a custom energy functional, not on CNNs, the system does not use any uncertainty visualization, and manual corrections are made completely freehand, as opposed to our intelligent interactions.

When discussing uncertainty visualization, we will distinguish between segmentation errors, which can be quantified by comparing a given segmentation to a reference ("ground truth"), and segmentation uncertainty, which has to be inferred without such a reference.

Examples of visualizing segmentation errors include work by Cárdenes et al. [13], who visualize segmentation quality with newly derived similarity measures. Von Landesberger et al. visualize how statistical shape model based segmentations converge to an expert segmentation [91] and present a system for comparative evaluation across larger datasets [92]. Geurts et al. [47] visually compare segmentation quality to identify the most suitable automated algorithms. Raidou et al. [137] developed a system for visually exploring segmentation errors both in individuals and in whole cohorts.

In contrast to these approaches, our goal is to guide the user towards images where corrections are most likely to be required, even in cases where no ground truth is available. In this sense, it is similar in spirit to the uncertainty-aware guided volume segmentation of Praßni et al. [135], or to work by Summa et al. [178] on quantifying uncertainty in 2D image segmentations. However, they do not directly apply to our case, since they are specific to different algorithms. Learning shape and appearance priors is another strategy to detect potential segmentation errors [151], but cannot easily be applied to segmenting retinal layers in the presence of pathological anomalies such as drusen.

Since we build on a CNN-based segmentation, we note that numerous recent works have used visualization to support better understanding of neural networks [38, 77, 114, 140, 176, 182, 197, 200, 208], to monitor their training [103], and to facilitate tuning of hyperparameters such as the number of neurons and layers [104, 130, 145]. However, none of them are concerned with using intermediate results to support correcting individual outcomes, which is our main focus.

Entropy is a natural building block for uncertainty measures, and will be used in Section 8.5.1. Related measures were used for visualizing segmentation uncertainty by Potter et al. [134], and another closely related measure was used by Al-Taie et al. [3]. We will adapt this measure to our needs, combine it with a second uncertainty measure, and validate it in the context of our application.

8.4 Background on Application and Underlying Segmentation Technique

On the application side, it is our goal to use OCT imaging for the epidemiological study of age related macular degeneration (AMD), which is known to be the leading cause for irreversible blindness in most of developed countries [1, 69]. Segmenting and measuring disease biomarkers in hundreds of OCT scans can help us to better understand its causes, and to find effective treatments. Drusen are one of the early appearing and important biomarkers, since their size, number, and location contribute to AMD staging and tracking of disease progression. Drusen and retinal layer segmentation are widely studied problems in the field of medical image analysis [192]. With the recent popularity of convolutional neural networks for image segmentation, various groups have started using deep neural networks for this application [6, 39, 51, 60]. Our system builds on our own prior work [51]. We previously demonstrated that it allows for a more accurate segmentation of drusen compared to a traditional state-of-the-art method [18].

Figure 8.2 illustrates the underlying computational pipeline for drusen segmentation. It finds the drusen by first segmenting the RPE and BM layers. In this pipeline, the convolutional neural network used for layer segmentation is a U-Net [146], which consists of two symmetric paths, a contracting one that is used to capture image context, and an expanding one to recover the original resolution. The network is set up to transform an input B-scan into four probability maps, which indicate the probability of a pixel belonging to the background (top left in Figure 8.2), RPE layer (bottom left), BM layer (top right), or to a special case in which the two layers overlap (bottom right). These probability values are obtained by first computing a score $s_{ij}^{(l)}$ for each label l and pixel (i, j), and then converting it into non-negative and normalized probabilities $p_{ij}^{(l)}$ via the softmax:

$$p_{ij}^{(l)} = \frac{\exp s_{ij}^{(l)}}{\sum_{l'} \exp s_{ij}^{(l')}} \tag{8.1}$$

These maps are the basis for the uncertainty visualization discussed in Section 8.5.

Next, our pipeline extracts a curve representation of the RPE and BM layers from the probability maps. To this end, the probability maps are first converted into cost maps using a cost function defined in [51]. In Figure 8.2, the cost map for the BM is shown on the left, for the RPE on the right. Layer representations are computed using Dijkstra's algorithm [167], as paths that connect the left and right image boundaries with minimum accumulated cost of the corresponding cost map.

The final steps detect drusen based on the layer segmentations. For this, the rectification step vertically shifts each column so that the BM layer forms a straight horizontal line. An ideal (drusen-free) RPE is then estimated by fitting a third degree polynomial to the rectified RPE. Finally, the area that lies below the RPE, but above the estimated ideal RPE is taken to represent drusen and everything is warped back into original image coordinates. In a final false positive elimination (FPE) step, falsely detected drusen with a height of 2 pixels or less are removed from the segmentation [51].

8.5 Visualizing Segmentation Uncertainty

The data that was available to train the CNN mostly contained OCTs with intermediate AMD. Therefore we found that the automated pipeline fails to create a plausible segmentation for OCTs that include advanced stages of AMD, and in the presence of additional pathological changes. Our system guides the user to such problems by estimating and visualizing segmentation uncertainty based on the network's probability maps.

8.5.1 Uncertainty Estimation

As in [50], two measures are used to estimate segmentation uncertainty. One is based on entropy (u_e), the other one on the probability (u_p).

For the entropy-based measure, we consider individual columns of the B-scan, which are called A-scans (axial depth scans). Ideally, the layer probabilities in an A-scan would be concentrated in a single pixel, indicating that the network has been certain regarding the exact location of the corresponding layer within that column. Entropy is a natural measure to quantify the degree to which probabilities are concentrated, with a high entropy indicating high uncertainty with respect to layer localization.

To compute per-column entropy, we first need to compute a column-normalized probability $q_{ij}^{(l)}$ at each pixel position (i, j) and for each label $l \in \{\text{RPE}, \text{BM}\}$. Given the raw values $p_{ij}^{(l)}$ from the probability maps, it is obtained as

$$q_{ij}^{(l)} = \frac{p_{ij}^{(l)}}{\sum_i p_{ij}^{(l)}}. \tag{8.2}$$

Using $q_{ij}^{(l)}$, our normalized entropy-based uncertainty measure $u_e^{(l,j)} \in [0, 1]$ is then defined as

$$u_e^{(l,j)} = 1 - G_\sigma * \exp\left(\sum_i q_{ij}^{(l)} \ln q_{ij}^{(l)}\right), \tag{8.3}$$

where the superscript (l, j) indicates column j of retinal layer l. Later on, per-layer/per-column uncertainties are aggregated into a per-B-scan uncertainty. For this, a small amount of regularization via Gaussian smoothing G_σ across neighboring columns becomes important to add robustness, so that isolated columns of high entropy do not receive an excessive weight. In all our experiments we set $\sigma = 2$, which corresponds to a Gaussian window of size 17 pixels. Over the boundaries, the values are reflected for the convolution operation.

Even if uncertainty is increased in a relatively small area, and in one of the layers, we would like to bring this to the expert's attention. Therefore, per-layer/per-column uncertainties are aggregated into a per-B-scan uncertainty u_e by taking the maximum over all $u_e^{(l,j)}$, i.e.,

$$u_e = \max_{l,j}\left(u_e^{(l,j)}\right). \tag{8.4}$$

The second uncertainty measure u_p is computed based on the exact value of the probability at each point where the final segmented layer is passing through, which indicates the network's confidence that the retinal layer exists at that point, i.e.,

$$u_p^{(l,j)} = 1 - G_\sigma * p_j^{(l)}, \tag{8.5}$$

where $p_j^{(l)} = p_{ij}^{(l)}$ are directly taken from the probability map of layer l, at the row index i where the shortest path traversed column j. Again, regularization via

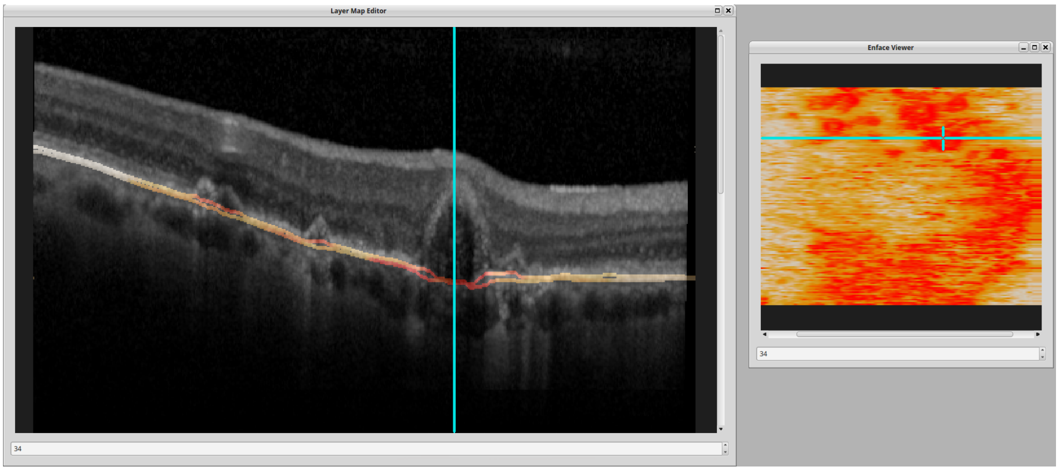


FIGURE 8.3: On the right, uncertainties are shown on an en-face image, which is generated by projecting the OCT volume along A-scans; one such A-scan is shown as a vertical line on the left. In this example, the entropy based uncertainty u_e is shown for the RPE, but other uncertainty measures or layers can be chosen.

Gaussian smoothing is used to add robustness, and per-layer/per-column values of $u_p^{(l,j)}$ are aggregated into a global u_p by taking the maximum of $u_p^{(l,j)}$ over l and j.

8.5.2 Visual Encoding of Uncertainty

The user can choose between uncertainty measures u_e or u_p for visualization. Figure 8.1 shows an example where the user has chosen u_e. We have assigned different color maps for u_e (orange/red) and u_p (blue) to clarify which measure is currently shown. The per-column uncertainty is color coded over each retinal layer. More saturated colors indicate stronger uncertainties to make them stand out from the grayscale background.

In the two navigation heatmaps below the subwindow, the ith cell corresponds to the ith B-scan. They visualize the per-B-scan uncertainties with respect to each measure. Cells with more saturated colors are more likely to require the user's attention for segmentation correction. By clicking on each cell, the corresponding B-scan is shown to the user in the subwindow.

We have decided to visualize both u_e and u_p because of occasional cases in which one was more sensitive to segmentation errors than the other [50]. In addition, in the evaluation (Section 8.8) it is observed that ranking B-scans with respect to both measures correlates better with a manual and error-based ranking, compared to the case where only one of the uncertainty measures is used.

In addition to the navigation heatmap, the user can visualize the uncertainty measures over the en-face projection image as shown on the right in Figure 8.3. The en-face view is a standard view in OCT that corresponds to a 2D projection of the volume along axial lines, resulting in a frontal view of the retina. En-face images are computed using selective summed-voxel projection [18], which first extracts a narrow sub-volume between two parallel curves that contain the RPE. In practice, we use the BM layer as the bottom curve; the top curve is determined by the tallest druse in all B-scans. To enhance the visibility of drusen in the en-face projection, before summing up voxel intensities of selected sub-volumes in axial direction, the

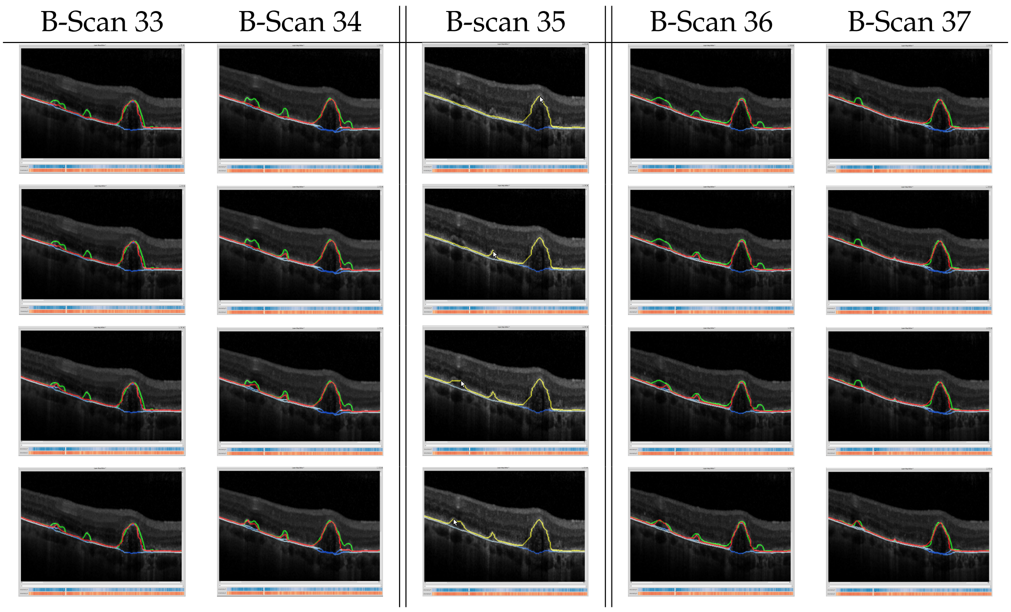


FIGURE 8.4: Illustration of the workflow for adding missed drusen with the constrained shortest path tool; each row shows the updated result after one click. This action not only changes the layer segmentation in the current B-scan (#35), but also automatically updates RPE suggestions in the neighborhood. Red is the suggested RPE segmentation; green is the gold standard RPE; blue is the CNN based RPE segmentation. Here, the gold standard is visualized only to highlight the accuracy of RPE suggestions; in real-world uses, it is of course unavailable.

intensity of voxels between RPE and BM are replaced by the maximum intensity in that column.

Visualizing layer uncertainties on the en-face image gives the user a complete overview of segmentation errors within the volume. Clicking on a point in the en-face viewer automatically takes the user to the corresponding B-scan. Within it, the corresponding A-scan is marked with a vertical line in the editor.

8.6 Methods for Layer Segmentation Editing

Our system provides three intelligent interaction methods for editing layer segmentations: Constrained shortest path, automated proposal of alternative segmentations based on 3D context, and local smoothing. In addition, our system implements all standard editing tools.

8.6.1 Constrained Shortest Path Method

Figure 8.4 shows multiple examples in which the shortest path algorithm failed to follow drusen in the RPE layer. This type of error is very common, and correcting it freehand is a substantial effort. Therefore, we designed a semi-automated method that facilitates correcting such errors by clicking on a single point that the desired layer segmentation must pass through.

To achieve this effect, when the user selects a point using the "constrained shortest path (CSP)" tool, the cost map of the B-scan, as illustrated in Figure 8.2, is modified based on the user's input. In particular, the cost is set to zero at the pixel the

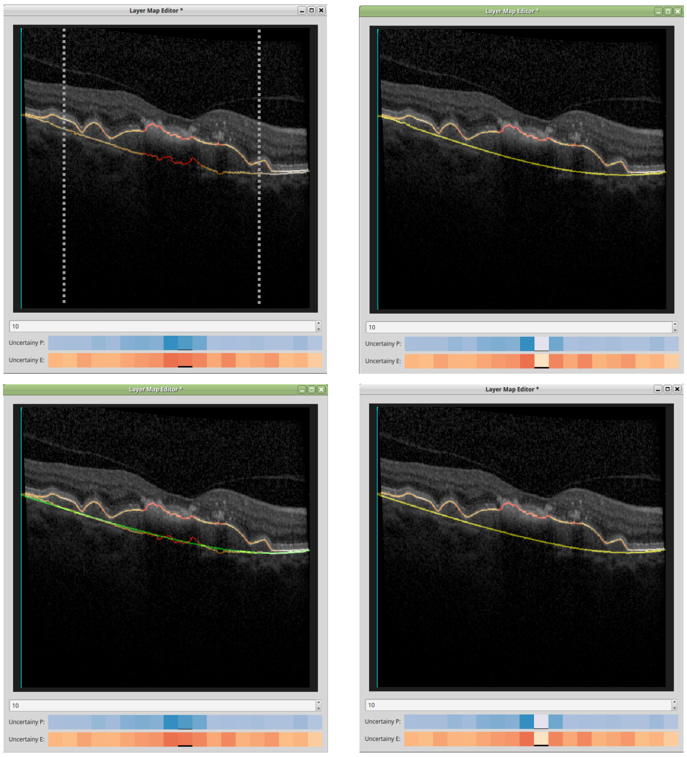


FIGURE 8.5: The first row shows layer correction with our local smoothing method: In the examples on the left, a lack of image contrast caused a noisy estimate of Bruch's membrane. On the right, this has been fixed using polynomial fitting in the region specified by the gray vertical lines by the user. In addition, the uncertainty color codes in the navigation heatmaps are updated. The second row shows the BM layer suggestion using 3D context. The user can completely replace the old layer (in red) with the newly suggested (in green).

user clicks on, and the cost of every other pixel in the same image column is set to a large number. When repeating the shortest path extraction, this modification forces the new layer segmentation to run through the selected point.

Using exactly the same cost function as in our prior work [51] within the CSP method sometimes led to implausible spiky results. We found that this is partly caused by the fact that in the original cost function, the layer probabilities at each point are divided by the maximum probability within the same column, which sometimes made vertical moves within the cost map much cheaper than horizontal or diagonal ones. To solve this, we made two modifications: first, we replaced the maximum over the column with the maximum over the full image,

$$\chi^{(l)} = -\log\left(\frac{p_{ij}^{(l+o)}}{\max p_{::}^{(l+o)}}\right), \tag{8.6}$$

where p_{ij} is the probability at position (i, j), $\chi^{(l)}$ is the cost image for layer l, and $\max(p_{::}^{(l+o)})$ is the maximum, taken over the full B-scan, of the combined probability of l and a special class o that is used to encode layer overlaps. Class o is defined separately, since the two layers might overlap in some pathological cases. Second, we allow the user to modify the weight of the cost of making vertical moves compared to horizontal ones in the software.

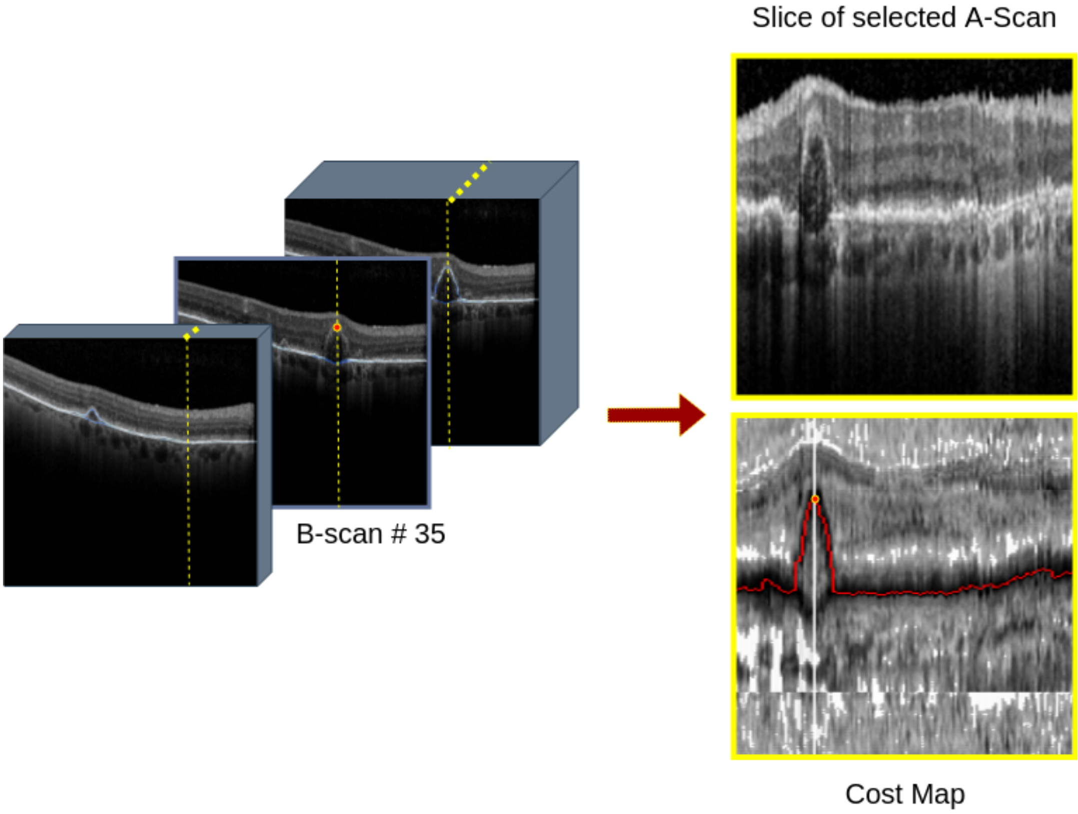


FIGURE 8.6: When the CSP tool is used in one B-scan, a shortest path in an orthogonal slice is extracted to create RPE suggestions for neighboring B-scans. The second column shows the shortest path found in the cost map of that slice for the point selected in B-scan #35.

To keep track of the changes made to the layer segmentations, we render modified layers in yellow. On the en-face projection of edited layers, a small yellow marker is added near the corresponding B-scan. In addition, as can be seen in Figure 8.5, the uncertainty values in the navigation heatmaps below the viewer are updated by setting them to the color code for absolutely certain.

8.6.2 Segmentation Suggestion Using 3D Context

Suggestions for RPE Layer

If a druse is hard to detect and is missed by the layer segmentation pipeline, it is highly likely to be missed in the neighboring B-scans as well, at least when B-scans are sampled densely enough. In such scenarios, we use the information provided by the user for correcting the RPE layer in current B-scan to suggest a segmentation in the adjacent B-scans. If the user finds the suggested RPE layer plausible, he or she can replace the existing segmentation by the newly suggested one.

The implementation of this is illustrated in Figure 8.6: When the CSP method forces the RPE layer to follow a druse at the selected point in a B-scan, in order to provide plausible segmentation suggestions in the neighborhood, another shortest path is computed in an orthogonal slice of the OCT volume (A-scan slice, shown in yellow).

The second column of Figure 8.6 shows the A-scan slice for the selected point in a sample B-scan. Using the shortest path in the A-scan slice, we can estimate where the RPE layer must go through in neighboring B-scans. Figure 8.6 shows the suggested RPE layers for 4 B-scans in the vicinity of 35th B-scan in the given OCT volume in red.

The spacing between B-scans is typically much larger than the in-plane resolution of B-scans. We found that, for this reason, the steep vertical moves permitted by the original cost function from [51] and discussed in Section 8.6.1 are actually beneficial here. Therefore, we do not use the modifications from Equation 8.6 in this case.

Suggestions for BM Layer

To provide suggestions for the BM layer, a 3D cubic B-spline is used to fit a surface to the original BM segmentation throughout a 3D neighborhood. In Figure 8.5 (bottom row), the green curve shows where the estimated surface passes through the current B-scan. This results in the suggested BM segmentation. In order to reduce the effect of erroneous segmentations on the estimation, segmentation uncertainties are used as weights during surface fitting, i.e., original segmentation points with higher uncertainty will have less effect in the fitting process. The user can choose between u_e or u_p as weights, and tune the size of the neighborhood used for estimation.

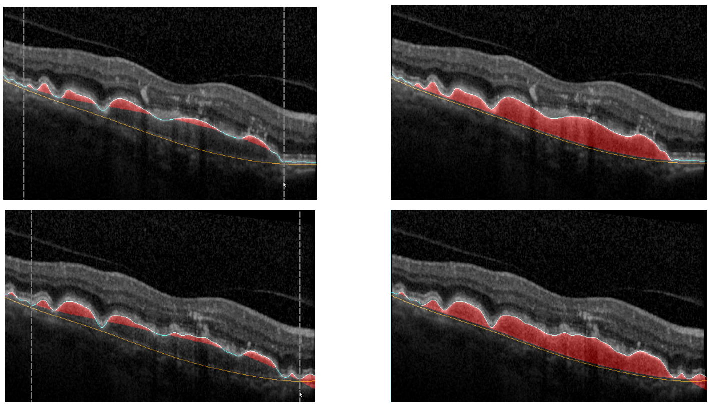
8.6.3 Local Smoothing

The local smoothing (LS) tool is introduced to correct automated layer segmentations that are in the correct location overall, but too “wiggly”. Usually, such segmentation errors happen when the B-scan is noisy or has weak contrast. For correction, the LS tool allows the user to pick an interval over an specified layer. The original layer segmentation in that interval is then replaced with a smoothed version, which is obtained by fitting a low-degree polynomial to the pixel positions of the original segmentation. The amount of smoothing can be controlled by the user by changing the polynomial order. The left image in the top row of Figure 8.5 shows a case where the user has selected an interval over the Bruch’s membrane. On the right, the resulting corrected layer segmentation is shown.

8.6.4 Basic Tools

Our system implements three basic tools, to be used as a baseline in our evaluation. The first one is a “pen” that marks or unmarks individual pixels as belonging to the selected layer. The second one is a “line” tool, which marks or unmarks every pixel along a line segment drawn by the user. To speed up correction, in both cases, all previous segmentations are automatically erased from the edited image columns. This is based on the assumption that the layers run from left to right without looping back, and agrees with the data format in which ground truth segmentations were given to us.

The third basic tool is spline-based editing, which fits a layer by a cubic B-spline. B-spline knots are visualized over the layer. The user can then drag, delete or add knots in order to reshape the retinal layer. We keep the number of knots of the initial B-spline small by permitting a certain amount of smoothing, as long as the sum of squared residuals with respect to the original layer is smaller than a given threshold. In our experiments, we set this threshold to 50.



Selecting drusen extent Resulting drusen segmentation

FIGURE 8.7: After selecting a region in which drusen occur (first column), our semi-automated drusen extraction tool automatically annotates them (second column).

8.7 Drusen Segmentation

Given the RPE and BM layers, drusen can be extracted automatically as described in Section 8.4. However, that approach only works well if drusen are not too wide. The first column of Figure 8.7 shows two cases where automated drusen segmentation has failed due to wide drusen. We again provide a specific tool for correcting these cases.

The drusen editing view not only overlays the drusen segmentation over the B-scans, but also allows the layer segmentation to be shown over the drusen segmentation (as in Figure 8.7). This works as an extra guide for the user on where the border of drusen must be. The user is provided with two separate sliders for setting the opacity of drusen or layer segmentations.

8.7.1 Semi-automated Drusen Extraction

Our semi-automated drusen extraction (SDE) tool annotates drusen based on simply marking their start and end points on a B-scan, as illustrated in Figure 8.7. It is based on the fact that the ideal RPE, which coincides with the lower drusen boundary, runs in parallel to the BM layer. Therefore, it can be extracted using the distance between RPE and BM layers at the beginning (d_b) and at the end (d_e) of the druse. In between, it is interpolated as

$$y_j^{idealRPE} = y_j^{BM} + \alpha d_e + (1-\alpha)d_s \,, \tag{8.7}$$

where α ranges from 0 to 1 from beginning to the end of the druse, $y_j^{idealRPE}$ is the location of the ideal RPE at each column j of the image, and y_j^{BM} is the location of the BM layer at each point. Then, the area between the lower drusen boundary and the RPE layer is automatically marked as druse. Images in the second column of Figure 8.7 show the drusen that are automatically segmented using this method.

8.7.2 Basic Tools

The first and second basic tools for correcting drusen segmentations are a pen and line tool that allow marking individual pixels, or pixels along a line, as belonging to drusen. The third tool is a bucket fill that marks all pixels in a closed area by a single

	Without guidance		With guidance
	Only basic	All	All
Time per B-scan (Sec)	11.13	8.13	5.77
Total time (Sec)	1614	1179	836
Total actions (clicks)	423	279	180
Pen	16.8%	0.5%	0%
Line	1.8%	0%	0%
Spline fitting	79.4%	14%	20.3%
CSP	-	17.6%	15.9%
LS	-	0%	0%
Keep Suggested RPE	-	41.5%	55.1%
Keep Suggested BM	-	21.2%	5.8%
Undo/Redo	2%	5.2%	2.9%
	User A		

	Without guidance		With guidance
	Only basic	All	All
Time per B-scan (Sec)	10.31	6.4	4.43
Total time (Sec)	1495	928	643
Total actions (clicks)	374	151	113
Pen	0%	0%	0%
Line	0%	0%	0%
Spline fitting	99.7%	3.1%	11.8%
CSP	-	8.3%	30.3%
LS	-	0.1%	0%
Keep Suggested RPE	-	69.8%	56.6%
Keep Suggested BM	-	17.7%	1.3%
Undo/Redo	0.3%	1%	0%
	User B		

TABLE 8.2: Each of two test users corrected OCT scans with basic and all editing tools, and with and without uncertainty guidance. The first row shows the corresponding average time needed to correct segmentations in one B-scan. The second row shows the time spent on correcting one full OCT. The third row shows the number of total actions (clicks) per experiment, and the remaining rows reveal the percentages of using the different tools in each experiment. To ensure that remembering the required modifications is not the main factor behind the observed efficiency gain, the following order of experiments is used 1) using guidance with all tools 2) without guidance, but with all tools 3) without guidance and with only basic tools.

click. For simplicity, the bucket fill tool only marks pixels up to the RPE layer. This feature allows the user to only draw the lower boundary of a druse and quickly fill in the area between the lower boundary and RPE. The filling tool can also be used to remove a false segmentation.

8.8 Evaluation

The major goal of our system is to make correction of layer segmentations and drusen more efficient. To evaluate the extent to which our prototype meets this goal, we set up multiple experiments in which users were asked to correct the automated segmentation of OCT scans which had been identified as failure cases of the automated method. The experimental setup is described and results are reported and discussed in Section 8.8.1.

In a separate experiment, we investigated the validity of our proposed uncertainty measures. These results are reported and discussed in Section 8.8.2.

8.8.1 Study of the Achieved Efficiency Gain

In order to quantify the benefit from our intelligent interaction tools and uncertainty visualization, segmentations were corrected repeatedly. In different runs, either all features of our software were available to users, or the features whose benefit we wanted to evaluate were excluded. In addition, our software was modified to measure the time needed to complete the task, and to record the number of times each of the available tools was used during the process.

In real-world uses of our framework, the human expert will continue to make corrections until he or she is satisfied with the result. However, in our study, letting the users decide when to stop would have introduced the possibility that faster task completion might go along with achieving lower accuracy. Therefore, we modified the software so that it would force the user to continue until a predefined level of accuracy was reached. Accuracy was computed with respect to a ground truth reference which had been created previously by an independent expert, using independent tools. Neither the reference nor the current measure of accuracy were visible to the user. We used a pilot experiment to determine a specific tolerance level that we could expect our users to achieve within a reasonable time without revealing the ground truth to them.

In order to ensure that differences in completion times are not related to differences in task difficulty, we asked users to repeatedly correct the same dataset. In our previous work [50], we observed that this introduced a learning effect, where users became faster in subsequent runs because they remembered the required corrections. Our current experiments were set up so that users could initially use all available tools, and were limited to more basic tools in subsequent runs. This way, we ensure that the learning effect can only lead us to underestimate the benefit from our system.

To avoid further confounding effects, results in our current work are from different users than in our previous workshop paper [50]. Both current users were previously unfamiliar with our system, and were given about 10 minutes of training on how to use our tools.

	Only basic	All	Only basic	All
Time per B-scan (Sec)	53.5	14.9	44.9	12.3
Total time (Sec)	535	149	449	123
Total actions (clicks)	150	24	146	23
Pen	0.1%	0%	0%	0%
Line	69.9%	0%	55%	0%
Fill	27.6%	0%	35.8%	0%
SDE	-	100%	-	100%
Undo/Redo	2.4%	0%	9.2%	0%
	User A		User B	

TABLE 8.3: Each of two test users corrected 10 B-scans in an OCT volume with basic and all drusen editing tools. The first row shows the corresponding average time needed to correct segmentations in one B-scan. The second row shows the time spent on correcting all 10 B-scans. The third row shows the number of total actions (clicks) per experiment, and the remaining rows reveal the percentages of using the different tools in each experiment. To ensure that remembering the required modifications is not the main factor behind the observed efficiency gain, the following order of experiments is used: 1) with all tools 2) with basic tools only.

Layer Editing Tools

For the layer correction task, we wanted to investigate the benefits from our intelligent interaction tools and the uncertainty visualization separately. Therefore, each segmentation was corrected three times. In the first round, users were allowed to use all of the available tools including the basic tools (pen marker, line marker, spline) and more advanced ones (CSP, LS, suggestions for RPE and BM). They could navigate through the data based on the per-B-scan uncertainty estimates, and local uncertainties were color coded on the layers, as discussed in Section 8.5.2. In the second round, all editing tools were provided, but without any guidance based on the uncertainty measures. In the final round, users were only allowed to use the basic tools, and were not shown any uncertainty information.

This experiment used an OCT scan consisting of 145 B-scans. A ground truth layer segmentation was created by an independent expert, using the OCT manufacturer's software. The target accuracy was considered to be achieved when the average segmentation error decreased to 5 pixels for the whole OCT scan, and when there was no B-scan left that included any segmentation errors greater than 12 pixels.

Table 8.2 summarizes the results of these experiments. Both users clearly benefited from our novel editing tools. They reduced the time required for the corrections by 27% (user A) and 38% (user B), respectively. This went along with a clear reduction in the number of actions (clicks). Usage statistics suggest that the speedup was mostly enabled by the constrained shortest path tool, and the associated tool for suggesting RPE corrections in adjacent B-scans. Suggestions for Bruch's membrane, which have been added compared to the previously presented version of our system [50], have largely replaced usage of the local smoothing tool, which confirms effectiveness of this new tool. Comparing users, A took more than 25% longer than B. This went along with a reduced usage of our advanced tools, and a more frequent use of undo and redo.

For both users, guidance from the uncertainty visualization gave an additional speedup, leading to an overall reduction by 49% (user A) and 57% (user B) of the time taken with basic tools only. We note that, even though this result demonstrates a clear benefit from our system, the speedup reported in our previous work [50] was even greater. Beside the above-mentioned learning effect, this is primarily due to introducing the spline-based "basic" tool, which both users relied on heavily when our intelligent tools became unavailable, and which substantially decreased the time that was required to perform corrections without them. Since a similar spline-based tool is also available in the OCT manufacturer's software, we consider our current result to be closer to reality.

We also note that part of the benefit from the uncertainty visualization might be due to the fact that our termination mechanism could have ended the experiment before the user viewed all B-scans. In practice, we believe that encouraging the expert to view only part of the data can be beneficial in order to improve the quality of a given large-scale training set (e.g., [51] used 52,377 B-scans) as much as possible within a limited time budget. For cases where it may be important to ensure the best possible accuracy within an arbitrary amount of time, the system may be modified to enforce inspection of each B-scan.

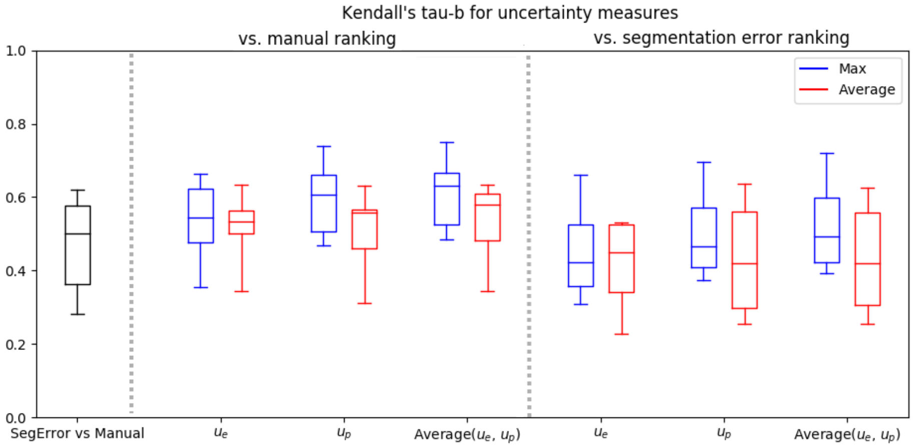


FIGURE 8.8: Comparing our uncertainty measures to a manual rating of segmentation quality or to segmentation errors with respect to a ground truth indicates a clear correlation, which shows that they are suitable for guiding users to problematic cases.

Drusen Editing Tools

To evaluate the newly added tools for drusen editing, we followed a similar paradigm as for the layer correction. In particular, users were asked to perform the correction twice. In the first round, they were allowed to use the basic tools (pen marker, line marker, flood fill) as well as our Semi-automated Drusen Extraction (SDE). In the second round, the SDE tool was excluded. In both cases, users had to correct the drusen segmentations on 10 B-scans, which were selected from an OCT volume different than the one used for layer correction. The experiments were terminated when the overall segmentation accuracy had reached at least 85% intersection over union (IOU) with respect to a ground truth that was created by an independent expert, without our SDE tool.

Table 8.3 summarizes the results. Both users exclusively used the SDE tool when it was available, which clearly demonstrates that they found it to be convenient and effective. We note that eliminating a false detection with a single click could be done either with the fill or the SDE tool. Our SDE tool greatly reduced the number of required actions, by around 84% of the value with basic tools, and the required time, by 73%.

8.8.2 Validity of Uncertainty Measures

The goal of the two uncertainty measures u_e and u_p that were introduced in Section 8.5 is to draw the expert's attention to B-scans that are likely to require corrections. To investigate their validity, i.e., the extent to which they measure what they are supposed to, we correlated them with human judgment, and with a more traditional measure of segmentation error.

In particular, we compared rankings of B-scans according to u_e and u_p to two other rankings: The first one is based on classifying segmentation quality in six different OCT scans with about 365 B-scans in total, on a scale from 1 to 5, 1 indicating that no further attention by the user is required, 5 indicating a strong need for correction. This judgment was made by a human rater considering the severity of segmentation failure. The second ranking is based on the segmentation error with respect to an expert-provided ground truth. For this, the column-wise Euclidean distance between the automated segmentation and the ground truth was taken. Since many layer segmentation errors only affect a small region in the data, our error measure averages the largest 10% of per-column distances in each B-scan.

We use Kendall's tau-b rank correlation coefficient [80] to compare the rankings based on our uncertainty measures to the manual and segmentation error based ones. Tau-b ranges between 1 (identical rankings) and -1 (inverted rankings), and is applicable to rankings that include ties. This is important since our manual ranking is based on a discrete scale.

From the six different OCT scans, we obtain six different rank correlation coefficients, which we summarize using boxplots in Figure 8.8. As a reference for the magnitude of correlations that we can reasonably expect, both plots include a direct comparison between segmentation error based and manual rankings (black). The remaining comparisons are between our uncertainty measures and manual rankings (left) or segmentation errors (right), respectively. As uncertainty measures, we consider u_e, u_p, and the average of the two. We aggregate them over the B-scan either using the max (as suggested in Section 8.5.1, shown in blue), or by averaging over all image columns (red).

These results confirm our choice of max-aggregation by illustrating that it leads to slightly stronger overall correlations than simple averaging. Also, slightly better results were obtained when combining u_e and u_p than when using them individually, which confirms our design decision to display both in the user interface.

The overall correlation between our uncertainty measures and manual or error-based rankings was similarly strong as the correlation between our two references. Together with the results from the user study, this validates their suitability for user guidance.

8.9 Conclusion

Convolutional Neural Networks are the current state-of-the-art approach to medical image segmentation. Despite their success, it is still quite common that the results of CNNs require corrections when they were trained on limited data, or when applying them to images whose characteristics differ from the training data, e.g., due to differences in acquisition devices, or in the presence of pathological anomalies. This establishes a need for systems that make it easier for human experts to localize regions in which the automated segmentation is likely to have failed, and to efficiently correct any remaining segmentation errors. To our knowledge, compared to the rapidly growing number of CNN-based segmentation algorithms [102], there is still relatively little effort in that direction.

We presented a system that implements uncertainty visualization and intelligent interaction for the specific use case of CNN-based segmentations of retinal layers and drusen in three-dimensional images from Optical Coherence Tomography. Our prototype implementation is based on Python and Qt, and is publicly available at `https://github.com/MedVisBonn/OCT-Annotation-Tool`. In a small user study, we found that our system reduced the time for layer correction by 53% compared to spline-based interaction, which we consider to be the previous state-of-the-art. For drusen segmentation, we even achieved a reduction by 73%. Moreover, our uncertainty measures were validated against manual quality measures and segmentation error with respect to an expert-provided ground truth.

We hope that our work can inspire similar tools also for other applications. Even though our system was originally motivated by the need to ensure reliability of medical segmentations, a promising use case of segmentation correction that we hope to investigate in the future is the iterative refinement of CNNs, i.e., using segmentation correction to more efficiently create large and high-quality training data. In this respect, it will have to compete against weakly and semi-supervised alternatives [125].

Chapter 9

Conclusion

9.1 Contributions to Drusen Segmentation

Ridge-enhancing filter

Our contribution to this topic was that we introduced a multi-scale anisotropic fourth order diffusion (MAFOD) filter, which enhances ridges and valleys. We showed that, particularly for RPE layer extraction in B-scans, this filter preserves RPE location better than the commonly used bilateral filter. MAFOD uses a vesselness measure to automatically estimate the underlying ridge scale in addition to a fourth order tensor that allows to sharpen creases in perpendicular direction and to smooth them in parallel direction. We compared our proposed filter to other alternative second-order and fourth-order diffusion filters and observed that the MAFOD filter better restores ridge locations. This filter can optionally be adjusted to only enhance ridges, valleys or both at the same time.

Fully-automated drusen segmentation

We needed to re-implement a state-of-the-art technique as a baseline for our drusen segmentation pipeline. For this reason we picked the algorithm introduced by Chen et al. [18]. Since we got much poorer results than reported in that paper, we invested additional effort to analyze the reasons for this, and to adapt the baseline technique to our data-set. We identified multiple factors that might explain this; First, our data-set had half the axial resolution compared to the reported data-set in the original paper, which made it more difficult to separate the RPE from the ellipsoid zone. In addition, our data-set included scans with a greater diversity of drusen load, making it a more heterogeneous and more challenging data-set compared to the reference data-set. Realizing these factors motivated us to refine the baseline using four additional regularization steps, and by replacing the bilateral filter with the MAFOD filter. Even though we still could not reproduce the previously reported results, our refined algorithm significantly improved the performance of the state-of-the-art algorithm. To ensure algorithm reproducibility we made our reproduction of Chen et al. algorithm as well as our proposed refined algorithm programs publicly available at `https://github.com/MedVisBonn/DrusenSegmentation-ModifiedChen`.

In our second contribution to this topic we designed a CNN-based pipeline for automated segmentation of drusen in OCT images. We tested three different approaches, i.e., 1) directly segmenting drusen using a CNN, 2) combining a CNN-based RPE and BM layer segmentation with post-processing steps, and 3) combining a CNN-based RPE/BM complex segmentation with the post-processing steps for drusen segmentation. All three produced better results than the state-of-the-art method by Chen et al. [18], and the second pipeline gave the overall best results. The drawback of the CNN approach is that it requires a large training data-set, but when available, the CNN-based approach showed much higher accuracy and robustness in presence of geographic atrophy.

We used our proposed drusen segmentation pipeline in a laser intervention in early age-related macular degeneration (LEAD) study [198], whose goal was to investigate the effect of laser treatment on changes in drusen volume. Initially our pipeline had a poor performance on LEAD data-set as it was collected using a different scanning device compared to the data-set that was used for training our CNN. Instead of retraining the CNN with the new data, we solved this issue by preprocessing each image by matching their histogram to the average histogram of the data-set that was used to train the CNN.

Semi-automated drusen segmentation

Despite being the current state-of-the-art method for medical image segmentation, still sometimes it can happen for CNNs to fail to correctly perform the segmentation. One reason might be that the characteristics of the training data-set might not cover all the possible pathological anomalies of a disease or the data-set can come from different acquisition devices. So far relatively little work has been done in the direction of detecting and correcting these failed cases. Therefore, for retinal layer and drusen segmentation, we implemented a system that uses a CNN for the initial layer segmentation. Then the output maps of the network are utilized to create an uncertainty visualization and intelligent tools for the user. The visualized uncertainty guides the user to those cases where segmentations by the network are highly likely to be erroneous. The intelligent tools can be used to correct these failures. Compared to the previous state-of-the-art spline-based interaction, our system reduced the needed correction time almost by half for retinal layers. For drusen, the time was reduced by 73%. We evaluated the effectiveness of our uncertainty measures by finding a large agreement between ranked B-scans with respect to the uncertainty measures, and a manual ranking of B-scans according to their segmentation quality, and using the numerical segmentation error with respect to the ground truth. The code and the pre-trained CNN network for layer segmentation are available at `https://github.com/MedVisBonn/OCT-Annotation-Tool`.

9.2 Contributions Beyond Drusen Segmentation

Some of the contributions of this dissertation are not limited to drusen and retinal layer segmentation, and can serve as useful tools in the field of image processing and analysis in general.

Our multi-scale fourth order anisotropic diffusion filter allows to selectively enhance ridge-like and/or valley-like structures in images. When compared to other filters, we show that this filter helps to more accurately restore the center-line of such structures. The MAFOD filter can be used in different applications, such as improving vessel center-line detection in angiographic images, or when extended to 3D, can be used as a great visualization tool. For instance, one can represent a brain's white matter skeleton in fractional anisotropy images by enhancing and extracting the ridge surfaces in the image.

Even though CNNs have greatly improved image segmentation, still they can produce erroneous segmentations. Specially in medical applications, detecting and correcting the failed cases can be crucial. In this dissertation we proposed two CNN-based uncertainty measures that quickly guide users to the cases that are more likely

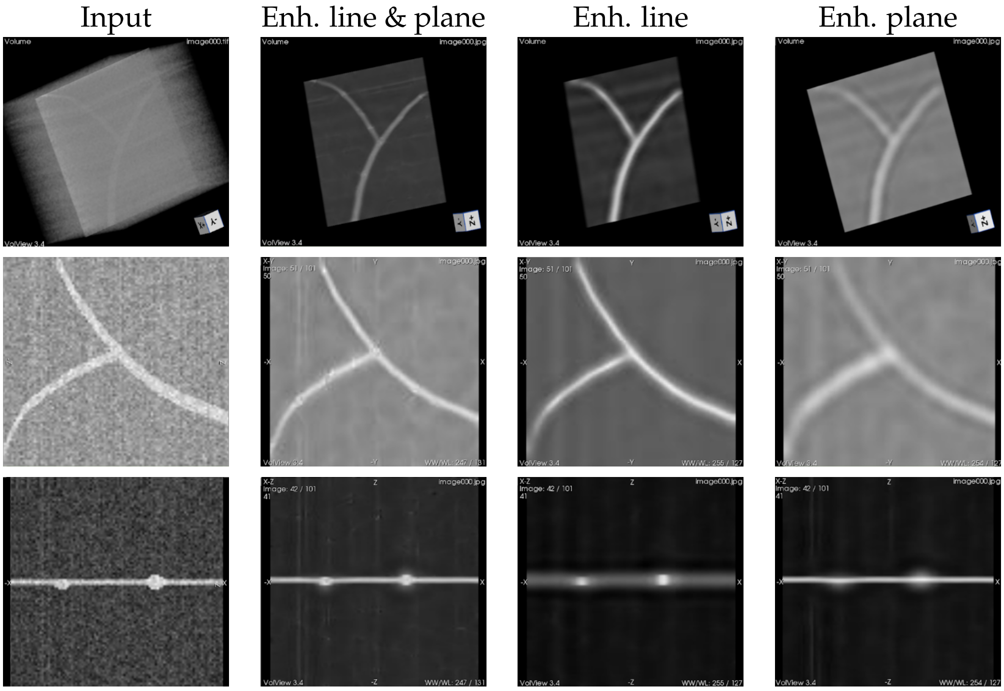


FIGURE 9.1: Selectively enhancing crease structures in 3D using the MAFOD filter. Top row is an overall view of the 3D volume. Second row is a XY 2D slice that passes through the middle of the planar structure. Bottom row represents a XZ cross-sectional slice. The MAFOD filter can be steered to enhance crease-lines and crease-surfaces at the same time (second column), or separately (third and fourth columns).

to have failed. These measures can be used to design similar guidance mechanism for other CNN-based image segmentation applications. Another transferable idea from this dissertation is the use of probability maps from the CNN to design fast segmentation correction tools.

In addition, our results from the evaluation of the Chen et al. [18] approach highlighted that the current way of evaluating algorithms does not allow to reliably infer how good the results are going to be on an independent data-set. This emphasizes that there is a need to do more work on the proper replication and validation of algorithms in the field of medical image analysis.

9.3 Future work

The MAFOD filter has the potential to be extended into higher dimensions. Once it is generalized to 3D, it opens up additional opportunities for fine grained steering of the filter to selectively enhance crease-lines or crease-surfaces with a steerable 3D fourth-order tensor. Figure 9.1 illustrates the results of selective crease enhancement using 3D MAFOD filter. The input is a synthetic 3D image containing a planar surface and a bifurcating tubular structure. The input is contaminated with Gaussian noise. This Figure shows that MAFOD filter can successfully reduce the noise and selectively enhance crease-lines and crease-surfaces. It could be also interesting to investigate filter's performance on the intersection point between crease-lines and crease-surfaces (see Figure 9.2).

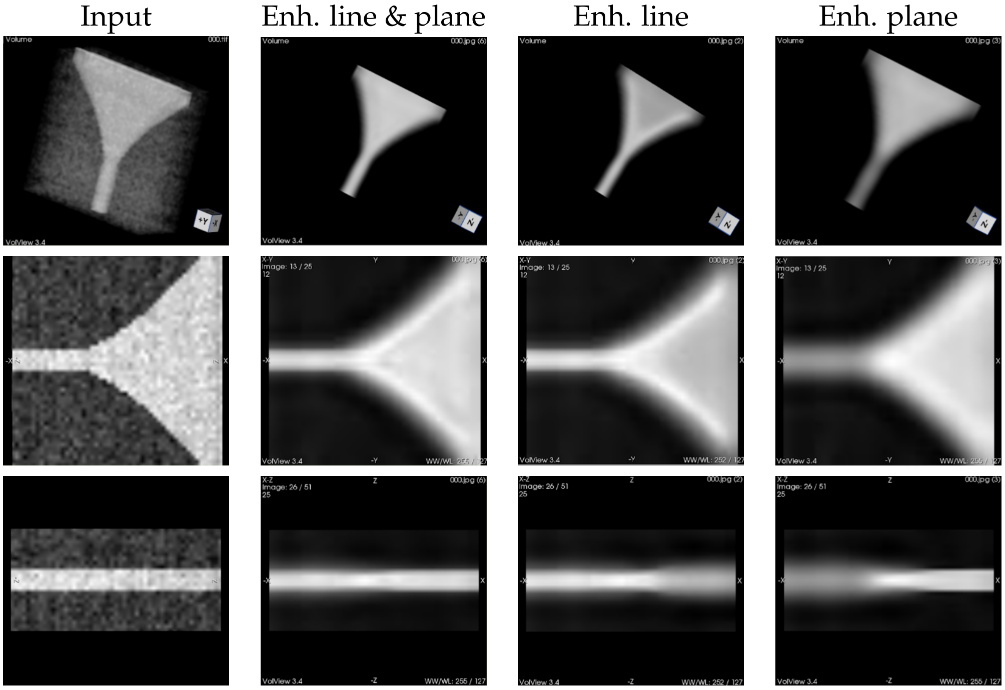


FIGURE 9.2: The effect of MAFOD filter on the intersection point between a ridge-line and a ridge-surface. Top row is an overall view of a tubular line that evolves into a planar shape. Second row is a XY 2D slice that passes through middle of the planar structure. Bottom row represents a XZ cross-sectional slice. The figure shows that enhancing only ridge-lines, enhances the boundaries of the ridge-surface as well.

Our proposed drusen segmentation pipeline consists of multiple engineered parts, such as shortest path finding in prediction maps, polynomial fitting or the false positive elimination steps. A general research direction could be to replace more of these steps with learned steps using neural networks. For instance the false positive elimination step can be learned by a CNN, using the height maps of the RPE and BM layers and the enface projection of the OCT volume. Or ideally one could design a deep neural network architecture to learn the drusen segmentation end-to-end, which could more effectively make use of the three-dimensional context of OCT images.

We hypothesize that our interactive framework for retinal layers and drusen segmentation can be used for iteratively creating large training sets. It could be interesting to evaluate the benefits of creating and improving training sets using this approach, and compare it to semi-supervised or weakly supervised alternatives [125].

Segmentation techniques similar to the ones developed within this dissertation can be used for the assessment of other AMD biomarkers in OCT. This opens up the opportunity of understanding the risk factors of AMD by exploring the correlation between these risk factors and AMD progression over time using thousands of automatically analysed OCT volumes. In addition, this can speed up the process of designing and studying possible treatments for AMD.

Bibliography

[1] Michael D Abràmoff, Mona K Garvin, and Milan Sonka. "Retinal imaging and image analysis". In: *IEEE reviews in biomedical engineering* 3 (2010), pp. 169–208.

[2] S. Agatonovic-Kustrin and R. Beresford. "Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research". In: *Journal of pharmaceutical and biomedical analysis* 22.5 (2000), pp. 717–727.

[3] Ahmed Al-Taie, Horst K. Hahn, and Lars Linsen. "Uncertainty estimation and visualization in probabilistic segmentation". In: *Computers & Graphics* 39 (2014), pp. 48–59.

[4] Roberto Annunziata, Ahmad Kheirkhah, Pedram Hamrah, and Emanuele Trucco. "Scale and curvature invariant ridge detector for tortuous and fragmented structures". In: *Proc. Medical Image Computing and Computer-Assisted Intervention (MICCAI), Part III*. Vol. 9351. LNCS. Springer, 2015, pp. 588–595.

[5] Syed Muhammad Anwar, Muhammad Majid, Adnan Qayyum, Muhammad Awais, Majdi Alnowami, and Muhammad Khurram Khan. "Medical image analysis using convolutional neural networks: a review". In: *Journal of medical systems* 42.11 (2018), p. 226.

[6] Stefanos Apostolopoulos, Sandro De Zanet, Carlos Ciller, Sebastian Wolf, and Raphael Sznitman. "Pathological OCT Retinal Layer Segmentation Using Branch Residual U-Shape Networks". In: *Proc. Medical Image Computing and Computer Assisted Intervention (MICCAI), Part III*. 2017, pp. 294–301.

[7] Rhona Asgari, José Ignacio Orlando, Sebastian Waldstein, Ferdinand Schlanitz, Magdalena Baratsits, Ursula Schmidt-Erfurth, and Hrvoje Bogunović. "Multiclass segmentation as multitask learning for drusen segmentation in retinal optical coherence tomography". In: *arXiv preprint arXiv: 1906.07679* (2019).

[8] Rhona Asgari, Sebastian Waldstein, Ferdinand Schlanitz, Magdalena Baratsits, Ursula Schmidt-Erfurth, and Hrvoje Bogunović. "U-Net with spatial pyramid pooling for drusen segmentation in optical coherence tomography". In: *International Workshop on Ophthalmic Medical Image Analysis*. Springer. 2019, pp. 77–85.

[9] Samer Barakat, N. Andrysco, and Xavier Tricoche. "Fast Extraction of High-quality Crease Surfaces for Visual Analysis". In: *Computer Graphics Forum* 30.3 (2011), pp. 961–970.

[10] Peter J. Basser and Sinisa Pajevic. "Spectral decomposition of a 4th-order covariance tensor: Applications to diffusion tensor MRI". In: *Signal Processing* 87 (2007), pp. 220–236.

[11] Neelakshi Bhagat and Christina J Flaxel. "Nonexudative macular degeneration". In: *Age-Related Macular Degeneration*. CRC Press, 2002, pp. 83–98.

[12] Lee Brandon and Adam Hoover. "Drusen detection in a retinal image using multi-level analysis". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2003, pp. 618–625.

[13] Rubén Cárdenes, Rodrigo de Luis García, and Meritxell Bach Cuadra. "A multidimensional segmentation evaluation for medical image data". In: *Computer Methods and Programs in Biomedicine* 96.2 (2009), pp. 108–124.

[14] P Carpineto, Marco Ciancaglini, A Aharrh-Gnama, D Cirone, and L Mastropasqua. "Custom measurement of retinal nerve fiber layer thickness using STRATUS OCT in normal eyes". In: *European journal of ophthalmology* 15.3 (2005), pp. 360–366.

[15] C Cañero and Petia Radeva. "Vesselness enhancement diffusion". In: *Pattern Recognition Letters* 24.16 (2003), pp. 3141–3151.

[16] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. "Hybrid task cascade for instance segmentation". In: *arXiv preprint arXiv:1901.07518* (2019).

[17] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. "Semantic image segmentation with deep convolutional nets and fully connected crfs". In: *arXiv preprint arXiv:1412.7062* (2014).

[18] Qiang Chen, Theodore Leng, Luoluo Zheng, Lauren Kutzscher, Jeffrey Ma, Luis de Sisternes, and Daniel L Rubin. "Automated drusen segmentation and quantification in SD-OCT images". In: *Medical image analysis* 17.8 (2013), pp. 1058–1072.

[19] Stephanie J. Chiu, Joseph A. Izatt, Rachelle V. O'Connell, Katrina P. Winter, Cynthia A. Toth, and Sina Farsiu. "Validated Automatic Segmentation of AMD Pathology Including Drusen and Geographic Atrophy in SD-OCT Images". In: *Investigative Opthalmology & Visual Science* 53.1 (2012), p. 53.

[20] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. "3d u-net: learning dense volumetric segmentation from sparse annotation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2016, pp. 424–432.

[21] Taco Cohen and Max Welling. "Group equivariant convolutional networks". In: *International conference on machine learning*. 2016, pp. 2990–2999.

[22] Matthew P Coleman. *An introduction to partial differential equations with MATLAB*. CRC Press, 2013.

[23] Johanna M Colijn, Gabriëlle HS Buitendijk, Elena Prokofyeva, Dalila Alves, Maria L Cachulo, Anthony P Khawaja, Audrey Cougnard-Gregoire, Bénédicte MJ Merle, Christina Korb, Maja G Erke, et al. "Prevalence of age-related macular degeneration in Europe: the past and the future". In: *Ophthalmology* 124.12 (2017), pp. 1753–1763.

[24] Christine A Curcio, Emma C Zanzottera, Thomas Ach, Chandrakumar Balaratnasingam, and K Bailey Freund. "Activated retinal pigment epithelium, an optical coherence tomography biomarker for progression in age-related macular degeneration". In: *Investigative ophthalmology & visual science* 58.6 (2017), BIO211–BIO226.

[25] George E Dahl, Tara N Sainath, and Geoffrey E Hinton. "Improving deep neural networks for LVCSR using rectified linear units and dropout". In: *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE. 2013, pp. 8609–8613.

[26] Jifeng Dai, Kaiming He, and Jian Sun. "Instance-aware semantic segmentation via multi-task network cascades". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 3150–3158.

[27] Matthew D Davis, Ronald E Gangnon, Li Yin Lee, Larry D Hubbard, BE Klein, Ronald Klein, Frederick L Ferris, Susan B Bressler, and Roy C Milton. "The Age-Related Eye Disease Study severity scale for age-related macular degeneration: AREDS report No. 17." In: *Archives of ophthalmology (Chicago, Ill.: 1960)* 123.11 (2005), pp. 1484–1498.

[28] Jeffrey De Fauw, Joseph R Ledsam, Bernardino Romera-Paredes, Stanislav Nikolov, Nenad Tomasev, Sam Blackwell, Harry Askham, Xavier Glorot, Brendan O'Donoghue, Daniel Visentin, et al. "Clinically applicable deep learning for diagnosis and referral in retinal disease". In: *Nature medicine* 24.9 (2018), p. 1342.

[29] Maxime Descoteaux, D Louis Collins, and Kaleem Siddiqi. "A geometric flow for segmenting vasculature in proton-density weighted MRI". In: *Medical image analysis* 12.4 (2008), pp. 497–513.

[30] Stephan Didas, Joachim Weickert, and Bernhard Burgeth. "Properties of higher order nonlinear diffusion filtering". In: *Journal of Mathematical Imaging and Vision* 35.3 (2009), pp. 208–226.

[31] Bruno Diniz, Ramiro Ribeiro, Florian M Heussen, Mauricio Maia, and Srinivas Sadda. "Drusen measurements comparison by fundus photograph manual delineation versus optical coherence tomography retinal pigment epithelial segmentation automated analysis". In: *Retina* 34.1 (2014), pp. 55–62.

[32] Timothy Dozat. "Incorporating nesterov momentum into adam". In: (2016).

[33] Cattleya Duanggate and Bunyarit Uyyanonvara. "A review of automatic drusen detection and segmentation from retinal images". In: *The 3rd International Symposium on Biomedical Engineering*. 2008, pp. 222–225.

[34] David Eberly, Robert Gardner, Bryan Morse, Stephen Pizer, and Christine Scharlach. "Ridges for image analysis". In: *Journal of Mathematical Imaging and Vision* 4.4 (1994), pp. 353–373.

[35] David H Eberly and Stephen M Pizer. "Ridge flow models for image segmentation". In: *Medical Imaging 1994*. International Society for Optics and Photonics. 1994, pp. 54–64.

[36] David Howard Eberly. "Geometric methods for analysis of ridges in n-dimensional images". PhD thesis. The University of North Carolina at Chapel Hill, 1994.

[37] Albert O Edwards, Robert Ritter, Kenneth J Abel, Alisa Manning, Carolien Panhuysen, and Lindsay A Farrer. "Complement factor H polymorphism and age-related macular degeneration". In: *Science* 308.5720 (2005), pp. 421–424.

[38] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. *Visualizing Higher-Layer Features of a Deep Network*. Tech. rep. 1341. Presented at ICML 2009 Workshop on Learning Feature Hierarchies. University of Montreal, 2009.

[39] Leyuan Fang, David Cunefare, Chong Wang, Robyn H. Guymer, Shutao Li, and Sina Farsiu. "Automatic segmentation of nine retinal layer boundaries in OCT images of non-exudative AMD patients using deep learning and graph search". In: *Biomedical Optics Express* 8.5 (2017), p. 2732.

[40] Sina Farsiu, Stephanie J Chiu, Joseph A Izatt, and Cynthia A Toth. "Fast detection and segmentation of drusen in retinal optical coherence tomography images". In: *Ophthalmic Technologies XVIII*. Vol. 6844. Proc. SPIE. 2008, p. 68440D.

[41] Marta S Figueroa, Angel Regueras, and Josefina Bertrand. "Laser photocoagulation to treat macular soft drusen in age-related macular degeneration." In: *Retina (Philadelphia, Pa.)* 14.5 (1994), pp. 391–396.

[42] Alejandro F Frangi, Wiro J Niessen, Koen L Vincken, and Max A Viergever. "Multiscale vessel enhancement filtering". In: *Proc. Medical Image Computing and Computer-Assisted Interventation (MICCAI)*. Vol. 1496. LNCS. Springer, 1998, pp. 130–137.

[43] Erik Franken and Remco Duits. "Crossing-preserving coherence-enhancing diffusion on invertible orientation scores". In: *International Journal of Computer Vision* 85.3 (2009), p. 253.

[44] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, Pablo Martinez-Gonzalez, and Jose Garcia-Rodriguez. "A survey on deep learning techniques for image and video semantic segmentation". In: *Applied Soft Computing* 70 (2018), pp. 41–65.

[45] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. "Image style transfer using convolutional neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2414–2423.

[46] Qichuan Geng, Zhong Zhou, and Xiaochun Cao. "Survey of recent progress in semantic image segmentation with CNNs". In: *Science China Information Sciences* 61.5 (2018), p. 051101.

[47] Alexander Geurts, Georgios Sakas, Arjan Kuijper, Meike Becker, and Tatiana von Landesberger. "Visual Comparison of 3D Medical Image Segmentation Algorithms Based on Statistical Shape Models". In: *Int'l Conf. on Digital Human Modeling (DHM)*. Vol. 9185. LNCS. Springer, 2015, pp. 336–344.

[48] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. 2010, pp. 249–256.

[49] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[50] Shekoufeh Gorgi Zadeh, Maximilian W.M. Wintergerst, and Thomas Schultz. "Uncertainty-Guided Semi-Automated Editing of CNN-based Retinal Layer Segmentations in Optical Coherence Tomography". In: *Proc. Visual Computing for Biology and Medicine*. 2018, pp. 107–115. DOI: 10.2312/vcbm.20181235. URL: https://doi.org/10.2312/vcbm.20181235.

[51] Shekoufeh Gorgi Zadeh, Maximilian WM Wintergerst, Vitalis Wiens, Sarah Thiele, Frank G Holz, Robert P Finger, and Thomas Schultz. "CNNs Enable Accurate and Fast Segmentation of Drusen in Optical Coherence Tomography". In: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer, 2017, pp. 65–73. DOI: 10.1007/978-3-319-67558-9_8. URL: https://doi.org/10.1007/978-3-319-67558-9_8.

[52] RE Graham. "Snow removal–A noise-stripping process for picture signals". In: *Information Theory, IRE Transactions on* 8.2 (1962), pp. 129–144.

[53] W Richard Green, Peter J McDonnell, and Julia H Yeo. "Pathologic features of senile macular degeneratlon". In: *Ophthalmology* 92.5 (1985), pp. 615–627.

[54] John B Greer, Andrea L Bertozzi, and Guillermo Sapiro. "Fourth order partial differential equations on general geometries". In: *Journal of Computational Physics* 216.1 (2006), pp. 216–246.

[55] Giovanni Gregori, Fenghua Wang, Philip J Rosenfeld, Zohar Yehoshua, Ninel Z Gregori, Brandon J Lujan, Carmen A Puliafito, and William J Feuer. "Spectral domain optical coherence tomography imaging of drusen in nonexudative age-related macular degeneration". In: *Ophthalmology* 118.7 (2011), pp. 1373–1379.

[56] Choroidal Neovascularization Prevention Trial Research Group et al. "Laser treatment in eyes with large drusen: short-term effects seen in a pilot randomized clinical trial". In: *Ophthalmology* 105.1 (1998), pp. 11–23.

[57] Yanming Guo, Yu Liu, Theodoros Georgiou, and Michael S. Lew. "A review of semantic segmentation using deep neural networks". In: *International Journal of Multimedia Information Retrieval* 7.2 (2018), pp. 87–93.

[58] Daniel Haehn, Seymour Knowles-Barley, Mike Roberts, Johanna Beyer, Narayanan Kasthuri, Jeff W. Lichtman, and Hanspeter Pfister. "Design and Evaluation of Interactive Proofreading Tools for Connectomics". In: *IEEE Trans. on Visualization and Computer Graphics* 20.12 (2014), pp. 2466–2475.

[59] Mohammad Reza Hajiaboli. "An anisotropic fourth-order diffusion filter for image noise removal". In: *International Journal of Computer Vision* 92.2 (2011), pp. 177–191.

[60] Jared Hamwood, David Alonso-Caneiro, Scott A. Read, Stephen J. Vincent, and Michael J. Collins. "Effect of patch size and network architecture on a convolutional neural network approach for automatic segmentation of OCT retinal layers". In: *Biomedical Optics Express* 9.7 (2018), p. 3049.

[61] Julius Hannink, Remco Duits, and Erik Bekkers. "Crossing-preserving multiscale vesselness". In: *Proc. Medical Image Computing and Computer-Assisted Intervention (MICCAI), Part II*. Vol. 8674. LNCS. Springer, 2014, pp. 603–610.

[62] Robert M. Haralick, Layne T. Watson, and Thomas J. Laffey. "The Topographic Primal Sketch". In: *Int'l Journal of Robotics Research* 2.1 (1983), pp. 50–72.

[63] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *Proc. IEEE Int'l Conf. on Computer Vision (ICCV)*. 2015, pp. 1026–1034.

[64] Frank Heckel, Jan Hendrik Moltz, Christian Tietjen, and Horst K. Hahn. "Sketch-Based Editing Tools for Tumour Segmentation in 3D Medical Images". In: *Computer Graphics Forum* 32.8 (2013), pp. 144–157.

[65] David Huang, Eric A Swanson, Charles P Lin, Joel S Schuman, William G Stinson, Warren Chang, Michael R Hee, Thomas Flotte, Kenton Gregory, Carmen A Puliafito, et al. "Optical coherence tomography". In: *science* 254.5035 (1991), pp. 1178–1181.

[66] Daniel P Huttenlocher, Gregory Klanderman, and William J Rucklidge. "Comparing images using the Hausdorff distance". In: *IEEE Trans. on Pattern Analysis and Machine Intelligence* 15.9 (1993), pp. 850–863.

[67] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *arXiv preprint arXiv:1502.03167* (2015).

[68] Daisuke Iwama, Masanori Hangai, Sotaro Ooto, Atsushi Sakamoto, Hideo Nakanishi, Takashi Fujimura, Amitha Domalpally, Ronald P Danis, and Nagahisa Yoshimura. "Automated assessment of drusen using three-dimensional spectral-domain optical coherence tomography". In: *Investigative ophthalmology & visual science* 53.3 (2012), pp. 1576–1583.

[69] Rama D Jager, William F Mieler, and Joan W Miller. "Age-related macular degeneration". In: *New England Journal of Medicine* 358.24 (2008), pp. 2606–2617.

[70] Bernd Jahne. *Spatio-temporal image processing: Theory and scientific applications*. Springer-Verlag New York, Inc., 1993.

[71] Anil K Jain, Jianchang Mao, and KM Mohiuddin. "Artificial neural networks: A tutorial". In: *Computer* 3 (1996), pp. 31–44.

[72] Nieraj Jain, Sina Farsiu, Aziz A Khanifar, Srilaxmi Bearelly, R Theodore Smith, Joseph A Izatt, and Cynthia A Toth. "Quantitative comparison of drusen segmented on SD-OCT versus drusen delineated on color fundus photographs". In: *Investigative ophthalmology & visual science* 51.10 (2010), pp. 4875–4883.

[73] Tim Jerman, Franjo Pernuš, Boštjan Likar, and Žiga Špiclin. "Enhancement of Vascular Structures in 3D and 2D Angiographic Images". In: *IEEE Trans. on Medical Imaging* 35.9 (2016), pp. 2107–2118.

[74] Zexuan Ji, Qiang Chen, Sijie Niu, Theodore Leng, and Daniel L Rubin. "Beyond retinal layers: a deep voting model for automated geographic atrophy segmentation in SD-OCT images". In: *Translational vision science & technology* 7.1 (2018), pp. 1–1.

[75] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. "Caffe: Convolutional Architecture for Fast Feature Embedding". In: *arXiv preprint arXiv:1408.5093* (2014).

[76] Vishal Jindal. "Interconnection between brain and retinal neurodegenerations". In: *Molecular neurobiology* 51.3 (2015), pp. 885–892.

[77] M. Kahng, P. Y. Andrews, A. Kalro, and D. H. Chau. "ActiVis: Visual Exploration of Industry-Scale Deep Neural Network Models". In: *IEEE Trans. on Visualization and Computer Graphics* 24.1 (2018), pp. 88–97.

[78] Yogesan Kanagasingam, Alauddin Bhuiyan, Michael D Abramoff, R Theodore Smith, Leonard Goldschmidt, and Tien Y Wong. “Progress on retinal image analysis for age related macular degeneration”. In: *Progress in retinal and eye research* 38 (2014), pp. 20–42.

[79] Andrej Karpathy and Li Fei-Fei. “Deep visual-semantic alignments for generating image descriptions”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3128–3137.

[80] M. G. Kendall. “The Treatment of Ties in Ranking Problems”. In: *Biometrika* 33.3 (1945), pp. 239–251.

[81] Gordon Kindlmann, D.B. Ennis, R.T. Whitaker, and C.-F. Westin. “Diffusion Tensor Analysis With Invariant Gradients and Rotation Tangents”. In: *IEEE Trans. on Medical Imaging* 26.11 (2007), pp. 1483–1499.

[82] Gordon Kindlmann, Raúl San José Estépar, Stephen M. Smith, and Carl-Fredrik Westin. “Sampling and Visualizing Creases with Scale-Space Particles”. In: *IEEE Trans. on Visualization and Computer Graphics* 15.6 (2009), pp. 1415–1424.

[83] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[84] Robert J Klein, Caroline Zeiss, Emily Y Chew, Jen-Yue Tsai, Richard S Sackler, Chad Haynes, Alice K Henning, John Paul SanGiovanni, Shrikant M Mane, Susan T Mayne, et al. “Complement factor H polymorphism in age-related macular degeneration”. In: *Science* 308.5720 (2005), pp. 385–389.

[85] Ronald Klein, Barbara EK Klein, and Kathryn LP Linton. “Prevalence of age-related maculopathy: the Beaver Dam Eye Study”. In: *Ophthalmology* 99.6 (1992), pp. 933–943.

[86] Ronald Klein, Stacy M Meuer, Michael D Knudtson, Sudha K Iyengar, and Barbara EK Klein. “The epidemiology of retinal reticular drusen”. In: *American journal of ophthalmology* 145.2 (2008), pp. 317–326.

[87] Jan J Koenderink and Andrea J van Doorn. “Local features of smooth shapes: Ridges and courses”. In: *SPIE's 1993 International Symposium on Optics, Imaging, and Instrumentation*. International Society for Optics and Photonics. 1993, pp. 2–13.

[88] Th M Koller, Guido Gerig, Gabor Szekely, and Daniel Dettwiler. “Multiscale detection of curvilinear structures in 2-D and 3-D image data”. In: *Computer Vision, 1995. Proceedings., Fifth International Conference on*. IEEE. 1995, pp. 864–869.

[89] Karl Krissian, Grégoire Malandain, Nicholas Ayache, Régis Vaillant, and Yves Trousset. “Model based multiscale detection of 3D vessels”. In: *Biomedical Image Analysis, 1998. Proceedings. Workshop on*. IEEE. 1998, pp. 202–210.

[90] P Kshirsagar and N Rathod. “Artificial neural network”. In: *International Journal of Computer Applications* (2012).

[91] Tatiana von Landesberger, Gennady L. Andrienko, Natalia V. Andrienko, Sebastian Bremm, Matthias Kirschner, Stefan Wesarg, and Arjan Kuijper. “Opening up the "black box" of medical image segmentation with statistical shape models”. In: *The Visual Computer* 29.9 (2013), pp. 893–905.

[92] Tatiana von Landesberger, Dennis Basgier, and Meike Becker. “Comparative Local Quality Assessment of 3D Medical Image Segmentations with Focus on Statistical Shape Model-Based Algorithms”. In: *IEEE Trans. on Visualization and Computer Graphics* 22.12 (2016), pp. 2537–2549.

[93] Max WK Law and Albert CS Chung. “Three dimensional curvilinear structure detection using optimally oriented flux”. In: *Computer Vision–ECCV 2008*. Springer, 2008, pp. 368–382.

[94] Yann LeCun, Yoshua Bengio, et al. “Convolutional networks for images, speech, and time series”. In: *The handbook of brain theory and neural networks* 3361.10 (1995), p. 1995.

[95] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. “Photo-realistic single image super-resolution using a generative adversarial network”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4681–4690.

[96] Cecilia S. Lee, Doug M. Baughman, and Aaron Y. Lee. “Deep Learning Is Effective for Classifying Normal versus Age-Related Macular Degeneration Optical Coherence Tomography Images”. In: *Ophthalmology Retina* (2017). In press.

[97] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. “Deeply-supervised nets”. In: *Artificial intelligence and statistics*. 2015, pp. 562–570.

[98] Hyungwoo Lee, Kyung Eun Kang, Hyewon Chung, and Hyung Chan Kim. “Automated segmentation of lesions including subretinal hyperreflective material in neovascular age-related macular degeneration”. In: *American journal of ophthalmology* 191 (2018), pp. 64–75.

[99] Noah Lee, Andrew F Laine, and Theodore R Smith. “Learning non-homogenous textures and the unlearning problem with application to drusen detection in retinal images”. In: *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. IEEE. 2008, pp. 1215–1218.

[100] Laurence S Lim, Paul Mitchell, Johanna M Seddon, Frank G Holz, and Tien Y Wong. “Age-related macular degeneration”. In: *The Lancet* 379.9827 (2012), pp. 1728–1738.

[101] Tony Lindeberg. “Edge detection and ridge detection with automatic scale selection”. In: *International Journal of Computer Vision* 30.2 (1998), pp. 117–156.

[102] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. “A survey on deep learning in medical image analysis”. In: *Medical image analysis* 42 (2017), pp. 60–88.

[103] M. Liu, J. Shi, K. Cao, J. Zhu, and S. Liu. “Analyzing the Training Processes of Deep Generative Models”. In: *IEEE Trans. on Visualization and Computer Graphics* 24.1 (2018), pp. 77–87.

[104] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu. “Towards Better Analysis of Deep Convolutional Neural Networks”. In: *IEEE Trans. on Visualization and Computer Graphics* 23.1 (2017), pp. 91–100.

[105] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. "Path aggregation network for instance segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8759–8768.

[106] Anat London, Inbal Benhar, and Michal Schwartz. "The retina as a window to the brain—from eye research to CNS disorders". In: *Nature Reviews Neurology* 9.1 (2013), p. 44.

[107] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[108] Antonio M López, David Lloret, Joan Serrat, and Juan J Villanueva. "Multilocal creaseness based on the level-set extrinsic curvature". In: *Computer Vision and Image Understanding* 77.2 (2000), pp. 111–144.

[109] Antonio M López, Felipe Lumbreras, Joan Serrat, and Juan J Villanueva. "Evaluation of methods for ridge and valley detection". In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 4 (1999), pp. 327–335.

[110] Der-Shan Luo, Michael King, Stephen Glick, et al. "Local geometry variable conductance diffusion for post-reconstruction filtering". In: *Nuclear Science, IEEE Transactions on* 41.6 (1994), pp. 2800–2806.

[111] Angelica Ly, Michael Yapp, Lisa Nivison-Smith, Nagi Assaad, Michael Hennessy, and Michael Kalloniatis. "Developing prognostic biomarkers in intermediate age-related macular degeneration: their clinical use in predicting progression". In: *Clinical and Experimental Optometry* 101.2 (2018), pp. 172–181.

[112] Marius Lysaker, Arvid Lundervold, and Xue-Cheng Tai. "Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time". In: *IEEE Trans. on Image Processing* 12.12 (2003), pp. 1579–1590.

[113] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. "Rectifier nonlinearities improve neural network acoustic models". In: *Proc. icml*. Vol. 30. 1. 2013, p. 3.

[114] Aravindh Mahendran and Andrea Vedaldi. "Understanding deep image representations by inverting them". In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 5188–5196.

[115] Lena Maier-Hein, Matthias Eisenmann, Annika Reinke, Sinan Onogur, Marko Stankovic, Patrick Scholz, Tal Arbel, Hrvoje Bogunovic, Andrew P Bradley, Aaron Carass, et al. "Why rankings of biomedical image analysis competitions should be interpreted with care". In: *Nature communications* 9.1 (2018), p. 5217.

[116] JB Antoine Maintz, Petra A van Den Elsen, Max Viergever, et al. "Evaluation of ridge seeking operators for multimodality medical image matching". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 18.4 (1996), pp. 353–365.

[117] Markus Mayer. "Automated Glaucoma Detection with Optical Coherence Tomography". PhD thesis. Friedrich-Alexander-Universität Erlangen-Nürnberg, 2018.

[118] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. “V-net: Fully convolutional neural networks for volumetric medical image segmentation”. In: *2016 Fourth International Conference on 3D Vision (3DV)*. IEEE. 2016, pp. 565–571.

[119] Nader Nassif, Barry Cense, B Hyle Park, Seok H Yun, Teresa C Chen, Brett E Bouma, Guillermo J Tearney, and Johannes F de Boer. “In vivo human retinal imaging by ultrahigh-speed spectral domain optical coherence tomography”. In: *Optics letters* 29.5 (2004), pp. 480–482.

[120] Yurii Nesterov. “A method for unconstrained convex minimization problem with the rate of convergence O (1/kˆ 2)”. In: *Doklady AN USSR*. Vol. 269. 1983, pp. 543–547.

[121] Anke Neumann and Cristian Lorenz. “Statistical shape model based segmentation of medical images”. In: *Computerized Medical Imaging and Graphics* 22.2 (1998), pp. 133–143.

[122] Anh Nguyen, Jason Yosinski, and Jeff Clune. “Deep neural networks are easily fooled: High confidence predictions for unrecognizable images”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 427–436.

[123] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. “Learning deconvolution network for semantic segmentation”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1520–1528.

[124] Sílvia Delgado Olabarriaga and Arnold W. M. Smeulders. “Interaction in the segmentation of medical images: A survey”. In: *Medical Image Analysis* 5.2 (2001), pp. 127–142.

[125] George Papandreou, Liang-Chieh Chen, Kevin P. Murphy, and Alan L. Yuille. “Weakly-and Semi-Supervised Learning of a Deep Convolutional Network for Semantic Image Segmentation”. In: *Proc. IEEE Int’l Conf. on Computer Vision (ICCV)*. IEEE, 2015, pp. 1742–1750.

[126] S Swarna Parvathi and N Devi. “Automatic drusen detection from colour retinal images”. In: *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*. Vol. 2. IEEE. 2007, pp. 377–381.

[127] Ronald Peikert and Filip Sadlo. “Height ridge computation and filtering for visualization”. In: *Visualization Symposium, 2008. PacificVIS’08. IEEE Pacific*. IEEE. 2008, pp. 119–126.

[128] Pietro Perona and Jitendra Malik. “Scale-space and edge detection using anisotropic diffusion”. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence* 12.7 (1990), pp. 629–639.

[129] Pascal Peter, Lilli Kaufhold, and Joachim Weickert. “Turning diffusion-based image colorization into efficient color compression”. In: *IEEE Transactions on Image Processing* 26.2 (2016), pp. 860–869.

[130] N. Pezzotti, T. Höllt, J. Van Gemert, B. P. Lelieveldt, E. Eisemann, and A. Vilanova. “DeepEyes: Progressive Visual Analytics for Designing Deep Neural Networks”. In: *IEEE Trans. on Visualization and Computer Graphics* 24.1 (2018), pp. 98–108.

[131] Boris T Polyak. “Some methods of speeding up the convergence of iteration methods”. In: *USSR Computational Mathematics and Mathematical Physics* 4.5 (1964), pp. 1–17.

[132] Susanne G Pondorfer, Maximilian WM Wintergerst, Shekoufeh Gorgi Zadeh, Thomas Schultz, Manuel Heinemann, Frank G Holz, and Robert P Finger. "Association of Visual Function Measures with Drusen Volume in Early Stages of Age-Related Macular Degeneration". In: *Investigative Ophthalmology & Visual Science* 61.3 (2020), pp. 55–55. DOI: 10.1167/iovs.61.3.55. URL: https://doi.org/10.1167/iovs.61.3.55.

[133] Dan P Popescu, Costel Flueraru, Youxin Mao, Shoude Chang, John Disano, Sherif Sherif, Michael G Sowa, et al. "Optical coherence tomography: fundamental principles, instrumental designs and biomedical applications". In: *Biophysical reviews* 3.3 (2011), p. 155.

[134] Kristin C. Potter, Samuel Gerber, and Erik W. Anderson. "Visualization of Uncertainty without a Mean". In: *IEEE Computer Graphics and Applications* 33.1 (2013), pp. 75–79.

[135] J.-S. Prassni, Timo Ropinski, and Klaus Hinrichs. "Uncertainty-aware guided volume segmentation". In: *IEEE Trans. Visualization and Computer Graphics* 16.6 (2010), pp. 1358–1365.

[136] Giuseppe Querques, Maria Vittoria Cicinelli, Alessandro Rabiolo, Luigi de Vitis, Riccardo Sacconi, Lea Querques, and Francesco Bandello. "Laser photocoagulation as treatment of non-exudative age-related macular degeneration: state-of-the-art and future perspectives". In: *Graefe's Archive for Clinical and Experimental Ophthalmology* 256.1 (2018), pp. 1–9.

[137] Renata G. Raidou, Freek J. J. Marcelis, Marcel Breeuwer, M. Eduard Gröller, Anna Vilanova, and Huub M. M. van de Wetering. "Visual Analytics for the Exploration and Assessment of Segmentation Errors". In: *VCBM 16: Eurographics Workshop on Visual Computing for Biology and Medicine, Bergen, Norway, September 7-9, 2016*. 2016, pp. 193–202.

[138] Konstantinos Rapantzikos and Michalis Zervakis. "Nonlinear enhancement and segmentation algorithm for the detection of age-related macular degeneration (AMD) in human eye's retina". In: *Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205)*. Vol. 3. IEEE. 2001, pp. 1055–1058.

[139] Fabian Rathke, Stefan Schmidt, and Christoph Schnörr. "Probabilistic intraretinal layer segmentation in 3-D OCT images using global shape regularization". In: *Medical image analysis* 18.5 (2014), pp. 781–794.

[140] P. E. Rauber, S. G. Fadel, A. X. Falcao, and A. C. Telea. "Visualizing the Hidden Activity of Artificial Neural Networks". In: *IEEE Trans. on Visualization and Computer Graphics* 23.1 (2017), pp. 101–110.

[141] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems*. 2015, pp. 91–99.

[142] Xiuxiu Ren, Yuanjie Zheng, Yanna Zhao, Chao Luo, Hong Wang, Jian Lian, and Yunlong He. "Drusen segmentation from retinal images via supervised feature learning". In: *IEEE Access* 6 (2017), pp. 2952–2961.

[143] "RETOUCH - Retinal OCT Fluid Challenge". (Last accessed 02/10/2020). URL: https://retouch.grand-challenge.org/.

[144] "ROCC - Retinal OCT Classification Challenge". (Last accessed 02/10/2020). URL: https://rocc.grand-challenge.org/.

[145] Isabelle Roesch and Tobias Günther. "Visualization of Neural Network Predictions for Weather Forecasting". In: *Proc. Vision, Modeling, Visualization (VMV)*. Ed. by Matthias Hullin, Reinhard Klein, Thomas Schultz, and Angela Yao. The Eurographics Association, 2017, pp. 61–68.

[146] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *Int'l Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer. 2015, pp. 234–241.

[147] Paul Rosenthal, Marc Ritter, Danny Kowerko, and Christian Heine. "OphthalVis-Making Data Analytics of Optical Coherence Tomography Reproducible." In: *EuroRV^3@ EuroVis*. 2016, pp. 9–13.

[148] Abhijit Guha Roy, Nassir Navab, and Christian Wachinger. "Concurrent spatial and channel 'squeeze & excitation'in fully convolutional networks". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2018, pp. 421–429.

[149] Sebastian Ruder. "An overview of gradient descent optimization algorithms". In: *arXiv preprint arXiv:1609.04747* (2016).

[150] David E Rumelhart, Richard Durbin, Richard Golden, and Yves Chauvin. "Backpropagation: The basic theory". In: *Backpropagation: Theory, architectures and applications* (1995), pp. 1–34.

[151] Ahmed Saad, Ghassan Hamarneh, and Torsten Möller. "Exploration and Visualization of Segmentation Uncertainty using Shape and Appearance Prior Information". In: *IEEE Trans. on Visualization and Computer Graphics* 16.6 (2010), pp. 1366–1375.

[152] Srinivas R Sadda, Robyn Guymer, Frank G Holz, Steffen Schmitz-Valckenberg, Christine A Curcio, Alan C Bird, Barbara A Blodi, Ferdinando Bottoni, Usha Chakravarthy, Emily Y Chew, et al. "Consensus definition for atrophy associated with age-related macular degeneration on OCT: classification of atrophy report 3". In: *Ophthalmology* 125.4 (2018), pp. 537–548.

[153] Shirley H Sarks. "Drusen patterns predisposing to geographic atrophy of the retinal pigment epithelium". In: *Australian Journal of Opthalmology* 10.2 (1982), pp. 91–97.

[154] Z Ben Sbeh, Laurent D Cohen, Gérard Mimoun, and Gabriel Coscas. "A new approach of geodesic reconstruction for drusen segmentation in eye fundus images". In: *IEEE Transactions on medical imaging* 20.12 (2001), pp. 1321–1333.

[155] Zakaria Ben Sbeh, Laurent D Cohen, Gerard Mimoun, Gabriel Coscas, and Gisele Soubrane. "An adaptive contrast method for segmentation of drusen". In: *Proceedings of International Conference on Image Processing*. Vol. 1. IEEE. 1997, pp. 255–258.

[156] Hanno Scharr and Kai Krajsek. "A short introduction to diffusion-like methods". In: *Mathematical Methods for Signal and Image Analysis and Representation*. Springer, 2012, pp. 1–30.

[157] O. Scherzer. "Denoising with higher order derivatives of bounded variation and an application to parameter estimation". In: *Computing* 60.1 (Mar. 1998), pp. 1–27.

[158] Thomas Schlegl, Sebastian M Waldstein, Hrvoje Bogunovic, Franz Endstraßer, Amir Sadeghipour, Ana-Maria Philip, Dominika Podkowinski, Bianca S Gerendas, Georg Langs, and Ursula Schmidt-Erfurth. "Fully automated detection and quantification of macular fluid in OCT using deep learning". In: *Ophthalmology* 125.4 (2018), pp. 549–558.

[159] U Schmidt-Erfurth, S Klimscha, SM Waldstein, and H Bogunović. "A view of the current and future role of optical coherence tomography in the management of age-related macular degeneration". In: *Eye* 31.1 (2017), p. 26.

[160] Ursula Schmidt-Erfurth, Hrvoje Bogunovic, Sophie Klimscha, Xiaofeng Hu, Thomas Schlegl, Amir Sadeghipour, Bianca S Gerendas, Aaron Osborne, and Sebastian M. Waldstein. "Machine learning to predict the individual progression of AMD from imaging biomarkers". In: *Proc. Association for Research in Vision and Ophthalmology*. 2017, p. 3398.

[161] Ursula Schmidt-Erfurth, Amir Sadeghipour, Bianca S Gerendas, Sebastian M Waldstein, and Hrvoje Bogunović. "Artificial intelligence in retina". In: *Progress in retinal and eye research* (2018).

[162] Ursula Schmidt-Erfurth and Sebastian M Waldstein. "A paradigm shift in imaging biomarkers in neovascular age-related macular degeneration". In: *Progress in Retinal and eye Research* 50 (2016), pp. 1–24.

[163] William J Schroeder and Kenneth M Martin. "Overview of visualization". In: *The visualization handbook* (2005), pp. 3–35.

[164] Thomas Schultz, Holger Theisel, and Hans-Peter Seidel. "Crease surfaces: From theory to extraction and application to diffusion tensor MRI". In: *IEEE Trans. on Visualization and Computer Graphics* 16.1 (2010), pp. 109–119.

[165] Thomas Schultz, Joachim Weickert, and Hans-Peter Seidel. "A Higher-Order Structure Tensor". In: *Visualization and Processing of Tensor Fields – Advances and Perspectives*. Ed. by David H. Laidlaw and Joachim Weickert. Springer, 2009, pp. 263–280.

[166] Luis de Sisternes, Noah Simon, Robert Tibshirani, Theodore Leng, and Daniel L Rubin. "Quantitative SD-OCT Imaging Biomarkers as Indicators of Age-Related Macular Degeneration ProgressionPredicting AMD Progression Using SD-OCT Features". In: *Investigative Ophthalmology & Visual Science* 55.11 (2014), pp. 7093–7103.

[167] S Skiena. "Dijkstra's algorithm". In: *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica, Reading, MA: Addison-Wesley* (1990), pp. 225–227.

[168] William E Smiddy and Stuart L Fine. "Prognosis of patients with bilateral macular drusen". In: *Ophthalmology* 91.3 (1984), pp. 271–277.

[169] Andres Fco Sóle, Antonio López, and Guillermo Sapiro. "Crease enhancement diffusion". In: *Computer Vision and Image Understanding* 84.2 (2001), pp. 241–248.

[170] Milan Sonka and Michael D Abràmoff. "Quantitative analysis of retinal OCT". In: *Medical Image Analysis* 33 (2016), pp. 165–169.

[171] Christoph W Spraul, Gabriele E Lang, Hans E Grossniklaus, and Gerhard K Lang. "Characteristics of drusen and Bruch's membrane in post-mortem eyes with age-related macular degeneration". In: *Der Ophthalmologe* 95.2 (1998), pp. 73–79.

[172] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. “Striving for simplicity: The all convolutional net”. In: *arXiv preprint arXiv:1412.6806* (2014).

[173] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. “Dropout: a simple way to prevent neural networks from overfitting”. In: *The Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.

[174] Giovanni Staurenghi, Srinivas Sadda, Usha Chakravarthy, Richard F Spaide, et al. “Proposed lexicon for anatomic landmarks in normal posterior segment spectral-domain optical coherence tomography: the IN• OCT consensus”. In: *Ophthalmology* 121.8 (2014), pp. 1572–1578.

[175] Julia S Steinberg, Arno P Göbel, Monika Fleckenstein, Frank G Holz, and Steffen Schmitz-Valckenberg. “Reticular drusen in eyes with high-risk characteristics for progression to late-stage age-related macular degeneration”. In: *British Journal of Ophthalmology* 99.9 (2015), pp. 1289–1294.

[176] H. Strobelt, S. Gehrmann, H. Pfister, and A. M. Rush. “LSTMVis: A Tool for Visual Analysis of Hidden State Dynamics in Recurrent Neural Networks”. In: *IEEE Trans. on Visualization and Computer Graphics* 24.1 (2018), pp. 667–676.

[177] Ingo Stuke, Til Aach, Erhardt Barth, and Cicero Mota. “Analysing superimposed oriented patterns”. In: *6th IEEE Southwest Symposium on Image Analysis and Interpretation, 2004.* IEEE. 2004, pp. 133–137.

[178] Brian Summa, Julien Tierny, and Valerio Pascucci. “Visualizing the Uncertainty of Graph-based 2D Segmentation with Min-path Stability”. In: *Computer Graphics Forum* 36.3 (2017), pp. 133–143.

[179] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. “Intriguing properties of neural networks”. In: *arXiv preprint arXiv:1312.6199* (2013).

[180] Andrey N Tikhonov and Vasiliy Y Arsenin. *Solutions of ill-posed problems*. Washington D.C.: V.H. Winston ; New York: J. Wiley, 1977.

[181] J. Tumblin and G. Turk. “LCIS: A boundary hierarchy for detail-preserving contrast reduction”. In: *SIGGRAPH ’99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1999, pp. 83–90.

[182] Fan-Yin Tzeng and Kwan-Liu Ma. “Opening the Black Box – Data Driven Visualization of Neural Networks”. In: *Proc. IEEE Visualization*. Ed. by C. Silva, E. Gröller, and H. Rushmeier. 2005, pp. 383–390.

[183] Waldo Valenzuela, Stephen J. Ferguson, Dominika Ignasiak, Gaëlle Diserens, Levin Häni, Roland Wiest, Peter Vermathen, Chris Boesch, and Mauricio Reyes. “FISICO: Fast Image SegmentatIon COrrection”. In: *PLOS ONE* 11.5 (2016). Ed. by Dzung Pham, e0156035.

[184] Guo W Wei. “Generalized Perona-Malik equation for image restoration”. In: *IEEE Signal processing letters* 6.7 (1999), pp. 165–167.

[185] Joachim Weickert. *Anisotropic diffusion in image processing*. Teubner Stuttgart, 1998.

[186] Joachim Weickert. “Multiscale texture enhancement”. In: *Computer analysis of images and patterns*. Springer. 1995, pp. 230–237.

[187] Joachim Weickert, Sven Grewenig, Christopher Schroers, and Andrés Bruhn. "Cyclic schemes for PDE-based image analysis". In: *International Journal of Computer Vision* (2016), pp. 1–25.

[188] Joachim Weickert, BM Ter Haar Romeny, Max Viergever, et al. "Efficient and reliable schemes for nonlinear diffusion filtering". In: *Image Processing, IEEE Transactions on* 7.3 (1998), pp. 398–410.

[189] Tino Weinkauf. "Extraction of Topological Structures in 2D and 3D vector fields". PhD thesis. Otto-von-Guericke-Universität Magdeburg, Universitätsbibliothek, 2008.

[190] E. T. Whittaker. "On a new method of graduation". In: *Proceedings of the Edinburgh Mathematical Society* 10 (Feb. 1922), pp. 63–75.

[191] Maximilian WM Wintergerst, Shekoufeh Gorgi Zadeh, Vitalis Wiens, Sarah Thiele, Steffen Schmitz-Valckenberg, Frank G Holz, Robert P Finger, and Thomas Schultz. "Replication and Refinement of an Algorithm for Automated Drusen Segmentation on Optical Coherence Tomography". In: *Scientific Reports* 10.1 (2020), pp. 1–7. DOI: 10.1038/s41598-020-63924-6. URL: https://doi.org/10.1038/s41598-020-63924-6.

[192] Maximilian WM Wintergerst, Thomas Schultz, Johannes Birtel, Alexander K Schuster, Norbert Pfeiffer, Steffen Schmitz-Valckenberg, Frank G Holz, and Robert P Finger. "Algorithms for the automated analysis of age-related macular degeneration biomarkers on optical coherence tomography: a systematic review". In: *Translational vision science & technology* 6.4 (2017), pp. 10–10.

[193] Andrew P Witkin. "Scale-space filtering: A new approach to multi-scale description". In: *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'84.* Vol. 9. IEEE. 1984, pp. 150–153.

[194] MacIej Wojtkowski, Tomasz Bajraszewski, Iwona Gorczyńska, Piotr Targowski, Andrzej Kowalczyk, Wojciech Wasilewski, and Czesław Radzewicz. "Ophthalmic imaging by spectral optical coherence tomography". In: *American journal of ophthalmology* 138.3 (2004), pp. 412–419.

[195] Ute EK Wolf-Schnurrbusch, Lala Ceklic, Christian K Brinkmann, Milko E Iliev, Manuel Frey, Simon P Rothenbuehler, Volker Enzmann, and Sebastian Wolf. "Macular thickness measurements in healthy eyes using six different optical coherence tomography instruments". In: *Investigative ophthalmology & visual science* 50.7 (2009), pp. 3432–3437.

[196] Wan Ling Wong, Xinyi Su, Xiang Li, Chui Ming G Cheung, Ronald Klein, Ching-Yu Cheng, and Tien Yin Wong. "Global prevalence of age-related macular degeneration and disease burden projection for 2020 and 2040: a systematic review and meta-analysis". In: *The Lancet Global Health* 2.2 (2014), e106–e116.

[197] K. Wongsuphasawat, D. Smilkov, J. Wexler, J. Wilson, D. Mane, D. Fritz, D. Krishnan, F. B. Viegas, and M. Wattenberg. "Visualizing Dataflow Graphs of Deep Learning Models in TensorFlow". In: *IEEE Trans. on Visualization and Computer Graphics* 24.1 (2018), pp. 1–12.

[198] Zhichao Wu, Chi D Luu, Lauren AB Hodgson, Emily Caruso, Kate H Brassington, Nicole Tindill, Khin Zaw Aung, Colin A Harper, Sanjeewa S Wickremasinghe, Sukhpal S Sandhu, et al. "Secondary and Exploratory Outcomes of the Subthreshold Nanosecond Laser Intervention Randomized Trial in Age-Related Macular Degeneration: A LEAD Study Report". In: *Ophthalmology Retina* 3.12 (2019), pp. 1026–1034. DOI: 10.1016/j.oret.2019.07.008. URL: https://doi.org/10.1016/j.oret.2019.07.008.

[199] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. "Empirical evaluation of rectified activations in convolutional network". In: *arXiv preprint arXiv:1505.00853* (2015).

[200] Jason Yosinski, Jeff Clune, Anh Mai Nguyen, Thomas J. Fuchs, and Hod Lipson. "Understanding Neural Networks Through Deep Visualization". In: *CoRR* abs/1506.06579 (2015). Presented at ICML 2015 Deep Learning Workshop. eprint: 1506.06579.

[201] Yu-Li You and Mostafa Kaveh. "Fourth-order partial differential equations for noise removal". In: *Image Processing, IEEE Transactions on* 9.10 (2000), pp. 1723–1730.

[202] Yu-Li You, Wenyuan Xu, Allen Tannenbaum, and Mostafa Kaveh. "Behavioral analysis of anisotropic diffusion in image processing". In: *Image Processing, IEEE Transactions on* 5.11 (1996), pp. 1539–1553.

[203] Fisher Yu and Vladlen Koltun. "Multi-scale context aggregation by dilated convolutions". In: *arXiv preprint arXiv:1511.07122* (2015).

[204] Shekoufeh Gorgi Zadeh, Stephan Didas, Maximilian WM Wintergerst, and Thomas Schultz. "Multi-scale Anisotropic Fourth-Order Diffusion Improves Ridge and Valley Localization". In: *Journal of Mathematical Imaging and Vision* 59.2 (2017), pp. 257–269. DOI: 10.1007/s10851-017-0729-1. URL: https://doi.org/10.1007/s10851-017-0729-1.

[205] Shekoufeh Gorgi Zadeh, Maximilian WM Wintergerst, and Thomas Schultz. "Intelligent interaction and uncertainty visualization for efficient drusen and retinal layer segmentation in Optical Coherence Tomography". In: *Computers & Graphics* 83 (2019), pp. 51–61. DOI: 10.1016/j.cag.2019.07.001. URL: https://doi.org/10.1016/j.cag.2019.07.001.

[206] Ehud Zamir and Narsing A Rao. "Histopathological characteristics of Age-related macular degeneration". In: *Age-Related Macular Degeneration* (2002), pp. 15–26.

[207] Matthew D Zeiler. "ADADELTA: an adaptive learning rate method". In: *arXiv preprint arXiv:1212.5701* (2012).

[208] Matthew D. Zeiler and Rob Fergus. "Visualizing and Understanding Convolutional Networks". In: *Proc. European Conf. on Computer Vision (ECCV)*. Vol. 8689. LNCS. Springer, 2014, pp. 818–833.

[209] Bo Zhao, Jiashi Feng, Xiao Wu, and Shuicheng Yan. "A survey on deep learning-based fine-grained object classification and semantic segmentation". In: *International Journal of Automation and Computing* 14.2 (2017), pp. 119–135.

[210] Yalin Zheng, Bryan M. Williams, Harry Pratt, Baidaa Al-Bander, Xiangqian Wu, and Yitian Zhao. "Computer aided diagnosis of age-related macular degeneration in 3D OCT images by deep learning". In: *Proc. Association for Research in Vision and Ophthalmology*. 2017, p. 824.

[211] Yi-Tong Zhou and Rama Chellappa. “Computation of optical flow using a neural network”. In: *IEEE International Conference on Neural Networks*. Vol. 1998. 1988, pp. 71–78.

[212] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. “Unet++: A nested u-net architecture for medical image segmentation”. In: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer, 2018, pp. 3–11.