Kent Academic Repository

Bai, Fangliang (2021) *Computational Methods for Image Acquisition and Analysis with Applications in Optical Coherence Tomography.* Doctor of Philosophy (PhD) thesis, University of Kent,.

Downloaded from

https://kar.kent.ac.uk/89414/ The University of Kent's Academic Repository KAR

The version of record is available from

https://doi.org/10.22024/UniKent/01.02.89414

This document version

UNSPECIFIED

DOI for this version

Licence for this version

CC BY-NC-SA (Attribution-NonCommercial-ShareAlike)

Additional information

Versions of research works

Versions of Record

If this version is the version of record, it is the same as the published version available on the publisher's web site. Cite as the published version.

Author Accepted Manuscripts

If this document is identified as the Author Accepted Manuscript it is the version after peer review but before type setting, copy editing or publisher branding. Cite as Surname, Initial. (Year) 'Title of article'. To be published in *Title of Journal* , Volume and issue numbers [peer-reviewed accepted version]. Available at: DOI or URL (Accessed: date).

Enquiries

If you have questions about this document contact ResearchSupport@kent.ac.uk. Please include the URL of the record in KAR. If you believe that your, or a third party's rights have been compromised through this document please see our Take Down policy (available from https://www.kent.ac.uk/guides/kar-the-kent-academic-repository#policies).

Computational Methods for Image Acquisition and Analysis with Applications in Optical Coherence Tomography

Fangliang Bai

School of Physical Sciences

A thesis submitted to the University of Kent
in the subject of physics for the degree of
Doctor of Philosophy

2021

Acknowledgements

Chasing the goal of this PhD degree was one of the most important things I have dedicated myself to do. During this journey, it would no just happens without the indispensable support from some intelligent people. First, I would like to give my sincere appreciation to Dr Stuart Gibson for his invaluable mentorship during the journey. He is a fountain of wisdom, if I may say so modestly, from where the insightful thought and considerate care always guide me out when I lost way in the maze. I couldn't have even hoped for a better supervisor. I also want to thank my co-supervisor, Dr George Dobre, who has given excellent support during the progress of my PhD. I am very grateful. Special thanks to and Dr Jinchao Liu, Dr Chao Wang, Dr Xiaojuan Liu and Dr Chaitanya Mididoddi, who shared their experiences and knowledge when I need hardware support. I am also in memory of Craig Douglas, who gave me support on hardware implementation. I also want to thank my coworkers, who gave me assistance when things got harder. They are Prof. Adrian Podoleanu, Dr Margarita Osadchy, Dr Manuel Marques and Dr David Pickup. In addition, none of this would have been possible without the help of the University of Kent and the East Kent Hospitals University NHS Foundation Trust. Lastly, I would like to thank my family. Although they probably didn't fully understand my studies, they invested the most in me with deep love. They give me the most important reason to continue. They are the ones who I devote this work to.

"The greatest opportunity offered by AI is not reducing errors or workloads, or even curing cancer: it is the opportunity to restore the precious and time-honoured connection and trust—the human touch—between patients and doctors. Not only would we have more time to come together, enabling far deeper communication and compassion, but also we would be able to revamp how we select and train doctors."

—Eric Topol, Deep Medicine: How Artificial Intelligence Can Make Healthcare Human Again

To my dear family

Abstract

The computational approach to image acquisition and analysis plays an important role in medical imaging and optical coherence tomography (OCT). This thesis is dedicated to the development and evaluation of algorithmic solutions for better image acquisition and analysis with a focus on OCT retinal imaging.

For image acquisition, we first developed, implemented, and systematically evaluated a compressive sensing approach for image/signal acquisition for single-pixel camera architectures and an OCT system. Our evaluation outcome provides a detailed insight into implementing compressive data acquisition of those imaging systems. We further proposed a convolutional neural network model, LSHR-Net, as the first deep-learning imaging solution for the single-pixel camera. This method can achieve better accuracy, hardware-efficient image acquisition and reconstruction than the conventional compressive sensing algorithm.

Three image analysis methods were proposed to achieve retinal OCT image analysis with high accuracy and robustness. We first proposed a framework for healthy retinal layer segmentation. Our framework consists of several image processing algorithms specifically aimed at segmenting a total of 12 thin retinal cell layers, outperforming other segmentation methods. Furthermore, we proposed two deep-learning-based models to segment retinal oedema lesions in OCT images, with particular attention on processing small-scale datasets. The first model leverages transfer learning to implement oedema segmentation and achieves better accuracy than comparable methods. Based on the meta-learning concept, a second model was designed to be a solution for general medical image segmentation. The results of this work indicate that our model can be applied to retinal OCT images and other small-scale medical image data, such as skin cancer, demonstrated in this thesis.

Contents

Acknowledgements iii

Abstract vii

List of Figures xiii

List of Tables xvii

List of Author's Publications xix

1 Introduction 1

2 Evaluation of compressed signal sampling and reconstruction 5

2.1 Introduction . . . 5

2.2 Compressed image sampling and reconstruction of SPC system . . . 6

2.2.1 Hardware design . . . 9

2.2.2 Saturation calibration and dynamic range conversion . . . 12

2.2.3 SPC control software . . . 14

2.2.4 Experimental results . . . 19

2.3 Compressed sampling and reconstruction of PTS-OCT optical signal 22

2.3.1 Evaluation of signal reconstruction for PTS-OCT . . . 26

2.3.2 Evaluation metrics . . . 27

2.3.3 Reconstruction accuracy analysis . . . 27

2.3.4 Computational efficiency analysis . . . 31

2.4 Conclusion . . . 34

3 Single pixel imaging with a mixed-weights neural network 37

3.1 Introduction . . . 37

3.2 Review of existing approaches . . . 40

3.2.1 Networks based on pre-generated patterns . . . 40

3.2.2 Networks based on learned patterns . . . 41

3.2.3 Networks based on binary patterns . . . 42

3.3 Super-resolution SPC imaging using a deep neural network . . . 44

3.3.1 Image reconstruction net . . . 46

3.3.2 Residual correction net 49
3.4 Network training 52
3.4.1 Loss function 52
3.4.2 Training strategy 53
3.4.3 Network parameters and training hyper-parameters 54
3.5 Experiment results and discussion 55
3.5.1 Datasets 55
3.5.2 Experiment results with simulation dataset 56
3.5.3 Model training analysis with fixed and binary sampling schemes 61
3.5.4 Analysis of the reconstruction efficiency 67
3.5.5 Experiment results of incorporating with SPC hardware . . 69
3.6 Conclusions 72

4 A review of retinal OCT imaging and its clinical applications 75
4.1 Introduction 75
4.2 OCT evolution for scanning human retina 76
4.3 Retinal structures and lesions in OCT images 79
4.3.1 Retinal structure of healthy eyes 79
4.3.2 OCT image of diabetic macular oedema 80
4.3.3 OCT images of central and branch retinal vein occlusions . . 82
4.3.4 OCT images of pigment epithelial detachments 83
4.4 Review of approaches to retinal structure and defects segmentation 83
4.4.1 Segmentation of retinal layer boundaries 85
4.4.2 Segmentation of oedemas in the retina 95
4.5 Conclusion 100

5 Superpixel guided active contour segmentation of retinal layers in OCT volumes 103
5.1 Introduction 103
5.2 Boundary detection using adaptive curves, superpixels and active contours 105
5.2.1 Detection of retinal region in vertical direction using the adaptive curve 105
5.2.2 Retinal region grouping using superpixel aggregation method 111
5.2.3 Self-adaptive multi-boundary detection using active contour 116
5.2.4 Layer estimation in OCT Volume using the Kalman filter 120
5.3 Retinal boundary detection of OCT volume and segmentation results 123
5.3.1 Detection framework 125
5.3.2 Retinal layer segmentation in B-scans 129
5.3.3 En-face image extraction of retinal layers 132
5.4 Discussion and conclusion 137

6 Cystoid macular oedema segmentation using transfer learning 141
6.1 Introduction . . . 141
6.2 CMO characteristics and segmentation framework . . . 143
6.2.1 CMO characteristics of the diabetic retinopathy dataset . . . 143
6.2.2 Problem statement . . . 145
6.2.3 FCN network model . . . 145
6.2.4 Fully connected CRFs classification . . . 147
6.3 Fine-tuning strategy . . . 148
6.3.1 Dataset and data augmentation . . . 149
6.3.2 Framework settings . . . 150
6.4 Experimental results . . . 153
6.4.1 Evaluation metrics . . . 153
6.4.2 Segmentation results . . . 153
6.5 Conclusions and future work . . . 159

7 Image retrieval-based few-shot medical image segmentation 161
7.1 Introduction . . . 161
7.2 Proposed method . . . 164
7.2.1 Problem setup for one-shot segmentation . . . 164
7.2.2 Network architecture . . . 166
7.2.3 Support branch . . . 168
7.2.4 Query branch . . . 170
7.2.5 Embedding matching strategy . . . 172
7.2.6 Loss function . . . 173
7.2.7 Training strategy . . . 174
7.3 Test results . . . 175
7.3.1 Test datasets . . . 175
7.3.2 Evaluation results . . . 178
7.4 Key components analysis . . . 182
7.4.1 Embedding distribution . . . 182
7.4.2 Image retrieval . . . 184
7.4.3 Attention masks . . . 186
7.5 Conclusion . . . 189

8 Conclusion 193
8.1 Summary . . . 193
8.2 Achievements . . . 196
8.3 Future work . . . 198

Bibliography 199

List of Figures

2.1 The diagram of SPC hardware setup. 10

2.2 The circuit noise. 10

2.3 The design diagram of the potentiometer. 11

2.4 The ratio of noise against potentiometer gain. 12

2.5 Linearity correction of photodetector output for finding its optimum position. 14

2.6 The dynamic range mapping diagram. 15

2.7 The measurement procedure of the single-pixel camera hardware. . 16

2.8 The architecture and measurement procedure of the single-pixel camera software (SPC toolbox). 17

2.9 The UI of SPC toolbox, designed using Matlab GUI interface. . . . 18

2.10 The target image of a pac-man doodle. 19

2.11 An example set of reconstruction results of the same object using four methods separately. The measurements was obtained under the same environment condition. The quantitative evaluation of each reconstructed image was measured with PSNR value comparing to the original image. 20

2.12 An example set of reconstruction results of the same object using CS-based algorithms. The measurements was obtained under three measurement ratios. The quantitative evaluation of each reconstructed image was measured with PSNR value comparing to the original image. 21

2.13 Results of the sample object. 22

2.14 Block diagram of the proposed compressive sensing PTS-OCT system. 23

2.15 Interference pattern diagram. 24

2.16 The optical signal processing of our single-layer PTS-OCT measurement. 25

2.17 The reconstructed frequency domain signals for all five algorithms. . 29

2.18 RMSE of reconstructed signal calculated over entire frequency range. 30

2.19 RMSE of reconstructed signal calculated for the frequencies spanned by the four dominant peaks only. 30

2.20 RFE of reconstructed signal calculated for the four dominant peaks. 31

2.21 The computational time as a function of the number of measurements. 33

3.1 Overview of the proposed network structure. 45

3.2 The image sampling and reconstruction operations. 47
3.3 The structure of the residual correction net. 50
3.4 The phase shift operation. 52
3.5 The reconstruction result from the five methods, including two of ours, at compression ratio $R = 0.01$. 58
3.6 The reconstruction result from the five methods, including two of ours, at compression ratio $R = 0.10$. 58
3.7 The reconstruction result from the five methods, including two of ours, at compression ratio $R = 0.25$. 59
3.8 The comparison of reconstructed image between our static-weights and learned-weights networks. 60
3.9 The evaluation of the learned binary model on the ImageNet test dataset. 61
3.10 The validation losses based on static and learned binary patterns . 63
3.11 Binary pattern leanring during training. 64
3.12 The amount of updated binary weights during training. 66
3.13 The examples of training-optimised binary pattern. First column: patterns at $R = 0.25$. Second column: patterns at $R = 0.1$. Third column: patterns at $R = 0.01$. 66
3.14 The average PSNR of test results and reconstruction time using different numbers of residual blocks in the recursive structure. . . . 69
3.15 The schematic of the hardware setup. 70
3.16 The reconstruction results of 9 random selected MNIST handwritten numbers using the hardware measurements. 71
3.17 The reconstruction results of 9 random selected Omniglot characters using the hardware measurements. 72

4.1 The optical coherence tomography measures the backscattering light intensity of the sample beam. 77
4.2 A normal retina at the fovea region. 80
4.3 The diabetic macular oedema in retinal OCT B-scan image. 81
4.4 OCT image of retinal vein occlusion. 83
4.5 A cross-sectional retinal image with early-stage pigment epithelium detachment. 84
4.6 Retinal layer segmentation steps in [1]. 87
4.7 The boundary segmentation results of [1]. 88
4.8 The nine-step segmentation algorithm schematic for eight retinal layer boundary in SD-OCT images. 88
4.9 The boundary segmentation algorithm schematic for individual layer [2] 89
4.10 Segmentation using Dijkstra's algorithm, automatic endpoint initialisation, and search space limitation. 90
4.11 Boundary correction at vessel region. 91
4.12 Boundary correction at the fovea point. 92

4.13 Segmentation algorithm schematic. 93
4.14 Boundary correction by the active contour model. 94
4.15 Boundary segmentation of a normal adult retinal layer boundaries. . 95
4.16 The anisotropic diffusion filtering is applying to an OCT image. . . 96
4.17 The fluid-filled regions contours extracted by the GVF snake algorithm. 98
4.18 B-scan image with DME and its denoised result [3]. 98
4.19 Automatic fluid detection errors. Top: original OCT image. 99

5.1 The visualisation of the random sampling point update by the adaptive curve model on a sine wave. 108
5.2 The visualisation of the random sampling point update by the adaptive curve model on a sine wave. 109
5.3 The Vitreous-IML boundary estimation. The circle indicates the 17 final sampling points. 110
5.4 The segmentation of vitreous-IML boundary by adaptive-curve detection method. 111
5.5 The 2D superpixel clustering of OCT images. 113
5.6 The diagram of the superpixel updating procedure at a single A-scan. 114
5.7 The segmentation results of RPEI-RPE boundary using the superpixel algorithm. 116
5.8 Four configuration types for three adjacent contour points. 118
5.9 The refining results of active contour at RPEI/RPE boundary. . . . 119
5.10 The prediction of Vitreous-IML boundary using the Kalman filter. . 122
5.11 The diagram of checking boundary continuity when converting the floating-point values to unsigned pixel values. 122
5.12 The outlier-refined results of Kalman filter prediction at the vitreous-IML boundary. 123
5.13 The schematic of boundary prediction using the Kalman filter. . . . 124
5.14 The optical noise of the OCT system. 126
5.15 The pre-processing procedure of the OCT volume data. 126
5.16 The volumetric data alignment. Top: raw 3D OCT dataset. 127
5.17 The schematic of the whole framework of retinal layer boundary detection. 130
5.18 The segmentation results of the vitreous-IML boundary. 131
5.19 The segmentation result of 12 boundaries in the single B-scan. . . . 132
5.20 The segmentation result of 10 boundaries in the 3D volume. 133
5.21 The segmentation result of 10 boundaries in the 3D volume. 134
5.22 En-face image (C-scan) of the IS-OS layer. 135
5.23 En-face image (C-scan) of the GCL layer. 136

6.1 A retinal OCT image from our experimental dataset with two DMO symptoms in the foveal region. 144

6.2 Ground-truth segmentations of DMO in OCT retinal image, carried out by two ophthalmologists separately. 145
6.3 A schematic of the FCN network with different skip connection schemes. 146
6.4 The segmentation results of macular oedema by two clinicians and our framework. 154
6.5 The segmentation results of retinal thickening by two clinicians and our framework. 156
6.6 Segmentation results for more severe oedema as carried out by clinicians and the framework. 157
6.7 A visual comparison of manual and automatic segmentation. 158
6.8 A comparison between Chiu's work and our results. 158
6.9 The segmentation result from inaccurate prediction. 159

7.1 The overall structure of the SegMetaNet network. 167
7.2 The support branch of the MetaSegNet. 169
7.3 The query branch of the MetaSegNet. 171
7.4 The target matching mechanism with the attention unit inside. . . . 172
7.5 The diagram of forming an training episode. 174
7.6 An example of a retinal OCT B-scan image showing all three types of oedemas in the retinal tissue. 176
7.7 The appearance of normal retinal tissue, captured by three different OCT scanning devices. 177
7.8 The images of the same melanoma tissue, captured by a conventional camera (left) and a dermoscopy (right). 178
7.9 The segmentation results of retinal OCT images. 179
7.10 The segmentation results of melanoma lesion in the dermoscopy images. 181
7.11 The embedding visualisation of all support OCT images. 183
7.12 The embedding visualisation of all support dermoscopy images. . . 184
7.13 The retrieved OCT images based on the query image content. . . . 185
7.14 The melanoma objects of the retrieved images. 187
7.15 The attention masks generated for the OCT images. 188
7.16 The attention masks of the dermoscopy images. 189

List of Tables

2.1 The average reconstruction time. 33

3.1 The summary of reviewed methods. 43
3.2 The PSNR of 11 test image in dB from recent six methods at three measurement ratios. 57
3.3 The sample images from reconstruction of the large scale test dataset. 62
3.4 The five network methods were compared when restoring an image of size 32 × 32. 68
3.5 The analysis of the number of weights in residual blocks of the 5 methods. 68

5.1 Contour search configuration 119

6.1 The architecture of the FCN-8 model used in the experiment. 152

List of Author's Publications

Journal articles

- **Fangliang Bai**, Jinchao Liu, Xiaojuan Liu, Margarita Osadchy, Chao Wang, and Stuart J. Gibson. "LSHR-Net: A hardware-friendly solution for high-resolution computational imaging using a mixed-weights neural network." Neurocomputing 406 (2020): 169-181.
- Mididoddi, Chaitanya K., **Fangliang Bai**, Guoqing Wang, Jinchao Liu, Stuart Gibson, and Chao Wang. "High-throughput photonic time-stretch optical coherence tomography with data compression." IEEE Photonics Journal 9, no. 4 (2017): 1-15.

Conference proceedings

- **Fangliang Bai**, Stuart J. Gibson, Manuel J. Marques, and Adrian Podoleanu. "Superpixel guided active contour segmentation of retinal layers in OCT volumes." In 2nd Canterbury Conference on OCT with Emphasis on Broadband Optical Sources, International Society for Optics and Photonics, 2018.

Non peer-reviewed articles

- **Fangliang Bai**, Manuel J. Marques, and Stuart J. Gibson. "Cystoid macular edema segmentation of Optical Coherence Tomography images using fully convolutional neural networks and fully connected CRFs." arXiv preprint arXiv:1709.05324 (2017).

Chapter 1

Introduction

This thesis describes novel developments of computational methods for image acquisition and analysis. The research focuses on two themes: *computational imaging* and *ophthalmic image processing and interpretation.*

Computational imaging is a research field concerned with the generation of images from indirectly sensed data. Computational imaging refers to any system for which computation plays an integral role in the image formation process"[1] and is hence based on the integration of photon sensing hardware with computer algorithms. Conventional digital cameras utilise a 2D light-sensitive detector, typically comprising millions of photo-sites (pixels), that directly records an object scene. Conversely, a computational imaging system indirectly records a light intensity pattern, using a 2D, 1D or even single-pixel detector, which is then fed into a computational step(s) to recover an image of the object scene. For example, computational ghost imaging utilises a single photodiode to detect light intensity, magnetic resonance imaging (MRI) uses a radio antenna to receive nuclei radio waves. Computational imaging has enabled a wide range of novel imaging modalities, especially in a clinical setting, that allow the non-invasive visualisation of internal body structures that would not be possible through conventional camera technology.

[1] Definition of computational imaging stated in the scope of the IEEE Transactions of Computational Imaging

A highly successful application of computational imaging is optical coherence tomography (OCT) which has become the go-to imaging tool for ophthalmologists. Retinal OCT imaging is a technique that uses low-coherence light interferometry to visualise tissue underneath the retinal surface in the form of two and three-dimensional images. The retina's cross-sectional structure can be indirectly captured by low-coherence light directed through the patient's pupil and transformed into a recognisable image using algorithms. This allows the clinician to diagnose various retinopathy types by viewing the cell layers beneath the retinal surface tissue. Such diseases are prevalent with the projected number of people with age-related macular degeneration alone, affecting an estimated 196 million people worldwide in 2020. OCT has revolutionalised patient diagnosis and attracted about five-hundred million dollars investment in science research funding raised from taxpayers globally between 2006 and 2016 [4].

The widespread availability of fast computing technologies, developments in algorithms, and advanced sensing hardware have dramatically improved the efficiency of computational imaging modalities. Huang's law states that the performance of GPU cards more than doubles every two years compared with Moore's law that predicted the number of transistors in a dense integrated circuit (IC) approximately doubles every two years. GPUs have been developed as a high-performance data-parallel computing accelerator platform in recent years, enabling the deep learning revolution in computer vision and facilitating many solutions in the field of medical image processing. *High-resolution* computational tomography and MRI techniques are underpinned by graphics hardware acceleration which is used to implement computationally-intensive algorithms for image reconstruction. GPUs have also been used for a variety of ultrasound applications allowing *real-time* processing and visualisation of 2D and 3D data [5].

The majority of the work set out in this thesis is underpinned by compressive sensing (CS) and deep learning, prominent techniques in signal processing and computer vision, respectively. CS provides an extremely efficient mechanism for sensing image data. According to CS theory, a signal, typically of either one or two dimensions, can be reconstructed from fewer sampled measurements than predicted by the Nyquist sampling rate (hence the term compressive sensing) subject to certain caveats. This feature of CS enables imaging systems to be developed that overcome constraints due to optics and associated sensing hardware. The continued development of CS has led to a wide range of applications such as radar imaging [6], neural signal acquisition [7], sound field reconstruction [8], and dimensionality reduction [9].

The second computational method explored in this thesis is deep artificial neural networks. In computer vision applications, where training data is abundant, deep learning is the dominant tool (especially in classification tasks). Deep neural network technology has also made an impact on the OCT imaging modality over the last five years. More than $1,800$ research papers were published in this field since 2020, according to The Web of Science statistics. This success is largely due to the applicability of the *convolutional* neural network for medical imaging processing tasks.

The thesis is organised into eight chapters, comprising an introduction, a review chapter, five research chapters and a conclusion. Chapters 2 and 3 concentrate on computational imaging for a single-pixel camera (SPC) architecture and photonic time-stretched optical coherence tomography (PTS-OCT) system. We present the first comprehensive performance evaluation of selected CS algorithms for PTS-OCT and separately develop computational imaging software for a simple and accessible SPC imaging system (Chapter 2). A novel low-sample rate, high-resolution, deep neural network (LSHR-Net) is then presented (Chapter 3) as an alternative to the basic CS methodology for SPC imaging which is shown to improve reconstruction efficiency. In Chapter 4 OCT retinal imaging and its application to clinical ophthalmology are reviewed. The chapters that follow (5-7), describe in detail novel research in ophthalmic image analysis. Retinal layer boundaries were automatically segmented from OCT images using our segmentation framework, which consists of key components as self-adaptive curve, superpixel, and active contours (Chapter 5). Then separate deep neural networks were developed for precise segment cystoid macular oedemas using 1) transfer learning (Chapter 6) and 2) image retrieval-based medical image segmentation based on few-shot learning (Chapter 7).

The work covered in this thesis made several contributions to the research field:

1. We implemented a general control software and data sampling hardware module for our single-pixel camera and made this open-source so others can benefit from our implementation. We provide a systematic evaluation to guide researchers on how to choose the best algorithm for time-stretched signal reconstruction of the time-stretch OCT hardware[10].

2. The novel neural network-based image reconstruction model (LSHR) obtained a set of self-optimised binary sampling patterns, which were obtained from the model training. The binary patterns are ideally suited for use on single-pixel camera imaging hardware architectures, [11]. Our model also achieved a state-of-the-art reconstruction accuracy with overall PSNRs of 33.68 dB, 28.67 dB, and 22.11 dB at three different measure ratios on the benchmark dataset. Our network also had the best computational efficiency over other compared algorithms.

3. Our novel retinal OCT boundary detection framework achieved superior performance on very thin, tightly bunched layer segmentation [12]. The framework achieved full retinal layer segmentation in 12 layers. It is more than the maximum number of layers segmented in previous work.

4. We developed two deep-learning-based algorithms for medical image segmentation using small-scale datasets. We evaluated our models on retinal oedema segmentation in OCT images and melanoma segmentation in dermoscopy images. The transfer-learning based model avoided the need for ad-hoc rules and manual intervention. It performed an excellent segmentation on retinal images representing different stages of macular oedema and achieved an overall dice coefficient of 0.60 ± 0.26, which is better than the performance quoted in previous work. The meta-learning based model was designed to find the target features (contained within the reference image) in a database and segments these features in query images. Our model achieved an 83% dice coefficient accuracy on retinal OCT data and 89% on dermoscopy data.

Chapter 2

Evaluation of compressed signal sampling and reconstruction

2.1 Introduction

Compressed sensing (or compressive sensing, CS) is a popular technique that has attracted great interest in the field of signal sampling and reconstruction. The CS theory was developed on the principle that an under-sampled signal, measured using a set of suitable sensing patterns (typically comprising random pixel intensities), may contain enough information for accurate signal reconstruction. Since its invention, this technology has played a starring role in a wide range of applications in computer science, applied optics and electrical engineering, where a low signal sampling rate is often desired. In this chapter, we focused on two advanced CS applications: the single-pixel camera (SPC) for 2D image reconstruction and the photonic time stretch based optical coherence tomography (PTS-OCT) for 1D serial signal reconstruction. We present the principle of our frameworks and hardware design and evaluate the signal reconstruction results using our SPC control software[1] based on different reconstruction algorithms.

[1] The open source code is at https://github.com/FangliangBai/spc-control-software-repo

2.2 Compressed image sampling and reconstruction of SPC system

Single-pixel cameras and ghost imaging are hot topics in computational imaging. The SPC system uses sensing patterns to sample a target scene and applies optimisation algorithms to reconstruct an image of the scene from the sampled measurements. This new technique can offer an inexpensive alternative to conventional imaging sensors, especially when imaging outside the visible wavelength range. An SPC hardware setup comprises three key components: a random pattern generator, a single-pixel photodetector, and an analogue-to-digital converter. They perform the SPC imaging in concert according to the following procedure: first, the random pattern generator illuminates a set of sensing patterns (structured light) on a target scene. Then, for each pattern, the photodetector records an electric response of the light intensity reflected from the object. Lastly, the analogue-to-digital converter converts the electric responses to digital measurements. The correlations between the sensing patterns and their corresponding, reflected light are used to compute an estimate of the image scene. The implementation of this technique does not require the complex multiple-lens systems that are used in conventional cameras. Therefore it has the potential to replace the optical lens with a coded aperture assembly, as demonstrated in [13].

Hence two key components required for SPC image reconstruction are: the 2D sensing patterns and intensity measurements of the backscattered light. In the computational imaging paradigm [14], each measurement is determined as a inner product between a sensing pattern and the image scene. This can be formulated as:

$$y = \Phi x + e \tag{2.1}$$

where $x \in \mathbb{R}^n$ is the image (rearranged as a vector), $\Phi \in \mathbb{R}^{m \times n}$, $m \ll n$, are m sensing patterns (again rearranged as a vector), $e \in \mathbb{R}^m$ are measurement errors and $y \in \mathbb{R}^m$ are the measurements. In practice, the measurement is the voltage output from the photodetector, which approximates the inner product of the image scene and patterns. The projector produces light even when displaying fully black patterns (comprising pixels with zero intensity). Consequently, the photodetector records a voltage corresponding to the zero offset measurement. Taking that offset into consideration, the relationship between the voltage measurements and patterns is formulated as

$$V(m) \propto \langle x, \phi(m) \rangle + V_{\mathrm{DC}}(m) \tag{2.2}$$

where $V(m)$ is the voltage reading of mth pattern from the detector, x is the 2D test object we want to reconstruct, $\phi(m)$ is the mth measuring pattern, $\langle \cdot \rangle$ is a dot product operation. The $V_{\mathrm{DC}}(m)$ is the zero offset light intensity of the projector, which needs to be subtracted before the measurements resulting from each sensing pattern in the set. According to the conventional Shannon-Nyquist sampling theorem, a signal must be sampled at a rate at least twice its highest frequency in order to be correctly reconstructed [15]. However, when the signal is sparse or has a sparse representation on a transform basis, the CS-based algorithm can reconstruct it if the number of samples is less than the Nyquist limit provided. The number of sensing patterns, m, required by an SPC system can be far fewer than the total number of pixels n of the reconstructed image, which results in a compressive measurement ratio $R = \frac{m}{n}$.

The CS theory benefits our SPC system in terms of the efficiency of data compression. For a conventional imaging system, the intensity is measured directly at each pixel location. Typically, the image data is then compressed (e.g. JPEG compression scheme), at which point a large proportion of the measured image data is discarded. In contrast, using the SPC imaging technique, redundant information is not measured at the sampling step. Hence all of the measurements made using CS are used in image reconstruction, resulting in highly efficient image acquisition. In addition, the implementation of the SPC system does not rely on complex lens systems, which are often used in conventional cameras. Benefited from this feature, the SPC system has the potential to replace the optical lens with a coded aperture assembly, as demonstrated by G. Hang [13].

In this section, we present an SPC imaging system that was developed with the aim of performing sampling-efficient image reconstruction. This system comprised our own hardware setup and SPC control software which is easily adaptable to different SPC hardware configurations. We optimised the sampling process for our system to enhance the quality of the reconstructed image. Optimisation focused on two aspects:

- **Photodetector saturation calibration**: the photodetector was used to generate a linear voltage response according to the back-scattered light intensity. The amount of light intensity received by the photodetector depended on the distance between the object and the photodetector when other settings were fixed. Because the SPC hardware had an upper limit on voltage measurement, a saturation calibration process was done to optimise the best measurement distance.

- **Dynamic conversion of analogue-to-digital response range**: the light intensity response taken from the photodetector was converted from an analogue voltage to a digital value. The default analogue-to-digital conversion was based on the converter's full response range with a fixed resolution, i.e. a fixed resolution was applied between the minimum and maximum voltage responses. However, the measurements taken by the system only exist at the middle section of the range, which left the top and bottom part unused. Therefore, we applied a dynamic conversion process automatically to remove those unused voltage ranges such that the same resolution was applied to the actual response range. By doing this optimisation, a more accurate measurement was achieved. We convert the light intensity response of the photodetector from an analogue voltage to a digital value. Conventionally, this is done by mapping the voltage range (0-5 V) to the digital range (0-1023). The sensitivity of the system (the minimal detectable voltage change) is 0.0049 V. However, the real responses taken by the system only exists at the middle section of the range, e.g. 2-4 V. Therefore, we applied a dynamic conversion process automatically to remove the unused voltage range such that the real response range (2-4 V) can be mapped to digital range (0-1023). By doing this optimisation, we increased the system sensitivity to 0.0020 V and a more accurate measurement was achieved.

Our imaging system contributes to the research community with the software tools for testing and validating novel compressive sensing algorithms. Our hardware design can provide a cost-effective educational tool for communicating the principles behind the compressive sensing technique. Our imaging system also provides us a fundamental platform for our own work of compressive sampling and reconstruction using a neural network approach, which will be described in Chapter 3.

In the remainder of this section, the development of the SPC hardware and reconstruction software are described in Section 2.2.1 and 2.2.3. At the end of this section, we demonstrate the imaging procedure and image reconstruction results.

2.2.1 Hardware design

Our SPC imaging system consists of readily adaptable sampling hardware that produces sets of sensing patterns for projection onto the image scene and records light intensity measurements. The setup diagram is illustrated in Figure 2.1. In our system, we use a projector (of the type typically used to make presentations) to project the pre-generated patterns. The projector provides the same functionality as a digital micro-mirror device (DMD). But it does not require setting up an extra professional optical table layout, which is a more cost-efficient approach. The light intensity is measured by our single-pixel photodetector, which consists of a photodetector (a silicon photo-diode), a purposely designed trans-impedance amplifier, and an Arduino micro-controller. It implements a linear conversion from the light intensity to the digital measurements.

The trans-impedance amplifier converts the current due to light incident upon the photo-diode into a voltage, which is then digitised by the Arduino. In this intermediate process, measuring accurate voltage is the primary factor that affects image quality. There are two noise sources that affect the measurement accuracy: the circuit noise and the amplifier noise. To ensure a high signal-to-noise ratio for our measurements, we optimised the system by suppressing the noise from those two sources.

First, we applied voltage smoothing to suppress circuit noise. We believe a significant component of this noise was due to electromagnetic radiation (outside the visible range) emitted from the projector. When the system took measurements, the voltage signal generated by the circuit noise was always mixed with the voltage signal generated by the light incident on the photo-diode. An example in Figure 2.2 shows that the true voltage signal presented by an oscilloscope is highly affected by the circuit noise. It is seen that the voltage converted from the photo-diode current (red line) was periodically distorted by spike noise. Using a different projector may have resulted in a different peak noise intensity. To ensure

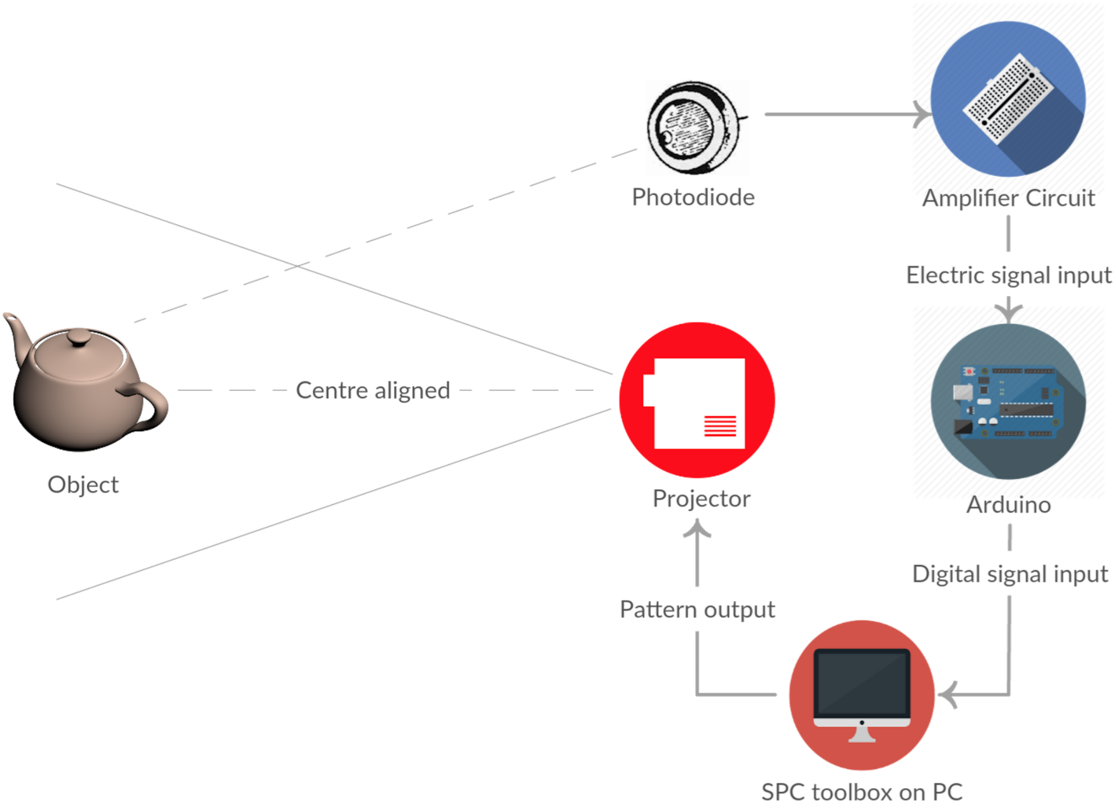


FIGURE 2.1: The diagram of SPC hardware setup. The hardware consists of a projector and a photodetector (photodiode, amplifier circuit, micro-controller). The hardware is controlled by our SPC toolbox software on a PC.

the voltage is stable over time, we smoothed out the noise signal by taking an average value over a number of measurements.

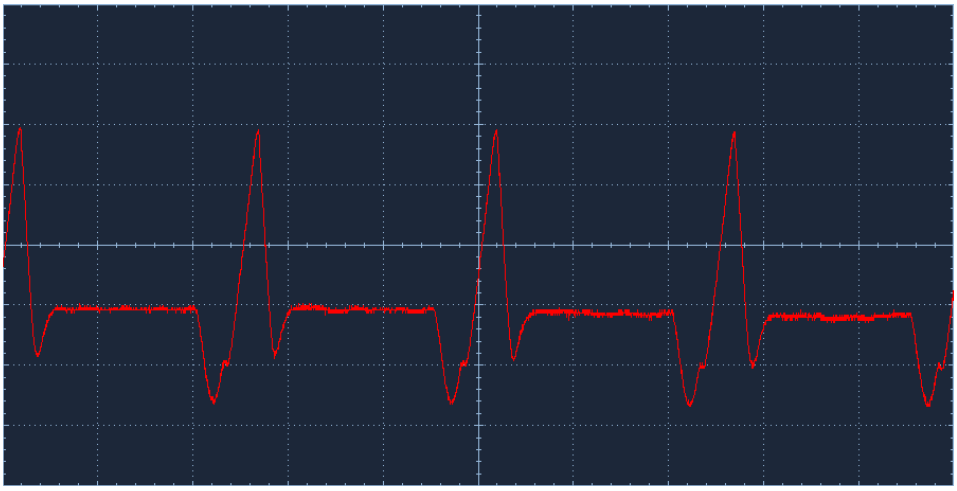


FIGURE 2.2: The circuit noise. The red line is the voltage of the photo-diode captured by an oscilloscope. The spike noise that appears periodically is the circuit noise introduced by the projector and the SPC circuit. This noise affects the measurements of real voltage values from the photo-diode.

Second, we optimised the parameter setting of the digital potentiometer. The original current signal generated by the photo-diode was very weak. When we convert the current signal to the voltage signal, we use a trans-impedance amplifier to add an additional gain to the voltage output. This gain was generated by an amplifier with an adjustable digital potentiometer. Figure 2.3 shows the current amplifier used in the SPC setup.

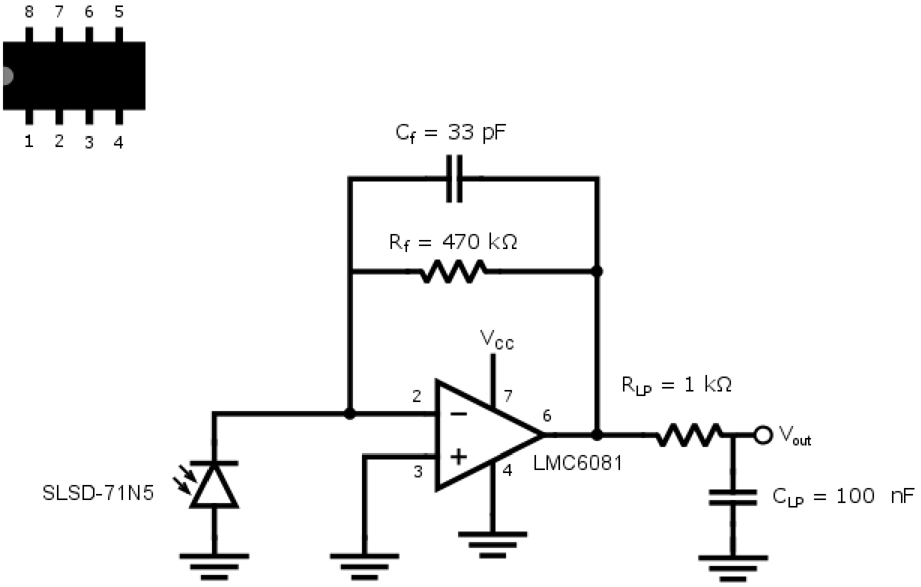


FIGURE 2.3: The design diagram of the potentiometer.

The digital potentiometer can generate different voltage gains by changing the value of the feedback resistor, denoted as R_f. The output voltage is proportional to the feedback resistor when we have stable current signals. This is formulated as

$$V_{\text{out}} = I_{\text{in}} \times R_{\text{feedback}} \tag{2.3}$$

where I_{in} is the input current generated by the photo-diode and V_{out} is the output voltage of the amplifier circuit. In principle, the higher the gain, the easier it was to accurately measure the voltage. In practice, however, when a high voltage was generated, this boosted the noise as well as the signal. Therefore, to get the best gain for our system, we aim for the smallest noise-to-gain ratio (noise over gain voltage, NGR) of the output voltage. Figure 2.4 shows the NGR values of the voltage outputs at 250 resistor scales, from minimum to maximum resistance values. Since the noise was not proportional to the voltage, it is seen that the higher resistance generated smaller NGR until scale 178 of the potentiometer. From here on, a significant converse trend is observed where the output of the amplifier circuit is no longer stable. To keep the NGR as small as possible and keep the amplifier stable, we set the feedback resistor scale to 170 for the digital potentiometer.

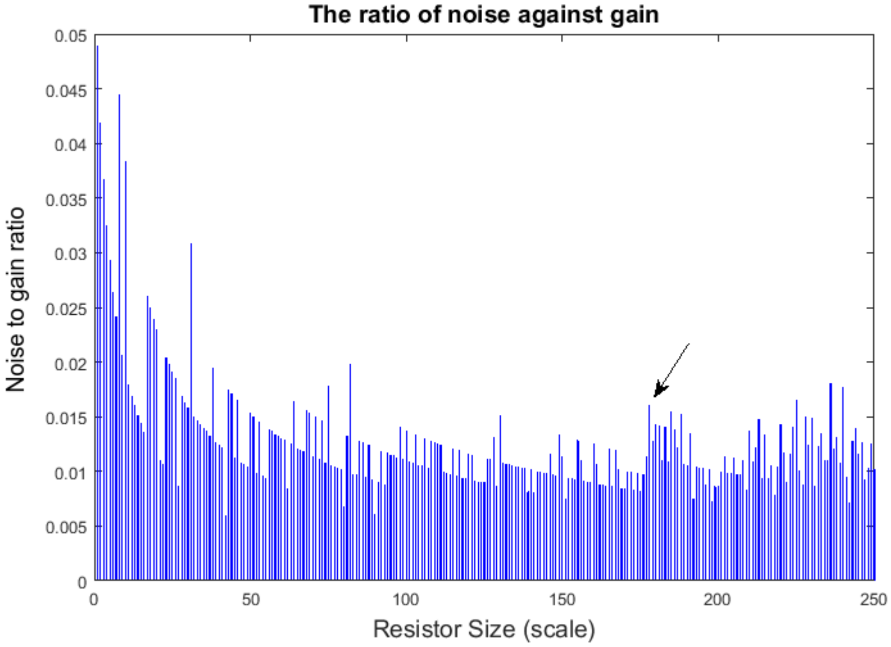


FIGURE 2.4: The ratio of noise against potentiometer gain. The potentiometer gain has 250 scales. It is seen that the noise-gain ratio decreases quickly from scale 0 to 170. After that (the arrow point) the ratio increases and voltage gain is not stable. Therefore we selected 170 as our best gain scale.

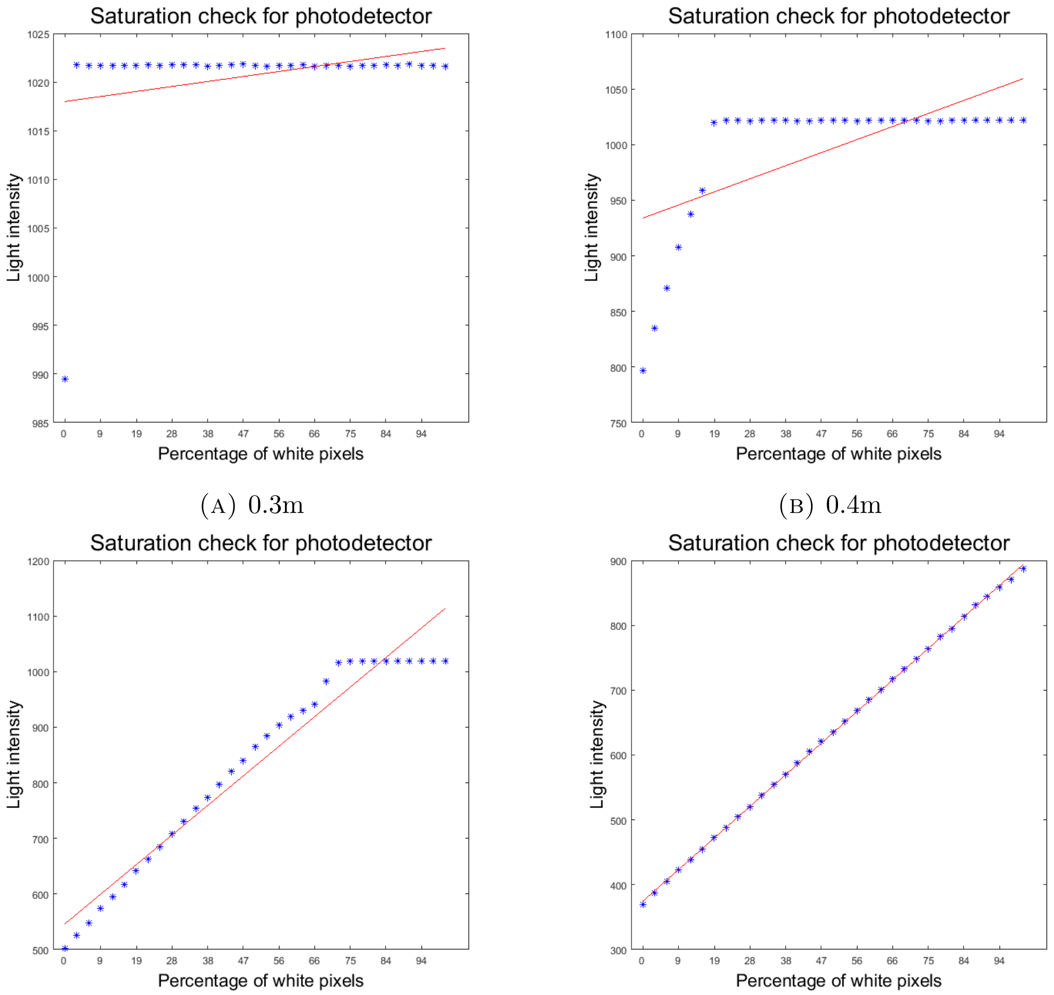
2.2.2 Saturation calibration and dynamic range conversion

The saturation calibration is a prerequisite step to ensure our system takes valid measurements during the sampling process. The maximum light intensity our photodetector can respond to is constrained by its ADC response range. When the photodetector is measuring the back-scattered light from the image scene, the light intensity recorded by the photodetector may be affected if the distance between itself and the image scene changes. If the photodetector is set too close to the image scene, the voltage signal generated from the photodetector may become saturated.

To find the optimal distance, we first used our control software to check the measure the saturation by projecting a series of sensing patterns, composed of black and white grid cells, onto a white screen and measured the back-scattered light intensity. In the sequence of patterns, the white cells were randomly placed on the grid, and the number of them was gradually increased from the first pattern (entirely black) to the last one (fully white), i.e. 0% to 100%. The measurements of the sequence of patterns were fitted by linear regression, and the result was used to determine the best distance between the camera and the image scene. Figure 2.5 shows the detector readings (in ADC units) against proportion of white pixels at four separate distance, 30cm, 40cm, 70cm, and 97cm. It is seen that the detector was completely saturated at 30cm since the measurement was fixed at the upper boundary of value 1024. By increasing the distance, the measurements fell into the valid range. Eventually, it is seen in Figure 2.5d that all the measurements were recorded in the valid range at a distance of 97cm. In addition, it is also seen that there is an offset voltage value, denoting the minimum measurement when the black screen was displayed. This value was removed by the dynamic range conversion before doing image reconstruction.

The dynamic range mapping is the second process, after the saturation calibration, to increase the measurement accuracy. The analogue-to-digital converter was the essential function of Arduino that was used to map voltage response into a digital signal. The Arduino on-board analogue pin supported an input range from 0 to 5 volts. The converter mapped this range into 0 to 1023 digit by default since its 10-bit bandwidth was able to provide a 1024 resolution at maximum. As a result, the default mapping resolution was 4.88 mV per digit.

However, the response range of the random patterns only covered the middle part of the entire voltage range since the offset voltage and the upper margin of the saturation calibration. This led to a condition that the bandwidth for converting the full voltage range was taken by both of the useful voltage range and two margins, seen Figure 2.6. To take a more accurate measurement, the converter was optimised by using a dynamic range mapping such that bandwidth was only used for the pattern response range, i.e. the range from the offset voltage to maximum pattern voltage was converted to 0 to 1023. For example, in Figure 2.5d, the mapping resolution after dynamic range mapping is 2.50 mV per digit (4.88 mV per digit without dynamic range mapping), covering from 1.76 V to 4.34 V. To remap the ADC range, first, the upper margins were removed by resetting the highest SPC voltage as the maximum reference voltage. Then an operational amplifier was used to subtract the baseline voltage so that the ADC input voltage always started from zero. Figure 2.7 shows the detailed procedure.



(A) 0.3m

(B) 0.4m

(C) 0.7m

(D) 0.97m

FIGURE 2.5: Linearity correction of photodetector output for finding its optimum position. Placing photodetector at different distances from the object reflects different results. A linear relation shown in plot (d) is the most expected.

After the saturation calibration and the dynamic range mapping, we used the SPC control software to do the image sampling and reconstruction.

2.2.3 SPC control software

The SPC control software is developed as a Matlab toolbox to control image sampling and reconstruction. The SPC imaging workflow is summarised in Figure 2.8. SPC imaging is achieved by three essential functions in our software: 1) random pattern projection, 2) measurement recording, and 3) image reconstruction,

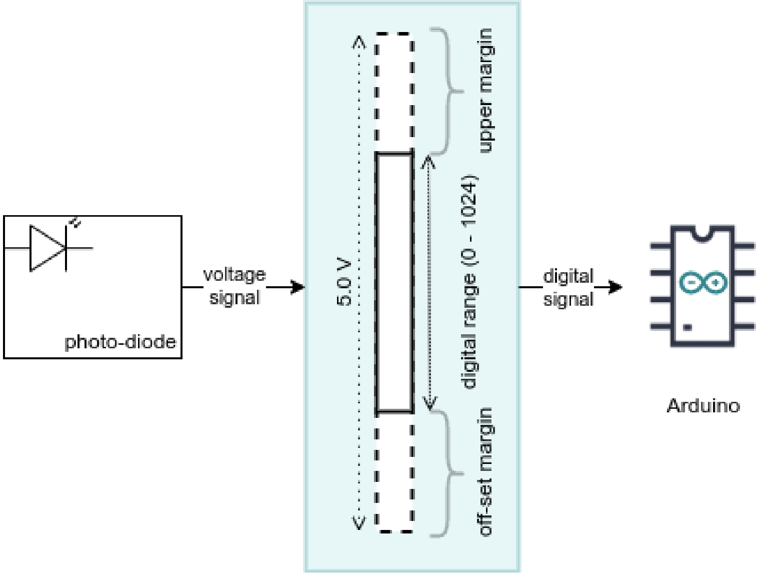


Figure 2.6: The dynamic range mapping diagram. The voltage signal was mapped to digital value thought this dynamic range mapping function such that only the valid voltage range are mapped to the digital range. As a result, a more precise measurement can be recorded compared to measurements without dynamic range mapping.

which are described in detail in this section. The interface of the software is shown in Figure 2.9.

In the sampling stage, the software controlled the projector to display the sequence of sampling patterns. It then triggers a light intensity measurement from the photo-diode to record the reflected light intensity associated with each sensing pattern. The software synchronised the projector with the photo-diode readings. Our software generated a choice of three types of sensing patterns: random Bernoulli matrix, Bernoulli fixed matrix, and raster scan matrix (each matrix containing a single white pixel).

In the reconstruction stage, the software took both the recorded measurements and the previously generated patterns to reconstruct the image using the chosen optimisation algorithm. The software offered four image reconstruction algorithms with associated constraints and sparse basis. Of these four algorithms, the L_2 minimisation, also known as the least-squares method, provides a conventional solution to the image reconstruction problem, which is fast and requires no predefined constraints. The reconstructed image x is calculated by

$$x = \left(\Phi^T \Phi\right)^{-1} \Phi^T y \tag{2.4}$$

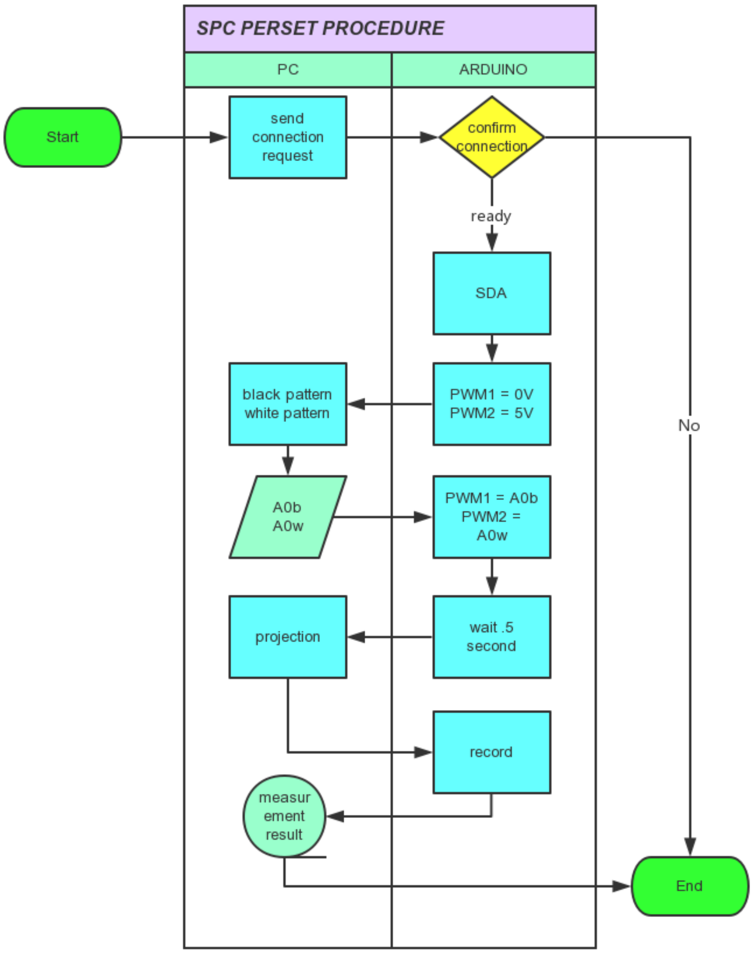


FIGURE 2.7: The measurement procedure of the single-pixel camera hardware. The preset configuration is defined by a PC and set in Arduino via serial data pin (SDA). The voltage mapping range is set by two analogue output (PWM) of the Arduino. After the mapping is done, the projection can be conducted.

where y is the measurements, Φ is the sampling patterns. Since it was not based on the CS theory, the number of the sampling patterns needed when using this algorithm theoretically should be greater than the number of image pixels in the reconstructed image. The L_2 minimisation was developed as a baseline method, which was used to compare with the other three CS-based algorithms. We implemented three CS-based reconstruction algorithms, the L_1 *minimisation*, the *total variation minimisation*, and the *orthogonal matching pursuit*. The L_1 minimisation and total variation minimisation (TV minimisation) were implemented using the L_1 magic package [16]. For the L_1 minimisation, we formulated the signal

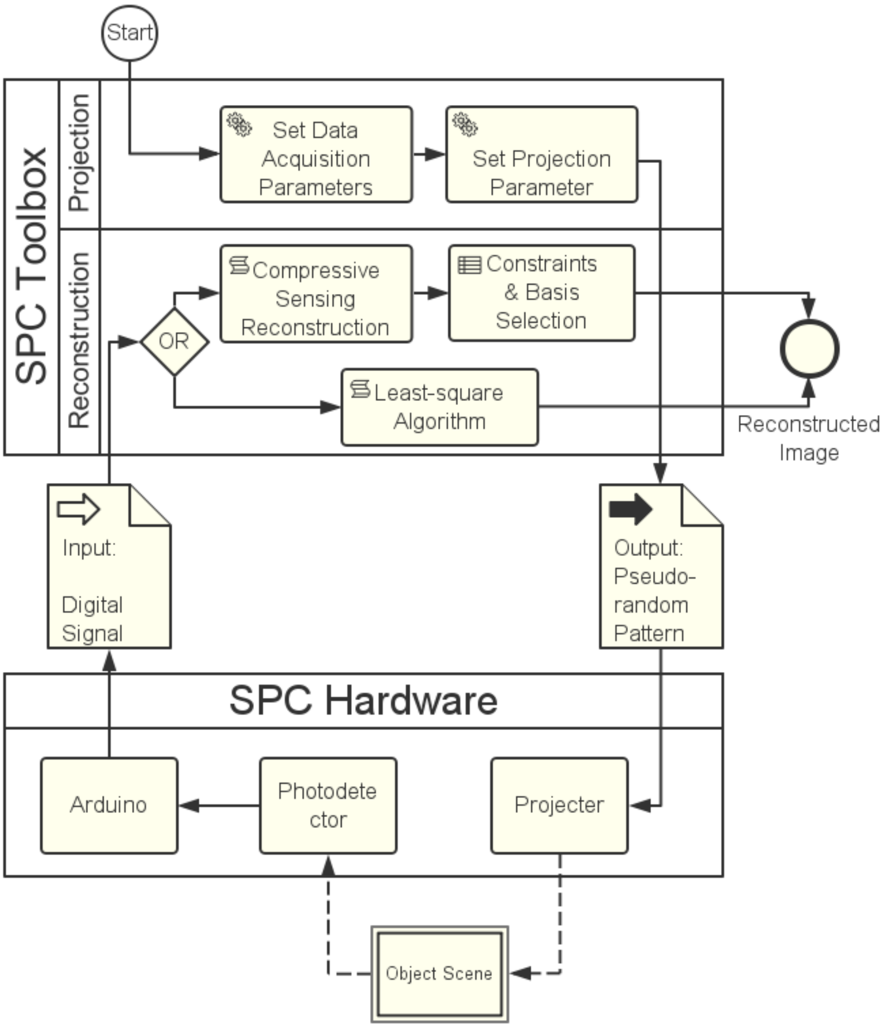


FIGURE 2.8: The architecture and measurement procedure of the single-pixel camera software (SPC toolbox).

reconstruction problem as

$$\min \|x\|_1 \quad \text{subject to} \quad \Phi x = y, \tag{2.5}$$

where we keep the same notation as Equation 2.4. For the total variation minimisation, we used the image gradient as target object and we formulate the problem with equal constrain as

$$\min \mathrm{TV}(x) \quad \text{subject to} \quad \phi x = y \tag{2.6}$$

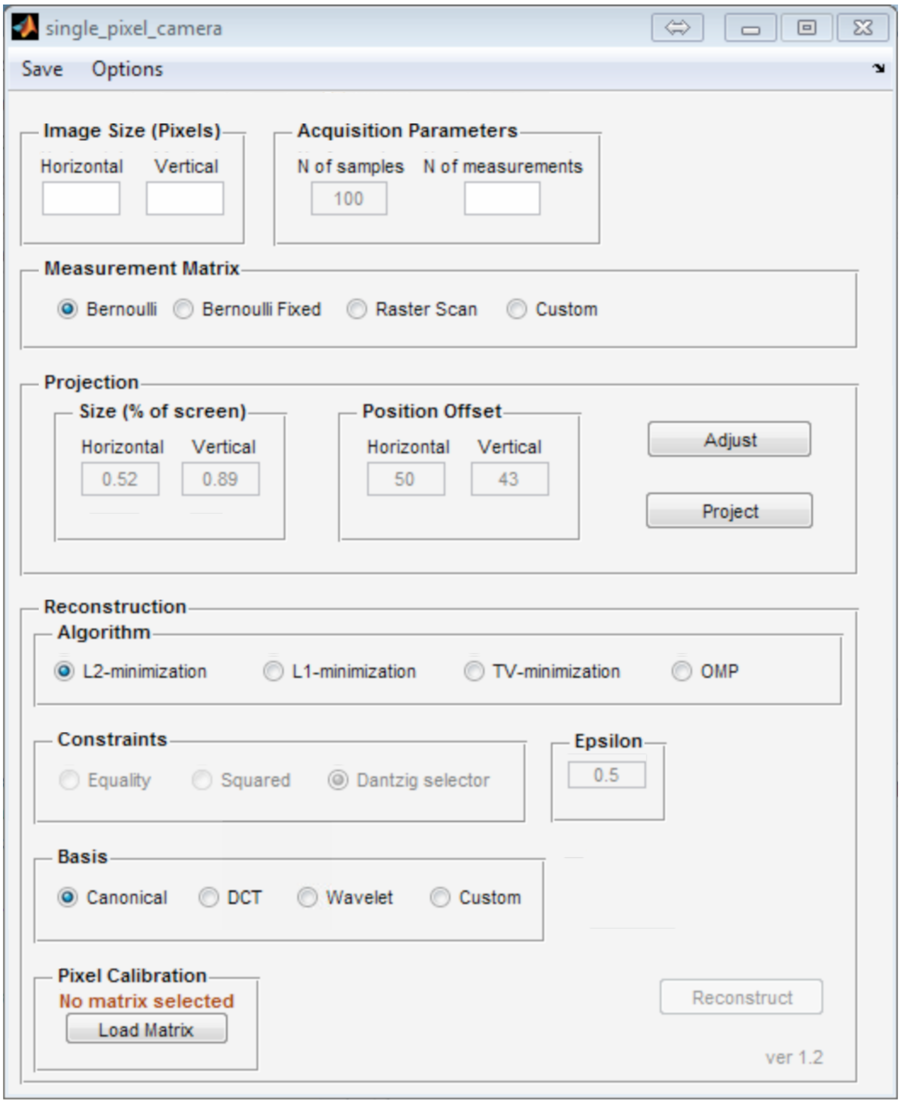


Image Size (Pixels)
Horizontal:
Vertical:

Acquisition Parameters
N of samples: 100
N of measurements:

Measurement Matrix
● Bernoulli ○ Bernoulli Fixed ○ Raster Scan ○ Custom

Projection
Size (% of screen) — Horizontal: 0.52 Vertical: 0.89
Position Offset — Horizontal: 50 Vertical: 43

Reconstruction
Algorithm: ● L2-minimization ○ L1-minimization ○ TV-minimization ○ OMP
Constraints: ○ Equality ○ Squared ● Dantzig selector
Epsilon: 0.5
Basis: ● Canonical ○ DCT ○ Wavelet ○ Custom
Pixel Calibration: No matrix selected

FIGURE 2.9: The UI of SPC toolbox, designed using Matlab GUI interface.

where TV(x) is the total variation of image x and it is the sum of the discrete gradient magnitudes at every pixel, which is formulated as

$$\mathrm{TV}(x) := \sum_{ij} \sqrt{(D_{h;ij}x)^2 + (D_{v;ij}x)^2} = \sum_{ij} \|D_{ij}x\|_2 \tag{2.7}$$

where

$$D_{ij}x = \begin{pmatrix} D_{h;ij}x \\ D_{v;ij}x \end{pmatrix} \tag{2.8}$$

and

$$D_{h;ij}x = \begin{cases} x_{i+1,j} - x_{ij} & i < n \\ 0 & i = n \end{cases} \quad D_{v;ij}x = \begin{cases} x_{i,j+1} - x_{ij} & j < n \\ 0 & j = n \end{cases}, \tag{2.9}$$

where x_{ij} denote the pixel in the ith row and jth column of an $n \times n$ image x. Both of them require a specific optimisation constraint and an error tolerance, which is denoted as Epsilon in Figure 2.9. The error tolerance value was generally set to a small positive value. This parameter determines the number of iterations needed for optimisation. The orthogonal matching pursuit (OMP) is a CS-based reconstruction algorithm designed with an equality constraint. In our software, we also used the L_1 object function in Equation 2.5. It recovers the signal from an under-determined matrix system by minimising the reconstruction error at each iteration. Just like L_1 minimisation, OMP also requires an error tolerance to be defined in advance.

2.2.4 Experimental results

In this section, we provide examples illustrating sampling and reconstruction of a target image using each of the four algorithms mentioned above. Here, the target image is a 2D greyscale image, as shown in Figure 2.10. When conducting the imaging process, the image was sampled in a dark room to avoid artefacts and noise from any other irrelevant light sources. Setting up the system entailed positioning the target image and the projector such that the sensing patterns covered the whole image scene. The detector was also placed facing the image scene, followed by the saturation calibration and dynamic range mapping, previously described (Section 2.2.2).

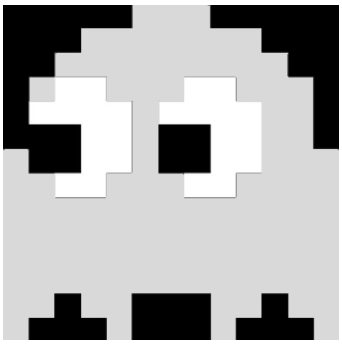
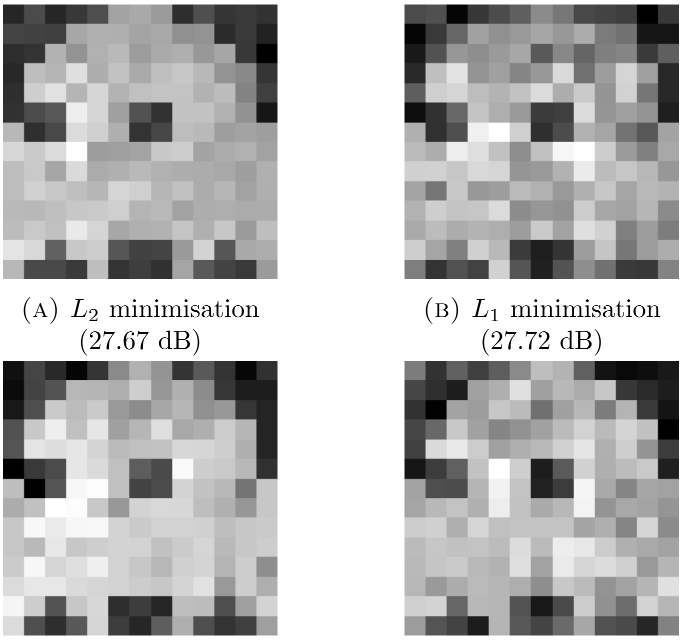


FIGURE 2.10: The target image of a pac-man doodle.

Next, we set sampling variables of the toolbox: the image size of the image was 14 × 13 pixels; the number of measurements was set to 91 (measurement ratio R = 50%); and the Bernoulli Matrix was selected as the pseudo-random pattern. After that, we started the image sampling procedure. When the image sampling



(A) L_2 minimisation (27.67 dB)

(B) L_1 minimisation (27.72 dB)

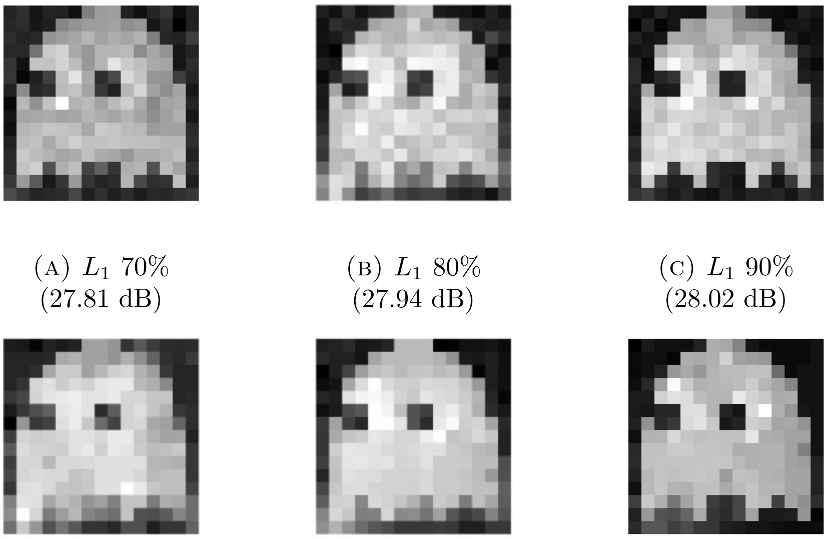
(C) TV minimisation (27.80 dB)

(D) OMP (27.44 dB)

FIGURE 2.11: An example set of reconstruction results of the same object using four methods separately. The measurements was obtained under the same environment condition. The quantitative evaluation of each reconstructed image was measured with PSNR value comparing to the original image.

was finished, we used the four algorithms configured with the same constraint and basis options to reconstruct the image. Figure 2.11 shows the reconstruction results by those four methods. It is seen that, among three CS-based algorithms, the TV minimisation yielded better results in terms of visual quality and measurement metrics. This is because the TV minimisation algorithm considered the image smoothness during the reconstruction. It is also worth noting that the L_2 minimisation results are visually better than the L_1 minimisation results, although the latter one yields a slightly higher metric value. The L_2 objective function aims at reconstructing the image that has a minimum average difference. Therefore, its results have lower contrast. The L_1 objective function aims for sparsity and hence more high-contrast pixels is observed.

We further evaluated the quality of the CS-based reconstructed image at different measurement ratios. Due to the compression factor, the CS-based reconstruction algorithm's performance is determined by the number of sampling patterns. The more sampling patterns we use, the better the reconstructed image quality is. We took the image sampling at measurement ratio R = 70%, 80%, 90% to see the



(a) L_1 70% (27.81 dB)

(b) L_1 80% (27.94 dB)

(c) L_1 90% (28.02 dB)

(d) TV 70% (27.84 dB)

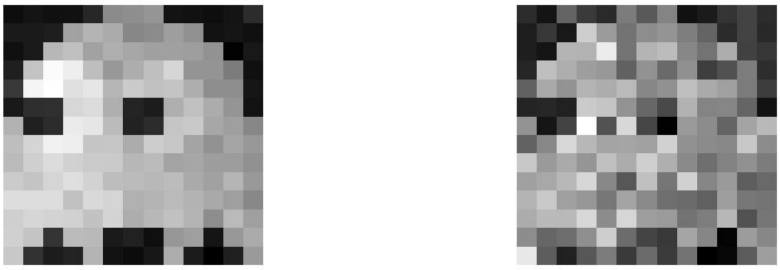
(e) TV 80% (28.01 dB)

(f) TV 90% (28.05 dB)

Figure 2.12: An example set of reconstruction results of the same object using CS-based algorithms. The measurements was obtained under three measurement ratios. The quantitative evaluation of each reconstructed image was measured with PSNR value comparing to the original image.

reconstruction quality. Figure 2.12 shows the comparison of the image reconstruction two CS-based reconstruction algorithm L_1 and TV minimisation algorithm. It is seen from the results that the image quality is positively related to the measurement ratio. We can also notice that the results of VT minimisation are smoother than the results of L_1 minimisation. This is mainly due to the difference between two object functions: the L_1 minimisation optimise the reconstruction to increase the sparsity of the image; The VT minimisation is to decrease the image gradient.

In addition, we demonstrate the results of image quality improvement by applying the dynamic range mapping. We conducted image sampling with and without dynamic range mapping under the same device layout. Then we selected the TV minimisation algorithm to reconstruct the images at measurement ratio $R = 85\%$. Note that this measurement ratio and setup layout of the photodetector and the projector is different from the setup for results in Figure 2.11. Therefore the quality of the result is not directly comparable. The results are shown in Figure 2.13. It is seen that the results generated are far more accurate, in terms of the contrast and grayscale level when the dynamic range mapping is applied. Dynamic range mapping enhanced sensitivity such that the system can detect



(A) Dynamic range mapping (27.88 dB)

(B) Original range (25.71 dB)

FIGURE 2.13: Results of the sample object, reconstructed with dynamic voltage mapping (A), and without it (B).

small changes in light intensity, and hence the measurements were recorded more accurately.

2.3 Compressed sampling and reconstruction of PTS-OCT optical signal

PTS-OCT system is a high-speed imaging system that captures an object's internal structure in the longitudinal section. The PTS technique, operating at a shorter wavelength range, offered the OCT system a better resolution in the axial direction (the direction of the light beam) and less water absorption in biological samples. Therefore, it can be used to capture the details of structural information over a long depth range. Although the PTS technique enables the high-resolution OCT measurement, the PTS instrument inherently produces the scan at an extremely high axial rate, at nearly 100 MHz. This deluge of OCT image data can overwhelm even the most advanced data acquisition circuits and the back-end, digital signal processors. To solve this sampling rate issue, the CS technique has previously been applied to OCT systems [17–19]. This application shows that a signal, sparse in the Fourier domain, such as a time-encoded OCT signal, can be recovered from a reduced number of measurements recorded by a single-pixel receiver scheme such as the PTS-OCT system. Therefore CS is a useful tool for compressing PTS-OCT high volume data streams.

Despite extensive studies on applying CS approach in various OCT systems, very little research work on data-compressed PTS-OCT has been reported so far, especially considering the fact that PTS-OCT suffers much more from massive data issues due to its high-throughput nature. To provide a better understanding of the proposed approach, we present an experimental verification and a comprehensive analysis of the reconstruction algorithms in this section. A number of optimisation algorithms for the reconstruction of the OCT signals were evaluated in terms of frequency reconstruction accuracy and efficiency. This will provide useful information in the selection of appropriate algorithms for this particular PTS-OCT scheme. As a result, we demonstrated that a data measurement ratio of 66% (a lower value is better) had been achieved in high throughput OCT measurements using a significantly reduced data sampling rate of 50 mega samples per second.

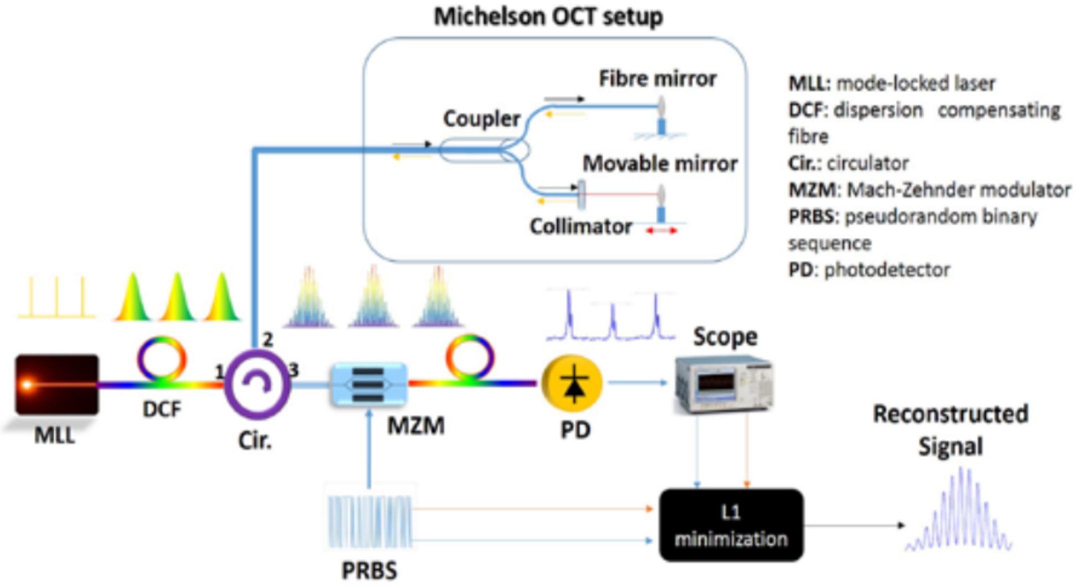


Figure 2.14: Block diagram of the proposed compressive sensing PTS-OCT system. The mode-locked laser produces ultra-short optical pulses, which are dispersed (stretched). The dispersed signal is then split by the coupler of the Michelson OCT system into a fixed mirror and movable mirror, which acts as a single layer object. The back-reflected interfered signal was captured by the circulator and then modulated with the PRBS patterns. Finally, the modulated signal was integrated by the single-mode fibre and measured by the photodetector. In the stage of signal reconstruction, we use the previous PRBS patterns and measurements of the photodetector to recover the original signal.

Our compressive sensing PTS prototype is presented in Figure 2.14. The mode-locked laser first generates a sequence of ultra-short optical pulses as input signals to the PTS system. The optical pulse is first stretched by the dispersion compensating fibre (the dashed line in Figure 2.15) and then sent to the Michelson OCT setup to sample the target object. The OCT's output signal is an interference pattern generated by the backscattered light of the fibre mirror and movable mirror. The interference signal encoded structure information of the target object in the frequency domain, which is shown as the solid line in Figure 2.15. Thereafter, the output signal is modulated with a series of pseudo-random bit sequences (PRBS, a vector which has a probability of 0.5 of being zero) by using a Mach-Zehnder modulator. This modulation process is demonstrated in Figure 2.16 (a-c). Then the modulated signal, 2.16 (c), is integrated by a length of single-mode fibre (SMF) to generate a single optical pulse, 2.16 (c), which is then recorded by the photodetector.

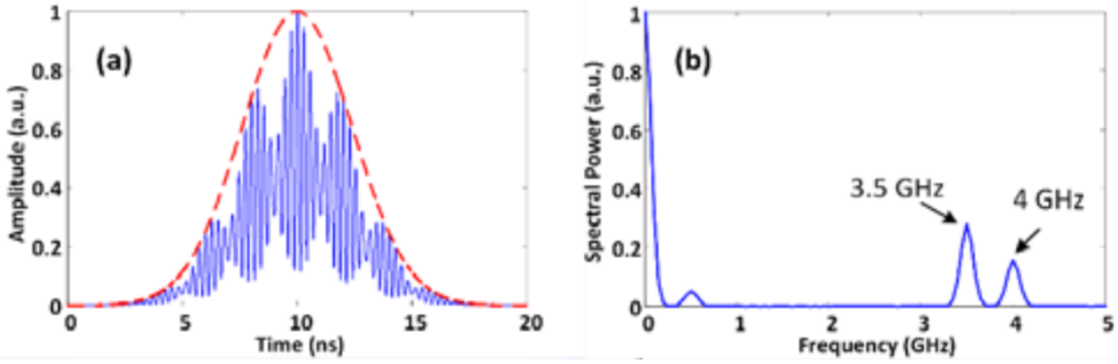


FIGURE 2.15: Interference pattern diagram. (a) Temporal interference pattern as a result of the path length difference. The time-stretched original pulse is shown in the red dotted line. (b) The spectrum profile of the optical interference pattern clearly showing two carrier frequencies of 3.5 GHz and 4GHz.

Our compressive sensing PTS method implements the signal compression in three successive steps: 1) signal modulation with pseudo-random patterns, 2) signal integration (converting a spectral signal to a single value response), and 3) low-rate response recording. Then, the original signal can be reconstructed from these response measurements by using a CS-based reconstruction algorithm. In this approach, Our proposed method overcomes the bottleneck of intensive data-sampling problems [20]. It also allows very low speed (50MHz) detectors to be used as an economical alternative to high-speed PTS-OCT data acquisition.

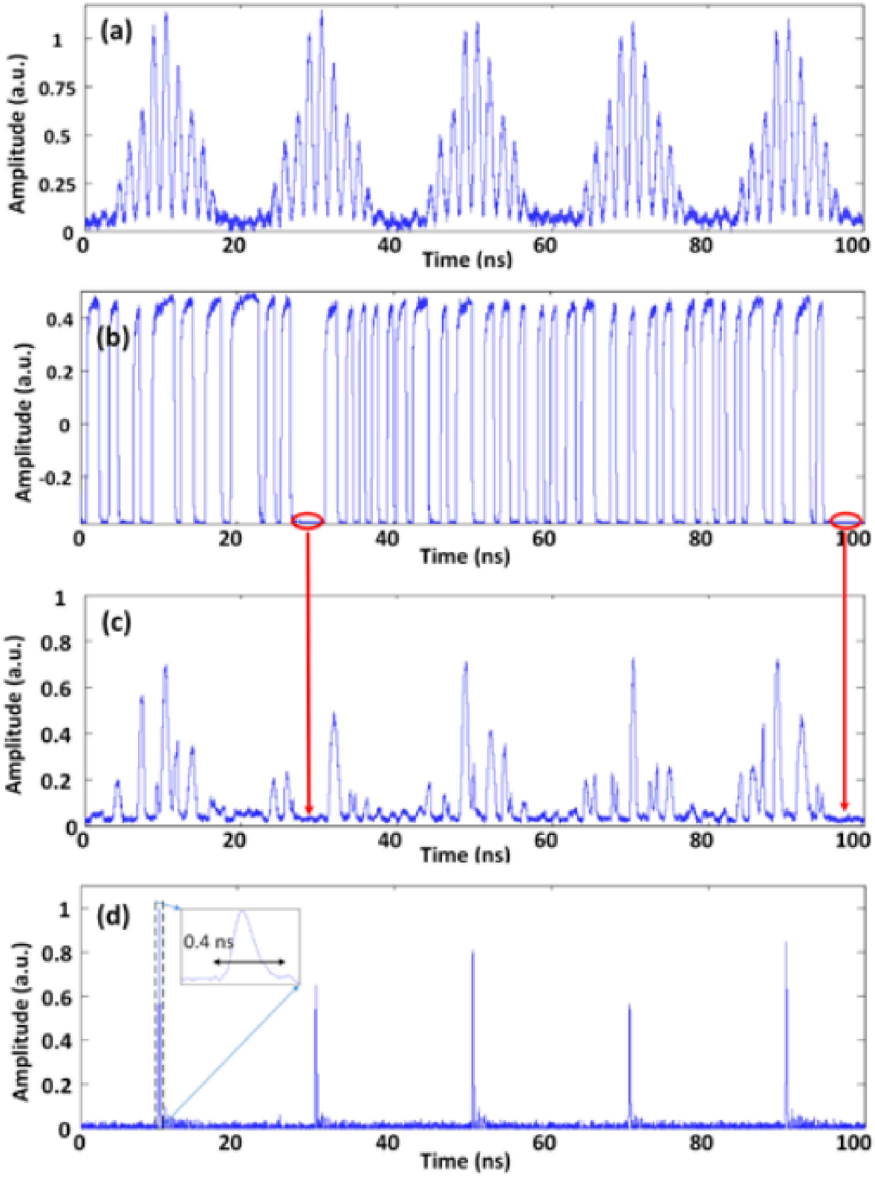


FIGURE 2.16: The optical signal processing of our single-layer PTS-OCT measurement. (a) The temporal interference pattern for five successive pulses. (b) The first 5 PRBS patterns. (c) The modulated waveforms with red marks showing no pattern for the exact amount of duration of a bit 0. (d) The compressed optical pulses using an SMF with an opposite dispersion profile. The peak power of compressed pulses produces the measurements.

2.3.1 Evaluation of signal reconstruction for PTS-OCT

Assuming that the time-stretched optical pulse y, which is sampled with N data points, is sparse in the Discrete Fourier Transform (DFT) domain $\Psi_{N\times N}$, the sparse signal s in DFT domain can be represented as,

$$s_{N\times 1} = \Psi_{N\times N} \times y_{N\times 1} \tag{2.10}$$

When the signal vector y is modulated with a set of PRBS sequence $\Phi_{m\times N}$, each randomly-mixed optical pulse is integrated into a single optical pulse by doing an inner product and recorded by the photodetector. This procedure generates a down-sampled $(m \times 1)$ measurement vector z, which can be represented as,

$$z_{m\times 1} = A_{m\times N} \times s_{N\times 1} \tag{2.11}$$

where $A = \Phi\Psi^{-1}$ and the measurement ratio is defined as m/N. Hence each measurement is the sum of approximately 50% of the points on the fully sampled signal. The measurements z can be obtained by taking the optical power of each compressed pulse using a low-speed photodetector. Finally, the reconstruction of the DFT domain signal from down-sampled measurements is achieved using the measurements and corresponding PRBS patterns as the inputs to a sparsity minimisation program. By solving the minimisation problem in Equation 2.12, the algorithms result in a sparse solution s, which represented the depth profile of the imaged object.

$$\min\left(\|s\|_1\right) \quad \text{subject to} \quad z = As \tag{2.12}$$

To achieve the best performance of this prototype PTS-OCT system, we evaluated five sparsity-promoting algorithms with the aim of achieving high accuracy of the PST-OCT reconstructed signal and reconstruction efficiency. They are 1) primal-dual interior point method (L1 Magic), 2) alternating direction method for multipliers for basis pursuit (ADMM BP), 3) lasso (ADMM Lasso) [21], 4) lasso method using coordinate descent (Matlab Lasso) and its standardised version [22], and 5) Nesterov's algorithm method (NESTA) [23].

2.3.2 Evaluation metrics

To evaluate the accuracy of the reconstructed signal structure and signal intensity, we compared the reconstructed signals to the original signals in the frequency domain. We used two metrics to calculate the reconstruction error: 1) the frequency reconstruction error and 2) the root mean square error. The frequency reconstruction error is defined as

$$\mathrm{RFE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(\mathrm{Idx}\left(s_i\right) - \mathrm{Idx}\left(\bar{s}_i\right)\right)^2} \tag{2.13}$$

where s_i and $\bar{s}_i$ are the local maximum of actual signal and reconstructed signal respectively, Idx is the index of the peak, and m is the number of peaks. Knowing the internal structure of the test object in advance is, of course, essential for evaluating the reconstruction accuracy of the CS PTS-OCT method. The frequency reconstruction error was used to quantify the total amount of structural discrepancy between reconstructed frequency components and the ground truth. We also considered the root mean square error:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(s_i - \bar{s}_i\right)^2}, \qquad i = 1, 2, \ldots, N \tag{2.14}$$

which indicates the accumulated energy error in all frequency components. Since the actual signal had a sparse structure, it only contained zeros or significant values at each frequency component. Based on these characteristics, a threshold was applied to the reconstructed signal to force the trivial reconstruction noise down to zeros before calculating the metrics.

2.3.3 Reconstruction accuracy analysis

We conducted our experiments with a four-layer test object. Each layer was detected by the presence of a peak in a spectrum where peaks at high frequencies correspond to deep layers embedded within the object. Therefore, the original PTS-OCT signal for our object had four peaks at frequencies of 2.6 GHz, 2.9 GHz, 3.6 GHz and 4.6 GHz (seen peaks of the red line in Figure 2.17). When our system received the reflected signal from the test object, it was modulated by a sequence of

PRBS vectors. The output is a sequence of measurements that we used thereafter for signal reconstruction.

For a given measurement ratio, the accuracy that our evaluated algorithms achieved varied. In Figure 2.17, we present our results illustrating the quality of the reconstructed spectra due to each algorithm at the measurement ratio of 40% (80/200 measurements). In this figure, the red lines correspond to the original spectra, and the blue lines represent the reconstructed spectra. It is seen that the NESTA (top left) and L1 Magic (top right) achieved good results, with the structure for all peaks being reconstructed. The ADMM Basis pursuit (middle left) and ADMM lasso (middle right) reconstructed a less noisy signal, but the energy at the frequency bands of interest is not fully reconstructed, which results in a less accurate result than that of NESTA and L1 Magic. Although the Matlab lasso (bottom left and right) produced less-noisy results than the previous four results, we observed a large frequency drift at the weakest frequency peak at 4.6 GHz.

To evaluate the reconstruction accuracy, we selected 31 subsets comprising measurements randomly sampled from the complete set of 200. Each of the subsets contained a different number of measurements. The smallest subset contained 50 measurements (equating to a measurement ratio of 25%), and the largest comprised all 200 measurements (measurement 100%). We applied the evaluated algorithms to reconstruct the signals for each subset and calculated the corresponding RMSE and RFE.

The RMSE error values of the reconstructed signals in the frequency domain were calculated, as shown in Figure 2.18. It is seen that the RMSE values for the ADMM lasso and basis pursuit show decreasing trends when increasing the number of measurements used for reconstruction. But the ADMM Lasso has higher error values at a low measurement ratio range (30-50 measurements), and its accuracy performance is less stable compared to other methods, which indicates that the ADMM lasso is not the best algorithm for reconstructing our signal. The Matlab lasso algorithms had a good performance at a low measurement ratio range (30-70 measurements). The standardisation pre-processing of Matlab lasso further reduced the error values. However, the error values did not decrease by a massive step when using more measurements for reconstruction. The error values that this method yielded with and without standardisation pre-processing converges when

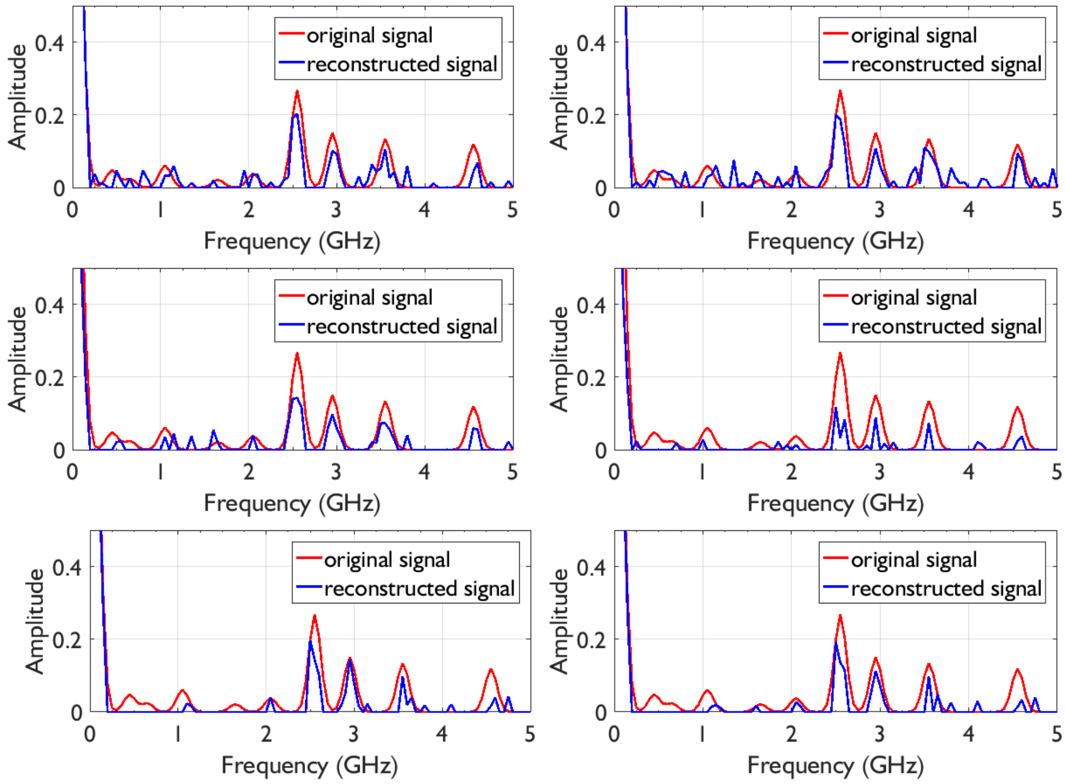


FIGURE 2.17: The reconstructed frequency domain signals for all five algorithms corresponding to a measurement ratio of 40%. The red line is the ground truth signal, and the blue line represents reconstruction. From left to right, top row: NESTA and L1 Magic; middle row: ADMM Basis pursuit and lasso; bottom row: Matlab lasso for non-standardised data and standardised data, respectively.

the number of measurements increases. Conversely, the L1 Magic and NESTA achieved far lower error values, and their performance is stable among all measurement ratio settings. When using half of the full measurements (100) or more, these two methods yielded the lowest error values, indicating that these two methods are suitable for our signal reconstruction.

Because the four dominant frequency bands contain the most significant information, RMSE was also calculated for the four dominant frequency peaks only. The RMSE results are shown in Figure 2.18. It is seen that the NESTA and the L1 Magic produce the lowest error over the frequency range, which is consistent with the results of Figure 2.19. However, the reconstruction error for ADMM basis pursuit is once again extremely unstable in relation to the number of measurements. At 70 to 90 measurements, ADMM basis pursuit produces comparable results to NESTA. This suggests that it has an advantage in terms of noise reduction. In comparison, at high sample densities, the ADMM lasso remains stable. Furthermore, Matlab lasso standardisation has no impact on RMSE.

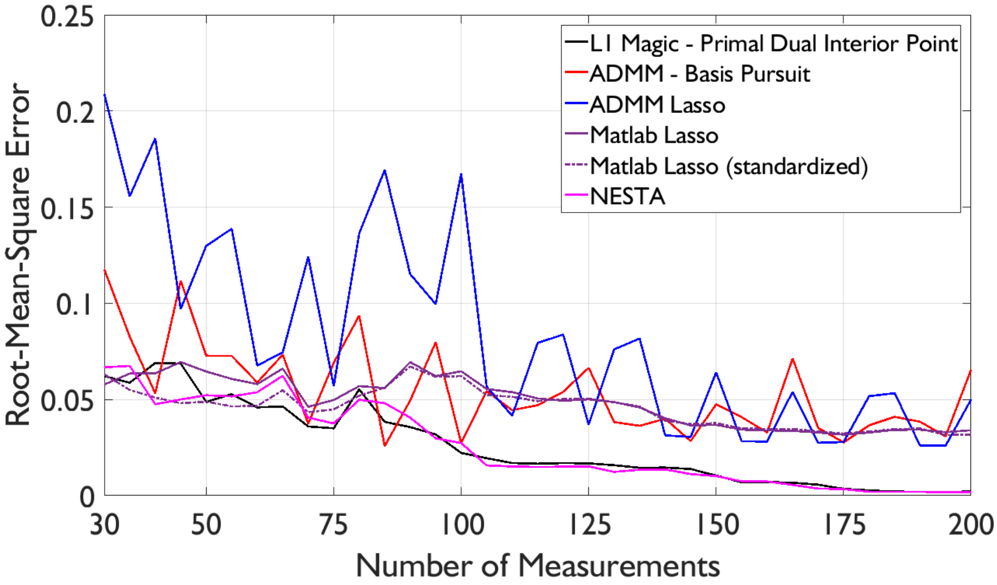


FIGURE 2.18: RMSE of reconstructed signal calculated over entire frequency range. All five candidate algorithms show a descending trend. NESTA and L1 Magic algorithm yield the smallest RMS error.

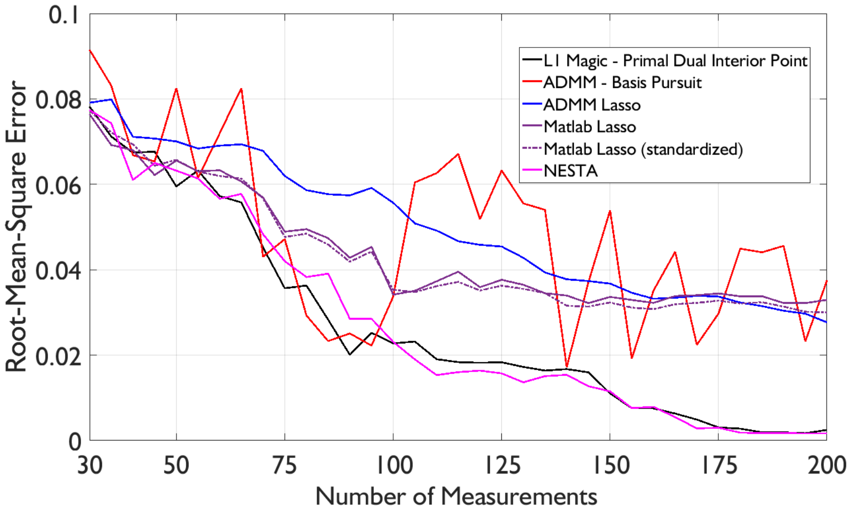


FIGURE 2.19: RMSE of reconstructed signal calculated for the frequencies spanned by the four dominant peaks only. Error rates similar for all five algorithms for the small number of measurements. The relative performance of NESTA and L1 Magic improves as the number of measurements increases.

We also calculated the RFE values to measure the reconstruction error at the four pre-defined main frequency bands, defined by the widths of the main peaks in the spectrum, which indicates the most important information. The results are shown in Figure 2.20, the error growing in either positive or negative direction

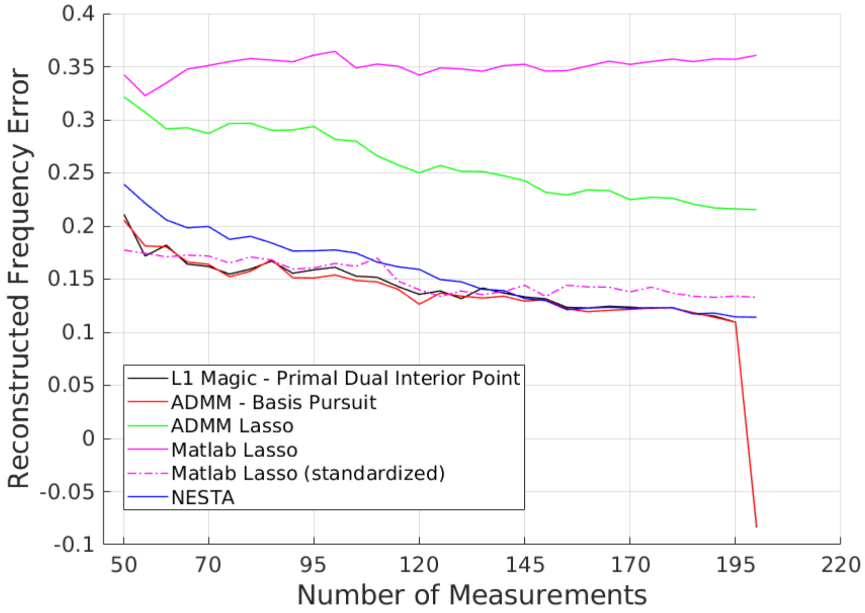


FIGURE 2.20: RFE of reconstructed signal calculated for the four dominant peaks. Error rates similar for all five algorithms for the small number of measurements.

denotes a larger discrepancy in comparison with the actual signal. Thus the desired error value should be close to zero regardless of its sign. The main tendency shows that a higher sampling rate results in a smaller error, except for Matlab Lasso and standardised ADMM Lasso. The Matlab Lasso algorithm has a negative error with a smaller subset than that of 60 measurements. This results in a stronger noise at an unwanted frequency band. In contrast, ADMM basis pursuit and L1 magic follow has similar results when handling a large number of measurements. The sharp decreasing trend implies that the amplitude of the introduced noise is over that of the actual signal. Therefore the structural information is affected by noise.

2.3.4 Computational efficiency analysis

In this section, we demonstrate the computational efficiency of the evaluated methods in terms of two criteria: 1) the reconstruction time, and 2) capability to reconstruct long signals consisting of a large number of data points. In practice, the PTS-OCT system often requires a quick scanning response and a short reconstruction time is desired for displaying results without delay. Therefore, when several algorithms yield similar reconstruction accuracy, we prefer to choose the most efficient one. In addition, we considered the ability to reconstruct a long signal as an essential factor because of the deep axial scanning range of the PTS-OCT system. To evaluate the algorithms with these two criteria, we upsampled and interpolated the original data (200 measurements) to produce signals of differing lengths and recorded the reconstruction time in each case. For each algorithm, the reconstruction time was measured using the Matlab Profile function, and the average time taken was determined over five repeat experiments. The experiments were run on a PC with 64-bit Windows 10 with an Intel Corei7 CPU @ 3.07GHz and an 8GB RAM.

First, we measured the reconstruction time of all algorithms based on the same signal length, 400. As shown in Figure 2.21, the reconstruction time by most of the algorithms is within one second. It is obvious that the ADMM lasso is the most efficient algorithm. The number of measurements has a minimal influence on its reconstruction time. The Matlab lasso and L1 Magic are two methods that consumed a longer time for reconstruction when the number of used measurements was increased. The L1 Magic is more efficient than the Matlab lasso by a small margin of 0.0083 on average. The NESTA method was slightly affected by the number of measurement; however, it generally took a much longer time than the other three aforementioned methods. The ADMM Basis pursuit is the only one that took longer than one second and the longest time it took can be 2.4 seconds.

The lengths of the interpolated signal used were 400, 800, and 1600 data points. To ensure a correct reconstruction at each scale and make fair comparison, we fixed the measurement ratio for all scales to be 50%, which we believe is an acceptable compression rate for a real application. Then we attempted to reconstruct the signal using each algorithm. Table 2.1 shows the average computational time for different data scales. It is seen that the ADMM basis pursuit, ADMM lasso and Matlab lasso could only reconstruct 400 data-point signal successfully. They failed to reconstruct signals of 800, and 1600 data points. In contrast, the L1 magic and NESTA, were able to reconstruct all larger-scale data. However, it should be noted that the reconstruction time of NESTA is increased to 54 seconds for reconstructing 1600 data-point signal.

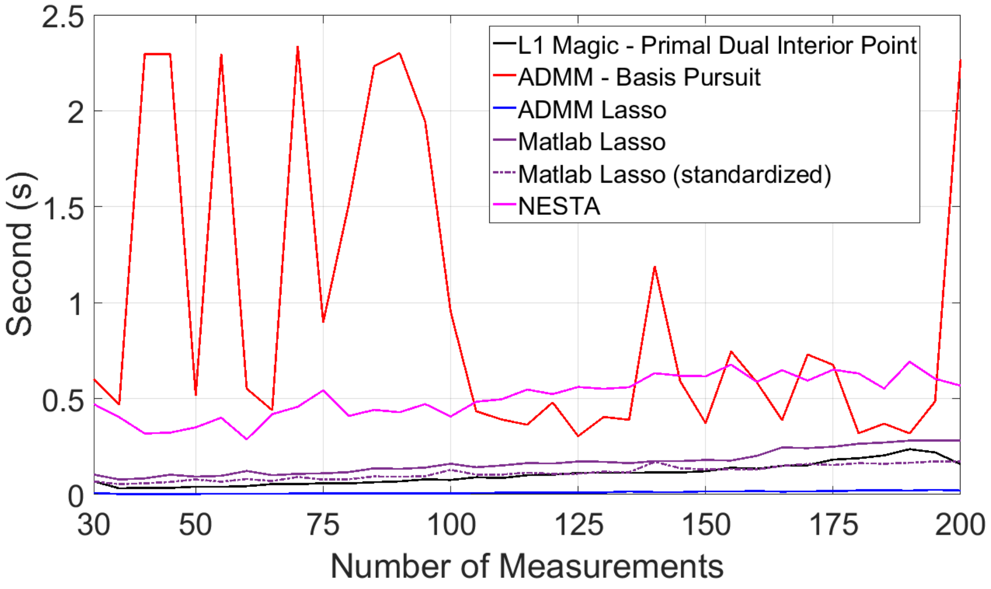


Figure 2.21: The computational time as a function of the number of measurements. ADMM Lasso, Matlab Lasso, Matlab Lasso (standardized) and L1 Magic show small linear increases with respect to the number of measurements. Basis pursuit is unstable for measurements $<$ 110 due to the slow convergence rate.

Table 2.1: The average reconstruction time. The original signal was interpolated exponentially. L1 Magic and Mesta successfully reconstructed the signal at three scales, while others only reconstruction the signal with 400 length and failed at other scales.

Average Reconstruction Time (second)			
Scale	L1 Magic	ADMM BP	ADMM Lasso
400	0.48	0.92	0.02
800	2.63	—	—
1600	23.32	—	—
Scale	Matlab Lasso	Matlab Lasso (standarded)	Nesta
400	0.04	0.05	1.08
800	—	—	3.58
1600	—	—	54.24

2.4 Conclusion

In this chapter, we presented compressive signal sensing and reconstruction for two systems: a single-pixel camera (SPC) system and a photonic time stretch optical coherence tomography (PTS-OCT) system. The SPC system, for 2D image reconstruction, was developed as a prototype that can be easily adapted to various hardware configurations. Our control software provides an optimised imaging pipeline and options for reconstruction algorithms. For our PTS-OCT prototype system, we evaluated, in terms of spectrum reconstruction accuracy and efficiency, six reconstruction algorithms for 1D PTS signal reconstruction.

Our SPC system is a simple and cost-efficient implementation of SPC technology that is suitable for both research and outreach activities. The hardware implementation also provided a facility for developing our deep learning-based reconstruction algorithm (Chapter 3). We attributed the good performance of our SPC hardware to two factors. The first factor is the hardware optimisation of removing the circuit noise and the voltage amplifier noise. As a result, we stabilised the photodetector's voltage signal and improved the accuracy of the measurement when converting the voltage signal to the digital signal. The second factor is improving the measurement accuracy by increasing the measurement resolution at our software end. Specifically, we implemented a two-step system setting-up operation before doing SPC imaging: the saturation calibration and the dynamic range mapping. They are essential to ensure a correct measurement record.

In our experiments, we demonstrated an example of image reconstruction by applying four different reconstruction algorithms. From the experiment, we conclude that four built-in algorithms have different capabilities of image reconstruction. The $L-2$ minimisation method had a good reconstruction capability in terms of the content structure. However, the energy of the reconstructed images is not sparsely distributed. As a result, its recovered image usually has lower contrast. The CS-based reconstruction methods, including L_1 and TV minimisation, has more high-contrast image content in the recovered image. Although the L_1 method has a relatively high PSNR value, its results have lower image quality in terms of content structure, which can also fund in OMP's results. In contrast, the TV method yields the best results by minimising image gradient as an objective function. When the measurement ratio increased, the CS-based methods can achieve a good result.

Our system provides a good introduction to the field of compressive sensing, and our open-source Matlab toolbox can be easily adapted by researchers wishing to evaluate their own compressive sensing algorithms. One can easily replicate the hardware system configuration using off-the-shelf components, including a photodetector and Arduino micro-controller. The design of our bespoke amplifier circuit board is freely available, and a replica of our board can be obtained quickly and at low cost using one of the many online PCB manufacturing businesses. Our SPC software was developed to control the image sampling and reconstruction procedure. We provide four well-known algorithms for this purpose. Our code could also be adapted for other experimental imaging architectures, such as coded aperture imaging.

In the second section, we presented a systematic evaluation of the PTS-OCT signal reconstruction using CS approaches. In this system, the essential sampling hardware performing the random pattern sampling was a signal modulator. It generated a series of pseudo-random bit sequences (PRBS) that were modulated with the optical signal and measured by a photodetector. Our work provides a mechanism for reconstructing a signal from a very fast data stream. For this purpose, we evaluated NESTA, L1 Magic, ADMM Lasso and ADMM Basis Pursuit algorithms. Our results indicated that the NESTA and L1 Magic algorithms produced the most reliable reconstruction performance for our PTS-OCT signal in terms of accuracy. Their error values were lower than values of other methods, and they provided a stable performance at different measurement ratios, indicating reliable signal compression. As a consequential benefit, the evaluation method in our PST-OCT approach inspired the community in this research area for further advanced development [24, 25].

Signal reconstruction times using these methods were, in general, long as is typically expected with the CS method. However, this is not the primary concern as we demonstrated that the key goal (to acquire a signal from a very fast PTS-OCT data stream) was accomplished using compressive sensing methodology. In the next chapter, we build on the CS work presented thus far and approach the signal reconstruction problem with an alternative, adaptive measurement technique based on deep-learning.

Chapter 3

Single pixel imaging with a mixed-weights neural network

3.1 Introduction

As discussed in Chapter 2, single-pixel imaging systems consist of three essential components: an image sampling hardware, a single-pixel photodetector, and a reconstruction algorithm. The sampling hardware is usually implemented using either a digital micro-mirror array or a light projector (e.g. of the type typically used in audio-visual presentations), which generates binary sampling patterns. A small number of sampling patterns (small measurement ratio) is desired to achieve good sampling efficiency. Then, a single-pixel photodetector is used to measure the intensity of light backscattered from the object scene. Lastly, these sampling patterns and their corresponding measurements are used by the reconstruction algorithm to recover the sampled image.

In the previous implementations described in Chapter 2, four approaches were evaluated to reconstruct the image. The CS reconstruction methods reduced the number of measurements required for a successful reconstruction. However, this reduction in the number of sampling is not significant, comparing to the full sampling number used by a conventional raster scanning. Usually, when the number of the measurements is fewer than half number of pixels in the reconstructed image (50% measurement ratio), the algorithms may cause incorrect reconstruction, and the image contained heavy noise. Moreover, those methods are time-consuming by using iterative optimisation. They required more iterations to reconstruct the image when only a few numbers of measurements are available, resulting in an inefficient reconstruction process. In Section 2.3.4 we stated that the reconstruction time for large-scale signals was significantly increased due to the large iterations of optimisation. The computational cost of signal recovery became prohibitively expensive. To solve the problems mentioned above, deep-learning-based methods of image reconstruction have been introduced recently as promising approaches with the advantages of reducing the number of measurements and reconstruction time.

Deep neural networks (DNNs) have become widespread in image processing tasks [26–28]. Especially, a DNN-based method has been proved to achieve favourable results in image recovery [29]. Motivated by this success in image reconstruction tasks, researchers subsequently applied DNN approaches to compressed sensing of images and corresponding reconstruction problems [30–37]. These methods proposed to train a DNN model with a large number of image samples such that a set of sampling patterns and model weights can be optimised for high-quality image reconstruction. These DNN-based solutions were reported to outperform the state-of-the-art of conventional compressed sensing algorithms in terms of speed, accuracy and data compression. However, none of them was applicable to the SPC sensing hardware due to several realisation difficulties, which we will describe in Section 3.2.

The disadvantage of current deep learning-based methods motivated us to develop a DNN-based model that is hardware-friendly to our SPC device. Specifically, the network should be able to learn a set of binary sensing patterns such that they can be used in the sampling device and improve the sampling efficiency by further increasing the reconstruction accuracy. To this end, we proposed a network architecture that had the following features:

- **Mixed-weights architecture**: the model is designed based on a mixed weights network, which leads itself naturally to hardware implementation. The binary patterns can be trained together with other floating-point weights in an end-to-end manner. Unlike the floating-point patterns used in previous work, our binary patterns are more appropriate for both sampling and measuring hardware. In respect of the sampling hardware, which uses the patterns to take sampling, the binary patterns can be represented on a DMD without the need for any additional device modulation. In terms of the measuring hardware, which takes measurements of the backscattering light, the binary patterns effectively increase the light-sensing sensitivity of the single-pixel photodetector. Compared to the real-valued patterns used in previous work, the binary patterns result in more significant light changes since the binary values are more discrete. These changes from pattern to pattern are more distinguishable when the photodetector converts the light intensity to digital value based on a fixed analogue to digital conversion resolution. To clarify how the network uses the binary patterns, we present more details of the network structure in Section 3.3.1.

- **LSHR sensing-reconstruction scheme**: the network uses a novel sensing-reconstruction scheme, which we term low-resolution sensing with super-resolution reconstruction (LSHR), to directly reconstruct super-resolution images from low-resolution sampled measurements. Specifically, it assumes two things: that the object is sampled using a low-resolution DMD array, and the network reconstructs a super-resolution image that has more pixels than the number of micro-mirrors in the DMD array. Compared with related methods, our LSHR scheme leads to a low-computation reconstruction model that has small intermediate feature maps and therefore, fewer convolutional operations are required for each feature map. Hence, it is more efficient than previously reported methods. We present the efficiency comparison in Section 3.5.4.

- **Recursive block structure**: Our network architecture implements image reconstruction by using long-term recursive blocks, where their weights are shared across different blocks. Our block structure further reduces the model size and number of weights required while yielding higher reconstruction PSNR accuracy.

In the remainder of this chapter, we first present a brief background in Section 3.2 reviewing several recently proposed approaches which implemented compressed sampling and reconstruction for images. Then, we describe the details of our proposed method in Section 3.3. The simulation results are presented in Section 3.5, and evaluated with respect to prior work, followed by a further analysis of the network's structure and efficiency. We present the experimental results of integrating the network with our imaging hardware setup in Section 3.5.5. Lastly, we conclude our work in Section 3.6.

3.2 Review of existing approaches

The problem of compressed sampling and reconstruction of images has recently been approached using deep neural networks. The prototype concept of neural network-based image reconstruction was implemented using a fully-connected network. Thereafter, this problem was addressed using convolutional neural networks, which avoided the fixed input image size requirement of the fully-connected network. In this section, these existing methods are organised into three categories according to different types of sensing pattern: 1) pre-generated patterns, 2) learned patterns, and 3) binary patterns. The details of these methods are discussed below, and their key characters are summarised in Table 3.1.

3.2.1 Networks based on pre-generated patterns

In the early work in this area, pre-generated, 'static' sensing patterns were used for network training. The first example was a stacked denoising auto-encoder (SDA) which was implemented using a network comprising only fully-connected layers [30]. A set of pre-generated random Gaussian patterns, used to sample images, and the corresponding measurements were used to train the network for reconstructing images. In this method, the measurements y were the network input, which represented as $y = \Phi x$ where Φ is the sampling patterns, and x is a target image. The network output is the reconstructed image that is compared with the ground-truth image to calculate the loss.

Inspired by the SDA method, ReconNet [34] was subsequently proposed. It improved the accuracy by using a deeper network that added a set of additional convolutional layers before the end of the network architecture. These layers were designed to have different kernel sizes such that both fine and coarse image content were learned. Although, its performance was still constrained by its fully-connected layer since the weights resulted in a large model size leading to a large computational burden. Due to this fact, the sensing area had to be constrained to small-size image reconstruction. For a large image, it had to be divided into small patches and processed individually. In a post-processing step, the reconstructed small patches were then concatenated to create the whole image. Lastly, the image was denoised by using a BM3D [38] to smooth the edges between patches.

Thereafter, the performance of the ReconNet was further improved by a DR^2-Net [35]. In this network, they replaced the convolutional layers of ReconNet with a set of residual blocks which was shown to make the network easier to train. However, the image sensing procedure was still achieved using small patches.

In the aforementioned methods, the neural networks were trained with a single set of pre-generated patterns. During the testing phase, the model used the same patterns to do the sampling. In contrast to those methods, DeepInverse [32] randomly generated a set of static patterns at each step of training. During the testing, it could use pre-generated patterns that were different from those in training, i.e. the model does not rely on a single set of patterns. The pre-generated random sensing patterns, denoted as Φ_{arb}, was used in both sampling and reconstruction stage. Specifically, the model first got measurements by sampling the image with Φ_{arb}. After the sampling stage, the network then got an initial image approximation, called an initial proxy, by calculating $\tilde{x} = \Phi_{\text{arb}}^T y$, where the $\tilde{x}$ is the proxy of the reconstructed image, Φ_{arb}^T is the pseudo inverse of Φ_{arb} and y is the measurements. Having the proxy, the network then used a convolutional network to reconstruct the final image.

3.2.2 Networks based on learned patterns

In the field of image sampling and reconstruction, it has been studied that specially designed sensing patterns can improve reconstruction accuracy compared with randomly generated patterns [39, 40]. Some of the work described in Section 3.2.1 were further developed using learned patterns to seek improvement of reconstruction accuracy. In these methods, the sampling patterns, as part of the neural network, were trained through a training process.

The structure of the SDA network was further modified to learn a set of patterns by adding a fully-connected layer as the first network layer such that its weights were used to sample an input image x directly. The sampling procedure can be represented as $y = \sigma(Wx + b)$ where the $\sigma(\cdot)$ is an activation function and W and b are the weights and bias of the fully-connected layer. During the training, the fully-connected layer was optimised with the goal of obtaining the best measurements y from sampling x to do the reconstruction. At the same time, another method having similar network architecture to SDA was proposed [31]. It applied a fully-connected neural network to do block-based compressed sensing, i.e. it works on small patches of high-dimensional images. The network was trained to optimise the sensing patterns and the reconstruction network weights jointly.

The DeepInverse network was also further developed with a new architecture named DeepCodec [33], which had an encoder-decoder architecture. Unlike the SDA network that used a single sampling layer to get measurements, the DeepCodec was trained to take measurements from images by using a serial of convolutional layers. The DeepCodec encoder took an input image as input and gradually reduced the spatial dimension of the intermediate feature maps to get the final encoded measurements of a single spatial dimension. They claimed the reconstruction efficiency was improved by using this encoder-decoder architecture. However, this multi-layer measurements encoding scheme was barely impractical in hardware application.

The authors of the ReconNet also modified the network to incorporate learned patterns [36]. In this extended network, the fully-connected layer was initialised with a set of random Gaussian patterns and optimised during the iterative training process. In the testing stage, the optimised patterns were used to perform the image sampling. They showed that a further improvement of reconstruction accuracy was achieved by using the learned patterns. In their recent work [37], they proposed an image sampling scheme by using a convolutional layer with a small stride step to avoid the blocking artefacts appearing in the reconstructed images. Although the accuracy improvement of these two work was demonstrated in the simulation results, implementing those sensing patterns, which contain high-precision floating-point values, in DMD sampling hardware is not straightforward and less efficient.

3.2.3 Networks based on binary patterns

Iliadis et al. [41] proposed a network using binary patterns for video reconstruction. The network applied a 3D binary sampling matrix to sample a sequence

Table 3.1: The summary of reviewed methods. The random Φ_r and learned Φ_b means the sensing matrices are randomly generated or further learned by the training. The Φ_r Φ_r is for real-valued weights and binary weights separately. The compression type denotes the axis along which the sensing is compressed. The post-processing indicates whether the image enhancement procedure is needed in addition to the network results.

Network	Sensing Matrices	Compression Type	Post-processing
SDA	random/learned Φ_r	spatial CS	Yes
BCS-Net	learned Φ_r	spatial CS	Yes
DeepInverse	random Φ_r	spatial CS	No
DeepCodec	learned Φ_r	spatial CS	No
ReconNet	random Φ_r	spatial CS	Yes
DR^2-Net	random Φ_r	spatial CS	Yes
Adp-Rec	learned Φ_r	spatial CS	Yes
Fully-Conv	learned Φ_r	spatial CS	No
DeepFully-Conn	random Φ_b	temporal CS	Yes
DeepBinaryMask	learned Φ_b	temporal CS	Yes

of temporal video frames, resulting in a set of measurements. The network was trained to learn a non-linear mapping between measurements and reconstructed frames via fully-connected layers. The same 3D binary matrix sampling strategy was used in their more recent work, DeepBinaryMask [42], for sampling video frames. However, in this work, matrices were generated through a training process using an encoder-decoder structure, which is similar to DeepCodec. Their sampling strategy achieved temporal compression of video frames, which is functionally different from the spatial compression task, which is the focus of our own work.

In summary, a variety of network architectures were proposed to improve the reconstruction accuracy from compressed sampling measurements. However, they were not designed to be integrated with the SPC sensing hardware due to two factors.

First, these frameworks were designed to use real-valued sensing patterns that were not suitable for SPC sampling. These patterns, used to train models, were represented in 32-bit floating-point format. Although good performance was demonstrated in image sampling simulations using modern GPUs, their high-precision values were impractical for implementation on the structured light sensing hardware such as DMDs, where instead, binary patterns were more suitable.

Second, previous methods assumed that the sensing patterns and the reconstructed images had the same resolution, i.e. the reconstructed image size should be the same as the pattern size. This sampling scheme was inefficient when reconstructing large images from the network for two reasons: 1) When the network reconstructs a large size image, the patterns also need to be increased to the same size. As a result, more measurements need to be taken such that the compression ratio defined by the number of measurements divided by the total number of pixels in the patterns remained constant. 2) An increase in pattern size dictates that the network's intermediate feature maps, which are processed by hidden layers, are expanded accordingly. This results in more convolutional operations being required for each layer. To overcome these drawbacks, it is beneficial to reconstruct a large image using sampling patterns that are smaller than the reconstructed image size. In this way, the network requires fewer measurements and has smaller intermediate feature maps.

Since the models described in previous work were constrained by the two aforementioned limitations, we are motivated to develop a hardware-friendly network that 1) can be applicable to SPC imaging and 2) can further improve the network efficiency by using low-resolution sampling patterns to reconstruct high-resolution images. The remainder of this chapter describes the details of our network implementation and experiment results.

3.3 Super-resolution SPC imaging using a deep neural network

In this section, we describe the details of our network structure and model training strategy. Figure 3.1 presents an overall schematic of the network structure. Conceptually, our network consisted of two parts: the convolutional net for initial image reconstruction (the green part) and the recursive residual-block net for image correction (the red part). In practice, they combine to operate as a single model during training and testing. The structures and functions of these two parts are described separately in Section 3.3.1 and 3.3.2.

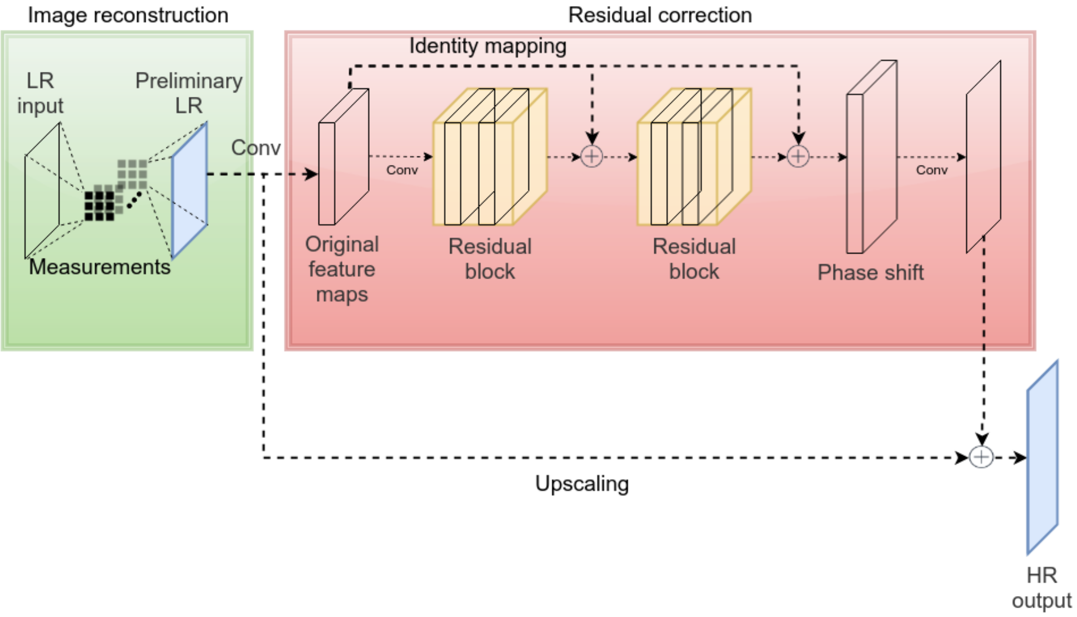


FIGURE 3.1: Overview of the proposed network structure. Our proposed network structure has two parts, which performs different functions: the first part (green block) implements image reconstruction, and the second part (red block) performs the reconstruction-residual correction. For the image reconstruction part, the network takes the low-resolution image patches as input data. It compressively senses them with binary patterns and reconstructs a preliminary image. After that, the residual correction net extracts the preliminary image features and corrects the reconstruction error using a sequence of recursive residual blocks. These blocks are connected to the original feature maps through identity mapping to learn the error correction. Finally, the preliminary image and residuals maps are up-scaled through two branches and combined element-wise to generate the final output image.

The development of the network architecture was based on our LSHR scheme, which is the fundamental idea behind this model. The LSHR scheme assumes that the object scene has been sampled at low-resolution and that the network uses these sampled measurements to recover the missing details. To this end, the image reconstruction net first sampled a low-resolution version of an original image, using binary patterns, to generate a corresponding set of measurements. These measurements were then deconvolved to generate a coarsely reconstructed image, which had the same size as the sampling patterns. At this stage, the details of the object scene were still missing. The coarsely reconstructed image was fed into the residual-block net, which estimated the image details and output the final image in high-resolution. By incorporating these two parts, the model was able to reconstruct the high-resolution image directly from the low-resolution sampling.

In practical applications, the ground truth, super-resolution, images are not known a priori. To evaluate the LSHR scheme, we trained and tested the network using paired-image datasets, in which the original image was used as the ground-truth, and its low-resolution version was used for sampling. These paired image sets enabled us to quantify the reconstruction accuracy at the training and testing stages. The training strategy is presented in Section 3.4, which involves details of the objective loss function and settings for training parameters.

We demonstrate the practical reconstruction accuracy of the network by incorporating actual SPC imaging hardware in its evaluation. In Section 3.5.5, we describe our SPC hardware setup, which was modified from our original setup (Section 2.2.1), and present the image reconstruction results.

3.3.1 Image reconstruction net

The image reconstruction net is responsible for image sampling and low-resolution image reconstruction. The net does the sampling process at the training and testing stages by using a convolutional layer where each 2D convolutional kernel acts as a small digital mirror array, and the kernel weights represent binary patterns. When it samples an image that has the same size as the kernel, sampling measurements are generated by doing a dot product of the kernel and the image. When the image is larger than the kernel, the measurements are generated by doing image convolution.

The image reconstruction from the measurements was implemented by using a set of transposed convolution layers, which had real-valued kernels. During the training, the network optimised both the binary and real-valued kernels such that an initial image was reconstructed from the measurements. When the trained model was integrated with the SPC hardware, the learned binary patterns were uploaded to the digital mirror device to do the sampling. The measurements of backscattered light intensity were recorded by a photodetector and then fed into that input layer of the network to obtain a reconstructed image.

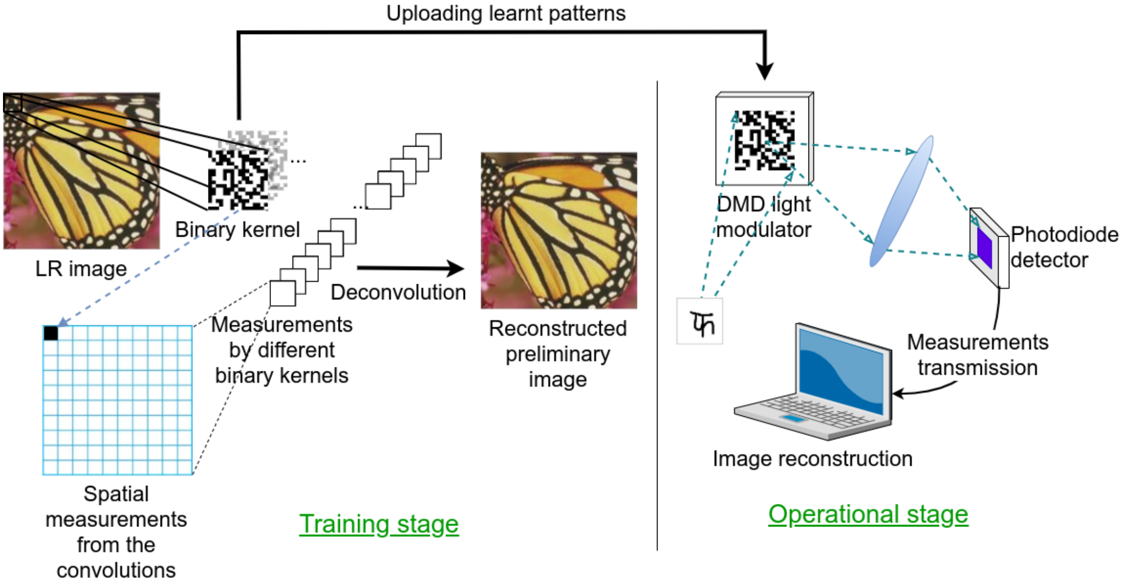

Uploading learnt patterns
LR image
Binary kernel
Spatial measurements from the convolutions
Measurements by different binary kernels
Deconvolution
Reconstructed preliminary image
Training stage
DMD light modulator
Photodiode detector
Measurements transmission
Image reconstruction
Operational stage

FIGURE 3.2: The image sampling and reconstruction operations. Training stage: the low-resolution image is sampled with binary kernels using a convolutional layer. Each convolution operation generates a measurement (the black element). By sliding the binary kernel through the image, the corresponding binary kernel's measurement map is generated. After convolution with all binary kernels, the transposed convolutional layer is used to reconstruct the low-resolution preliminary image from the measurements. Operational stage: the learned patterns are uploaded to the DMD hardware to do the sampling, the measurements recorded by the photodetector are sent back to the network to compute the reconstructed image.

A schematic diagram of the image reconstruction net is shown in Figure 3.2. It is seen that the measurements are obtained by doing the convolution of the binary kernels on a low-resolution image, constructing a measurement matrix. The sampling and reconstruction can be formulated as

$$\begin{aligned}\tilde{x} &= \mathcal{F}_d(y, W_r) + b \\ &= \mathcal{F}_d(\mathcal{F}(\varphi(x), W_b), W_r) + b\end{aligned} \tag{3.1}$$

where $\tilde{x}$ is the reconstructed preliminary image. The $\mathcal{F}_d(\,\cdot\,)$ is the transposed convolution, the W_r and b are the real-valued kernels and bias respectively. The $\varphi(\,\cdot\,)$ down-scales the original images for simulating the sampling process. The measurements y are generated by the convolution $\mathcal{F}(\,\cdot\,)$ of image x and the binary kernels W_b where each kernel corresponds to a sensing pattern. In terms of image sampling, the binary patterns can be obtained in two ways: 1) fixed pre-initialised patterns, and 2) learned patterns which were optimised in the network training procedure. The details of their implementation is described in below. To evaluate their effect on reconstruction accuracy, we study the model training for both types of pattern and discuss their performance in Section 3.5.3.

Randomly pre-generated binary weights: In this approach, the patterns were initialised with random values, which remained fixed during the training. Before the training, the binary weights generated from the random Bernoulli distribution with $\Pr(1) = 0.5$. For each kernel, the initialisation was applied independently. During the training process, the fixed kernels were used to generate the measurements, and the kernel weights in the rest of the network were updated. By freezing the sampling kernels, the network was trained to fit a specific set of binary patterns.

In our experiments, this approach was used to obtain a benchmark performance for training, compared with training using the learned weights method.

Learned binary weights: In this approach, the generation of learned kernel weights was achieved in two steps. First, kernels were initialised with floating-point weights following the uniform distribution within the range of $[-1, 1]$. This method ensured the weights were equally assigned to positive and negative values within the range, which facilitated the binarisation in the second step. These values were kept for gradient calculation during the backpropagation since the floating-point weights were necessary for the network optimisation. In the second step, the floating-point weights were mapped to binary values and applied to the sensing kernels during forward propagation. This binarisation scheme is formulated as

$$w_b = \begin{cases} 1 & if\ \ w_r > 0, \\ 0 & if\ \ w_r \leq 0, \end{cases} \tag{3.2}$$

where the w_b are the 0, 1 binary weights and the w_r are the real-valued weights. It should be noted that in the scheme, the binary kernels were only involved in the convolution operations. Furthermore, we implemented two additional operations during the network training to facilitate the binary weights learning process. First, we clipped the real-valued weights within the range $[-1, 1]$ after each training step. This ensured that the floating-point weights can be used effectively for the binarisation mapping since the very large values, which were out of the range $[-1, 1]$, did not have a significant impact on the binarisation process. Second, the network also applied an ℓ_2 norm regularisation to the weights such that the large weights values and the risk of gradient explosion were avoided. It is noticed that a Sigmoid activation function can do the same computation. However, to optimise the patterns towards binary values by this approach, the absolute values of weights in the previous layer have to be very large. It adds a risk of gradient explosion and prevents the pattern updating, i.e. changing from 0 to 1 means changes of weights from a very small negative value to a very large positive value. Therefore, it is not suitable for our network.

3.3.2 Residual correction net

Using the output of the image restoration net as input, the residual correction net produces the final reconstructed image in high resolution and recovers the image details. A schematic of the residual correction net is shown in the red block in Figure 3.1. This net can be briefly summarised into two branches: 1) residual correction and 2) image upscaling.

The residual correction branch used a set of recursive residual blocks to learn the reconstruction of missing details between the low-resolution input image and the high-resolution ground-truth image. To this end, a sequence of convolutional residual blocks was developed. It had a long-term identity that mapped original features, extracted from the input image, to the output of each block. This link allowed the original features to be directly added to the residuals at each stage, seen in Figure 3.3. To better resist over-fitting due to a large number of weights, the kernel weights are designed to be shared across different blocks, which gives the network a recursive structure and reduces the total amount of learnable parameters significantly. At the end of the residual blocks, we applied a phase shift operation to upscale the feature maps into higher resolution.

In the image upscaling branch, the preliminary images, which were reconstructed by the image reconstruction net, were up-scaled to the super-resolution size. These images were then merged with the output of the residual mapping branch to reconstruct the final super-resolution output. In the remainder of the section, we first explain the residual block and then describe the image upscaling.

The residual block structure was initially proposed in ResNet [43], in which the block consisted of a set of convolutional layers followed by activation functions. For each block, an identity mapping branch was used to add the input feature maps with blocks output, thereby forming a more detailed feature map. With this

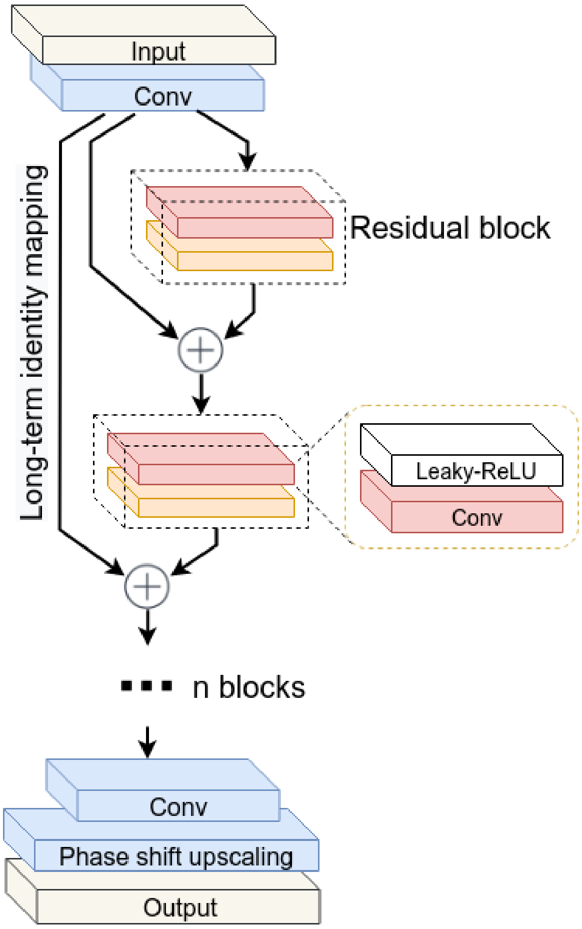


Figure 3.3: The structure of the residual correction net. The network feeds in the reconstructed preliminary image as an input node and then extracts the original features. The feature maps are then passed to the recursive residual blocks, shown as dashed green lines. Each residual block has an identity branch that connects the original features with its output. Thereafter the residuals and the original features are added, element-wise, to generate the input to the next residual block. For each residual block, we applied leaky ReLU as the pre-activation function. At the end of the network, an extra convolutional layer and an upscaling layer is added to generate the residual output.

structure, the blocks were designed to learn the residual between the input and output features. The residual block is formulated as

$$\hat{a} = \mathcal{R}(a) = \mathcal{F}(a, W) + h(a) \tag{3.3}$$

where a and $\hat{a}$ are the input and output of the residual block, W indicates the weights of the residual block, $\mathcal{F}(a, W)$ learns the residual mapping between the input and the output and $h(a)$ is the identity mapping function. In a residual block with two convolutional layers and two activation functions, the residual mapping function is

$$\mathcal{F}(a, W) = \sigma(W_2\sigma(W_1 a)) \tag{3.4}$$

where σ denotes the post-activation function. In a recent research [44], it was shown that a pre-activation generally made the model easier to be trained. The sequence of the blocks in our network is shown in Figure 3.3. We used two convolutional layers with a pre-activation function in each block. For the identity mapping, the network connected the feature maps associated with the low-resolution input, which was generated by the first convolutional layer, to the output of each block. This long-term connection directly related these features with the outputs of the deep residual blocks. This can be formulated as

$$\hat{a}^j = \mathcal{R}^j(\hat{a}^{j-1}) = \mathcal{F}(\hat{a}^{j-1}, W^j) + h(a^0) \tag{3.5}$$

$$\mathcal{F}(\hat{a}^{j-1}, W^j) = W_2^j\sigma(W_1^j\sigma(\hat{a}^{j-1})) \tag{3.6}$$

where $\mathcal{R}^j$ is the residual mapping function of the j-th block, a^0 is the initial features, and $\hat{a}^j$ is the output of j-th block. W^j is the weight and σ is the Leaky ReLU activation function [45]. The ith-layer in each block shared the same weights W_i where $i \in 1, 2$. This formed a recursive structure and reduced the total amount of model parameters significantly. In the experimental results discussed in Section 3.5.2, it is shown that this structure yielded better reconstruction accuracy.

After the recursive block sequence, an image upscaling operation was implemented at the end of the residual correction net to generate the final reconstructed image. This procedure consisted of a phase shift operation, performed on the output of the recursive residual blocks, and an image merging branch, which is shown under the residual blocks in Figure 3.1.

The phase-shift layer [46] was used to enlarge the size of the learned residual feature maps by a factor of s. In our experiment, we set the upscaling factor s as 2. The phase shift operation, shown in Figure 3.4, rearranged the input feature maps $H \times W \times C \cdot s^2$ into $H \cdot s \times W \cdot s \times c$ such that the output feature maps had a super-resolution size.

The idea behind this operation is that the missing details between the pixels, which was denoted as sub-pixel information, was learned by the residual correction

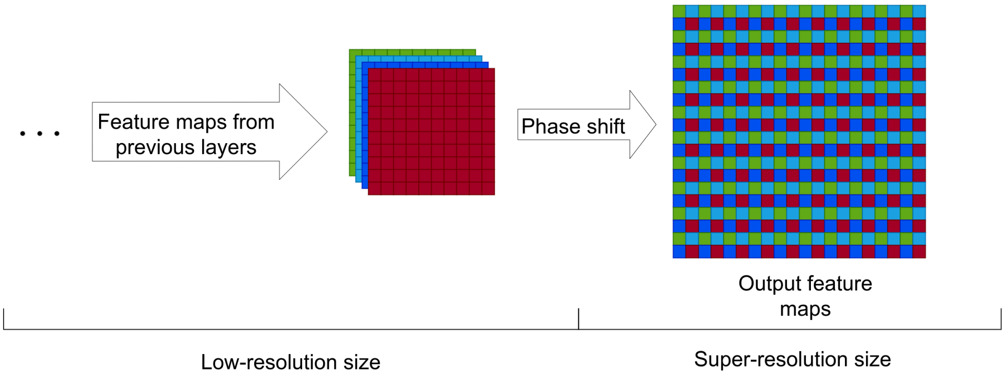


FIGURE 3.4: The phase shift operation rearranges the elements in the low-resolution feature maps and forms a high-resolution feature map. Each feature map, showing in different colours, fills the sub-pixel location of the high-resolution feature maps.

net with $S \cdot c$ kernels. These feature maps were then rearranged such that the learned details were filled in the sub-pixel location in each channel of c. Note that the channel c is determined by the final image (one for grayscale and three for RGB) such that the rearranged residual feature maps can be added to the image branch.

In the image up-scaling branch, we enlarge the image size by the same factor with the transpose convolution. Then the residual and the image are added together element-wise to generate the output image in the super-resolution size. Thus, the image details in the super-resolution image were reconstructed.

3.4 Network training

3.4.1 Loss function

Denoting the original image as x, the loss function aims to train the whole network f to reconstruct a super-resolution image $\tilde{x} = f(x, \theta)$, where the θ denotes the parameters of the model. In contrast to the general ℓ_2-norm loss function used in previous work, the network was trained using a Charbonnier loss function, which is a variant of the ℓ_1-norm function. From our empirical results, it was indicated that images generated using the Charbonnier loss function were usually sharper than the results obtained using an ℓ_2 norm loss function.

The loss function is formulated as

$$\mathcal{L}(x, \tilde{x}, W) = \frac{1}{N} \sum_{i=1}^{N} \sum_{s=1}^{S} \omega\alpha \left(x_i^s - \tilde{x}_i^s\right) + \frac{\lambda}{2N} \sum_{W} w^2 \tag{3.7}$$

$$\alpha(\mu) = \sqrt{\mu^2 + \epsilon} \tag{3.8}$$

where, in the first term, N is the batch size, S is scale, ω is the balance weight at each scale, $\alpha(\,\cdot\,)$ denotes the Charbonnier penalty with a pre-defined residual parameter ϵ. The second term is the ℓ_2-norm regularisation for the network weights w, balanced with weight λ.

This loss function supervised the training of both sub-nets simultaneously by associating the total loss with the output of both sub-nets, i.e. the reconstructed low-resolution image $\tilde{x}^1$ at the up-scaled super-resolution image $\tilde{x}^2$. For $\tilde{x}^1$, the ground truth image x_i^1 was generated by downsizing the original image using the bi-cubic interpolation method. For $\tilde{x}^2$, we used the original image for ground-truth. The scalar weight ω controlled the influence of each x_i^s in the loss function. In our experiment, we set $\omega = 2^s$ for each part.

3.4.2 Training strategy

The proposed network consists of two sub-nets that contain different types of weights, namely the mixed-valued weights for image reconstruction net and the real-valued for residual correction net. To train such a heterogeneous network, a straightforward strategy would be training two parts separately in a pipeline manner, i.e. the image-reconstruction net is trained at the first step and then used as a pre-trained model to train the whole network. This process can be viewed as either a two-step training strategy or as a semi-pretraining strategy for transfer learning. This method was adopted in previous work [35] and was claimed to result in a smoothly diminishing training loss function.

However, it was shown in a range of work that the pipeline strategy that optimises two parts of the network in a semi-decoupled way is usually inferior to the pure end-to-end training where all the components of the network are jointly optimised during training [26, 47, 48]. Therefore, in this work, we propose training the heterogeneous network in a pure end-to-end fashion: the optimiser updated the binary and real-valued weights together with the aim of loss decreasing. Our results demonstrated that this training strategy also led to a smoothed training procedure (Figure 3.10) and yielded a model with better overall performance (Table 3.2).

During the training procedure, it was observed that the loss for image reconstruction net was decreased quicker than the rest of the network. To balance the training process, we applied two learning-rate schemes during the joint training, where different learning-rate decays were applied to each part. Specifically, for the image reconstruction net, we set a larger initial learning rate with a faster decay. This encouraged a more rapid updating of the binary weights in the early stages of training, such that the sampling patterns were learned at the beginning. These optimised weights became stable in the later stage, where a greater emphasis was placed on the residual correction net with subsequently improved capability for fine image detail estimation. In our experiment, we empirically used a larger learning rate of 1×10^{-4} with a decay rate of 0.25 for the image reconstruction net. For the residual correction net, the initial learning rate and the decay rate were 1×10^{-5} and 0.75, respectively. The learning rate decayed every $200,000$ training steps.

3.4.3 Network parameters and training hyper-parameters

For the image reconstruction net, the pattern size 16×16 was applied to both the sensing kernels and the transposed convolution kernels. For each recursive residual block, the kernel size of the convolutional layers was set to 3×3, and each layer consisted of 64 channels. The leaky ReLU activation function with leaky rate $p = 0.2$ was used. The network training was implemented in 300 epochs by using Adam optimiser with a batch size of 16. The proposed network was trained on an Nvidia GeForce GTX 1080Ti GPU.

In our experiment, the network was trained with different measurement ratios, $R = \frac{m}{N}$, of 0.01, 0.10 and 0.25, where m is the number of sampling kernels and N is the number of pixels in the sensing patterns. Accordingly, the number of binary kernels for the 128×128 benchmark sampling images are 164, 1638 and 4096.

3.5 Experiment results and discussion

In this section, we present a series of experiments to demonstrate the model performance and study the advantages of our network. First, we evaluated the image reconstruction quality (see Section 3.5.2) on three datasets. Based on the results of this study, our learned and fixed-pattern binary models showed the first and second highest peak signal-to-noise ratio (PSNR) respectively compared to the four methods reviewed in Section 3.2. Next, in Section 3.5.3, we analyse how fixed and learned patterns affected the model training process. It was observed that the model trained with learned binary patterns yielded a faster training procedure and led to a lower validation loss. Finally, we assess the reconstruction efficiency of the network in Section 3.5.4, in comparison with other tested methods. Based on the comparison results, we demonstrated that our model outperformed other models with respect to balancing the reconstruction accuracy and the computational cost.

3.5.1 Datasets

To achieve good model robustness (to variation in image texture and scene content) and generality for the image reconstruction performance, the network training was based on a large-scale image dataset. In the experiment, we used the DIV2K image dataset [49] for training and validation. DIV2K contains 800 super-resolution images with a large diversity of contents. Two data augmentation processes were implemented on the training images. First, each training image was randomly cropped to form 50 small patches of size 256×256. Small patches sped up the training process compared with training on the original 2K-resolution images. The random cropping procedure generated $40,000$ training images in total. Second, random flipping and rotation of the original patches were also applied after cropping. These original resolution image patches were used as ground truth, and the corresponding low-resolution images were obtained using a down-scaling factor of 2.

At the testing stage, three datasets were used to test the model's performance. First, we used a benchmark dataset of 11 test images, which has been used in compared work [34–37] to evaluate the reconstructed image quality and compare it with the results obtained from previously described methods. Second, we evaluated our method on a much larger dataset – the test set of ILSVRC2017, comprising $50,000$ natural images from 1000 classes [50]. Tests on this dataset provided a good indication of the generality of the model. Lastly, we modified the DCT coefficients of the images contained in the ILSVRC2017 test set. It is known that natural images are often approximately sparse in the domain of the discrete cosine transformation (DCT) and the wavelet transform [51], and CS is an efficient method for approximate recovery of such images. Since our method is an alternative to CS, we have also evaluated the performance of our structured signal recovery method with images of various levels of sparsity. For this experiment, we generated a DCT-sparse version of the ILSVRC2017 test set, and we controlled the sparsity of the DCT coefficients as follows: Each image was first transformed into the DCT domain where the coefficients were reordered based on their magnitude, then we set 5 percentage threshold cases for coefficient magnitude such that 100%, 20%, 10%, 5% or 1% of the coefficients were retained and all other coefficients were set to zero. Finally, they were transformed back to the spatial domain to form a DCT-sparse image.

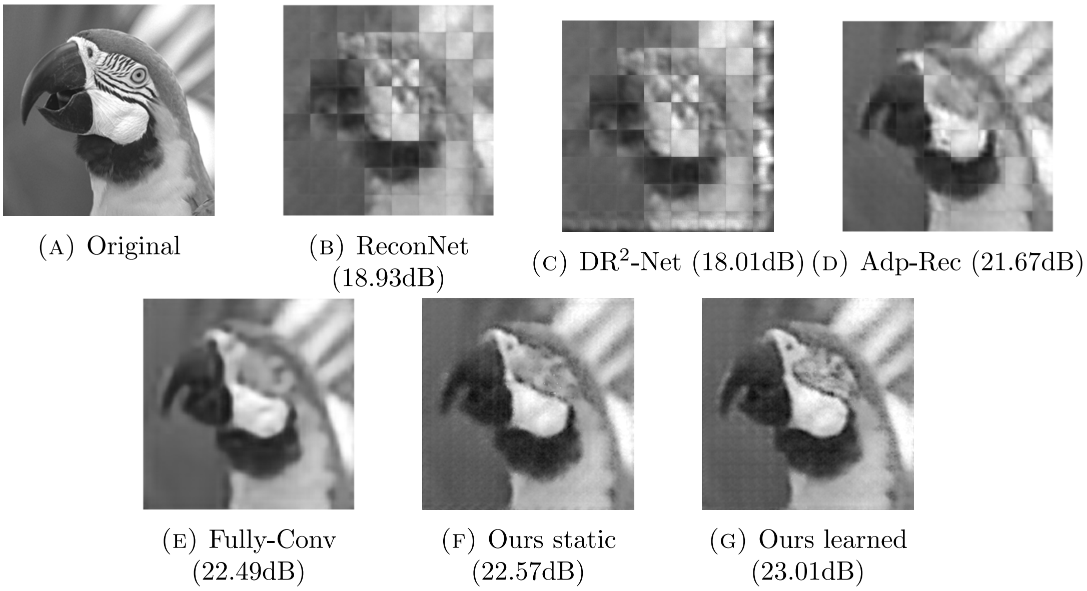
3.5.2 Experiment results with simulation dataset

The reconstruction results of the proposed network were presented and compared with four recently proposed methods: ReconNet [34], DR^2-Net [35], Adp-Rec [36] and Fully-Conv [37]. To be consistent with previous work, the peak signal-to-noise ratio (PSNR) was used as the evaluation metric.

The overall comparison results based on the benchmark image set are summarised in Table 3.2, which indicates the advantage of the proposed network. From the table, it can be seen that our network with the learned patterns achieved the highest average PSNR at all three measurement ratios. Our network with static binary weights yields the second-highest average value at measurement ratios of 0.01 and 0.10, and the third-highest at 0.25. The figures show that our approach of reconstruction from low-resolution sampling achieved a better overall reconstruction accuracy than the competing methods. At the same time, it should be noticed that the low-resolution image sampling employed in our method has the additional

TABLE 3.2: The PSNR of 11 test image in dB from recent six methods at three measurement ratios. The reported mean is the average PSNR value for all images. The red figures and the blue figures denote the first and second highest value among all the methods. Our network based on learned binary weights yields the highest average PSNR at all three measurement ratios.

Image	Methods	measurement ratio			Image	Methods	measurement ratio		
		R=0.25	R=0.1	R=0.01			R=0.25	R=0.1	R=0.01
Barbara	ReconNet	23.58dB	22.17dB	19.08dB	Boats	ReconNet	27.83dB	24.56dB	18.82dB
	DR^2-Net	25.77dB	22.69dB	18.65dB		DR^2-Net	30.09dB	25.58dB	18.67dB
	Adp-Rec	27.40dB	24.28dB	21.36dB		Adp-Rec	32.47dB	28.80dB	21.09dB
	Fully-Conv	28.59dB	24.28dB	22.06dB		Fully-Conv	33.88dB	29.48dB	22.3dB
	Ours (static)	27.52dB	24.57dB	22.03dB		Ours (static)	32.05dB	29.55dB	22.59dB
	Ours (Learned)	31.11dB	24.56dB	22.34dB		Ours (Learned)	34.13dB	29.59dB	23.31dB
Fingerprint	ReconNet	26.15dB	20.99dB	15.01dB	Cameraman	ReconNet	23.48dB	21.54dB	17.51dB
	DR^2-Net	27.65dB	22.03dB	14.73dB		DR^2-Net	25.62dB	22.46dB	17.08dB
	Adp-Rec	32.31dB	26.55dB	16.22dB		Adp-Rec	27.11dB	24.97dB	19.74dB
	Fully-Conv	32.91dB	27.36dB	16.33dB		Fully-Conv	28.99dB	25.62dB	20.63dB
	Ours (static)	30.36dB	26.07dB	17.10dB		Ours (static)	28.68dB	26.53dB	20.84dB
	Ours (Learned)	33.38dB	26.40dB	17.23dB		Ours (Learned)	30.63dB	26.56dB	21.35dB
Flinstones	ReconNet	22.74dB	19.04dB	14.14dB	Foreman	ReconNet	32.08dB	29.02dB	22.03dB
	DR^2-Net	26.19dB	21.09dB	14.01dB		DR^2-Net	33.53dB	29.20dB	20.59dB
	Adp-Rec	27.94dB	23.83dB	16.12dB		Adp-Rec	36.18dB	33.51dB	25.53dB
	Fully-Conv	30.26dB	24.98dB	16.92dB		Fully-Conv	38.10dB	34.00dB	27.26dB
	Ours (static)	28.00dB	24.34dB	16.81dB		Ours (static)	35.34dB	33.13dB	26.36dB
	Ours (Learned)	31.01dB	24.66dB	17.27dB		Ours (Learned)	36.91dB	33.45dB	27.13dB
Lena	ReconNet	27.47dB	24.48dB	18.51dB	House	ReconNet	29.96dB	26.74dB	20.30dB
	DR^2-Net	29.42dB	25.39dB	17.97dB		DR^2-Net	31.83dB	27.53dB	19.61dB
	Adp-Rec	31.63dB	28.50dB	21.49dB		Adp-Rec	34.38dB	31.43dB	22.93dB
	Fully-Conv	33.00dB	28.97dB	22.51dB		Fully-Conv	36.22dB	32.36dB	23.67dB
	Ours (static)	31.60dB	29.37dB	23.13dB		Ours (static)	34.80dB	32.55dB	24.82dB
	Ours (Learned)	34.18dB	29.57dB	23.52dB		Ours (Learned)	36.61dB	33.73dB	25.12dB
Monarch	ReconNet	24.95dB	21.49dB	15.61dB	Peppers	ReconNet	25.74dB	22.72dB	17.39dB
	DR^2-Net	27.95dB	23.10dB	15.33dB		DR^2-Net	28.49dB	24.32dB	16.90dB
	Adp-Rec	29.25dB	26.65dB	17.70dB		Adp-Rec	29.65dB	26.67dB	19.75dB
	Fully-Conv	32.63dB	27.61dB	18.46dB		Fully-Conv	32.90dB	28.72dB	21.38dB
	Ours (static)	31.51dB	28.71dB	20.09dB		Ours (static)	31.20dB	28.23dB	21.52dB
	Ours (Learned)	34.20dB	29.07dB	20.79dB		Ours (Learned)	33.51dB	28.61dB	22.10dB
Parrot	ReconNet	26.66dB	23.36dB	18.93dB	Mean	ReconNet	26.42dB	23.28dB	17.94dB
	DR^2-Net	28.73dB	23.94dB	18.01dB		DR^2-Net	28.66dB	24.32dB	17.44dB
	Adp-Rec	30.51dB	27.59dB	21.67dB		Adp-Rec	30.80dB	27.53dB	20.33dB
	Fully-Conv	32.13dB	27.92dB	22.49dB		Fully-Conv	32.69dB	28.30dB	21.27dB
	Ours (static)	32.64dB	29.84dB	22.57dB		Ours (static)	31.25dB	28.44dB	21.62dB
	Ours (Learned)	34.75dB	30.18dB	23.01dB		Ours (Learned)	33.68dB	28.67dB	22.11dB



(A) Original

(B) ReconNet (18.93dB)

(C) DR^2-Net (18.01dB)

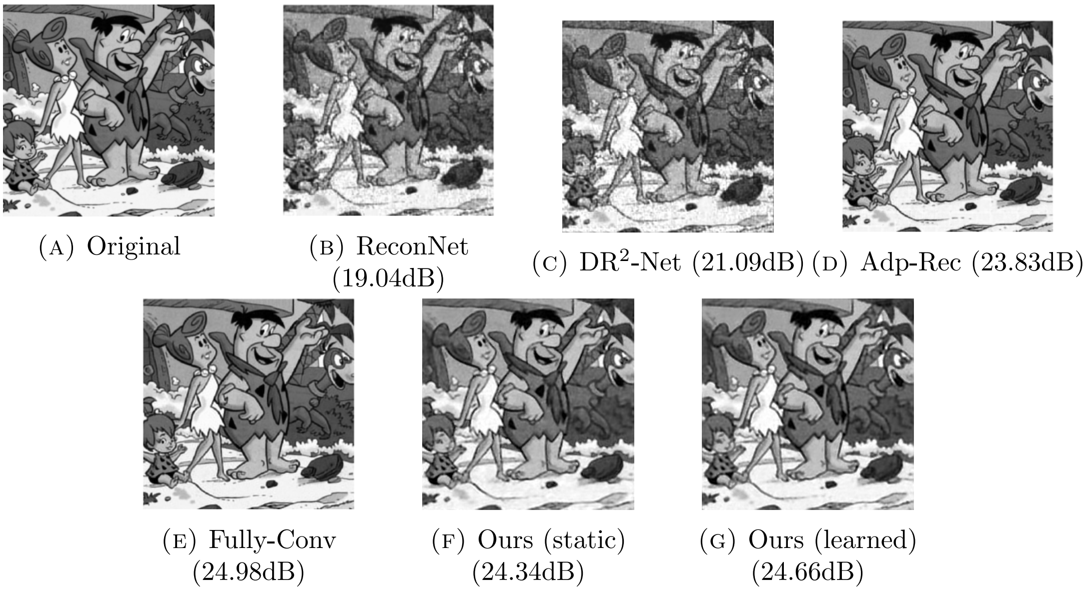
(D) Adp-Rec (21.67dB)

(E) Fully-Conv (22.49dB)

(F) Ours static (22.57dB)

(G) Ours learned (23.01dB)

FIGURE 3.5: The reconstruction result from the five methods, including two of ours, at compression ratio $R = 0.01$.



(A) Original

(B) ReconNet (19.04dB)

(C) DR^2-Net (21.09dB)

(D) Adp-Rec (23.83dB)

(E) Fully-Conv (24.98dB)

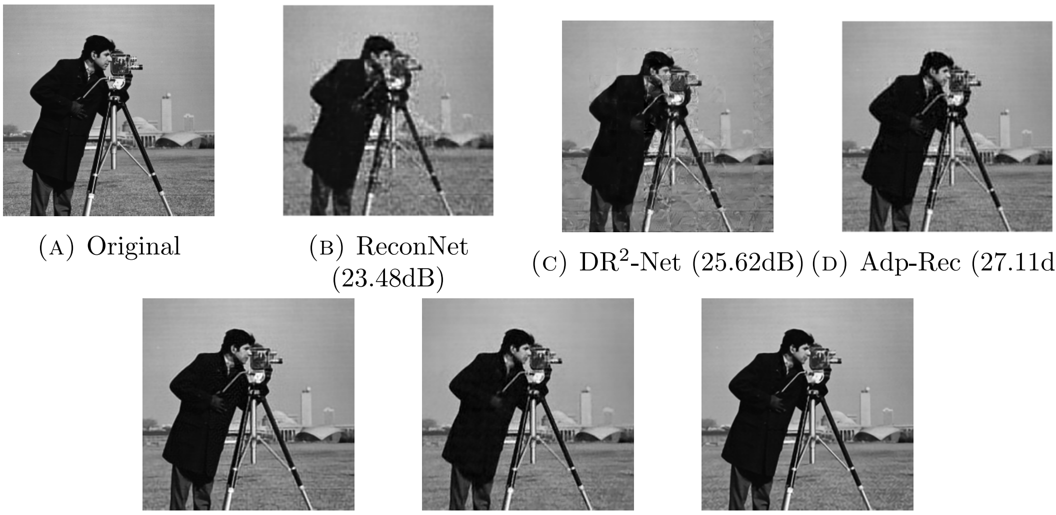
(F) Ours (static) (24.34dB)

(G) Ours (learned) (24.66dB)

FIGURE 3.6: The reconstruction result from the five methods, including two of ours, at compression ratio $R = 0.10$.

benefit of generating smaller intermediate feature maps, thereby reducing the computational burden at the image reconstruction stage.

Three sets of selected examples in Figure 3.5, 3.6 and 3.7 presents the benchmark images and their reconstruction results using different methods at measurement ratios of 0.01, 0.10 and 0.25 respectively. It is seen that our proposed network reconstructed more details than the other methods. This resulted in images that



(A) Original

(B) ReconNet (23.48dB)

(C) DR^2-Net (25.62dB)

(D) Adp-Rec (27.11dB)

(E) Fully-Conv (28.99dB)

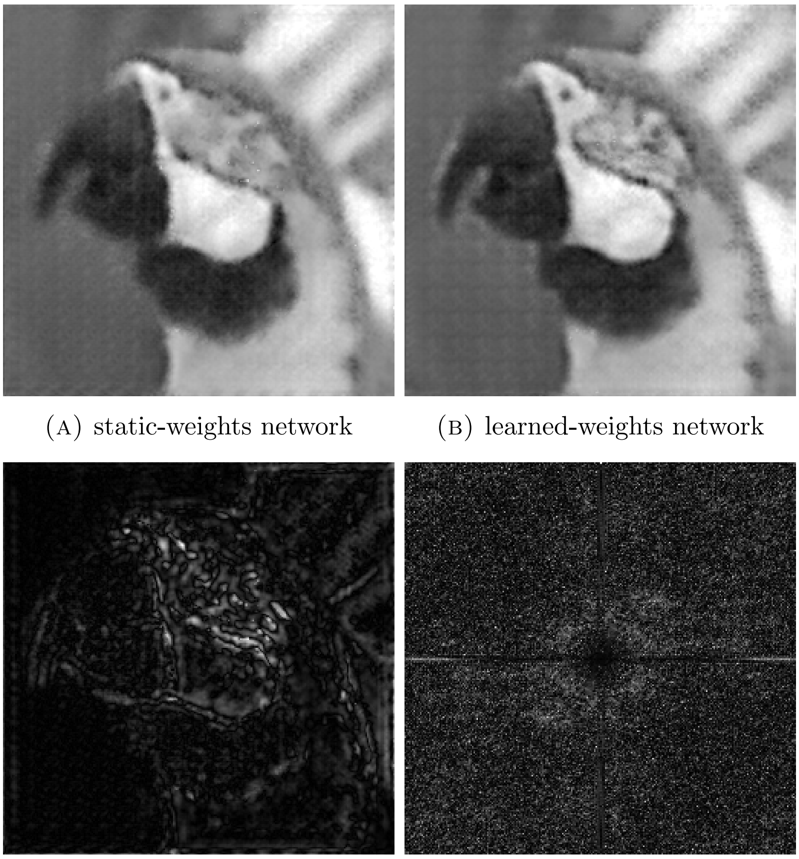
(F) Ours (static) (28.68dB)

(G) Ours (learned) (30.63dB)

FIGURE 3.7: The reconstruction result from the five methods, including two of ours, at compression ratio $R = 0.25$.

are visually sharper. At the lowest measurement ratio of 0.01, the block-based reconstruction strongly affected the quality of output images generated by the ReconNet, DR^2-Net and Adp-Rec. This effect was not observed in the results obtained from the Fully-Conv network and our proposed network. The reason for this is, both of these methods applied the fully convolutional layer to implement the image sensing and reconstruction. This approach brings two advantages to the network. First, in contrast to the fully-connected layer used in the former methods, it does not require an image splitting step. Therefore the network could be trained in an end-to-end fashion such that the reconstructed image was obtained with post-processing to smooth the output images. Second, the network does not require a fixed input-image size. This advantage introduces more flexibility to the network. It is also noticed that the blocking effect was eliminated for all results at the measurement ratio of 0.10 and 0.25. This indicates that the number of measurements was sufficient to achieve adequate reconstruction quality.

For the proposed network, the difference between the results due to the static patterns and the learned patterns is significant at the measurement ratio of 0.01. Figure 3.8 shows the reconstructed images for both cases their differences in both the spatial and frequency domains. It is observed, in both Figure 3.8 (C) and (D), that the main differences are in high-frequency image content. This result implies that learning binary weights can help preserve more detail and make the model



(A) static-weights network (B) learned-weights network

(C) image difference (D) difference of image FFT

Figure 3.8: The comparison of reconstructed image between our static-weights and learned-weights networks at compression ratio $R = 0.01$. Since the learned network yield a higher PSNR of 22.92, more details are reconstructed. The difference image (c) and the difference of their Fourier transform (d) show that the significant improvement of the learned-weights net is contributed to the high-frequency component under the same training procedure.

converge more quickly based on the same measurement ratio and training process, thereby reducing the training time.

Next, we evaluated our model on the ILSVRC2017 test dataset. Figure 3.9 shows the mean PSNR values of the reconstructed images on the large ILSVRC2017 test dataset. The figure depicts five bar charts; one for each of the five DCT-sparsity levels. At each sparsity level, the test images were reconstructed at three measurement ratios. Each bar represents the mean PSNR values calculated over all test images. For original images, it was seen that the model achieved good performance with mean PSNR values of 32.11 dB, 28.45 dB, and 22.69 dB, which is comparable to the results on the benchmark test images.

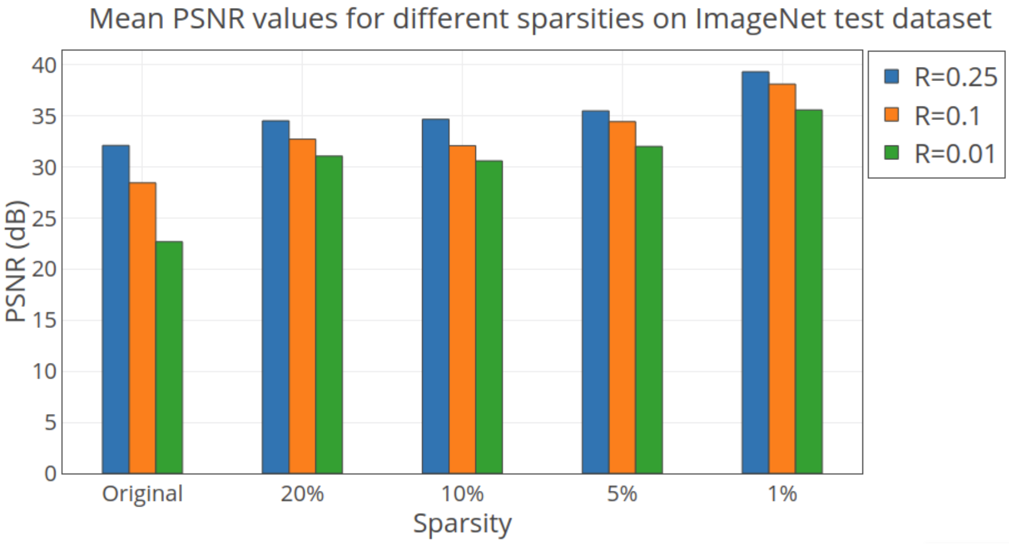


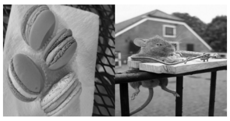
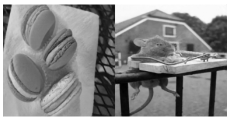
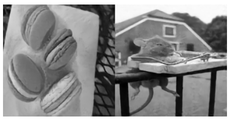
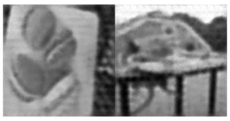
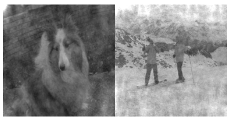
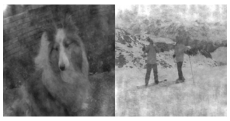
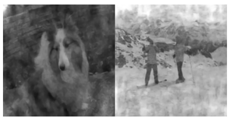
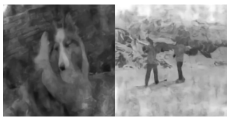
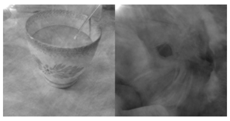
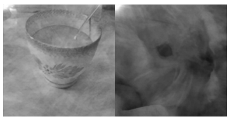
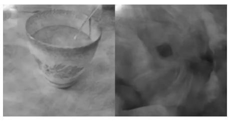
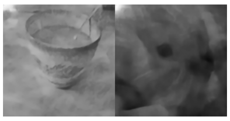
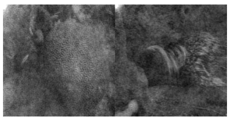
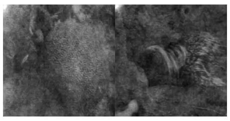
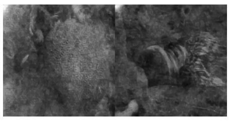
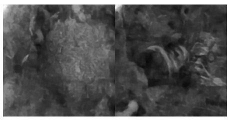
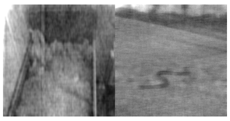
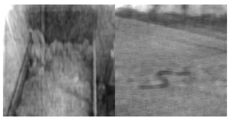
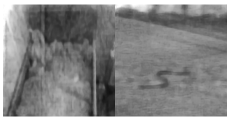
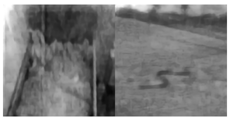
FIGURE 3.9: The evaluation of the learned binary model on the ImageNet test dataset. The trained learned-binary model was tested on the original ImageNet test dataset and the DCT-sparse images in three measurement ratios (R = 0.01, 0.1 and 0.25). For the dataset, we controlled the sparsity of the images in the DCT domain. Specifically, we fixed the sparsity of the original images in the DCT domain such that 20%, 10%, 5% and 1% of the original DCT coefficients were retained. The results show that the trained model works well on the large-scale image datasets, indicating the model's ability to generalise. It is also observed that the mean PSNR values increase with increasing sparsity. This denotes that the model also performs well on DCT-sparse images.

For DCT-sparse images, it was found that the PSNR values increase with increasing sparsity of images. This result implies that the DCT-sparse images are more easily reconstructed by our proposed network, which is aligned with the performance of the conventional CS reconstruction. Examples of reconstructed images are shown in Table 3.3, in which each row represents a different sparsity level, and each column is the reconstruction result for a particular measurement ratio.

3.5.3 Model training analysis with fixed and binary sampling schemes

This section provides an analysis of the model training process with the static and learned sampling patterns schemes. First, we analysed the training efficiency by monitoring the validation loss in both sampling schemes. The model validation was done for each of the 2000 training steps with the DIV2K validation image

Table 3.3: The sample images from reconstruction of the large scale test dataset. The rows denote the reconstruction at different sparsity in the DCT domain. The first row is the reconstruction of the original images. The second to the last rows show the reconstruction of the sparsity-controlled images. Specifically, the sparsity of the images is at 100%, 20%, 10%, 5% and 1% of the original images. The first column shows the ground truth images, and the second to the last columns show the reconstruction at the compression ratio of 0.25, 0.1 and 0.01.

Sparsity	Raw image	Reconstruction results at different measurement ratios		
		$R = 0.25$	$R = 0.1$	$R = 0.01$
original				
20%				
10%				
5%				
1%				

set, which was separate from the training set. Figure 3.10 shows the loss trend of the network, which was trained separately with the two different sampling patterns scheme at measurement ratios of 0.01, 0.1 and 0.25. We found that training with the learned patterns produced a faster loss reduction for all three measurement ratios than training with fixed patterns. When the measurement ratio was increased, the discrepancy between the losses of the two networks also increased. Furthermore, the network with learned patterns yielded a lower final loss than the fixed patterns network, especially for R of 0.1 and 0.25.

Even though the learned patterns network showed some instability compared to the fixed patterns network, it is still beneficial since it can be trained more

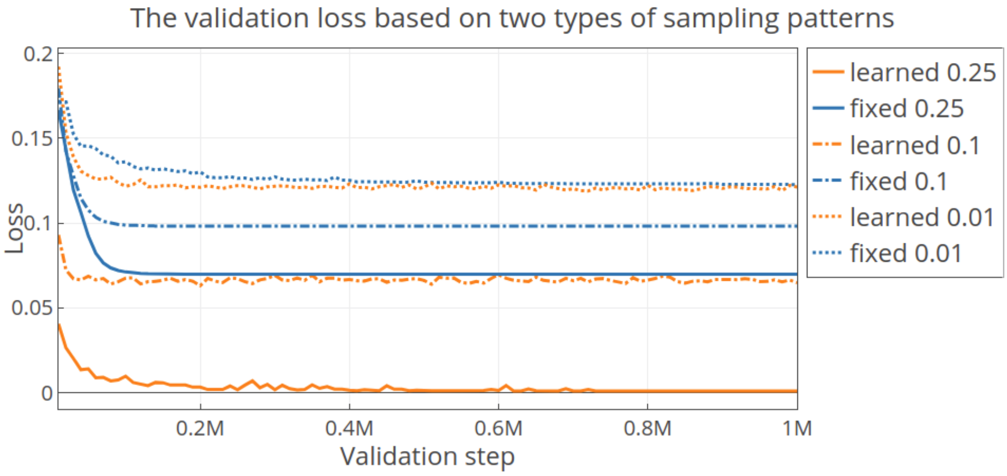


FIGURE 3.10: The validation losses based on static and learned binary patterns are in blue and orange traces, respectively. Each pattern type was validated for three measurement ratios (R = 0.01, 0.1 and 0.25). The training process for the learned binary weights was increased by updating the binary weights. As a result, its validation loss drops quicker than for the static patterns. Two losses at R = 0.01 are close at the end of the training, while the difference became significant when the measurement ratio increased. For $R = 0.25$, the loss of the learned-weights model was much lower at the beginning. Also, the training loss associated with the learned patterns was not as smooth as the static case. This is because the binarisation function mapped the real-valued weights to binary values and introduces perturbation to the training.

quickly. The instability was caused by the binarisation operations. Specifically, in the static-pattern scheme, only the real-valued weights of the network were not involved in the calculation of backpropagation, but the binary sampling patterns were not updated. In contrast, in the learned-pattern scheme, the binary weights were obtained from the binarisation of the real-valued version of the network after each updating step. This binarisation function introduced fluctuations in the gradient calculation and made the training progress less stable. However, the benefit of learning the binary weights is that it reduces the loss at a faster rate compared to the static one. As a result, the validation loss based on the learned weights was lower than that of the static weights at the same training step.

Next, we analysed the sparsity of learned patterns by exploring the percentage of pixels with value one during pattern update. In compressive sampling theory, we typically use a small number of dense sensing patterns (equal numbers of ones and zeros) in contrast with raster scan sensing in which each pattern is maximally sparse (contains one on pixel) and records the intensity of single pixel values one at a time. Conversely, the sparse patterns are more efficient for single pixel imaging hardware as they require less on-board memory usage. Our approach effectively adapts the sparsity of patterns according to the measurement ratios and hence finds an optimal compromise between sensing efficiency and hardware performance. We initialised all patterns using a single precision uniform distribution within the range $[-1, 1]$ (as required for model optimisation), which were subsequently binarised to form patterns with a similar number of ones and zeros. However, the number of ones decreased dramatically during training since the model at large sampling rates does not necessarily need dense patterns. In contrast, for a relatively small measurement ratio of $R = 0.01$, the number of ones remained consistently high, which suggests that more information was sampled by each pattern. As a result, the sampling patterns at $R = 0.1$ and $R = 0.25$ contain fewer ones compared to the patterns at $R = 0.01$, as seen in Figure 3.11. This variation, due to R, implies that the learning process can generate efficient binary sampling patterns that adapt to different measurements.

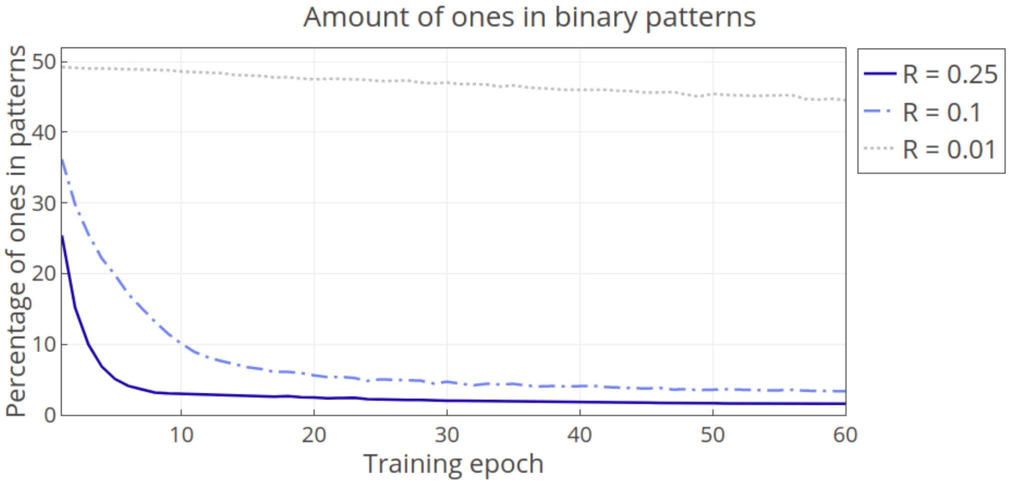


FIGURE 3.11: During training, the binary patterns adapt differently for each measurement ratio. Notice that the fraction of ones contained in the binary patterns is inversely at $R = 0.25$, while for the very small measurement ratio $R = 0.01$, the fraction of ones remains constant because more information needs to sensed by each pattern. This trend is closely related to Figure 3.12, and it implies that the initial binary patterns can be trained to be adaptive to the number of patterns.

The patterns learning process largely depend on the training data energy, which is determined by the Charbonnier loss. The variation of patterns according to the measurement ratio is currently self-optimised based on training with general real-world image data. It is foreseen that when dealing with images of a low signal-to-noise ratio (SNR), the noise energy may affect the reconstruction results. This implies, under that condition, the image energy (pixel intensity) may be recovered but the structure of image content might not be exactly maintained. As a result, the visual quality may be contaminated by the background noise. The potential approach of improving the model generalisation, especially for recovering low SNR images, is to fine-tune the model with an additional multi-scale structural similarity (MS-SSIM) loss [52], which has been demonstrated to have a good performance on image super-resolution task.

The pattern updating speed is determined by the measurement ratio, i.e. the number of patterns used. Figure 3.12 shows the percentage of the updated weights at each epoch. To simplify the visualisation, the chart only presents the updating track within the first 60 epochs. The quantity of updated weights in the remainder of the training becomes very small, which is not very informative and has therefore not been included in the plot. From the figure, we found that the updating of weights adapted to the number of sampling patterns. More sampling patterns were used in the network at the large measurement ratio; therefore, more weights needed to be updated after the random pattern initialisation. The number of weights for $R = 0.25$ was very large, and hence updating of the weights was intensive within the first ten epochs. Conversely, at smaller measurement ratios, patterns contained fewer pixels with value one compare with the initial state, and therefore the time taken to update weights was negligible. This behaviour implies the updating rate of the binary patterns adapted to the measurement ratio.

In addition, we present a set of training-optimised binary sampling patterns at three measurement ratios. It is seen in Figure 3.13 that the sparsity of the valid sensing pixel of the patterns is decreased while the measurement ratio decreases, i.e. the number of "on" pixels is decreased. It is worth noting that the patterns at high R tend to measure with the line pattern, while the patterns of low R is doing block sense. Although the interpretability of the patterns is not straightforward, a possible explanation, which is highly related to the general CNN training, is that the single high-R pattern focuses on fewer image features than the low-R pattern, which senses more mixed features in each pattern.

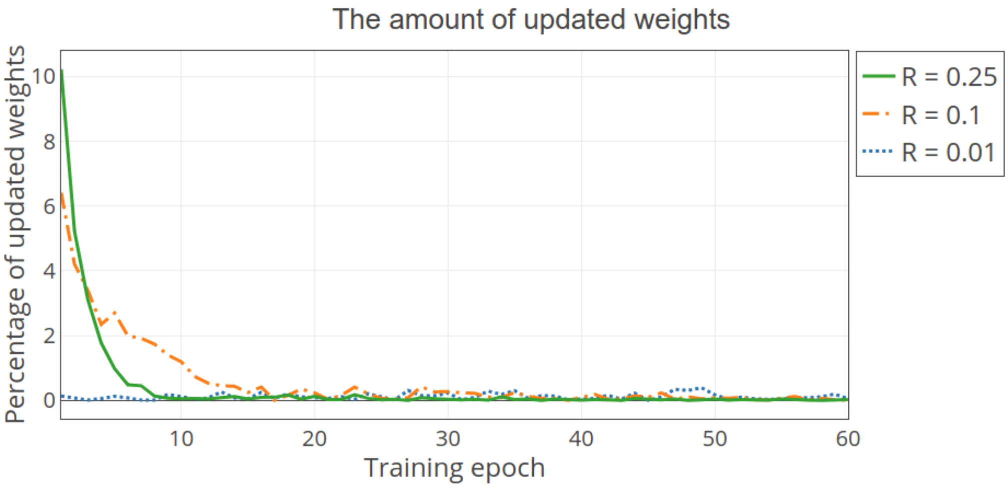


FIGURE 3.12: The amount of updated binary weights during training. The training process updated the patterns according to the total number of weights at different measurement ratios. Notice at $R = 0.25$ that the trace starts with a large figure which denotes the large update quantity of the sampling patterns. Most of the weights were updated frequently at the beginning and quickly became stable. On the contrary, the trace of $R = 0.01$ shows that the quantity of updated weights at $R = 0.01$ was consistently at low percentage.

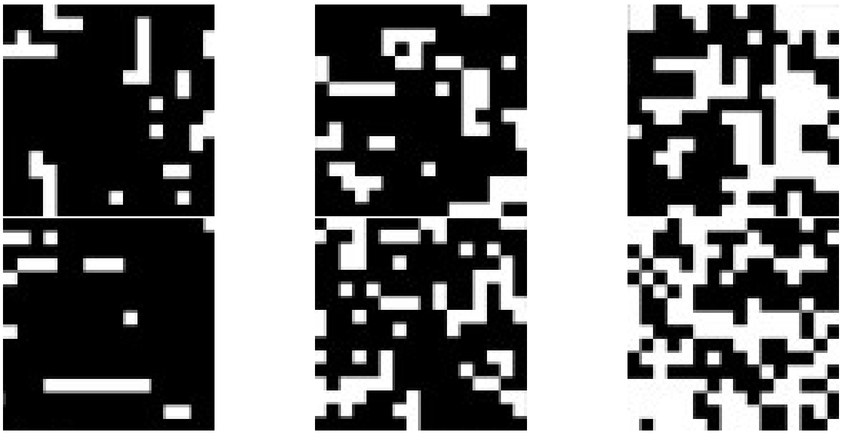


FIGURE 3.13: The examples of training-optimised binary pattern. First column: patterns at $R = 0.25$. Second column: patterns at $R = 0.1$. Third column: patterns at $R = 0.01$.

3.5.4 Analysis of the reconstruction efficiency

We analysed the computational efficiency of the network by calculating the time and space complexity. The results demonstrated that our model achieves the state of the art image quality at a low computational cost compared with competing methods.

Relative computational efficiency, with respect to the four other networks used in prior work (see Table 3.4), was measured in terms of model size (space complexity) and the number of operations (time complexity) of our network's image reconstruction layer. The comparison was based on the reconstruction of a single channel (greyscale) image of size 32 × 32 with a measurement ratio of $R = 0.01$ and is valid for any image size. The time and space complexity is formulated as

$$\textbf{Time} \sim \mathcal{O}(M^2 \cdot K^2 \cdot C_{in} \cdot C_{out}) \tag{3.9}$$

$$\textbf{Space} \sim \mathcal{O}(K^2 \cdot C_{in} \cdot C_{out}) \tag{3.10}$$

where M is the size of the feature map, K is the size of the kernel, C_{in} and C_{out} are the number of input and output channels separately.

Our network has the smallest model size among all the tested networks and lower time complexity than the Fully-Conv network. Note that the ReconNet, DR^2-Net and Adp-Rec perform fewer operations because these networks use fully-connected layers to reconstruct the image. However, the fully-connected layer can only be trained for a specific image size, which is less practical. Another disadvantage is that the fully-connected structured networks showed lower reconstruction quality. In contrast, the networks that adopt convolution layers can avoid this disadvantage by using the kernels with a small moving stride.

In the Fully-Conv network [37], images were directly reconstructed to the super-resolution size. The network then corrected the reconstruction error by applying convolution to the feature maps that had the same size as the super-resolution test image. However, the computational cost of the Fully-Conv network increases quadratically when the output image size is doubled since the time complexity is directly related to the square of the image size M^2. Instead of the previous approach, the proposed network firstly reconstructs the image at low resolution, and thereafter, the convolutional operation can be performed on small

Table 3.4: The five network methods were compared when restoring an image of size 32 × 32, with the sampling measurement ratio of R=0.01. The kernel size used in the Fully-Rec network and Our network is 16 × 16. The $\mathcal{O}_{space}$ and $\mathcal{O}_{time}$ denotes the space and time complexity of the reconstruction layer.

The image-restoration part of 5 methods				
Name	$\mathcal{O}_{space}$	$\mathcal{O}_{time}$	# Weights	Format
ReconNet	$1.024e^4$	$1.024e^4$	1024	32-bit
DR^2-Net	$1.024e^4$	$1.024e^4$	1024	32-bit
Adp-Rec	$1.024e^4$	$1.024e^4$	1024	32-bit
Fully-Conv	$2.560e^3$	$2.621e^6$	256	32-bit
Ours	$2.560e^3$	$6.553e^5$	256	1-bit

feature maps. The images are then up-scaled back to the original size only at the last layer. Therefore, the number of operations required by our network is related to $\frac{M^2}{4}$, which is four times less than the Fully-Conv network.

Table 3.5: The analysis of the number of weights in residual blocks of the 5 methods. The ReconNet and the Adp-Rec methods use plain convolutional structure and therefore have no blocks. Our LSHR scheme results in the deepest structure with the fewest weights.

The Residual-correction part in 4 methods				
Name	# Conv layers	# Weights	# Blocks	feature size
ReconNet	6	15650	N /A	32 × 32
DR2-Net	12	31300	4	32 × 32
Adp-Rec	6	15650	N/A	32 × 32
Ours	12	1152	6	16 × 16

The last part of our analysis evaluated the performance of the network for different numbers of residual blocks in our recursive structure. The depth of the recursive residual block affects the reconstruction accuracy. In principle, adding more residual blocks could improve the capability of the residual mapping, but in practice, training a deeper network is harder. It is seen in Figure 3.14 that the image quality increases by adding more blocks, and the best performance (time and accuracy) is obtained with the six-block structure. Adding more than six blocks leads to a degradation in image quality. It is also observed in Figure 3.14 that the reconstruction time increases linearly with the addition of blocks. It should be noted that the number of blocks does not affect the total number of weights since weights are shared between blocks. This contributed to an improvement in efficiency compared with the methods listed in Table 3.5.

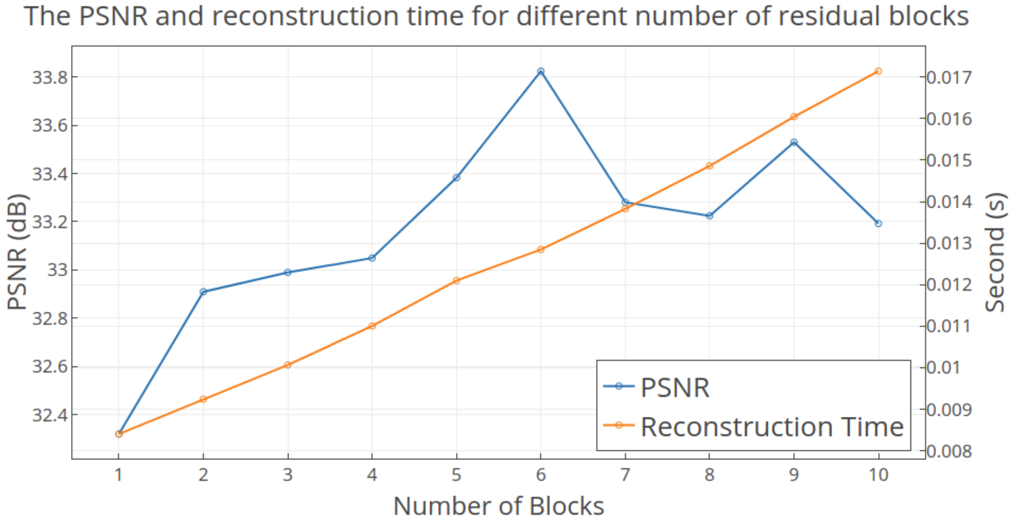


FIGURE 3.14: The average PSNR of test results and reconstruction time using different numbers of residual blocks in the recursive structure. The number of blocks used influences the performance. The PSNR value was maximized when six blocks were used in the recursive structure. The average reconstruction time increased approximately linearly.

3.5.5 Experiment results of incorporating with SPC hardware

For the purpose of demonstrating practical reconstruction accuracy, the proposed network was integrated with an SPC hardware setup that embodies the image degradation processes associated with single-pixel cameras. The hardware was modified from the previous setup, which is described in Section 2.2.1. The updated hardware was shown in Figure 3.15. It comprised a silicon planar photodetector with a purposely designed amplifier circuit, lenses and a light projector. The photodetector had a peak sensitivity at the wavelength of 930 nm, and its sensitive area was 93.6 mm^2. the circuit was connected to an Arduino Uno board which performed 10-bit analogue-to-digital conversion (1024 scales).

For evaluation purposes, the SPC hardware was set to measure the light intensity of test images from a database as an alternative to setting up unique object scenes. To this end, the test images were multiplied, in software, with each of the sampling patterns to form the modulated images. Then they were projected using

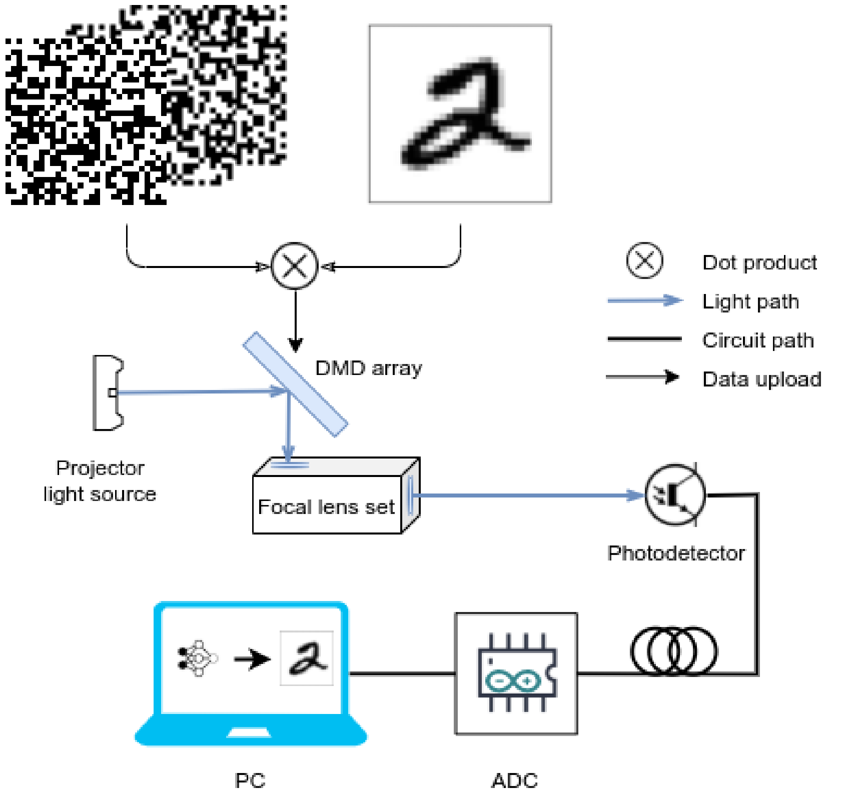


FIGURE 3.15: The schematic of the hardware setup. ADC: analogue to digital converter. The image was multiplied by the learned binary patterns and uploaded to the DMD array to do imaging sampling, and the photodetector records measurements.

a TI DLP LightCrafter evaluation module which consisted of a built-in DMD plane with a 608 × 684 array. The size of the sampling patterns is determined by the sensitivity of the photodetector and the analogue-to-digital conversion resolution. In the experiment described in this section, a good practical resolution for the sampling patterns was found to be 16x16 pixels. During the measuring process, each of the modulated test images was focused onto the photodetector using a set of lenses with focal lengths of 40 mm, 50 mm and 100 mm. In addition, a pair of polarisation filters with a fixed light attenuation rate was used to reduce the light intensity in front of the photodetector, thereby avoiding saturation of the detector's response. With this setup, the SPC hardware recorded the light intensity of the modulated images and sent these measurements as input data to the model.

The aim of the following experiment was to obtain super-resolution written characters from low-resolution samples. First, the MNIST dataset [53], which contains 10, 000 images of a single digit number, was used to train the network with the same training settings described in Section 3.4. Then the trained model was evaluated using 18 randomly selected test images of handwritten characters, nine-digit images from MNIST and nine images from the Omniglot datasets. The Omniglot dataset [54] consists of a set of natural language characters. The reason for using this dataset is to demonstrate that the trained model performs well on characters and writing styles not included in the training phase.

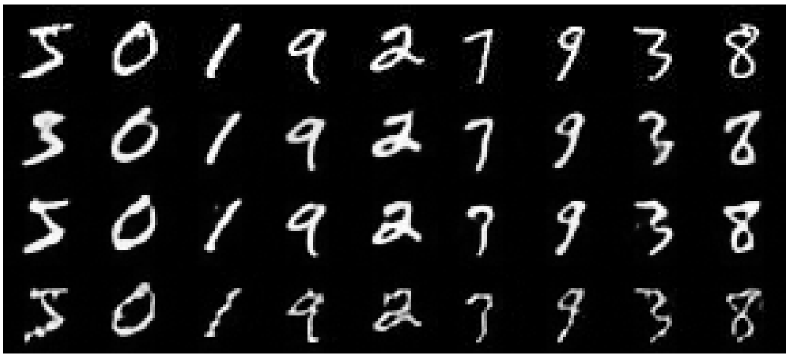


Figure 3.16: The figure shows the reconstruction results of 9 random selected MNIST handwritten numbers using the hardware measurements. The images at the top row are the original ground truth images, and the images from the second to the last row are reconstructed results at $R = 0.01$, 0.1 and 0.25.

The model reconstructed images directly from the photodetector measurements and output super-resolution images of size 32 × 32. We evaluated performance at the same measurement ratios used in Section 3.4.3. Results on MNIST and Omniglot are shown in Figure 3.16 and 3.17 respectively. It is observed that the reconstruction quality of the characters improved by increasing the number of measurements. However, artefacts in the reconstructed images can be seen. These are caused predominantly by noise in the hardware setup (e.g. by the amplifier circuit). The average SNR of the recorded measurement signal was 15.7dB. Moreover, it is noticed in the Figure 3.16 and 3.17 that the reconstructed images of $R = 0.25$ are more pixelated than those of $R = 0.1$ and $R = 0.01$, the latter resulting in better perceptual reconstruction quality. This is because when doing the down-sampling of the original image, the down-sampled image has a pixelating effect compared to the original image. The reconstruction at high measurement ratio is actually captured this high-frequency information, but the reconstruction at high measurement ratio did not. This was also seen in Figure 3.5, and Figure 3.8 that the high-frequency components were not fully recovered, resulting in a smoothing visual effect.

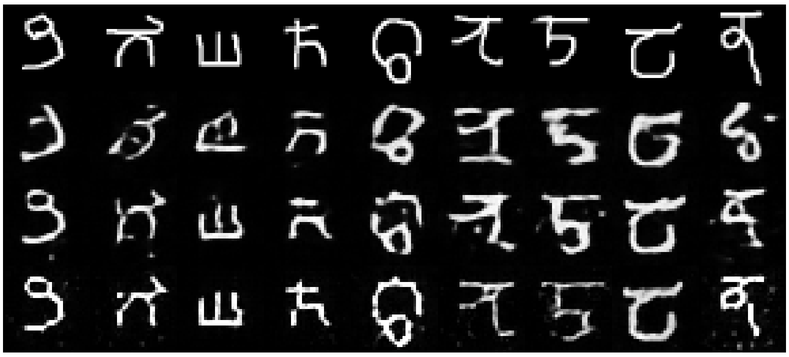


FIGURE 3.17: The figure shows the reconstruction results of 9 random selected Omniglot characters using the hardware measurements. The images at the top row are the original ground truth images, and the images from the second to the last row are reconstructed results at $R = 0.01$, 0.1 and 0.25.

3.6 Conclusions

This chapter presented a mixed-weights deep neural network solution for image reconstruction from compressively sensed measurements. The adaption of the neural network model to imaging sampling hardware is a challenging task while it is important to a real application of the compressive sensing image system. Although the simulation results of existing approaches demonstrated good performance, none of them was suitable for the real application of single-pixel imaging. It is due to the challenge when applying the real-valued patterns to perform high precision sensing We demonstrated that our network is more suitable than previous related methods, for the single-pixel camera application due to the following two factors.

First, the binary patterns used in our mixed-weights model are better suited for the digital mirror array. It simplifies the production of the sampling device. Our trained binary patterns only contain 1-bit values. These patterns can be easily displayed with the digital mirror array because there is no need to present greyscale values with mirror vibration using pulse with modulation.

Second, using binary patterns reduced the sensitivity requirement of the photodetector. Capturing the changes of light intensity due to the small changes of high-precision floating-point pattern values requires a high-sensitive photodetector. It is also a risk that this change may be overwhelmed by the circuit noise. Using the binary patterns instead makes the changes to be larger between different measurements of the light intensity. Therefore the sensitivity required for the photodetector can be less critical.

The scheme of low-resolution sampling and high-resolution reconstruction (LSHR) brings advantages to our model in terms of reconstruction accuracy and efficiency. With the LSHR, the network learned a set of low-resolution binary patterns to do a spatial sampling of the image and reconstructed the image with super-resolution details. The number of sampling patterns is directed related to the resolution of the image. We disentangle the image reconstruction task into low-resolution sampling and high-resolution refining. The low-resolution sampling assumes the initial reconstruction of the target image requires less information, and therefore quadratically reduced the required number of sampling patterns. It consequently saves more time at the sampling stage and leaves the details refining at the later stage. Once the initial reconstruction is done. The sub-net of high-resolution reconstruction refined the details using the optimised weights from the data-driven training. Therefore it overcomes the limitation of existing techniques that the reconstructed image must have the same size as the patterns on the sensing device.

Our end-to-end training strategy is beneficial to the network in terms of the adaptive pattern learning process. The model training led learned binary patterns to become progressively sparse and improves sampling efficiency. Updating the binary patterns adaptive to the different measurement ratios allows the network to be trained more quickly. We concluded from our evaluation that the self-learned binary sampling patterns demonstrated a better reconstruction accuracy comparing to the pre-generated static patterns. It also indicated a good generalisation of our model on the large-scale image dataset.

Though, this network is particularly proposed for a hardware realisation of the SPC imaging. It is not limited to this application and can be transferred to be the image reconstruction solution for other similar imaging modalities such as coded aperture imaging and structured light sensing.

Chapter 4

A review of retinal OCT imaging and its clinical applications

4.1 Introduction

The retina is a structured tissue located inside the posterior surface of an eyeball. It is often observed as a multi-layer structure. Each layer contains photoreceptor cells that are responsible for detecting the light intensity in different colours. The light information gathered by these photoreceptor cells is then sent to the brain via the optic nerve head, which is located next to the retina. The retina is an essential part of the optic nerve system. Many types of retinal damage can cause visual degradation or even permanent blindness. For example, diabetic macular oedemas can cause significant shape deformation of the retinal layers, thereby severely affecting the retina's ability to collect and process light. The important role of the retina in vision motivated early work resulting in the invention of the ophthalmoscope to check the retinal surface. Subsequently, ophthalmology developed as a discipline, followed by various additional imaging modalities for observing the finer details of the retinal structure.

Optical Coherence Tomography (OCT) is an approach to eye testing that offers a comprehensive 3D view of the tissue covering the retina region. Different from the ophthalmoscope, the OCT can produce images of the layers structure underneath the retinal surface. This imaging approach enables clinicians to diagnose eye diseases, such as glaucoma and diabetic retina injury, that are located deep in the tissue.

OCT also enables ophthalmologists to perform a quantitative analysis of the retinal structure. Specifically, OCT imaging allows a clinician to measure the thickness of the ganglion cell layer, which can be used to detect the early stages of ganglion cell loss [55]. Furthermore, the thickness measurement of the nerve fibre layer is often used as an essential metric to distinguish between retinas suffering from glaucoma and healthy retinas [56]. The presence of fluid-fill oedema inside the retinal layers is often used as a strong marker to determine the level of diabetic retinopathy [57].

In this chapter, we introduce the optical characteristics of healthy and diseased human retinas in OCT B-scan images and the state-of-the-art in morphology-based diagnosis methods for retinal diseases. We review five recent approaches to OCT image processing. Three of these methods are used for detecting retinal layer boundaries, and the other two for segmenting fluid-filled defects. We also discuss the advantages and limitations of these techniques, which motivated us to develop more accurate methods for the segmentation of retinal layers and defects.

4.2 OCT evolution for scanning human retina

OCT has emerged at the forefront of ophthalmic imaging since its introduction in clinics during the 1990s. It is a non-invasive imaging technique that generates cross-sectional images of the internal retinal structure in very high depth resolution (in range of 5-15 μm) and can achieve three-dimensional volumetric imaging [58]. The retinal structure, and defects within it, can be imaged in real-time. These distinctive properties of OCT make it a precise scanning modality in applications ranging from essential scientific and biological studies to various clinical practices in ophthalmology and, more recently, skin melanomas [59].

The OCT system produces cross-sectional and volumetric scanning that obtains the depth information from backscattered light. It utilises a broadband light source to generate a light beam that is separated by two optical arms into a reference beam and a sample beam. The sample beam travels through the pupil and vitreous and is aimed at the retinal region in vivo. The light backscattered from the different depths of the retinal tissue interfere with the reference beam. The resulting interference pattern is finally captured by the system detector to measure the reflected light intensity and thereafter the depth profile of the retina can be obtained. This is termed an axial scan, or A-scan [58, 60]. By moving the incident sample beam transversely in the retinal field and performing sequential A-scans, we can obtain a 2D cross-sectional image, as shown in Figure 4.1. This image, referred to as a B-scan, represents the optical backscattering from a cross-sectional region of the retinal tissue. The backscattering is usually converted into a grayscale or pseudocolour image, which facilitates the visualisation of the retinal tissue. The volumetric data is generated by scanning the sample beams in a raster, zig-zag, or other two-dimensional patterns. The 3D OCT displays the sequential cross-sectional images as a volume of 3D voxels. It contains comprehensive volumetric information of the retinal structure by combining multiple B-scan frames. Specifically, a 2D transverse view of the volume data, which is perpendicular to the axial direction, can provide a surface image at different depths and is referred to C-scan.

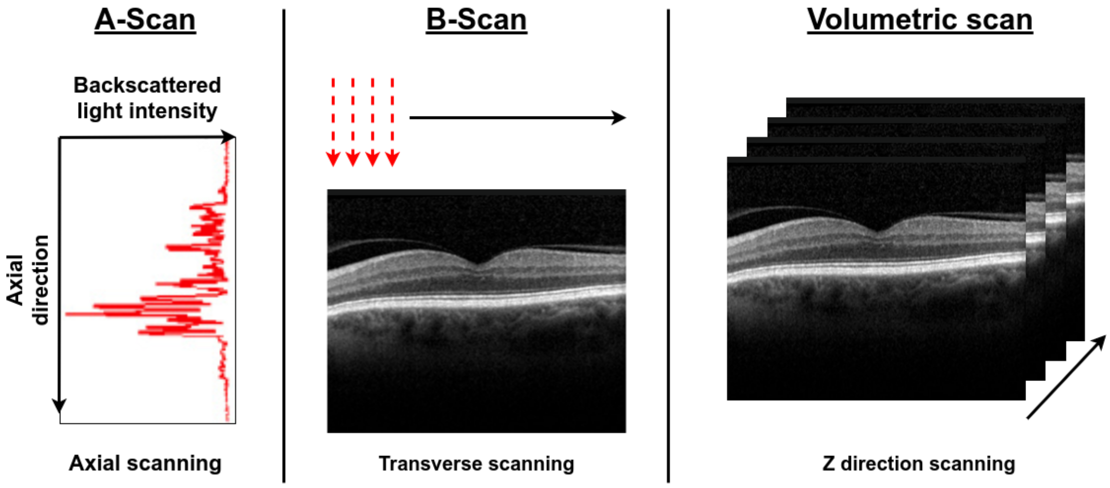


Figure 4.1: The optical coherence tomography measures the backscattering light intensity of the sample beam. The measurements of light versus the depth are known as axial scans (A-scans). The cross-sectional image (B-scan) is created by moving the beam to acquire a series of axial scans in a transverse direction. It produces a two-dimensional dataset that can be presented by using a grey or a false colour. The volumetric dataset can be created by moving the sampling beam in a 2D pattern across the retinal area. The figure shows volume data generated by a raster scanning, which produces a series of B-scan images. Other OCT system may implement it in a radial pattern.

Various OCT imaging modalities have been invented for capturing the detailed structure of the human retina. At the early stage, the time-domain OCT (TD-OCT) system dominated clinical applications. A typical TD-OCT can perform 400 A-scans per second. The A-scan can achieve an axial resolution of 8–10 μm in tissue [60]. Often, a TD-OCT system will scan a small region of the retinal area, producing six radial B-scans. Then it generates a retinal thickness map of the target region by estimating the remaining area using these six B-scans. However, this approach may miss the details of a retinal lesion if it is located between B-scans. In addition, subtle eye movement is inevitable during the scanning procedure. The scan rate of TD-OCT is not fast enough to avoid image artefacts due to eye movement. These factors present a barrier to imaging large target areas within the retina.

The limitations of TD-OCT were partly addressed by the introduction of spectral-domain OCT (SD-OCT or Fourier-domain OCT) and swept-source OCT. These configurations offer significant improvements in scan rate and axial resolution. These techniques can produce scanning speeds of 20,000–40,000 A-scans per second with an axial resolution of 5–7 μm in tissue [61, 62]. Benefiting from the high scanning speed, high-resolution 3D reconstructions of the retina can be generated. This can give a better representation of the optical disk topography and the thickness map of different retinal layers, such as nerve fibre layer and retinal ganglion cell layer [63]. As a result, spectral-domain OCT instruments reduce the risk of missing defects such as subretinal fluid regions, pigmented epithelial lesions or retinal oedema regions in AMD patients, by performing more intensive sampling. It also reduces the risk of generating unwanted artefacts due to involuntary eye movements. Compared with TD-OCT, SD-OCT can provide more useful guidance for surgical planning, such as vitreoretinal interface abnormalities [64].

This technique has been widely adopted in hospitals and high-street optician stores. Additionally, the SD-OCT system is extensively applied to scientific research fields of posterior segment diseases, which are associated with tissues like macular, choroid, retinal nerve fibre layer, and optic nerve. Based on SD-OCT images, ophthalmic studies demonstrated valuable morphological and functional information of individual layers [65–74]. Specifically, they presented the SD-OCT B-scan image of symptoms of photoreceptor layer impairment, such as macular holes, diabetic macular oedema, age-related macular degeneration, and macular dystrophies. Some clinical trials also report that SD-OCT is used to track treatment and contribute to a better knowledge and experience of disease pathogenesis [66, 68–70].

4.3 Retinal structures and lesions in OCT images

Since the SD-OCT is used as a non-invasive method to provide detailed imaging of the retina, it has led to a step-change in ophthalmic practice. It allows the detection of various macular and choroidal diseases and can be used to determine the severity of these conditions. Typical regions for clinical analysis are the retinal fovea and optical nerve head, where most symptoms of retinal diseases appear. In this section, we describe the retinal structure and the appearance of retinal lesions in these regions, which are associated with our work on OCT image analysis in Chapter 5, 6, and7.

4.3.1 Retinal structure of healthy eyes

A healthy retinal fovea consists of multiple layer-structured tissues that have different optical reflectivity in OCT images. In a grayscale OCT image, the region of the retinal fovea is generally observed having twelve layers, seen in Figure 4.2. The region of most interests to clinicians are layers between the inner retinal surface (the upper boundary of the nerve fibre layer) and the choroid surface (the bottom boundary of the pigment epithelium). Among them, both the nerve fibre layer and the retinal pigment epithelium usually present the highest reflectivity. The outer nuclear layer has the lowest reflectivity. Other layers in this range have medium reflectivity.

Apart from the retinal region, some other tissues are appearing in OCT B-scan images that worth noticing.

OCT images also contain normal structures found in healthy subjects such as the vitreous region, located above the inner retinal surface. The vitreous gel, within the vitreous region, is usually invisible due to its optical transparency. However, in high-resolution OCT images, the posterior cortical vitreous, which touches the retinal surface, may appear due to its high optical reflectivity, introducing variety to

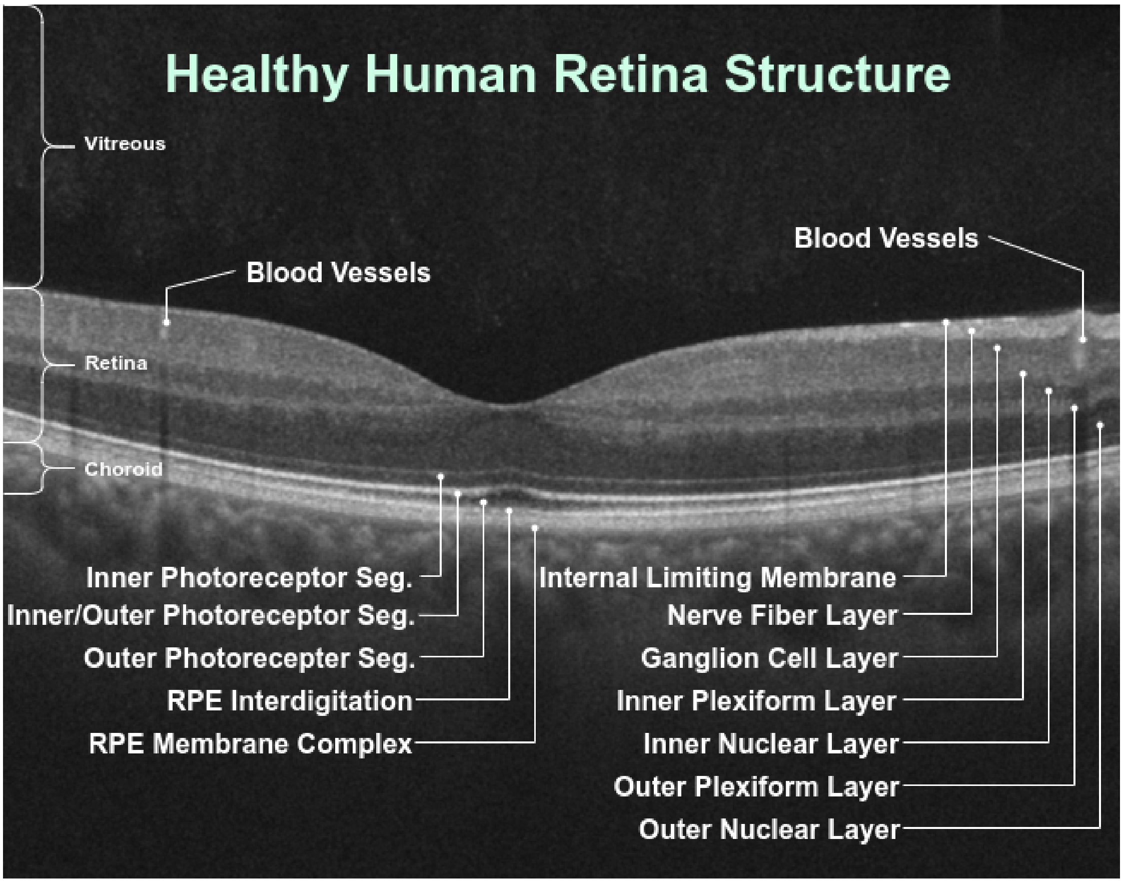


Figure 4.2: A normal retina at the fovea region. Twelve layers are presented in the OCT B-scan image.

the retinal structure presented in the image. Furthermore, the choroidal junction, a region at the outer choroid, can also be imaged by SD-OCT or swept-source OCT with improved depth imaging. The choroidal junction has variable reflectivity and shows notable changes in texture. Since it touches the choroid, it may affect the measurement accuracy of the choroidal thickness, which has also been the focus of recent studies [75–77].

4.3.2 OCT image of diabetic macular oedema

Diabetic macular oedema (DMO), occurring as a complication of diabetes, is a common ocular disease that causes vision loss due to retinal vascular damage. Before the use of OCT imaging, a clinician would identify the DMO characteristics from the retina surface image, captured using fundus microscopy. Now, the OCT imaging enables more reliable identification of the DMO by presenting oedemas in a cross-sectional view. Consequently, three DMO characteristics can be identified, which are only observable in OCT images. These three symptoms are diffuse retinal thickening, cystoid macular oedema (CMO), and serous macular detachment, seen Figure 4.3.

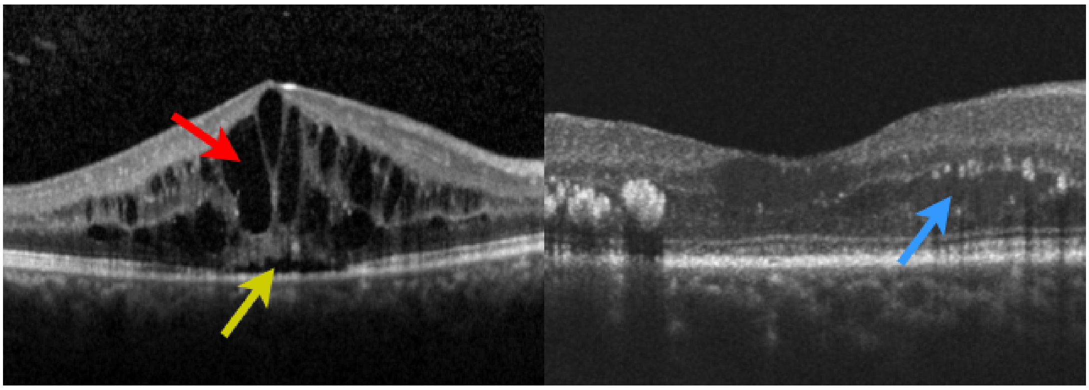


FIGURE 4.3: The diabetic macular oedema in retinal OCT B-scan image. The figure presents three types of lesions related to diabetic retinopathy. The diffuse retinal thickening is indicated by the blue arrow in the right figure. It appears as a thickened outer plexiform layer in the intraretinal region. The cystoid macular oedema and the serous macular detachment are indicated in the left figure by the red and green arrows separately.

These three characteristics can be observed at different stages of DMO. The diffuse retinal thickening is an early sign of diabetic retinopathy. It usually appears in the INL and (or) OPL layers. The affected layers usually become thicker because of the swelling, and their light reflectivity is attenuated consequently. This thickening may further develop to visually sponge-like oedemas. The CMO is formed by the accumulation of intraretinal fluid in the OPL layer. The cysts can merge to larger cavities during the development. It is widely agreed that the reflectivity of cyst content, as well as cyst size, has a prognostic significance of the severity level [57]. In later stages of the disease, serious retinal detachment may occur after a long-term development of diabetic retinopathy. The fluid may separate the intraretinal layers from the subretinal layer, resulting in a gap between them.

The non-invasive benefit of OCT imaging has led to a reduction in the use of fluorescence angiography. Therefore, the approaches to identification and the quantification of lesions in the OCT image becomes essential for predicting oedema development. Usually, lesions share a common feature in OCT images; namely, they result in a visible thickness change of the retina. However, diffuse oedemas have higher light reflectivity than the other two types of lesions [78, 79]. These differences require us to develop new analytical approaches applicable to OCT images. The quantitative analysis of the macular oedema is also of importance to clinical diagnosis. Compared to fluorescence angiography-based quantity estimation, the OCT-based quantity measurement offers a more precise assessment of retinal thickness at different macular regions, and this data may be useful in analysing treatment efficacy [80]. In addition to these two aspects, the ability to distinguish the relationship between the posterior vitreous and the retina is also required in some cases. It is because the existence of vitreous-macular adhesion in patients with DMO is a critical factor in the decision to perform vitreous surgery.

4.3.3 OCT images of central and branch retinal vein occlusions

OCT images can clearly present most of the lesions of the central and branch retinal vein occlusion (CRVO, BRVO), such as macular oedema, retinal haemorrhages, and cotton wool spots. These lesions, appearing with low reflectivity, are formed by the accumulation of tissue fluid at the centre and outer retinal region. These lesions have similar appearances as each other, and they may change the appearance of the retinal structure due to their complex pathology. Specifically, the severe leaking of fluid from oedema may cause a macular hole that affects a vast inner retina region, changing the reflectivity of other layers, seen in Figure 4.4. Subretinal detachments may appear as a low-reflective layer, which is located above the retinal pigmented epithelium layer. These two regions may have a similar appearance. Subretinal detachments may also have a similar appearance to shadows generated by retinal haemorrhaging. Due to their similarity in appearance, there is a risk of confusing the two conditions. In addition, the cotton wool spots, which is a strong-reflective structure, can attenuate the underlying reflectivity. This shadow effect is very similar to the shadows generated by vessels, which also leads to a high risk of confusion.

In the clinic, the severity of these lesions is assessed by measuring the retinal thickness and oedema volume. A 3D-volume OCT imaging of the CRVO and BRVO provides a visualisation of the macular thickness and oedema volume, which is helpful for the medical or surgical treatment and followup treatment tracking.

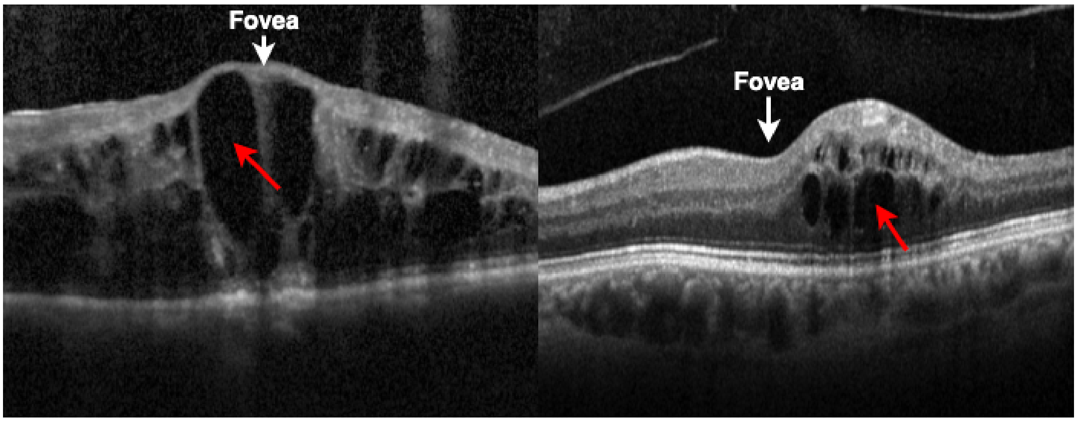


Figure 4.4: OCT image of retinal vein occlusion. The CRVO is shown on the left, and the BRVO is on the right. The CRVO presented large fluid oedemas at the fovea point. The BRVO is located at either branch of the fovea.

4.3.4 OCT images of pigment epithelial detachments

The pigment epithelial detachment (PED) is an easily recognisable symptom among macular diseases. It is a common disorder that occurs in company with age-related macular degeneration. Usually, the PEDs are shown as clear and optically empty space under the RPE layer, seen Figure 4.5. The blood on its surface causes a highly reflective appearance, and therefore the deeper tissue is under a shadow. Identifying the PEDs has significant diagnostic value since it indicates that the existence of new choroidal vessels at PEDs is related to inferior visual performance [81, 82].

4.4 Review of approaches to retinal structure and defects segmentation

Fast, high-resolution OCT scanning provides accurate visualisation data for retinopathy analysis. To do an analysis systematically, clinicians often require further information extracted from the image, such as layer thickness and oedema size. However, the annotation and segmentation of retinal structure is always a challenging and time-consuming task for the clinician. There are three main challenges for manual delineation of the retinal content. First, OCT systems with

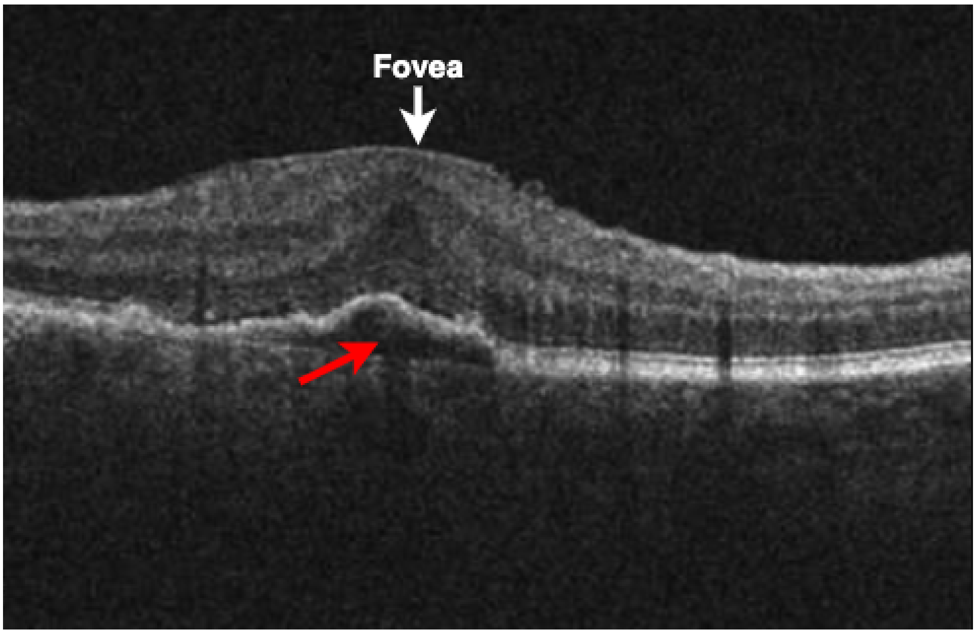


Figure 4.5: A cross-sectional retinal image with early-stage pigment epithelium detachment (PED), indicated by the red arrow. The PED happens in the sub-pigment area.

high acquisition rates yield large amounts of 3D data. These data consist of hundreds of B-scan images in each volume. A well-considered quantification analysis by manual annotation requires significant processing time, which is impractical for clinical applications. Second, the manual segmentation of CMO areas can become very subjective. This issue generally occurs in severe retinopathy and can result in a high diffusion of boundaries between healthy and diseased tissue. This diffusion region increases the risk of generating inconsistencies in manual annotation between different clinicians. Third, the imaging acquisition process presents deviation due to involuntary movements of the eyes and head of the subject. These movements affect continuity and smoothness within interval frames of the OCT volumes. This issue can also introduce difficulties to manual annotation. Hence, various automated segmentation approaches have been developed to produce efficient and stable segmentation of retinal content [2, 3, 83, 84].

In this section, we review the recently proposed segmentation algorithms in the literature. These approaches focused on the OCT image processing of the retinal fovea region. The B-scan image at this region presents a foveal structure where the horizontal boundaries of the inner retinal layers merge at the fovea point. The retinal defects, caused by intraretinal oedemas or layer detachment, often cause structural changes in this area. Therefore, our review mainly focuses on two aspects: the retinal layer segmentation and macular oedema segmentation. We also systematically discuss the features and limitations of each of these approaches.

4.4.1 Segmentation of retinal layer boundaries

As a primary and important feature, determining the locations of the retinal layer boundaries has always been the first step in clinical diagnosis on the retinal thickness [85]. Early OCT technology was only able to detect high-reflectivity layers. This was possible due to the profile of such layers being conspicuous in low-resolution TD-OCT images. At the same time, the speckle noise, which commonly exists in OCT images, is also an inevitable factor affecting the performance of segmentation methods and often contaminates the clear view of the retinal structure of the low-reflectivity layers. Koozekanani *et al.* [86] proposed a method for layer thickness measurement. It applied an edge-detection kernel on B-scans to extract the upper and lower retinal boundaries. Although this algorithm was demonstrated to be robust to noise, it was specifically optimised only for the detection of boundaries of the retinal region. Accurate segmentation of inner boundaries was significantly affected by the speckle noise. Benefiting from the continuous improvement of imaging resolution, Chan *et al.* [87] described an approach to segment retinal layers from Stratus OCT and ultra-high-resolution OCT (UHR-OCT) images. The proposed algorithm determined the boundary locations by applying several adaptive thresholds of pixel intensity to each A-scan of the cross-sectional images. However, this algorithm can only perform a four-layer segmentation, which is a relatively coarse segmentation of the retina structure.

In recent research, many approaches have been proposed to achieve better performance in terms of segmentation accuracy. We reviewed three representative methods and described these in detail in the remainder of this subsection.

Ahmet *et al.* reported an algorithm that enhances the content of the retinal structure and segments the inner layers [1]. The inevitable speckle noise of the OCT imaging system may decrease the sharpness of the retinal layer boundaries in OCT images. However, the proposed method can automatically detect boundaries of seven layers within the noisy retinal content and measured their thickness. This achievement was attributed to two image enhancement techniques. First, a 2D edge enhancement scheme was applied to the OCT image. This scheme applied a customised 2D filter to amplify the image pixel gradient of a retinal boundary in the axial direction. Second, a greyscale-level mapping technique was used to suppress speckle noise while preserving the image contrast between different retinal layers, resulting in clearer image content. This algorithm detected the retinal layer boundaries in five steps: 1) alignment of A-scans, 2) greyscale-level mapping, 3) directional filtering, 4) edge detection, and 5) model-based decision making.

The first three steps were designed for image enhancement. At the first step, the algorithm realigned each A-scan within a B-scan image based on the position of the RPE layer, which was recognised as the most prominent peak. It is worth knowing that this alignment process is only needed for the TD-OCT image due to the significant artefacts generated by eye movement. Conversely, SD-OCT provides high-speed scanning, and therefore does not require this step. At the second step, two greyscale-level mapping functions were used to enhance the visibility of inconspicuous boundaries. These boundaries were usually located between two adjacent layers, such as the region of RPE and ILM layers, that have a similar reflectivity. The mapping functions, which are denoted as G1 and G2, were calculated by an expectation-maximisation algorithm. Figure 4.6 shows two calculated functions and resulting images. It is seen that, after the mapping process, the weak layer boundaries become clearer. At the third step, a 2D directional filter, based on the previously published method [88] was employed to address edge blurring. The 2D frequency response of the filter is shown in Figure 4.6, top right. Since the edge direction is mostly horizontal, this filter preserved the image content in the axial direction and decreased the speckle noise by smoothing the image in the horizontal direction.

After the first three preprocessing steps, the boundaries were segmented with a two-step operation. First, an edge detection kernel was applied to the image. The kernel was designed based on the first derivative of the pixel gradient in the axial direction. It first detected the boundaries between bright and dark layers with bright on the top, and then moved to the boundaries with dark on the top. After the kernel process, non-maximum suppression and hysteresis thresholding were used to mark the local peak-value pixels as boundaries. At the second stage, the first and last boundaries, the ILM and RPE boundaries, respectively, were labelled. These two boundaries were taken as reference points to label other boundaries

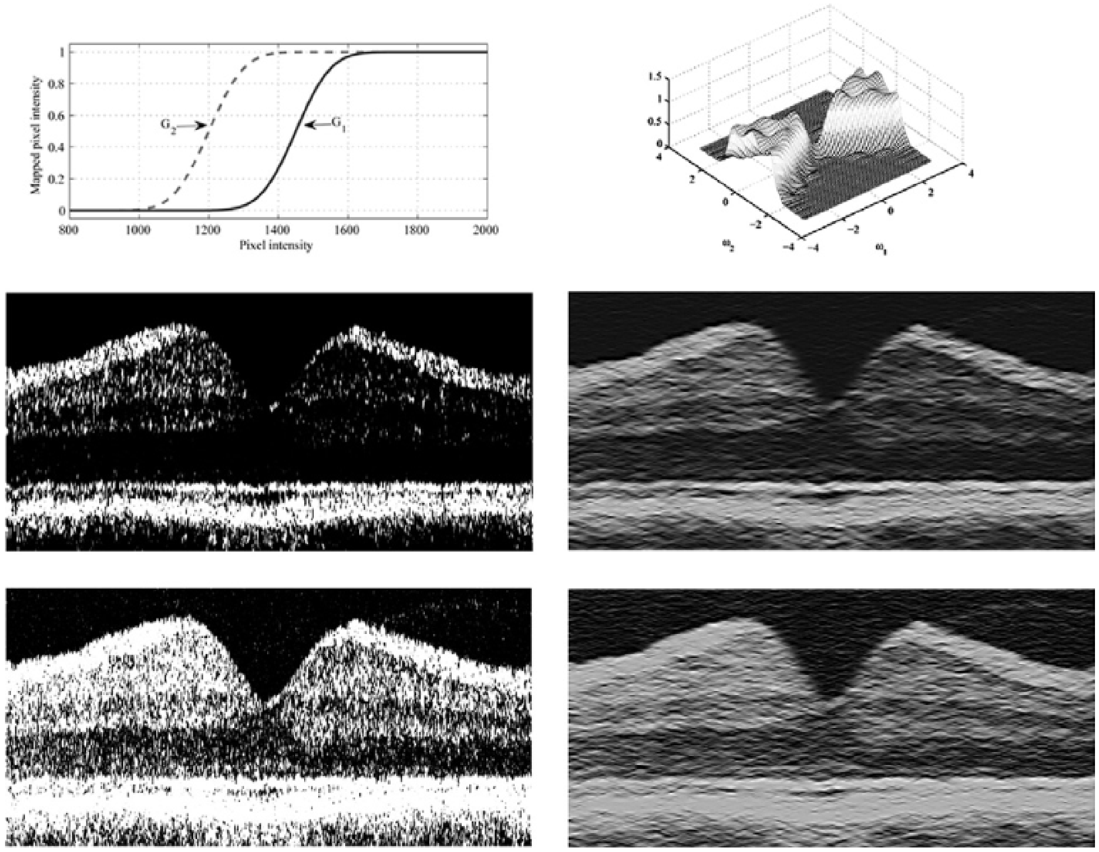


FIGURE 4.6: Retinal layer segmentation steps of grey-level mapping and directional filtering applied to a typical TD OCT image. Figure reproduced with permission from [1]. The first row, left: Two functions (G1 and G2) used for grey-level mapping. The second row, left: Image after grey-level mapping with G1 depicts boundaries of high reflectivity layers (NFL, IPL, GCL, PIOS, RPE). The third row, left: Image after grey-level mapping with G2 depicts the remaining boundaries. The first row, right: The frequency response of a wedge-shaped 2D directional filter. The second row, right: The image displayed in the second row left after directional filtering. The third row, right: The image displayed in the third row, left, after directional filtering.

with respect to their location. The final output was six layers segmented with seven boundaries, as shown in Figure 4.7.

This method had the advantage of addressing the issues of variation in image brightness between different retinal layers. However, the kernel-based edge detection mainly depends on the gradient variance. Therefore it was not able to perform the detection of boundaries with similar reflectivity. For example, the boundaries between IPL and GCL were not detected. In addition, it is seen from the segmentation result (Figure 4.7) that the algorithm can not detect the boundaries of very thin layers, such as the ellipsoid zone at the outer retinal region.

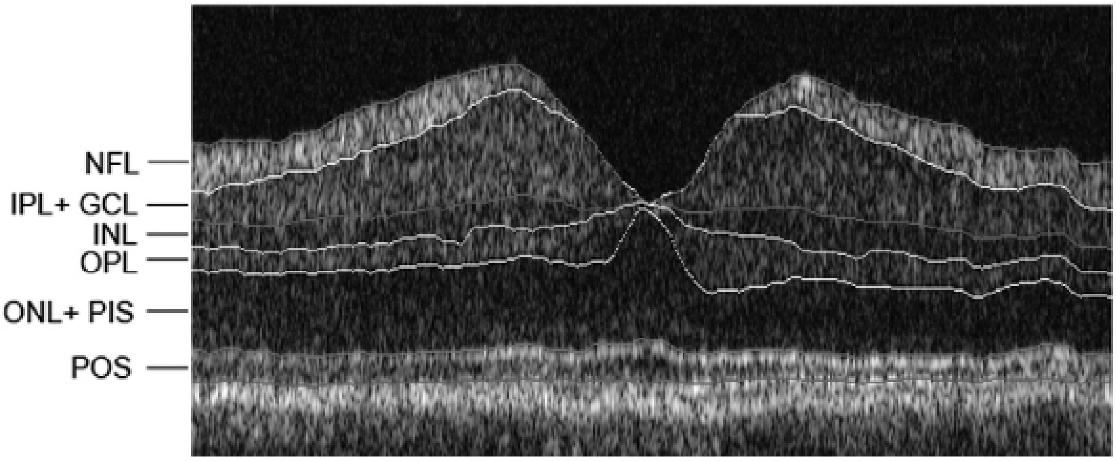


FIGURE 4.7: The boundary segmentation results from [1], presented with re-usage permission. Six retinal layers were segmented and labelled.

Stephanie *et al.* proposed an approach to retinal layer segmentation using graph theory and dynamic programming, which was proven to be successful in many image segmentation tasks [2]. Comparing this approach to the previously introduced method [1], that used fixed detection kernels, this method demonstrates the advantage of applying graph theory and dynamic programming, which offers a more flexible and accurate approach to the retinal layer segmentation problem.

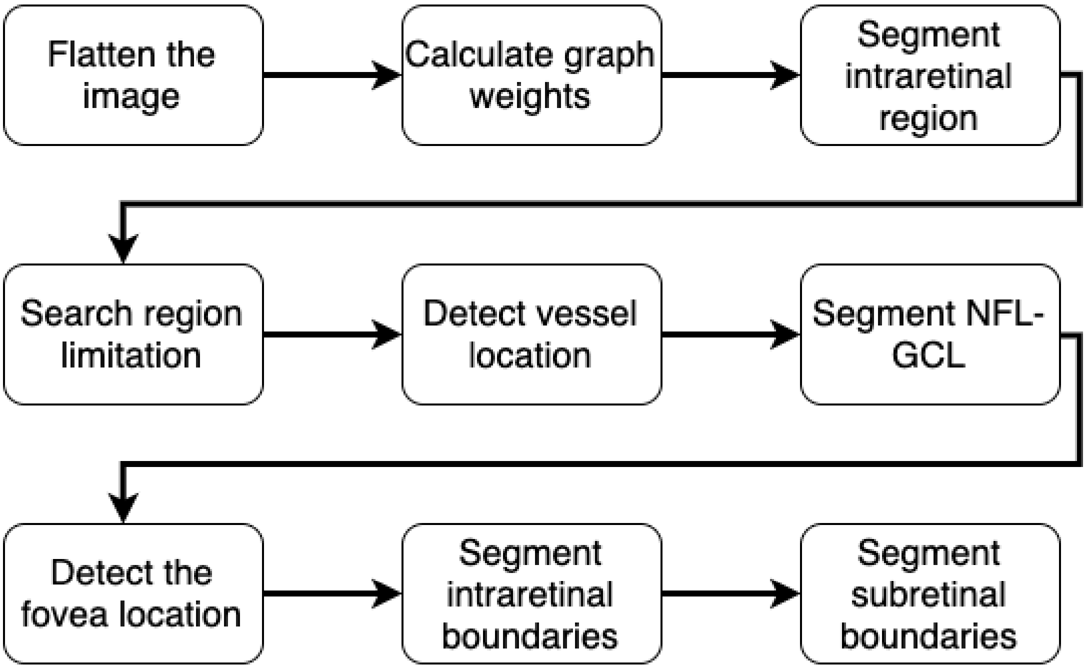

Flatten the image
Calculate graph weights
Segment intraretinal region
Search region limitation
Detect vessel location
Segment NFL-GCL
Detect the fovea location
Segment intraretinal boundaries
Segment subretinal boundaries

FIGURE 4.8: The nine-step segmentation algorithm schematic for eight retinal layer boundary in SD-OCT images.

This algorithm successfully detected eight boundaries and therefore separated seven retinal layers between those boundaries. The whole process of the eight-boundary detection was implemented by using a tailored procedure in 9 main steps, seen in Figure 4.8 for the details. For the boundary detection of each layer, the proposed graph-based segmentation can be summarised in 4 steps, seen in Figure 4.9.

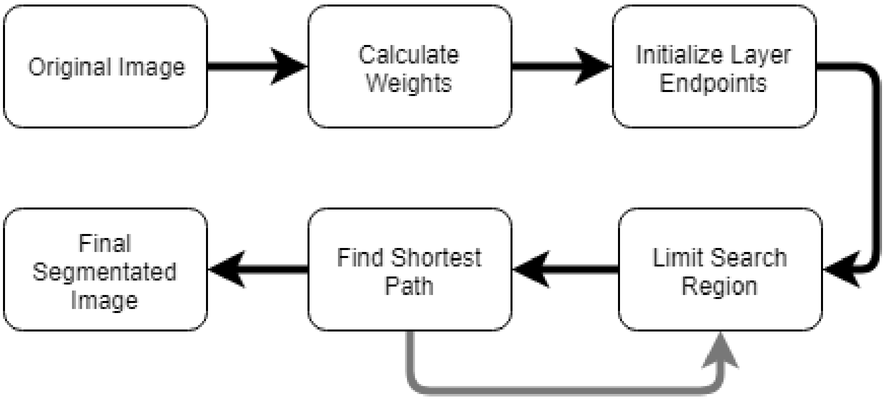


FIGURE 4.9: The boundary segmentation algorithm schematic for individual layer [2]

The graph search algorithm, which is used to determine the location of an individual boundary, is the core function in the tailored procedure. It first represented the OCT image as a graph. Each pixel was converted to a graph node, and the differences between intensity values of adjacent pixels were converted to their edge weight. The task of boundary segmentation was solved by finding a preferred path such that the total sum of the edge weights on this path is minimised. To achieve this goal, they utilised the Dijkstra's algorithm [89] with some constrains to search the optimal path. The algorithm assumes that the layer boundary should cross the entire OCT image in the horizontal direction. Based on this assumption, they first initialised the start and end search points to be at the left and right edge of the image. Second, they limited the search area to prevent the searched path diverting from the target area to the neighbouring region, which contains similar characteristics. With these two search constrains, the Dijkstra's algorithm can be applied to find the final boundary position. Once the first boundary was segmented, the procedure was repeated to detect other boundaries in depth ordered sequence. However, not all boundaries in the retinal OCT images, were correctly detected using this method (for example, see Figure 4.10).

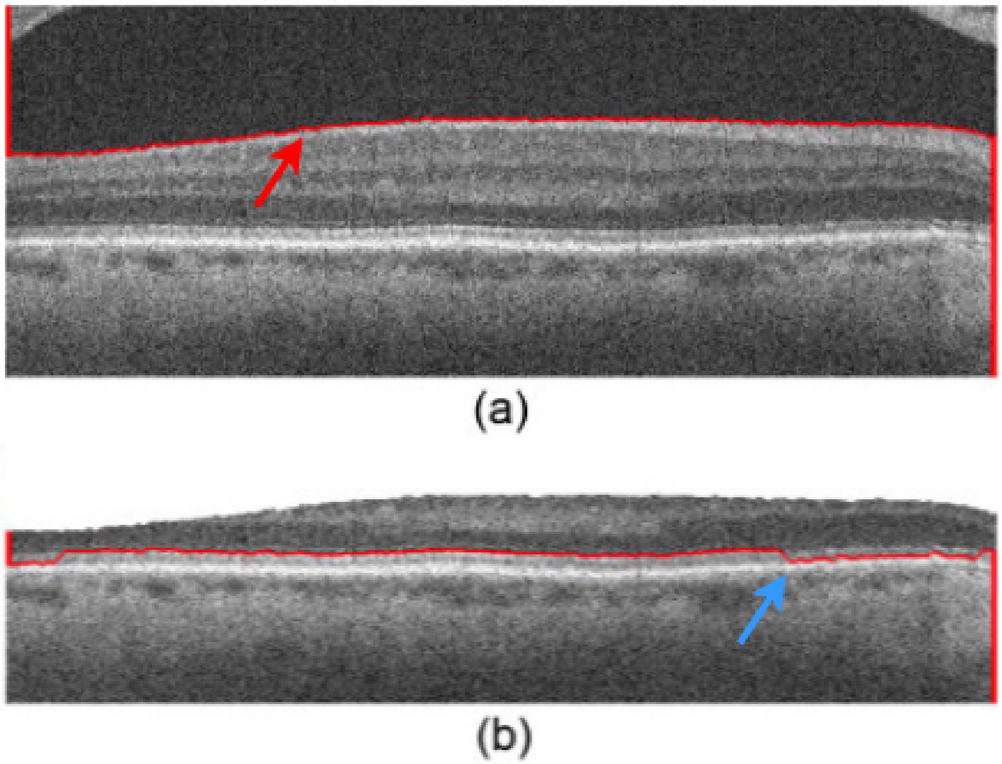


Figure 4.10: Segmentation results using Dijkstra's algorithm, automatic endpoint initialisation, and search space limitation. (a) The vitreous-NFL layer boundary segmented, indicated by the red arrow. (b) The pilot IS-OS layer boundary segmented. The detected path was diverted from the target boundary, shown in the blue arrow. The figure is presented with re-usage permission from [2].

To implement the whole eight-layer segmentation, they tailored the scheme with additional image operations. First, they estimated the RPE layer location based on the pixel intensity and used its location to realign all A-scans (adjusting the A-scan vertically). This operation can make the RPE layer and all other underlying layers flattened to nearly horizontal and thereby simplifies the detection. Second, the boundary search was done in priority order according to the pixel gradient, i.e. the high-gradient boundaries, such as vitreous-NFL boundary and IS-OS boundaries, has high priority being searched. Third, the algorithm applied a minimum edge weight to pixels of vessel tissue. This operation addressed the path diversion issue due to the uneven pixel intensity of vessels, seen Figure 4.11. The vessels embedded in the inner layers usually have a higher reflectivity than surrounding areas. This effect was problematic to the graph-cut algorithm because the vessel shadow may mislead the path to different layers. To solve this issue, they first determined the vessel location by accumulating the pixel intensity of

A-scans between the RPE and choroid layer. As a result, the low-value A-scans are determined as vessel shadows. When it segments the boundary, the minimum weights of the graph were set for these A-scans to ensure that the graph path is not diverted incorrectly. Lastly, the algorithm refined the boundaries at the fovea point. The fovea point was first determined by finding the shortest distance between the tentatively detected NFL and OPL layers. Then the NFL, GCL-IPL, INL, and OPL layers were forced to merge at the fovea point, seen Figure 4.12.

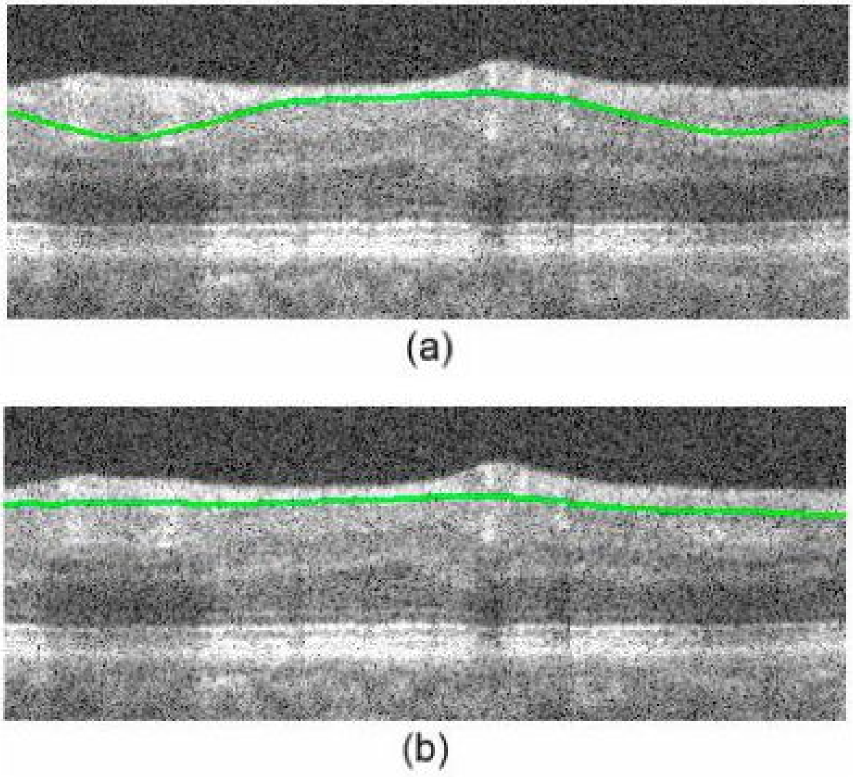


Figure 4.11: Boundary correction at vessel region. (a) Boundary detection of NFL-GCL without vessel correction. (b) Boundary detection of NFL-GCL after vessel correction. The figure is presented with re-usage permission from [2].

This algorithm effectively addressed the shape and texture variance of the retina tissue introduced by the nature of the retinal structure, such as the layer merging at the fovea point and uneven tissue reflectively of blood vessels. However, the number of layers which can be segmented were still limited to a maximum of eight. In addition, this algorithm was focused on single B-scan processing. Although the algorithm can be applied repetitively to the multiple B-scans, it didn't take into consideration the relationships between consecutive B-scans, which is also important for 3D OCT volume analysis.

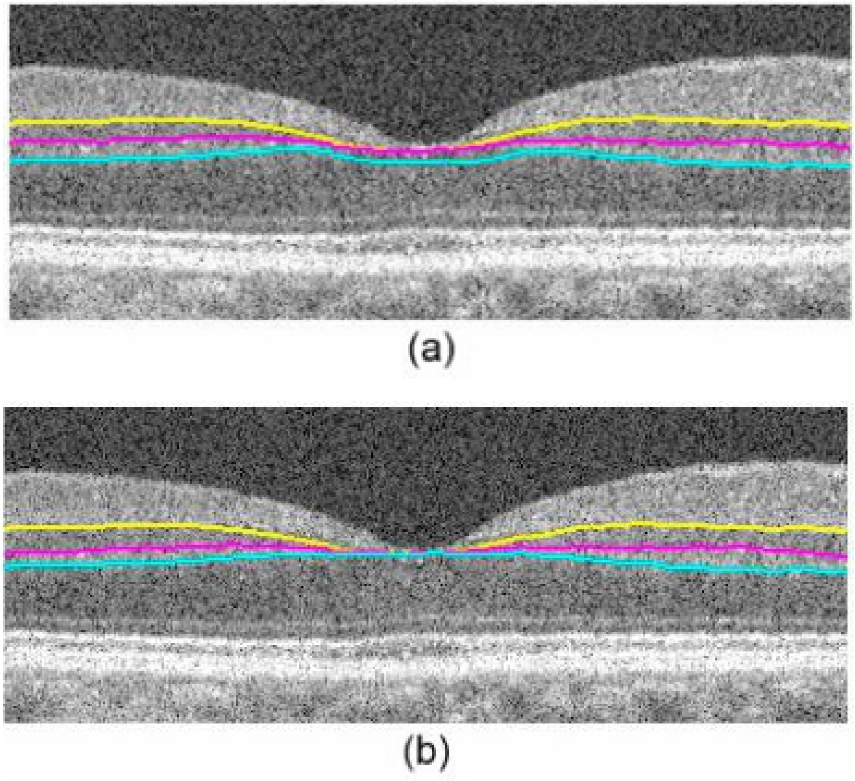


FIGURE 4.12: Boundary correction at the fovea point. (a) Detected boundaries before correction. (b) Detected boundaries before correction. The figure is presented with re-usage permission from [2].

More recently, Zhang *et al.* proposed a method for fast retinal layer segmentation [90]. They demonstrated a framework for the segmentation of eight macular layers from SD-OCT images. This method claimed two main features. First, the algorithm incorporated the layer segmentation in consecutive B-scan frames of the 3D OCT volume. Second, it extended the segmentation technique to real-time image processing of retinal OCT images. As they demonstrated, the algorithm decreased the processing time significantly while retaining accuracy. In their experiments, eight layers were segmented from a data volume within 5.8 seconds, which makes it 37 times faster than the other methods included in their study.

This method implemented the B-scan layer segmentation in an eight-step procedure, which is summarised in Figure 4.13.

The first two steps, which are for image preprocessing, are image projection and image filtering. The image projection is applied for searching the region of interest (ROI), including the retinal layers but not the vitreous. This was done

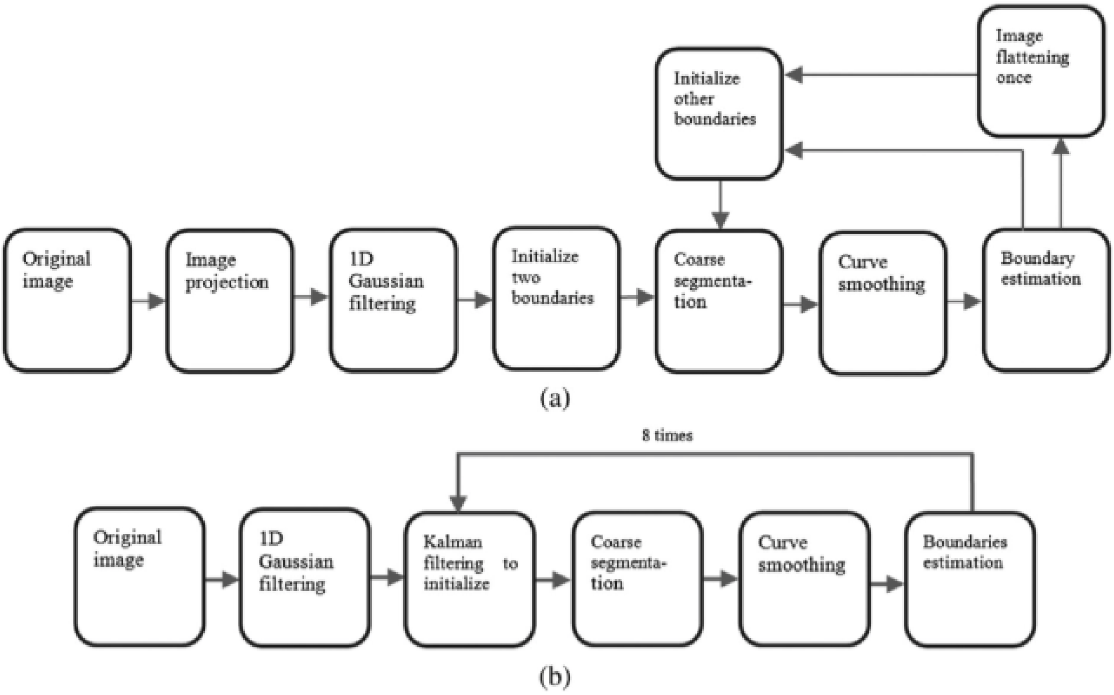


FIGURE 4.13: Segmentation algorithm schematic. (a) Schematic for single B-scans or the first frame of a volumetric OCT data. The initialisation is achieved from image projection and prior knowledge. (b) Schematic for the non-first frame of volumetric data. The initialisation is attained from previously segmented frames.

by averaging the image in the B-scan direction (the transverse direction in image). Consequently, it generated a 1D image projection with two peak points, which determined the region between the NFL layer and the RPE layer. Then, the ROI was defined by extending that region above and below with a predefined distance. This operation removed the outside areas and sped up the later process. After that, a 1D Gaussian filter was applied to suppress speckle noise in the lateral direction.

After the preprocessing, the boundaries of vitreous-NFL (between vitreous and NFL) and IS-OS was estimated in a down-sampled version of the original OCT image. In the down-sampled image, those two boundaries were conspicuous while other layers were smoothed out. Therefore, they applied a pixel intensity threshold to determine the pixels having higher intensity values, which denotes the target boundaries. This operation increased the robustness of the boundary location estimation.

In the next step, the location of detected boundaries was further optimised by using an active contour model. Specifically, an energy function of the model

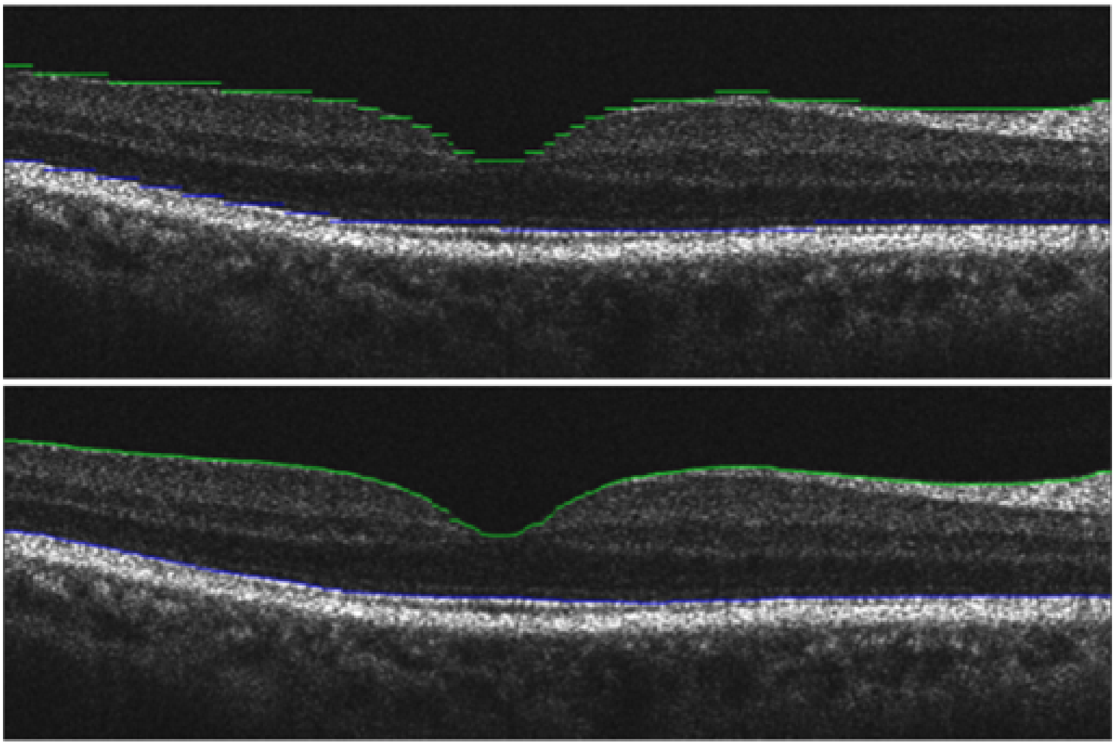


FIGURE 4.14: Boundary correction by the active contour model of [90], presented with re-usage permission. The first row shows the initial boundary detection from the down-sampled image. The green line segments for vitreous-NFL and blue for IS/OS. The second row shows the boundary corrected by the active contour model. The boundary sections are connected, and the boundary continuity is maintained.

transformed each pixel of the boundaries to its local max-gradient edge in the axial direction. However, the customised energy function didn't consider the smoothness of the boundary. As a consequence, the boundary estimation may be discontinuous, whereas the real retinal layers are not. Therefore, in the following step, a Savitzky–Golay smoothing filter [91] was applied as a complementary operation to guarantee boundary continuity. In this step, the boundaries were filtered with a smoothing window of 20 to 30 pixels, determined by different target boundaries. The boundary detection before and after the active contour is shown in Figure 4.14.

After that, the image was flattened based on the IS/OS boundary. The remaining boundaries were detected by iteratively applying the layer initialisation, the active contour model, and curve smoothing filters to each layer in a designated order. Note that the layer initialisation was estimated based on the prior knowledge of retinal structure.

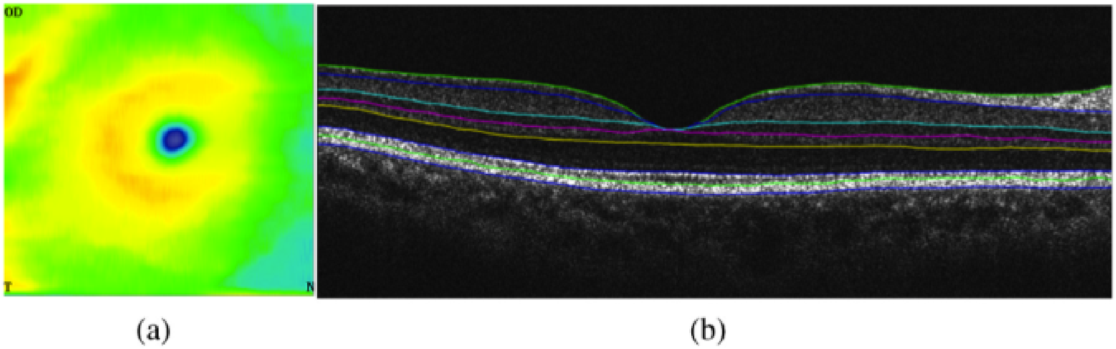


Figure 4.15: Boundary segmentation of a normal adult retinal layer boundaries. (a) Retinal thickness maps of the 3D OCT data. (b) Segmentation for the single B-scan frame of the volumetric data. The figure is presented with re-usage permission from [90].

Once the boundary detection was completed on the first B-scan slice, it was subsequently used to predict the initial position of boundaries in the adjacent B-scans. This is based on the assumption that the layer positions of close by retinal regions should be smooth and without significant change in shape. In the procedure of volume segmentation, a Kalman filtering function was used to track the layer boundaries in the previous B-scans and calculate an approximation of these boundaries in the next frame. By doing so, the algorithm achieved an efficient boundary initialisation and simplified the boundary detection in other frames. The boundary segmentation of the volumetric data generated the retinal thickness map, seen Figure 4.15.

4.4.2 Segmentation of oedemas in the retina

The commercial development of retinal OCT systems, has led to the segmentation of retinal structures using various approaches. Most of the existing segmentation methods focus on the detection of retinal layer boundaries. However, increasing interest in the pathological analysis of macular oedema, has also led to the development of methods for fluid oedema segmentation. Some approaches were adapted from the layer segmentation methods, while others were based on new frameworks. In this section, we review recent approaches to segmentation of fluid-filled region boundaries and discuss their limitations.

Fernandez *et al.* proposed a segmentation model consisting of anisotropic diffusion filtering and an active contour. This model was used to segment instances of cystoids and sub-retinal fluid spaces associated with age-related macular degeneration [92].

The OCT images were first processed using an anisotropic diffusion filtering [93] to improve the pixel gradient at the oedema boundary. Their experimental results demonstrated that the anisotropic diffusion filtering with selected parameters reduced the speckle noise significantly and simultaneously preserved the edges of fluid-filled regions. Figure 4.16 shows the filtered results, compared with the results of the general Gaussian smoothing filter.

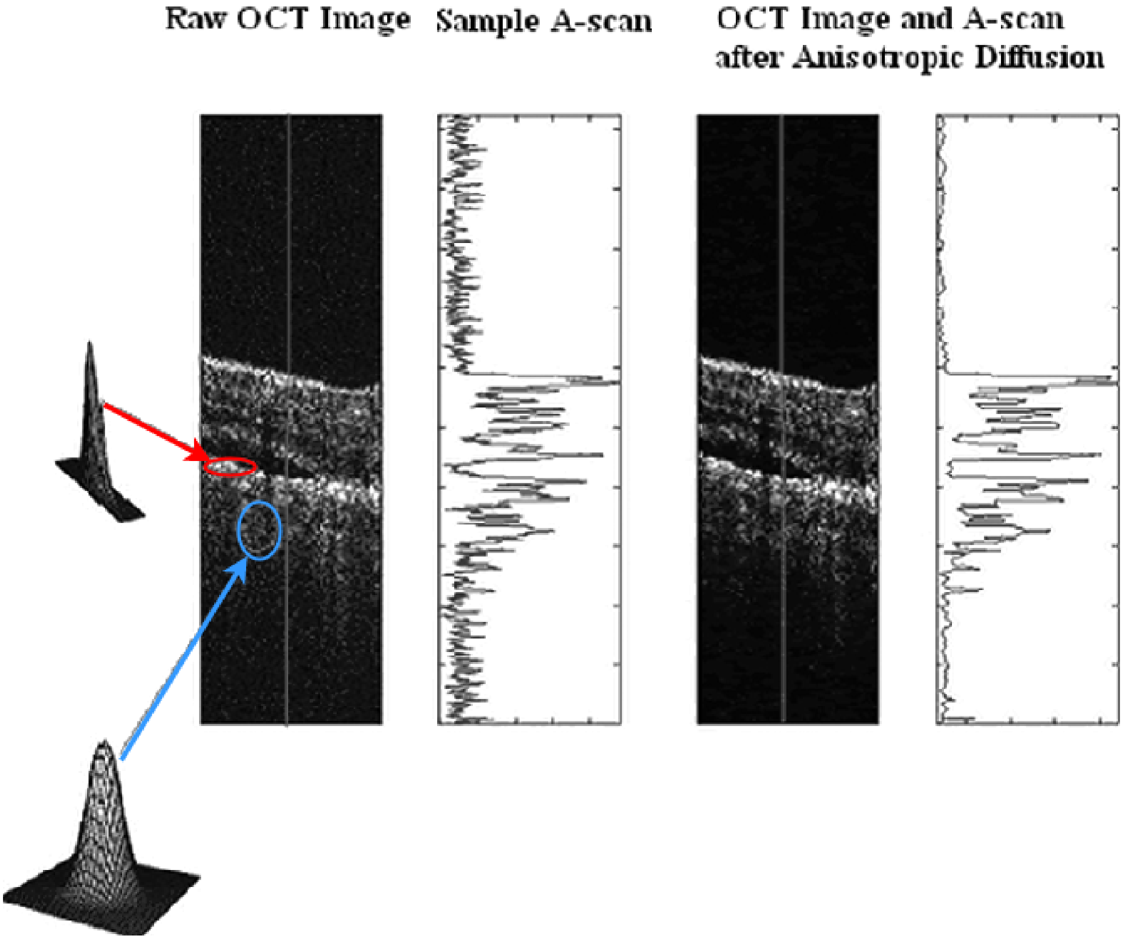


FIGURE 4.16: The anisotropic diffusion filtering is applying to an OCT image. A small section of the image containing 100 A-scan is shown. Left: the structure of two smoothing kernels are adapted to different image contents, shown in red and blue circles. Right: the image after the kernel smoothing. The figure is presented with re-usage permission from [92].

After the suppression of the speckle noise, a gradient vector flow (GVF) model powered by an active contour was used to find the final boundary position. The initial location contour was obtained from the GVF external force field of the OCT image, where the inner side of the fluid regions was detected. Then, the initial position of the active contour was manually set inside the original oedema region. Thereafter, the initial active contour was transformed iteratively until the contour reached its optimal position, i.e. touched the real oedema boundary. Due to the variation in the OCT image quality, their experiment indicates that the accuracy of this model depends on the parameter settings. The final variable settings of the final model were determined from their experiment. The active contour model was used to detect the boundaries of retinal lesions based on the initial boundary estimation. Due to the nature of the active contour model, this detection relies on two assumptions. First, it assumes the oedema is a closed region. Second, the oedema has a homogeneous cavity shape when comparing it to the surrounding area.

Though the segmentation results demonstrated the model's advantages on oedema segmentation, several issues limit its performance. The GVF model shows two advantages. First, it can attract the active contour to the cystoid's edge from a comparatively long distance, i.e., it can fit well to the large cystoids. Second, the contour can fit into elongated holes without exceptional contour occlusion. However, it is noticed that the performance of these models relies largely on the initial contour positions for the segmented object. Figure 4.17 shows two segmentation results. It is seen that the two small regions were not detected. Another issue can be seen in the same figure that the lesions, due to the macular thickening, were not detected, which may limit the application of the algorithm.

More recently, Chiu *et al.* proposed an automatic approach to the segmentation of macular oedema areas and seven retina layers on the SD-OCT images of DMO [3]. To this end, they first created a classification method based on kernel regression (KR) for determining the positions of fluid-filled regions and retinal layers. Its results were used as guidance for the second step; the boundary segmentation was improved by using graph theory and a dynamic programming framework (GTDP), which was also used in the method discussed earlier [2].

The proposed method was implemented in two stages. In the first stage, a set of selected features was used to train a supervised feature classifier. At the second stage, the trained classifier was used to classify each pixel of the OCT image. In the generated label image, the pixels that were classified as DMO formed the segmented fluid regions.

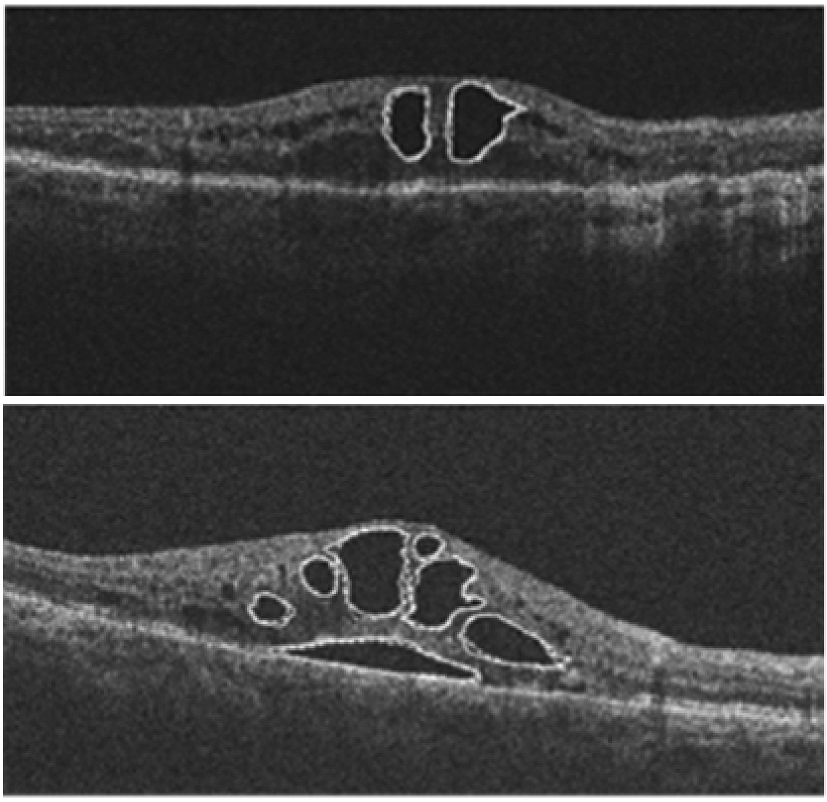


Figure 4.17: The fluid-filled regions contours extracted by the GVF snake algorithm. Top: AMD subject presenting two lesions in the foveal region. Bottom: AMD subject showing multiple lesions in the central area of the retinal structure. The figure is presented with re-usage permission from [2].

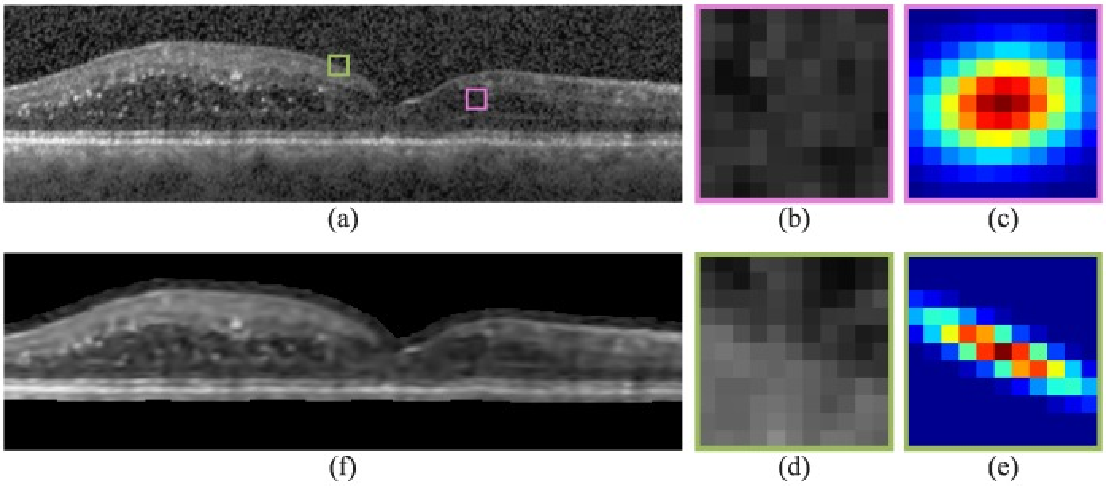


Figure 4.18: B-scan image with DME and its denoised result [3]. a: flattened image. b,d: zoomed-in images of the pink and green boxes in (a). c,e: Gaussian steering kernels used to denoised the central pixel of (b). d, f: the denoised image of (a). The figure is presented with re-usage permission from [2].

In the training stage, OCT images with manually segmented layer boundaries and DME regions were pre-processed with size standardisation and retina flattening. Then the GTDP method was applied to segment out the retinal region, i.e., detecting the ILM and BM layers and remove all image content outside the range bounded by these layers. Next, a denoising method based on an adaptive iterative Gaussian steering kernel was performed. In this method, a directional kernel was applied to each pixel. Features were then extracted from the resulting image for the purpose of classification. In their algorithm, 17 features were selected to train the feature classifier. Figure 4.18 shows two kernels adapted to the inner layer region (c) and layer boundaries (e), respectively. By following this process, the pixels belonging to the fluid-filled region will be classified, as shown in Figure 4.19. Eventually, all the pixels were classified into eight classes, including seven layers and the DMO oedema. The segmentation accuracy was calculated using the Dice coefficient, which measures the portion of the corrected region against whole regions. In their presented results, the method demonstrated a mean Dice coefficient of 0.78, including both retinal layers and oedema. However, the accuracy for oedema segmentation was only 0.432.

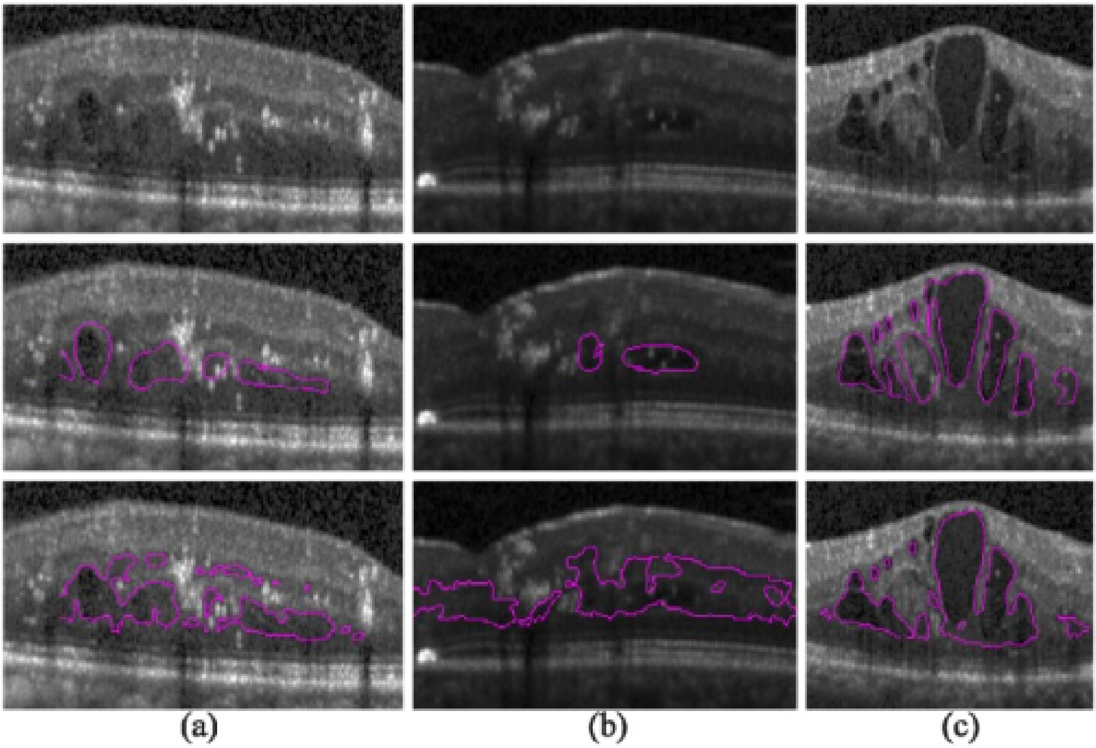


FIGURE 4.19: Automatic fluid detection errors. Top: original OCT image. Middle: manual segmentation of fluid-filled regions by a grader. Bottom: automatic classification results. The figure is presented with re-usage permission from [2].

To summarise, the methods reviewed in this section were developed to segment the fluid sections embedded within the retinal layers. They addressed the issues of speckle noise and oedema feature variation with a set of built-in rules. These rules were designed based on the nature of the retinal layer structures and lesion features, and they were developed in a particular order in the framework. When the character of the target object becomes varied and complex, more conditions should be considered to maintain accuracy. It may result in the model becoming complicated and less efficient. With more specifically designed rules, the application scope of the approach is narrowed, meaning that it is no longer applicable to multi-purpose diagnosis.

4.5 Conclusion

This chapter presented three aspects of retinal image analysis: the retinal structure, retinal layer segmentation, and retinal oedema segmentation. The retinal structure and its lesions are generally recognisable in the OCT images. Due to the optical reflectivity of different retinal tissues, the OCT imaging system can show the whole structure of the fovea region. A healthy retina has 12 layers and they merge at the fovea point. Every single retinal layer was usually presented by similar pixel intensity. The normal retinal structure may be affected by different disorders, including diabetic macular oedema, central and branch retinal vein occlusion, and pigment epithelial detachments. These defects typically appear in OCT images as a fluid cavity embedded within the retinal layers, where the cavity is characterised by a different reflectivity than the surrounding tissues. Recognising different retinal layers and segmenting these cavities are essential for retinal image analysis. We reviewed several mainstream frameworks for delineating the retinal structures regarding two aspects: segmenting the retinal layer boundaries, and segmenting the defect regions.

For segmenting the retinal layer boundaries, the reviewed frameworks consist of multiple standard image processing operations in sequence, including speckle noise reduction, retinal structure flattening, boundaries enhancement, and region limitation. These operations were particularly designed to make the boundaries more prominent based on the characteristics of the retinal structures. These methods used three different core functions to perform the layer segmentation, including edge detection kernel, graph search, and active contour. The edge detection kernel required the retinal layers between the boundaries to have a high contrast ratio. However, due to the nature of tissues, some layers, such as IPL and GCL layers, usually have similar optical reflectivity, which results in a visually blurred boundary. In the proposed method, the authors used the edge detection kernel to detect only seven boundaries within the retinal region. In addition, this method considered limited structural variations inside the retina, which can not be generalised for more complicated cases. The presence of vessels may also cause incorrect segmentation. The graph search is a more advanced method for detecting the boundaries. Their framework used a graph search to segment eight layers successfully. This method requires each type of layer tissue to be homogeneous, i.e., the reflectivity should be similar. However, this is not always ideal, especially for the GCL layer with high vessel reflectivity. To overcome this limitation and enhance the robustness of their approach, they applied a set of rules to the framework for dealing with these variations. The active contour was also used for boundary detection. It is a flexible method to adapt a deformable contour to the region edge. However, the framework found the prominent boundary position by down-sampling the image. This operation makes it have to discard the information of thin layers. Eventually, eight boundaries were segmented. Furthermore, in its framework, the initial contour was placed based on prior knowledge rather than the image content. This strategy means that objects of significantly differing variance are not segmented reliably.

For segmenting the defect regions, the core techniques used in these reviewed methods are the gradient vector flow with active contour and kernel-regression based pixel classification. In addition, common image preprocessing operations were also implemented to suppress speckle noise. The GVF active contour was able to segment fluid-filled oedema regions, which are usually homogeneous with low reflectivity. The proposed method was demonstrated to have good segmentation accuracy on large and elongated oedema regions. However, this method was not able to perform a good segmentation on retinal thickening areas and small oedema regions. This is because the retinal thickening often presents a very blurred boundary. Additionally, the small oedema regions often have similar reflectivity to the surrounding tissue. The kernel-regression based method segmented the oedema region by classifying each pixel into a set of pre-defined classes. The accuracy of this classification approach mainly depends on the feature vectors of each pixel. The classification was performed by using 17 selected features. However, the manually selected features still suffer from handling the feature variation. As a result, the method classified the healthy tissues to oedema when the retinal thickening regions appeared blurry as blurry texture features, which were very similar in appearance to the healthy retinal tissue.

Although the reviewed methods showed good capability for segmenting the retinal structures, these methods are not the optimal choice to fully and reliably segment the retinal layers and oedema regions due to the reasons discussed. Hence, we developed several approaches with the aim of accurate segmentation of more layer boundaries and oedema regions. These improvements include the segmentation of the very thin POS layers and appearance-varying retinal thickening and detachment.

Chapter 5

Superpixel guided active contour segmentation of retinal layers in OCT volumes

5.1 Introduction

Since 2017, there are about 1.93 million people in the UK affected by retinal diseases. The number of people with eye defects is estimated to rise to 2.70 million by 2030 [94]. With the extensive use of OCT in eye disease monitoring, it is becoming more critical for the clinician to do OCT image analysis and make a diagnosis. In this chapter, we describe our method for automatic segmentation of retinal layers in OCT images to provide better retinal analysis for the clinician. Many fast and accurate automatic methods for the segmentation of retinal layers and eye defects have been proposed. Approaches are based on a wide range of algorithms, and these have been discussed in Chapter 4. The performance of these models depended on various image pre-processing steps, including: enhancement, speckle noise suppression and edge gradient strengthening. However, the majority of prior work in this area was not able to reliably detect thin cell layers within the retina.

Recent work [90] reported state-of-the-art segmentation of retinal layer boundaries using active contours. As discussed in section 4.4.1, this deformable contour can change its shape and length, thereby migrating towards edges where the image intensity gradient is high. To improve detection accuracy for thick cell layers and reduce processing time, the authors applied pixel averaging, followed by down-sampling. However, these operations adversely affect the ability of the active contour to detect thin layers.

In order to achieve an automatic, full retinal layer, segmentation, a more accurate detection method is required. In addition, it should be time-efficient and amenable to subsequent 3D boundary profile location within OCT volumes. To this end, we developed a superpixel guided, active contour framework for the segmentation of retinal layers in OCT volumes. The framework comprises three main components: an adaptive-curve, superpixel aggregation and an active contour. These combine to make the framework self-adapting to local and global features of the retinal structure. Our approach is capable of segmenting up to 12 boundaries of healthy retina layers. This framework was developed in collaboration with the Applied Optics Group at the University of Kent, who provided the cross-sectional retinal image data using SD-OCT.

This chapter first describes the function of each component within the framework and then provides the segmentation results for each step in the image processing pipeline in Section 5.2. The function of each component is summarised as follows:

- *Adaptive curve*: The active curve function was used to automatically detect the retinal regions from partially selected A-scans of the OCT B-scan, which is more efficient than previous work that processed A-scans of an entire B-scan image.
- *Superpixel*: The superpixel algorithm was used to aggregate the pixels of the OCT B-scans into a single region where the pixels inside were contributed to the same retinal tissues.
- *Active contour*: A customised active contour function was used to refine the boundaries detected by the superpixel algorithm. By doing so, the framework was able to fit a smoothed boundary to the real layer boundaries.

The remainder of this chapter demonstrates the overall framework, including image pre-processing steps and the detection sequence for all retinal layers. We demonstrate the boundary segmentation results in both single B-scan images and in 3D retinal volumes and discuss the effectiveness of our approach.

5.2 Boundary detection using adaptive curves, superpixels and active contours

The proposed method is designed for boundary segmentation in the central fovea area. In OCT scanning of the retina, the structure of the optical disc varies at different locations. The structure of the periphery region (w.r.t. the fovea point) is relatively simple since retinal layers are almost parallel to one another in the cross-sectional, B-scan, images. Therefore, it is usually straightforward to detect their boundaries. In contrast, the central fovea has a more complex structure. Its characteristics cause the inner retinal layers to converge into the fovea point. Clinicians are especially interested in cell layers in the region close to the fovea point. In this section, we explain how the framework segments layers at the central fovea. However, it can be easily applied to the periphery regions, which is demonstrated in the later section.

5.2.1 Detection of retinal region in vertical direction using the adaptive curve

Localising the vertical retina region is essential for retinal image analysis. To ensure an efficient detection of the retinal layers, it is important to remove redundant regions in advance. The retinal structure presented in OCT B-scans is shown in Section 4.3.1. Image content above and below the upper and lower cell layers respectively (the vitreous and outer choroid tissues) have little diagnostic value and are segmented out. Since the A-scans are intensively sampled within a small retinal area, the consecutive A-scans usually contain similar depth profiles of the retina, causing redundancy in the representation. Hence, the detection of one A-scan may be used to estimate its adjacent A-scan.

Motivated by the need for accurate and efficient layer detection, we propose an adaptive curve method to determine the retinal region. This method aims to fit a curved boundary to inner and outer retinal surfaces automatically. To this end, the algorithm only selected a relatively small number of A-scans to estimate the boundary onset points and boundary curvature efficiently. This approach is amenable to removing the vitreous and outer choroid tissues from the image data.

The procedure of finding both the NFL layer and the RPE layer is the same. In this section, the NFL layer is taken as an example to explain the procedure. At the first step, the algorithm initialised an original boundary that is near the location of the NFL layer. For a single B-scan image, the algorithm selected a set of A-scans that were equally distributed in the lateral direction. In practice, it usually selected less than five A-scans, including left most and right most scans, which provide anchor points. Then, an onset search function was applied to these A-scans to search for the transition from the vitreous to the NFL layer. Since there is a high-intensity gradient between the NFL layer and the vitreous, the onset point was determined by applying an intensity threshold, which was calculated using the following equation:

$$T_{\text{onset}} = \frac{1}{N}\sum_{i=1}^{N} x_i + \beta\sigma(x_i^N), \quad i \in [1, ..., N] \tag{5.1}$$

where N is the total number of pixels in each A-scan, $\sigma(\,\cdot\,)$ is the standard variation calculated overall pixel intensities $\{x_i\}$, and β is a scalar factor that balances the two terms in the summation and ensures that the surface layer is detected. Hence, the optimal setting for β ensures that the threshold is not triggered by speckle noise in the vitreous region or by layers inside the surface of the retina. In our algorithm, two values for β, corresponding to two different intensity thresholds, were set for detecting the NFL (top) layer and RPE (bottom) layer.

An onset point was detected for each of the selected A-scans, and B-splines were interpolated through these points to provide initial estimations of the NFL and RPE layer positions in the B-scan images. Once the initial boundary was formed, its shape was then further refined through a number of iterations, thereby providing more accurate positioning of the spline curves.

Specifically, the evolution of the boundary shape entailed selecting further A-scans, where sampling density was increased based on the curvature of the splines. In retinal B-scan images, layer boundaries are mostly smoothly varying, and large changes in curvature only occur in a few small regions. In each iteration of the evolution, the algorithm relocated the positions of previously selected A-scans, moving them towards the high curvature regions on the estimated boundary. This process was repeated until changes in percentage change in boundary shape, between consecutive iterations, dipped below a predefined threshold.

To illustrate the iterative updating of points, here we demonstrate how the adaptive curve function works using a simple sine wave example. The sine wave was initially sampled using a set of equally spaced points over its lateral range. In this case, the points have a sequential order. The aim of this demonstration is to show the updating of the sampling points during the iteration. It can be seen (Figure 5.1) that point locations are updated such that the sampling density is highest for the peak and trough where maximum curvature occurs. Points move to optimal positions after a small number of iterations.

Sometimes, the retinal surface was not in a regular shape due to any artefact and the selected A-scans might not be in sequential order. Our adaptive curve function can handle this situation due to its order-irrelevant updating processing, i.e. the point sequence does not affect the point-updating result. In the sine wave example, the points were ordered sequentially. In this case, however, the points can be initialised in random order. The procedure is summarised in Figure 5.2. For a comprehensive visualisation, the neighbour points were shown with a coloured link. It is seen that the neighbour points stay in the high-curve region, and they were rearranged back to the sequential order. This advantage can still ensure a correct boundary in case the points allocation was disordered in some iterations.

The method for selecting A-scans at each iteration is described as follows:

$$r(t) = \delta + \frac{K(t) - K_{\min}}{K_{\max} - K_{\min}} \tag{5.2}$$

where $r(t)$ is the curvature weight of each selected A-scan point, which is later used in the point updating iterations in Equation 5.5. t is a set of neighbour A-scans of a selected A-scan, $T = [t_1, \cdots, t_m]$ of m points, i.e., the position of the A-scan in the lateral direction. The δ is a factor that controls the updating step length. The smaller δ results in a large step for each point updating. The $K(t)$ is the curve rate. It is calculated by

$$K(t) = \frac{|c'(t) \times c''(t)|}{|c'(t)|^3} \tag{5.3}$$

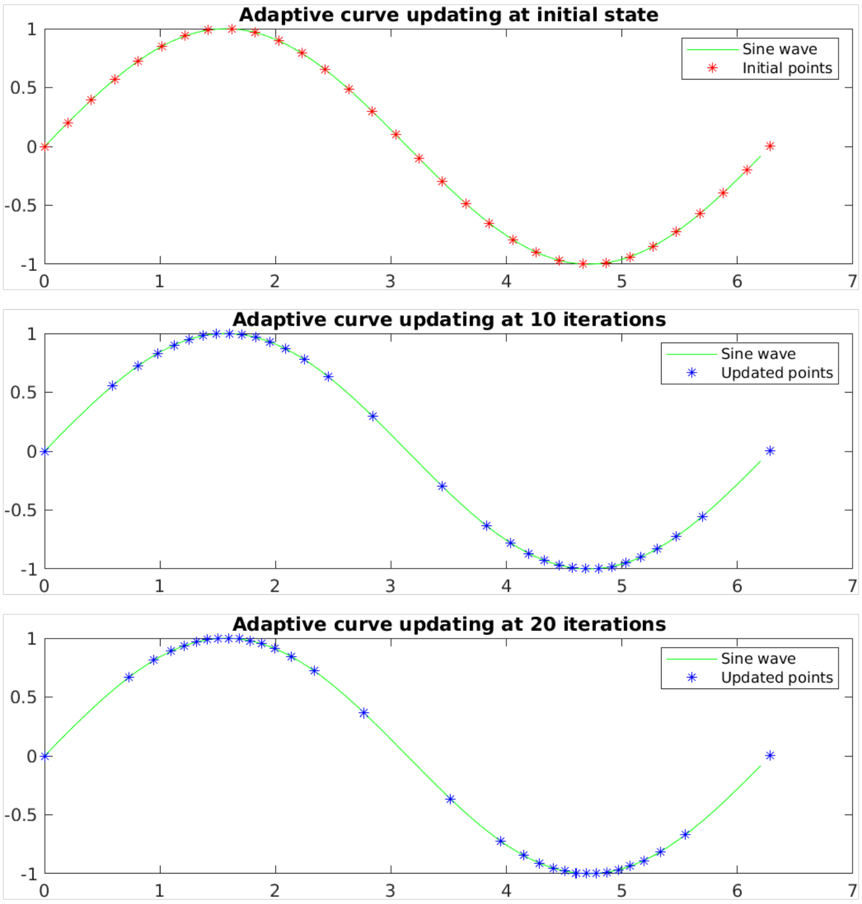


FIGURE 5.1: The visualisation of the random sampling point update by the adaptive curve model on a sine wave. Top: the initial points in sequence order. Middle: The update state of the sampling points at ten iterations. Bottom: The update state of the sampling points at 20 iterations.

where $c'(t)$ and $c''(t)$ are the first and second derivative of the estimated boundary at t. Given the curvature information $r(t)$ of all t, the adaptive curve functions can be formulated such as the position of each t should satisfy

$$\sum_{j \in N_i} (t_i - t_j)\, r\,(t_i) = 0, \quad t_i \in T \tag{5.4}$$

where $N_i = \{t_{i-1}, t_{i+1}\}$ is the neighbour sampling points of point t. The updated position of t_i can be calculated using Equation 5.5 in an iterative way.

$$t_i^{k+1} = \sum_{j \in N_i} \omega_j^k t_j^k, \quad \omega_j^k = r\left(t_j^k\right) / \sum_{j \in N_i} r\left(t_j^k\right) \tag{5.5}$$

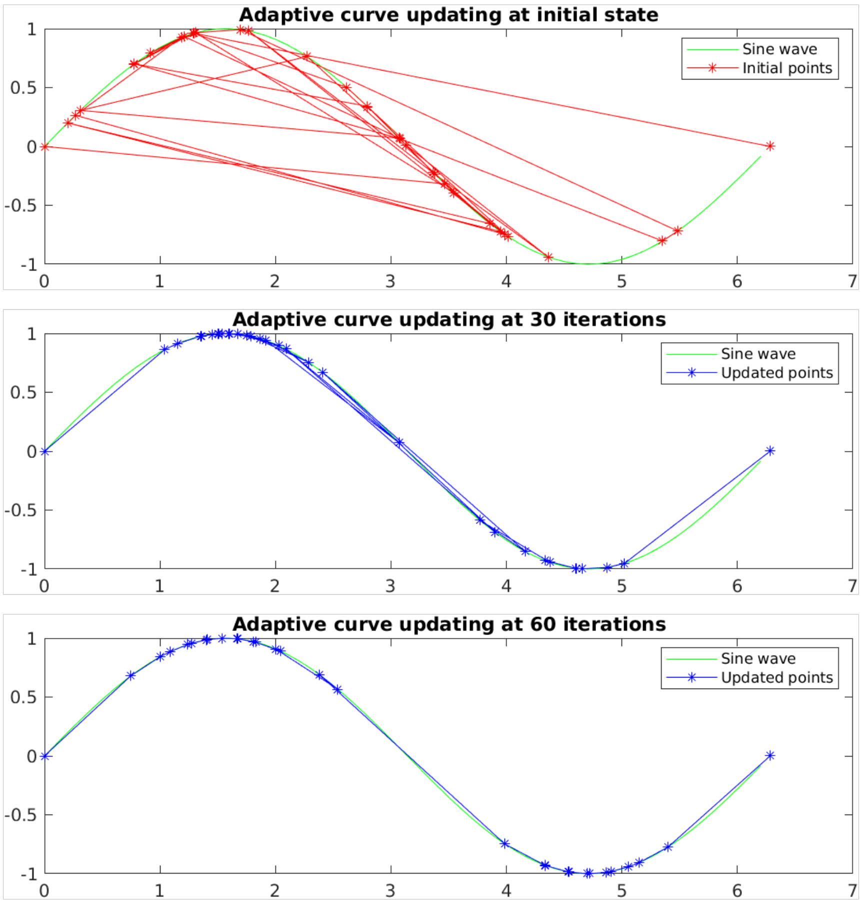


Figure 5.2: The visualisation of the random sampling point update by the adaptive curve model on a sine wave. Top: the initial points in random order. Middle: The update state of the sampling points at 30 iterations. Bottom: The update state of the sampling points at 60 iterations.

with the stop criterion as

$$\sum_{i=1}^{m} \left| t_i^{k+1} - t_i^k \right| < \varepsilon \tag{5.6}$$

where the ε is the change tolerance and k is the number of iterations. In this iteration procedure, the ratio of the curvature weights of two points, ω_j^k, acts as a balance term such that the updated points make the neighbouring points balanced in terms of curvature and distance.

When the A-scans were selected during each iteration, the threshold function 5.1 was used to find both the NFL and RPE layer boundaries. Specifically, the onset points for the NFL layer were determined by searching the first point to exceed the intensity threshold on the top edge, and also for the RPE layer on the bottom edge of the region of interest.

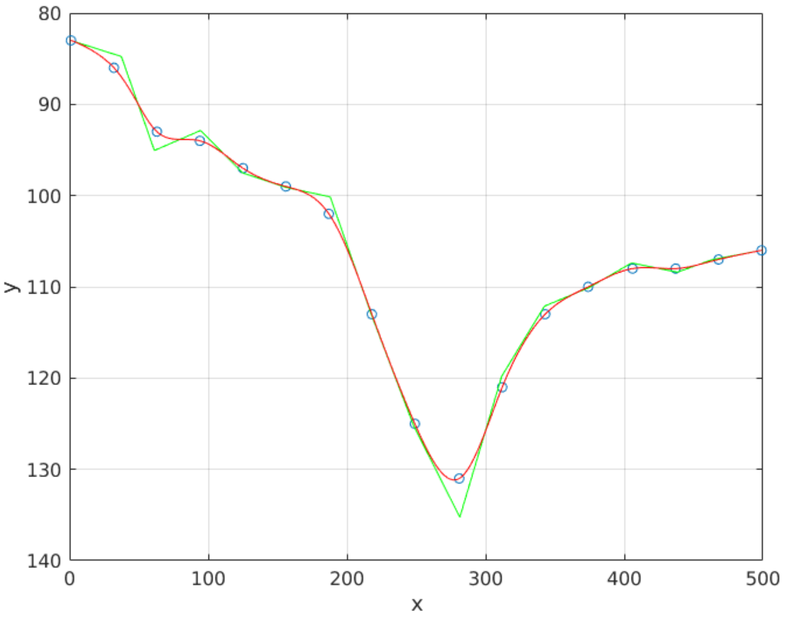


Figure 5.3: The Vitreous-IML boundary estimation. The circle indicates the 17 final sampling points. The green line is the initial boundary estimation. The red line is the final estimated boundary using B-spline interpolation.

Figure 5.3 demonstrated the detection processing of Vitreous-IML boundary by sampling 17 out of 500 A-scans through the image. The circle marks denoted the selected A-scans, the yellow line linked the detected boundary positions at each A-scan, and the red line is the final estimated boundary by refining the original boundary with B-spline interpolation. It is seen that the sampling points were accurately selected from the high curvature positions where it denotes the change of the surface boundary.

Figure 5.4 demonstrated the adaptive-curve updating of the top surface boundary in a test B-scan image. In this demonstration, only three initial A-scans were selected. It is seen that the adaptive curve only detected the boundary at selected A-scans and the boundary fitting was not accurate at the beginning. During the iterations, seen in Figure B, the detection at the right wing was accurate while the boundary at left wing was missed. However, this was corrected with more iterations because it attracted more sampling points, and the corrected boundary detection in those A-scans controlled the boundary shape. Therefore the final correct segmentation was maintained.

After the detection of the retinal region in the B-scan image, we replace the

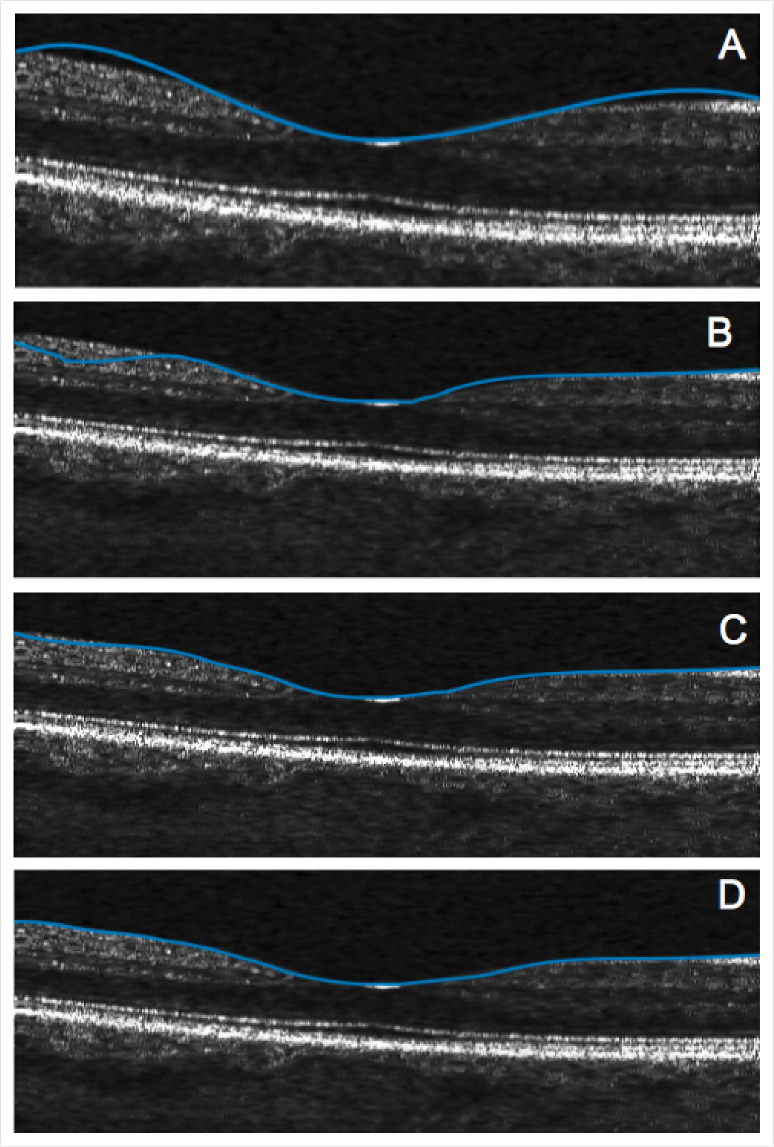


Figure 5.4: The segmentation of vitreous-IML boundary by adaptive-curve detection method. The four figures show the process of iterative boundary updating.

non-retinal content with zero-value pixels such that the framework only focuses on the retinal layers within the region.

5.2.2 Retinal region grouping using superpixel aggregation method

Once the retinal region has been detected by the adaptive curve, our framework then uses the pixel intensity and vertical gradient of B-scan images to estimate the coarse position of retinal layer boundaries, which is subsequently refined using an active contour in each case. The coarse detection is implemented using a customised simple linear iterative clustering (SLIC) superpixel algorithm [95].

The generic superpixel method is designed for 2D images. However, there are several features of OCT images that may negatively affect its out-of-the-box performance. It may be affected by the non-homogeneous tissue and speckle noise. Tissue layer boundaries are formed due to the differences in optical reflectivity (Section 4.3.1). In the absence of noise and artefacts, regions of continuous pixel intensity represent single-cell layers. However, for real data, pixel intensity also varies due to non-homogeneous tissue and speckle noise. For example, blood vessels often appear within the inner retinal layers. Due to the density and light absorption properties of blood vessels, shadows may be projected onto the underlying layers. As a consequence, affected layers exhibit discontinuities in light intensity. Figure 5.5 shows an example of inaccurate segmentation when the out-of-the-box superpixel clustering algorithm was applied directly to the 2D OCT images. In this case, it is seen that the same retinal layer was clustered into different superpixels due to the variation in reflectivity. Note that this type of incorrect segmentation happened in previous work (shown in Figure 4.11) due to the same reason.

To overcome these problems, we modified the original SLIC algorithm, especially for segmenting the retinal structure, comprising lateral cell layers with transitions between them in the axial direction. We adopt a simple 1D superpixel approach in the A-scan direction, rather than the usual 2D implementation of the superpixel algorithm. The advantage of our customised SLIC algorithm is its use of local A-scan intensity and distance information to form homogeneous areas. It achieves efficient layer segmentation over the entire B-scan image by considering localised regions independently. For example, when detecting layers affected by blood vessel shadowing, the layer intensity differences in the shadow area and normal area were considered separately to cluster corresponding pixels, and they were not affected by each other.

The process of the superpixel forming procedure is summarised in Figure 5.6. Specifically, at the first stage, for each A-scan, the retinal region was divided into a number of equally sized segments, corresponding to the number of layers. Each segment was a pre-allocated superpixel. We assigned a seed to each superpixel at the geometrically central pixel of each superpixel that was treated as its seed. To

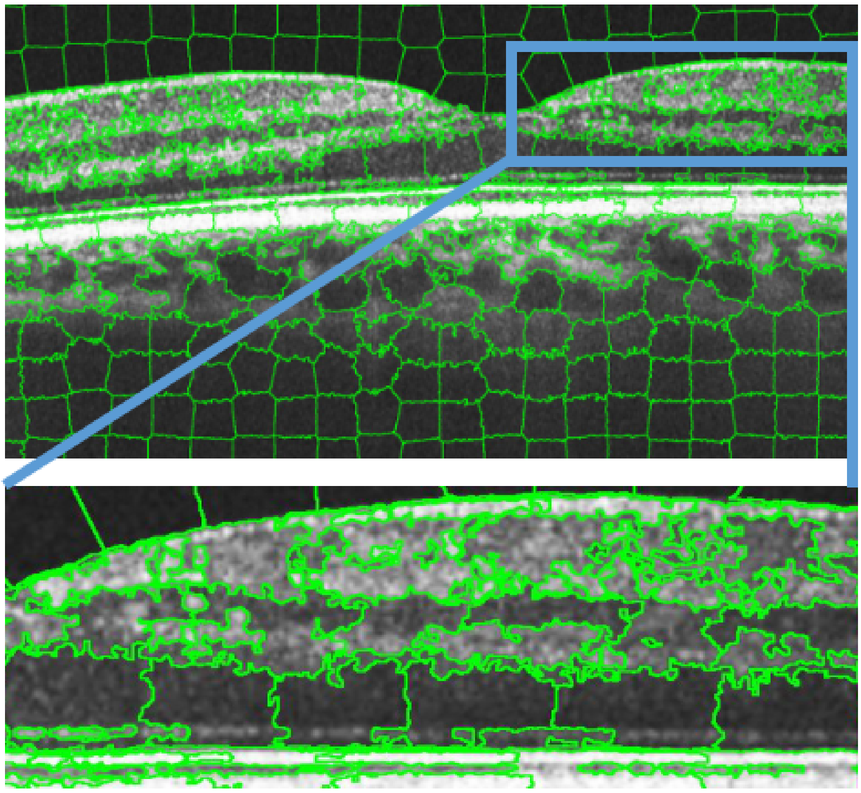


FIGURE 5.5: The 2D superpixel clustering of OCT images. Top: the superpixel applied to the whole image. Bottom: the zoomed-in view of the blue frame in the top image. The pixels of different layers are grouped into the same superpixel.

ensure no seeds were accidentally set at the boundaries, the vertical gradient at the seed pixel and its two neighbours were calculated by using Equation 5.7.

$$G(x, y) = \|\mathbf{I}(x, y+1) - \mathbf{I}(x, y-1)\|^2 \tag{5.7}$$

where $\mathbf{I}(x, y)$ is the grayscale intensity at position (x, y), and $\| \cdots \|$ is the L_2 norm. The seed was relocated to the pixel of the lowest gradient.

Secondly, for each superpixel, the algorithm calculated a similarity distance (Equation 5.8) between the seed and the remaining pixels in a pre-defined spatial range above and blow it. By comparing the distance between the pixels with different seeds, we assigned the pixels to one superpixel, whose seeds had the shortest distance. This is formulated as

$$P(x, y) = \min\left(D_p\right), \quad P \in [P(x, s-r), \ldots, P(x, s+r)] \tag{5.8}$$

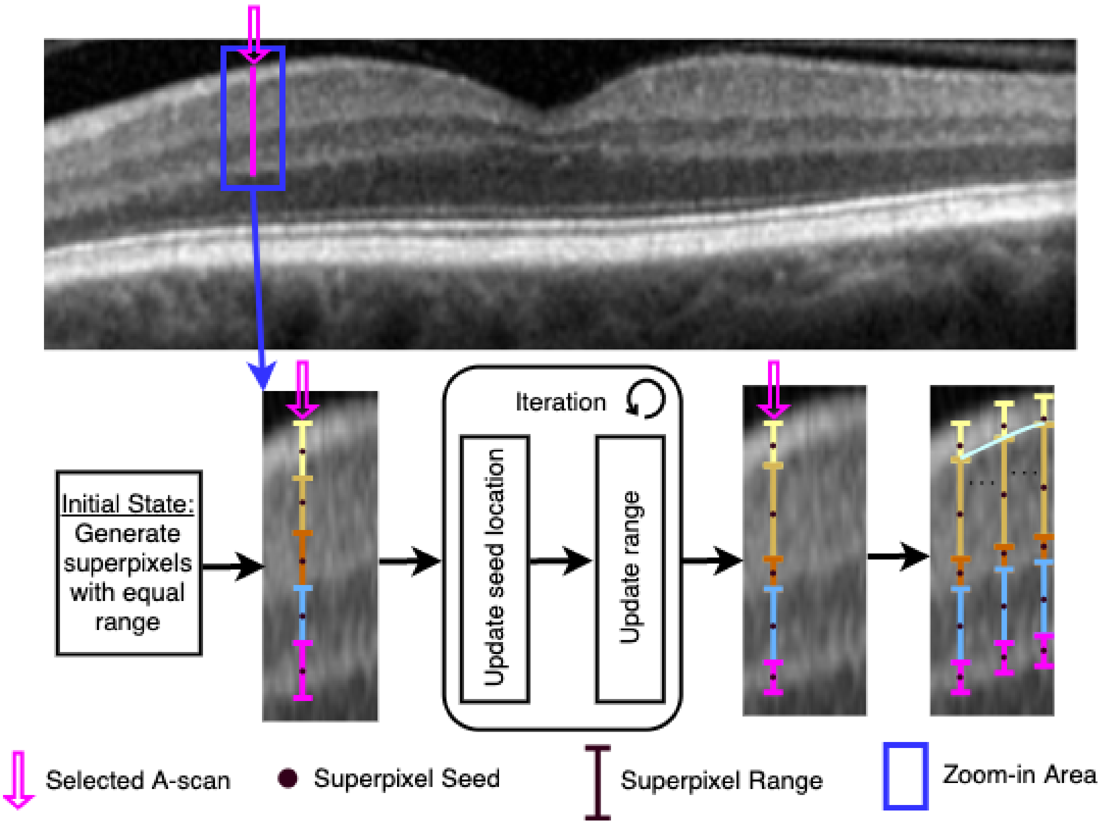


FIGURE 5.6: The diagram of the superpixel updating procedure at a single A-scan. The superpixels were initialised and updated iteratively to fit the image content in the A-scan area.

where r is the scope of seed pixel $P(x, s)$ and D_x is the similarity distance calculated by

$$D_P = d_I + \frac{m}{r} d_y \tag{5.9}$$

where D_P is formed as the sum of pixel intensity distance d_I and geometric distance $d_{x,y}$. r is the search scope, same as the r in Equation 5.8 and m is a balance factor that controls the compactness of the superpixel. For the segmentation of different boundaries, separate m were used. A high m take more geometric relation of the layers, and it is good at detecting weak boundaries between layers. The intensity information is known a priori, and it is good at detecting high-contrast boundaries and those with in-homogeneous vessels. The intensity distance d_I and geometric distance $d_{x,y}$ are calculated as

$$\begin{aligned} d_I &= \sqrt{(b_s - b_i)^2} \\ d_y &= \sqrt{(y_s - y_i)^2} \end{aligned} \tag{5.10}$$

where the b is the grayscale intensity in lab colour space and s is the seed. This pixel clustering process is repeated for each seed, and the resulting superpixels are formed.

After all pixels were grouped to their corresponding superpixels, the algorithm recalculated the spatial centre of each updated superpixel and the location of their seeds. This updating iteration was terminated once the overall change in seed position was less than a predetermined number of iterations. In our work, twenty iterations were sufficient for achieving stable clustering. Subsequently, the shape of the cell layer was approximated by connecting corresponding boundaries between superpixels in adjacent A-scans. The overall procedure is summarised in Algorithm 1.

Algorithm 1: Superpixel boundary detection

```
Data: iteration number $Iter$, number of layers $N$, A-scan $I$
Result: Superpixel clusters for $I$
Initialise the seed position for $n$ superpixels in $I$;
Adjust the seed to the local lowest-gradient position;
for $i < Iter$ do
    for each pixel in $I$ do
        Convert the intensity to CIELAB colour space;
        Calculate the similarity distance to the seeds in scope;
        Assign pixel to the superpixel where its seed had the smallest similarity
          distance;
    end
    Update the superpixel seed location to its geometric centre. ;
end
Connect the boundary point of corresponding superpixels in each $I$ to form
  the coarse layer boundary.
```

Figure 5.7 shows the segmented results of the RPEI/RPE layer. In the limited search region, the algorithm generated two superpixels that form an approximated boundary, separating two thin layers in the narrow region. It can be observed that the approximation of the target boundary is not smooth and continuous. The superpixels perform well in normal cell layer regions but less well in regions affected by shadows caused by blood vessels. In addition, one should note that the A-scans close to the fovea contain fewer layers than other bilateral regions, causing some

A-scans to be over segmented. To counter this problem, we applied lateral limits, after the segmentation of each layer boundary, to restrict the search area. This is described in Section 5.3.1.

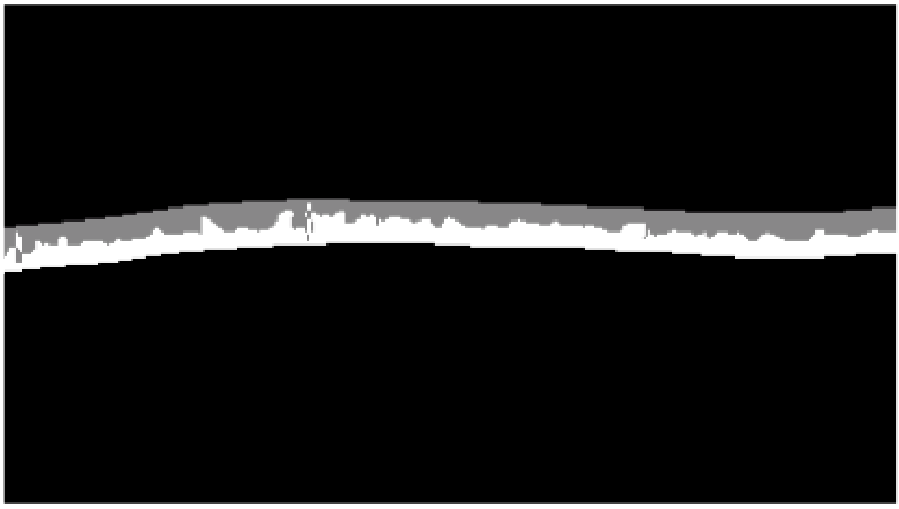


FIGURE 5.7: The segmentation results of RPEI-RPE boundary using the superpixel algorithm. In the limited search area, Two superpixels separate the image content within A-scan. The formed boundary in the B-scan image is not smooth and continuous. Therefore it cannot be accurately fitted to the actual position.

5.2.3 Self-adaptive multi-boundary detection using active contour

As discussed in the last section, the retinal layer boundaries were determined by using superpixel clustering in A-scans. The boundaries in the B-scan image were formed by connecting border points of the corresponding superpixels throughout the A-scans. However, this boundary was a coarse estimation and did not precisely fit the real boundary since it didn't consider the boundary continuity in the lateral direction. Therefore, a set of customised active contours were used to refine the coarse boundary to ensure the boundary smoothness and accuracy.

In this stage, the gradient-guided active contours were particularly customised for different boundaries based on their boundary type. Based on the contrast, the layer boundaries were classified into three types: dim-to-bright, bright-to-dim and neutral. The dim-to-bright boundaries were presented between low-intensity layers and high-intensity ones, such as the Vitreous-NFL boundary. It yielded a positive vertical gradient. Inversely, the bright-to-dim boundaries had a negative vertical gradient. The neutral type was applied to the weak contrast boundaries. The active contour was applied separately to these three boundary types.

The evolving contour procedure used in our framework was different from the conventional one in which the contour is initialised with a closure circle. For detecting each boundary, the active contour was first initialised by fixing two endpoints on the left and right edge of the B-scan. Then the contour shape was optimised by minimising its energy function, which is formulated as

$$\begin{gathered} E_{\text{contour}} = E_{\text{inter}}(v(x)) + \alpha E_{\text{exter}}(v(x))) \\ \alpha = \begin{cases} -1 & for \quad \text{dim-to-bright and neutral} \\ 1 & for \quad \text{bright-to-dim} \end{cases} \end{gathered} \tag{5.11}$$

where $v(x)$ represents a vector of all pixel positions, the E_{inter} and E_{exter} are the internal and external energy terms to control the shape of the contour, which were defined as

$$\begin{aligned} E_{\text{inter}} &= E_{\text{cont}} + E_{\text{curv}} \\ E_{\text{exter}} &= \begin{cases} G(v(x)) & for \quad \text{dim-to-bright} \\ |G(v(x))| & for \quad \text{neutral} \end{cases} \end{aligned} \tag{5.12}$$

where the $G(v(x))$ is the vertical gradient of the contour in each A-scan. The E_{cont} controlled the pixel to only move one-pixel distance at once. This guarantees no disconnection on the contour. The E_{curv} controlled the overall bending shape. They are formulated as

$$\begin{aligned} E_{\text{cont}} &= \begin{cases} 0 & if \ x = \min(\text{x}),\ \max(\text{x}) \\ \sqrt{\left|v(x)^{k+1} - v(x)^{k}\right|^2 + 1} & \text{otherwise} \end{cases} \\ E_{\text{curv}} &= \begin{cases} \left|v(x)^{k+1} + v(x+1)^{k}\right| & x = \min(x) \\ \left|v(x)^{k+1} - v(x+1)^{k}\right| & x = \max(x) \\ \left|v(x \pm 1)^{k+1} - v(x \pm 1)^{k}\right| + & \text{otherwise} \end{cases} \end{aligned} \tag{5.13}$$

It is noticed that Equation 5.13 controlled the local contour shape in four conditions, seen Figure 5.8. These conditions made each pixel of the contour to be spatially linked with the neighbour pixels. Each control point (pixel in the contour) could only move in vertical steps of one pixel per iteration to enforce the connectivity since the retinal layer is continuous.

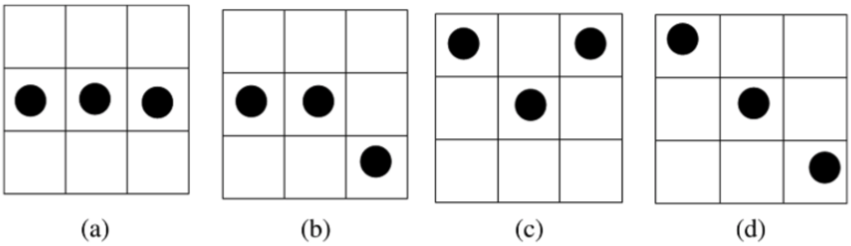


FIGURE 5.8: Four configuration types for three adjacent contour points. (a) Horizontal, (b) inflectional,(c) V-shaped, and (d) diagonal.

Because the boundary does not fold or overlap, it was assumed that each A-scan only contains a single point of the boundary. Based on this feature, the search space of each pixel was constrained to its A-scan. When updating the contour location, the pixel was updated in a back and forth sequence. Each evolving iteration started from the pixel at the left-edge A-scan and then travelled throughout all other A-scans and went back to the starting point. This was because the updating of each pixel was based on the position of the last pixel. The one-way updating sequence, which is the conventional method, may cause the contour position to be diverted from the real boundary without evoking the stopping criterion. This can cause a problem because one-way directional updating would cause incorrect updating of the pixels in the adjacent A-scan. Conversely, the back and forth sequence update the pixel twice (forward and backward) by considering pixels on either side. Therefore the wrong updating from one side was corrected by the other.

On the B-scan image, the space into which the contour is permitted to evolve into was defined by three factors: contour condition, search window size, and search direction, which are listed in Table 5.1. As described in this section, three contour gradient types were used to form three types of boundary condition. The search direction was used to define the limits of the search space. In particular, for the bi-directional search direction, the contour updating space was expanded both above and below the location of the coarse boundary according to the specified window size. The up search direction only allows searching above the boundary. These three factors make the contours adapt well to the specific conditions of each boundary. For example, a small window size guarantees that the very thin layers

Table 5.1: Contour search configuration

Layer boundary	**Search direction**	**Contour condition**	**Window size**
IS/OS	bi-directional	dim-to bright	8
OPRLayerU	bi-directional	dim-to bright	2
ISLayerU	bi-directional	bright-to-black	2
OSLayerU	up	dim-to bright	2
ILMLayerU	bi-directional	bright-to-black	2
ONLLayerU	bi-directional	bright-to-black	8
ONLLayerL	bi-directional	mix	2
ELMLayerL	bi-directional	bright-to-black	2
IPLLayerL	bi-directional	bright-to-black	2
OPLLayerL	bi-directional	mix	2
IPLLayerU	bi-directional	mix	2

are completely segmented, avoiding the layer-crossing error described in graph-cut based approaches 4.4.1.

The segmentation results for our active contour and the superpixel steps can be seen in Figure 5.9. It is clearly seen that the active contour method smooths the boundary between cell layers, whereas the superpixel method alone does not. Hence the active contour corrects for local errors in boundary estimation.

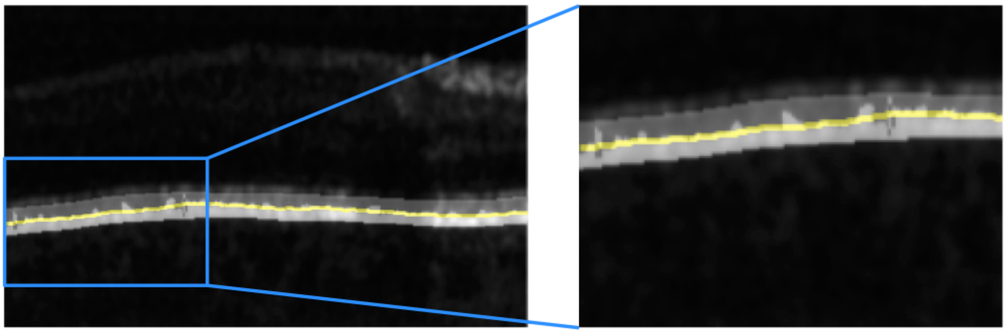


Figure 5.9: The refining results of active contour at RPEI/RPE boundary. The B-scan image is overlapped by the segmentation results by superpixel and active contour. As Seen in the zoomed window, the error generated by the superpixel is corrected by active contour that results in a smooth and continuous line fitting well to the actual boundary shape.

5.2.4 Layer estimation in OCT Volume using the Kalman filter

In clinical practice, the thickness measurement in the B-scan image is used for a preliminary examination of a patient, while measurements of the 3D scan volume can provide a more comprehensive diagnosis. The boundary segmentation in a single B-scan was obtained using the superpixel clustering method and active contour. To achieve segmentation of the 3D OCT volume, we predict, sequentially, the boundary locations in neighbouring B-scans using the preceding adjacent scan.

The boundary location in the adjacent B-scan was then predicted using the Kalman filter and active contour smoothing function. This works because the contents in adjacent B-scans are strongly correlated due to the fact that common SD-OCT systems obtain 3D volumes using fast raster scanning of closely packed B-scan images. In a patient B-scan, assuming minimal eye movement, the boundaries defining the retinal structure vary smoothly. These local changes in boundary shape are predicted using the Kalman filter and then updated by the active contour to ensure boundary smoothness.

Each pixel position in the adjacent B-scan is therefore calculated by

$$\hat{x}_k = \hat{x}_k^- + K_k \left(z_k - H \hat{x}_k^- \right) \tag{5.14}$$

where z_k is the known boundary location in the previous B-scan. The $\hat{x}_k^-$ is a priori estimation of boundary location, H is the measurement matrix, and K_k is the correction matrix. The $\hat{x}_k^-$ was calculated as

$$\hat{x}_k = A \hat{x}_{k-1} + B u_k \tag{5.15}$$

where the A was system matrix, and B and u_k are control variables. In this framework, the system matrix was set to 1, and u_k 0 since no external control was introduced into the prediction. We set the initial $\hat{x}_k$ as the boundary location in the first frame. To update the pixel prediction, the Kalman gain K_k was first calculated by

$$\begin{aligned} K_k &= P_k^- H^T \left(H P_k^- H^T + R \right)^{-1} \\ P_k^- &= A P_{k-1} A^T + Q \end{aligned} \tag{5.16}$$

where P_k^- is the covariance of priori measurement error, which was initialised as 10, R is the measurement covariance, and Q the noise covariance. In this system, the R and Q were set 0.1 and 1^{-6} separately. After each prediction, the covariance of priori measurement error P_k was updated by

$$P_k = (I - K_k H)\, P_k^- \tag{5.17}$$

The prediction of all boundaries was performed in a sequence. After the processing of each boundary, a smoothing function was applied to improve the boundary shape by filtering out vertically outlying pixels. Figure 5.10 demonstrates a detection result of the Vitreous-NFL layer boundary (the top retinal surface), predicted from corresponding boundaries in the nine preceding B-scan images. The figure shows the 3D layer without smoothing, where the vertical dimension corresponds to the A-scan direction. The lower plot shows the predicted boundary at the tenth frame.

It is seen that the noise present in the initial boundary prediction, based on detection in preceding B-scans, results in some outliers. We refine the boundary using local regression, which assigned zero weight to the data outside six standard deviations from the mean. From the experiment, the smoothing window was set to be 40 pixels for the boundaries, except for the Vitreous-NFL layer and OPL-ONL layer, which was 20. These values were set empirically to remove the outliers while maintaining a good fit to the true boundary (Figure 5.12).

The last step is to convert the floating-point estimated boundary positions to unsigned 8-bit integers that are commensurate with the image data. To avoid the discontinuity introduced by this conversion, the pixel distance was checked, seen in Figure 5.11.

Figure 5.13 summaries the overall Kalman prediction pipeline. This procedure was based on the ideal tracking condition that the eye movement was negligible. However, in practice, this was not always guaranteed, such as in cases where the head was tilting. To deal with this issue, each time when the top boundary was predicted and corrected by the active contour, the image was realigned to the first frame (B-scan) using two endpoints consisting of the first and last A-scans within the frame.

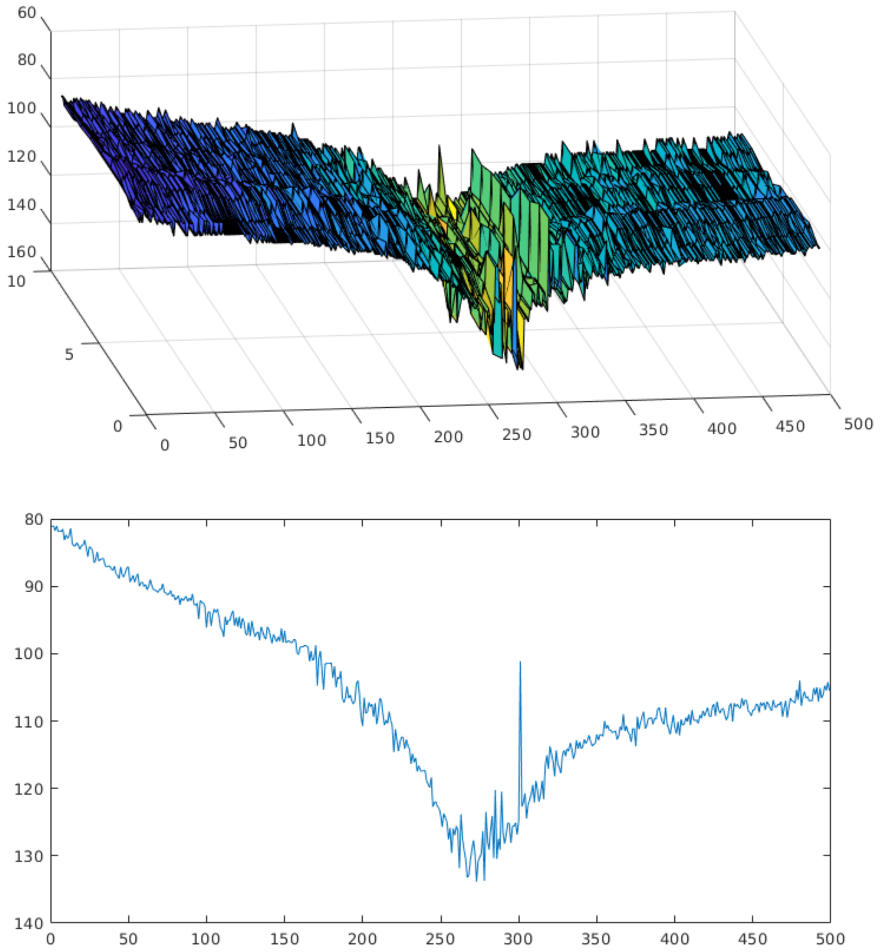


Figure 5.10: The prediction of Vitreous-IML boundary using the Kalman filter. Top: the detected boundary in nine frames. Bottom: the predicted boundary on the tenth frame. The Kalman filter's predicted boundary contains outliers, which was caused by the noise in the previous frames.

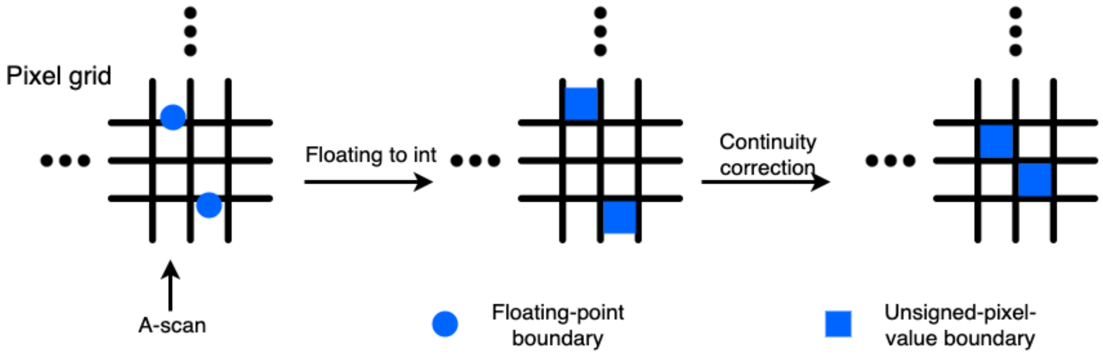


Figure 5.11: The diagram of checking boundary continuity when converting the floating-point values to unsigned pixel values.

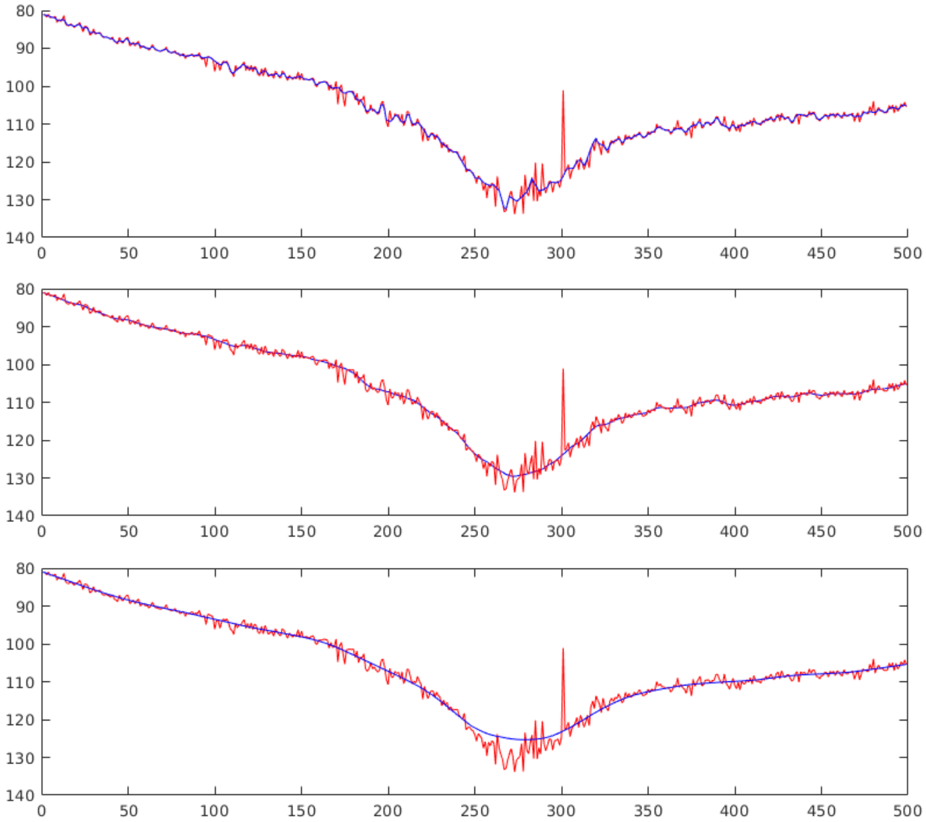


FIGURE 5.12: The outlier-refined results of Kalman filter prediction at the vitreous-IML boundary. Blue line: the boundary generated by using the Kalman filter. Red line: smoothed line by local regression. Top, middle, bottom: the refining results using smoothing windows of 10, 20, 40 separately.

5.3 Retinal boundary detection of OCT volume and segmentation results

In the previous section, the four essential algorithms for boundary detection were described: adaptive curve, superpixel clustering, active contour, and Kalman filter prediction. This section explains how these algorithms are integrated into our framework to perform 3D OCT volume segmentation.

The validation of the framework was implemented on OCT data generated by the Applied Optics Group, University of Kent. To test the generalisation of our framework, we collected data from two SD-OCT imaging systems with different scanning settings for spatial resolution and depth focus. The first dataset has a high axial resolution with depth focus is on the foveal point, and the second one has a lower resolution, and the depth focus is in the vitreous. The data contains

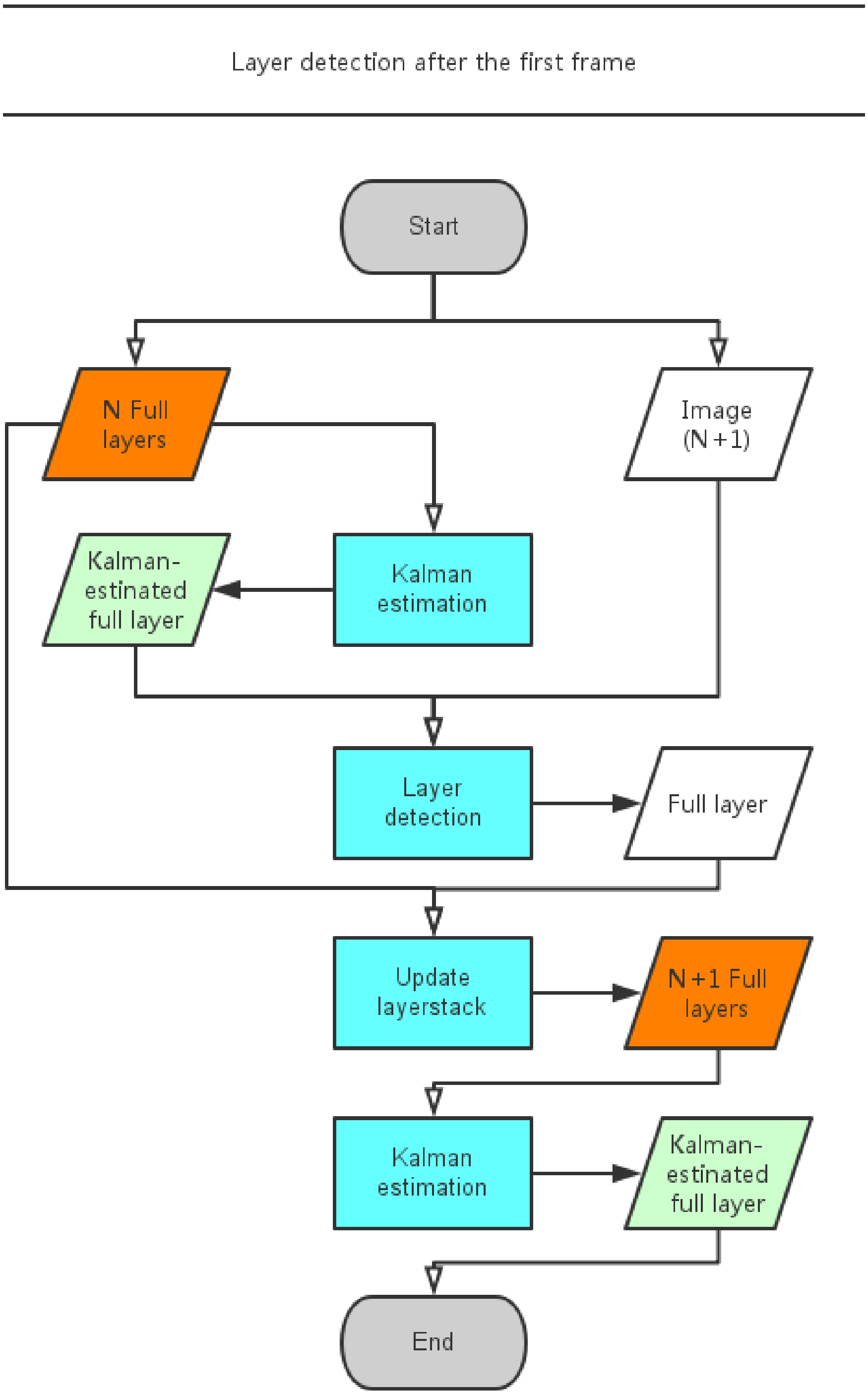


FIGURE 5.13: The schematic of boundary prediction using the Kalman filter.

OCT volumes of the fovea regions from the healthy eyes of two volunteers. Each volume contains 192 frames (B-scans). The size of each B-scan is 500×500 pixels, covering the areas from the vitreous to the choroid. The framework was assessed on both the original and a denoised dataset. In the denoised version, the speckle noise was suppressed using a 3D median filter. The framework was implemented using MATLAB.

5.3.1 Detection framework

Three main steps form the detection framework: 1) retinal area detection, 2) internal layer boundary detection, 3) volume segmentation. As described in section 5.2.1, the retinal area was first determined in the B-scan to limit the search range of the internal boundaries. For detecting the internal boundaries, the segmentation sequence was set based on three aspects: the relative positions of the layer structure, the differences in tissue reflectivity and layer shape. Specifically, it first focused on the layers within the sub-retinal region (the lower-section in the B-scan). Thereafter, we limited the search to the inter-retinal region and segmented the remaining layers. The 3D segmentation was initiated from either the first, or last, B-scan image. The first frame was taken as a reference for baseline layer detection. Thereafter the remaining B-scans were processed, in each case taking the previous frame as the reference.

Because the B-scans used for the test were directly generated from the OCT system, several essential pre-processing operations, summarised in Figure 5.15, were performed before the detection. First, OCT artefact removal was implemented. The raw B-scan was affected by two systems generated artefacts: the baseline noise and the high band noise. The baseline noise, seen in Figure 5.14, was always present and had a similar appearance to a genuine layer boundary. Since the layer search begins at the bottom of the B-scan, the presence of the baseline noise can be falsely recorded as a boundary position. High band noise may or may not be present but always appears at the same vertical position within the frame for a given OCT system. These artefacts also affected the detection of the retinal region. To resolve this issue, the pixel values corresponding to these linear artefacts were replaced by the average pixel values of the vitreous area.

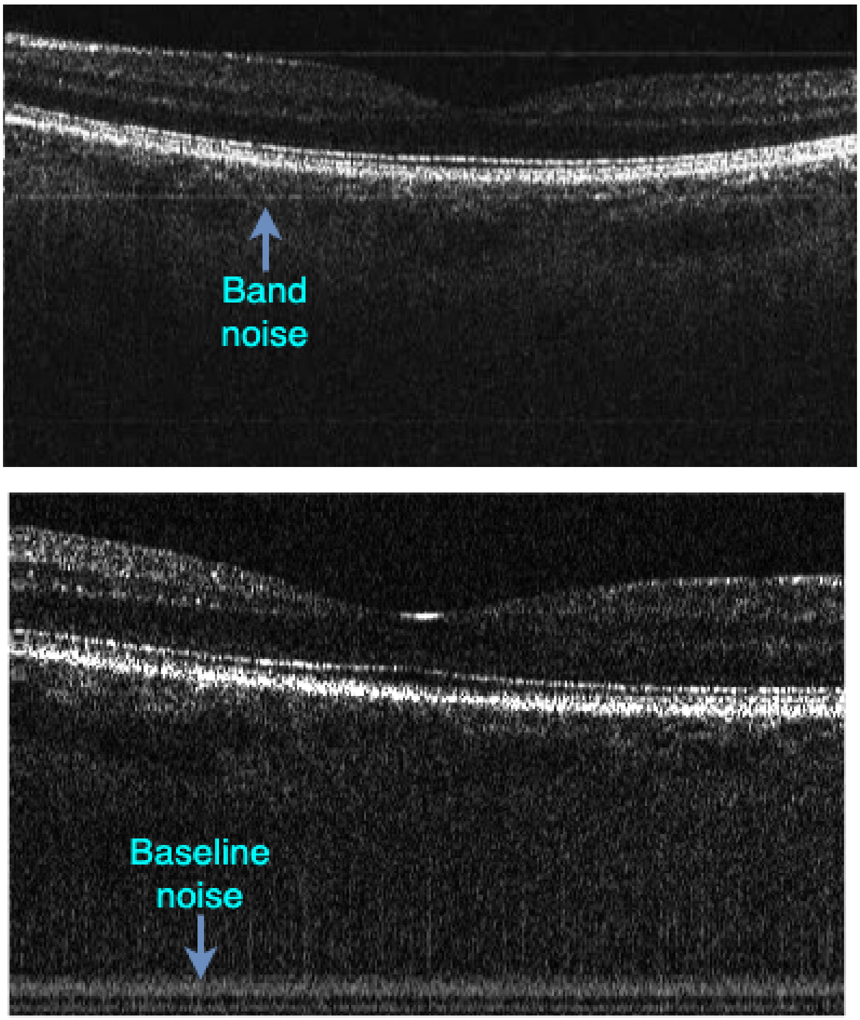


FIGURE 5.14: The optical noise of the OCT system. Top: two band-shaped noises are presented near the top and bottom of the retinal region in the OCT image. They always appear at the same location. Bottom: The baseline noise resented at the bottom of the image. It always appears in the OCT image during the scanning.

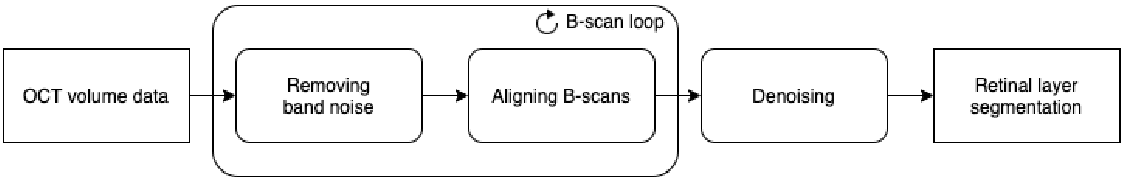


FIGURE 5.15: The pre-processing procedure of the OCT volume data.

The second step was the B-scan registration. Due to the spontaneous movement of the eye or head, the resulting B-scans were poorly registered in the 3D volume. This caused a ripple deformation of the retinal structure, seen Figure 5.16. To counter this problem, we register the B-scans using a TurboReg registration approach [96]. Specifically, we selected either the first, or last, B-scan of the volume as a source image. The rest B-scans, as target images, were automatically aligned to the source image in a rigid-body transformation. i.e. the retinal structure doesn't have deformation changes. This process guaranteed the smoothness of the layer surfaces in the 3D volume representative of the true retinal structure. It facilitated subsequent accurate boundary prediction using the Kalman filter at a later stage in the layer detection framework.

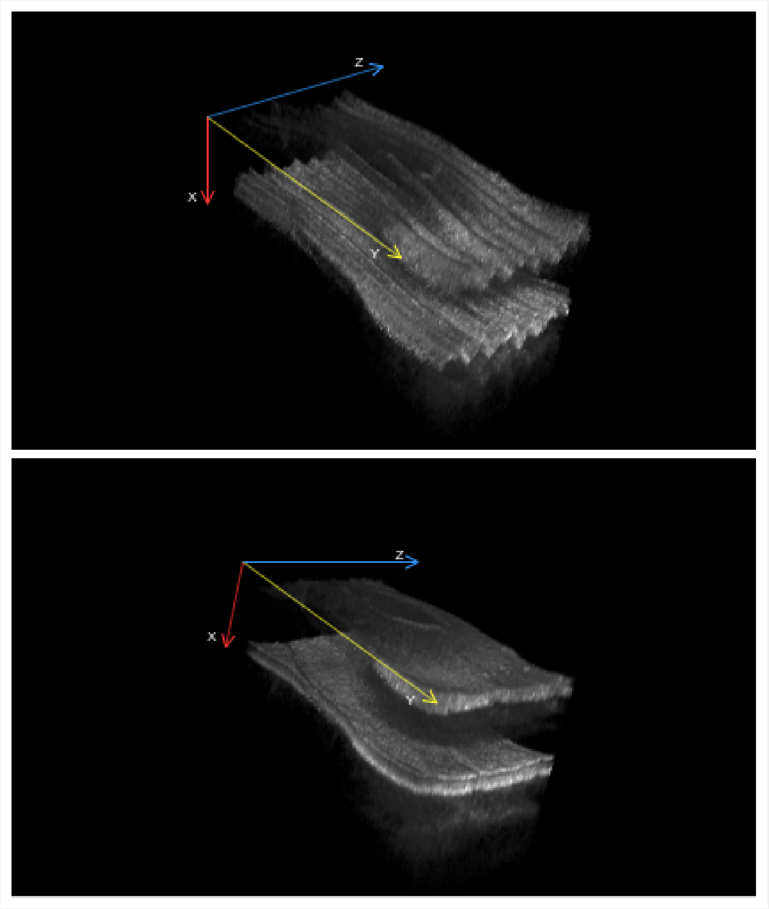


FIGURE 5.16: The volumetric data alignment. Top: raw 3D OCT dataset. The fast scanning direction is along the Y-axis, and slow scanning is along the Z-axis, in which direction the surface is aligned. Bottom, the dataset after the alignment. The surface wobbling effect is removed.

The boundary segmentation sequence is summarised in Figure 5.17. It illustrates the procedure of the overall 3D volume segmentation, including the retinal upper and lower boundary and ten internal layer boundaries. Its grayscale colour represents the layer position in the axial direction. The dimer, the deeper the layer lies.

After pre-processing, the first step within the framework was detecting the most salient (high-intensity response) IS/OS boundary. Then this boundary was used as a spatial search limit when identifying the layers above or below it. The IS/OS layer was segmented by initialising two superpixels corresponding to top and bottom regions. Each superpixel represented the inner and outer retinal region, where the inner region is from the retinal surface to the IS/OS layer, and the outer region is from IS/OS to Choroid. The interface between these superpixels approximately identifies the boundary position, which is subsequently refined by applying an active contour with a positive external energy term to smooth the boundary.

Next, the framework segmented the low curvature layers in the *outer* retinal region. The retinal layers located at the bottom of the B-scan had a higher intensity response and relatively flat structure, compared with the upper layers. Such layers also vary smoothly in the fovea region and hence can be easily and efficiently segmented. A number of superpixels were initialised, corresponding to the number of remaining layers in this region. After the superpixel clustering, the boundaries between each two adjacent superpixels were formed to do the active contour. By iterating this process, the remaining boundaries were detected in order of top to bottom within the gradually narrowed search region. Instead of using the boundaries formed by the first few superpixels, the new superpixels that were initialised each time provided more robust results.

In the second stage, the *inner* retinal layers were segmented. The retinal structure of the region above IS/OS, unlike the region below it, changes significantly in the foveal region. Generally, in the OCT B-scan image of the fovea, the internal layers merge into a single layer at the fovea region. B-scans acquired outside the fovea region comprise stacked, approximately horizontal and parallel layers. The procedure of detecting layers located in the region between the vitreous-NFL boundary (top retinal surface) and IS/OS boundary (inner retinal layers) was performed. Instead of processing from top to bottom, the boundaries that were closer to the region edge (top and bottom boundaries) were segmented with higher priority. This was because the limits of the search region produced by this sequence guaranteed the inner layers merged at the fovea point. First, the framework detected the OPL-ONL boundary, which was just above the IS-OS boundary. Then the active contour with negative external energy term was applied to refine the shape. This procedure was repeated for the remaining layer ending with the IPL-INL boundary.

5.3.2 Retinal layer segmentation in B-scans

The detection of the vitreous-NFL boundary was achieved using eight iterations of the adaptive curve algorithm. The shape of the boundary was updated at each iteration. The Figure 5.18 top shows the boundary detection in a 3D OCT volume where the individual B-scan at the fovea point is presented separately. It is seen in that the active contour enforced the continuity and smoothness of the retinal surfaces between adjacent B-scans. The Figure 5.18 bottom shows the segmentation result of active contour in a single B-scan, selected from the fovea point. The results show that the band noise removal resulted in an apparent discontinuity in biological tissue. However, the active contour was not affected by this artefact since its energy function ensures its smooth shape.

For the dataset of the high axial resolution, we achieved a segmentation of 12 boundaries on a raw B-scan image, shown in Figure 5.19. The B-scan of the retinal fovea regions was directly segmented by the framework without any pre-denoising process. The high axial resolution of the OCT system allowed the thin layers of the sub-retinal region (beneath the surface) to be successfully identified. The high-intensity response at the fovea point is due to the focusing point of the OCT system. The foveal region is shown in Figure 5.19 (B) provides a closer look at the merging of layer boundaries at the fovea. The thin layers in the central foveal region were differentiated from the surrounding tissue by relative intensity magnitude, and the boundaries were accurately segmented using our method. Figure 5.19 (C) shows the segmentation of the left-hand side of the fovea region, where our method provides stable segmentation results in the upper layers, which have lower contrast than the bottom layers.

For the dataset of the low axial resolution, we achieved a segmentation of 10 boundaries on a B-scan image, shown in Figure 5.20. This image was selected from the near fovea, where only part of the layers was merged. The image preserved

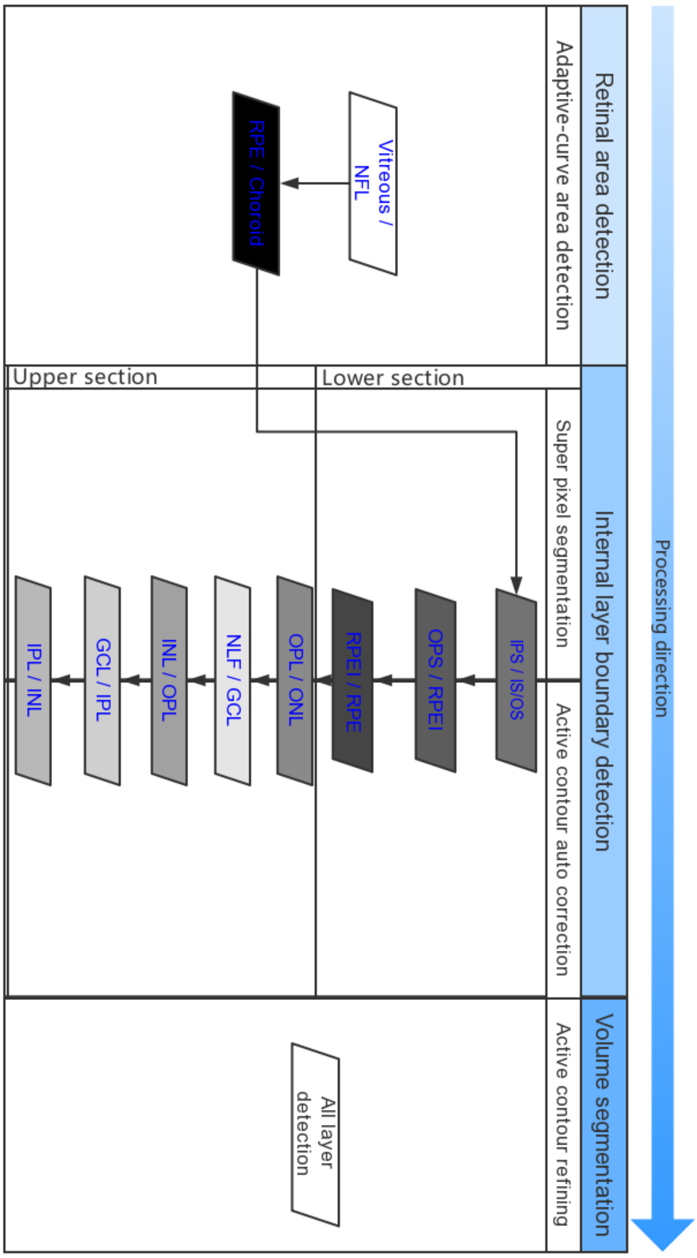


FIGURE 5.17: The schematic of the whole framework of retinal layer boundary detection. The framework is formed by three sections: 1) ROI region detection, 2) individual B-scan image segmentation, 3) volumetric data segmentation from the initial frame. The grey-level of the blocks indicate their boundary location in depth.

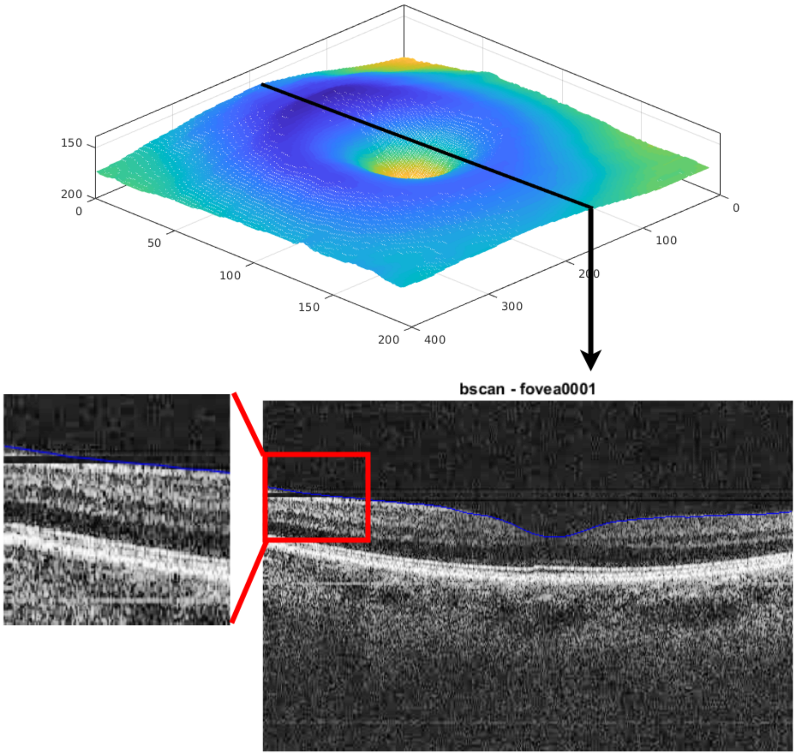


Figure 5.18: The segmentation results of the vitreous-IML boundary. Top: the inner retinal surface generated from the volumetric data segmentation. The black line indicates the location of the example B-scan image. Bottom: the example B-scan image. The zoomed-in of the red box shows the boundary detection is not affected by the band noise removal.

the thick retinal layers, but the thin layers in the outer retina were not preserved, including ELM, IPS, and RPEI layers. Therefore, the framework was not able to segment these layer boundaries. It is seen at the left fovea branch (the left area of the fovea) that the framework was able to detect the boundaries even when the image contrast of the NFL layers is poor. Figure 5.20 (B) provides the segmentation results at the right branch of the fovea region. It is seen that the GCL layers contain multiple blood vessels. This caused the inconsistent tissue appearance across the B-scan and projected shadow gaps on the outer retinal layers. However, the boundary detection was accurately performed, and no miss-segmentation was found. Figure 5.20 (C) shows the segmentation at the left branch of the fovea, where our method provides stable segmentation results on tissues with more uneven reflectivity. It is seen that the outer retinal layers have a long low-reflectivity gap. These inconsistencies in tissue appearance did not affect the ability of our

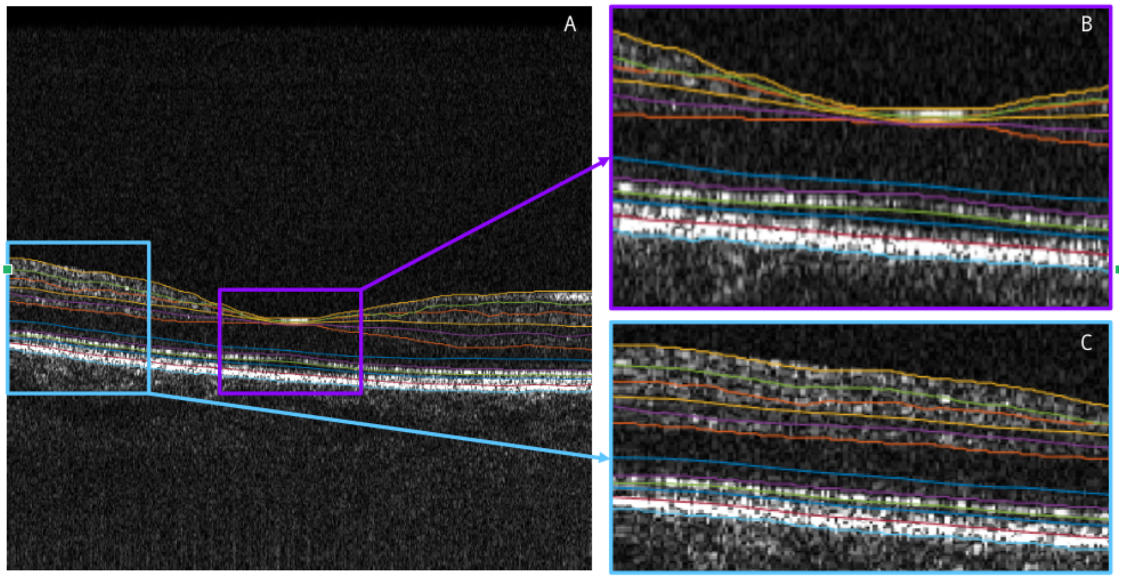


FIGURE 5.19: The segmentation result of 12 boundaries in the single B-scan. (A): The segmentation was processed on the original OCT image without pre-denoising. (B, C): the zoomed view of (A).

framework to detect boundaries since the combination of the superpixel clustering and active contour boundary correction can bridge such reflectivity gaps.

5.3.3 En-face image extraction of retinal layers

En-face OCT is routinely applied to different aspects of ophthalmology, including the imaging of the anterior, glaucoma, and retinal disease [97–99]. It provides a more detailed image of the eye structure beneath the surface than fundus ophthalmoscopy, which is only able to image the retinal surface. En-face OCT offers additional benefits to clinicians. A significant advantage of En-face OCT image is its ability to image retinal layers on the transverse plane. Using this method, it is possible to identify defects and visualise the vessel network accurately, in particular the sub-retinal layers, using their axial position on OCT cross-sections. In addition, the En-face OCT images can be overlaid onto ophthalmic images captured with retinal angiography to provide clear blood vessel information without the effect of blood leaking.

The conventional En-face image is a horizontal slice of the OCT volume data (C-scan). However, it is not able to provide a comprehensive view of the tissue since the slice is not along the layer boundary. The image generated in this way

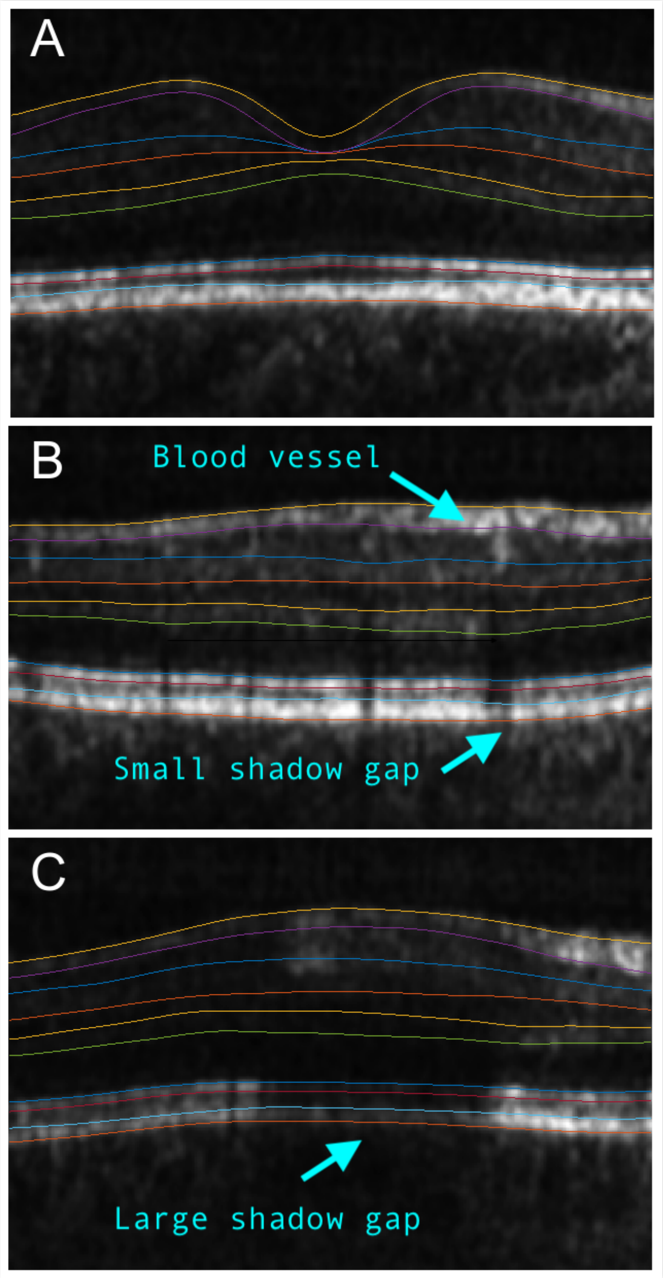


Figure 5.20: The segmentation result of 10 boundaries in the 3D volume. Top: the B-scan frame near the fovea point. The framework is able to process the very low-intensity retinal tissue. Middle: the B-scan at the right branch of the fovea. The framework is able to segment the retinal vessel area. Bottom: the B-scan at the left branch of the fovea. The boundary detection was not affected by the large shadow gap.

may contain content representing a cross-section of multiple layers. Therefore, it is not accurate for presenting a surface image of the retina. Hence, accurate detection of boundaries is essential for displaying retinal layers in a manner that can be easily interpreted by a Clinician.

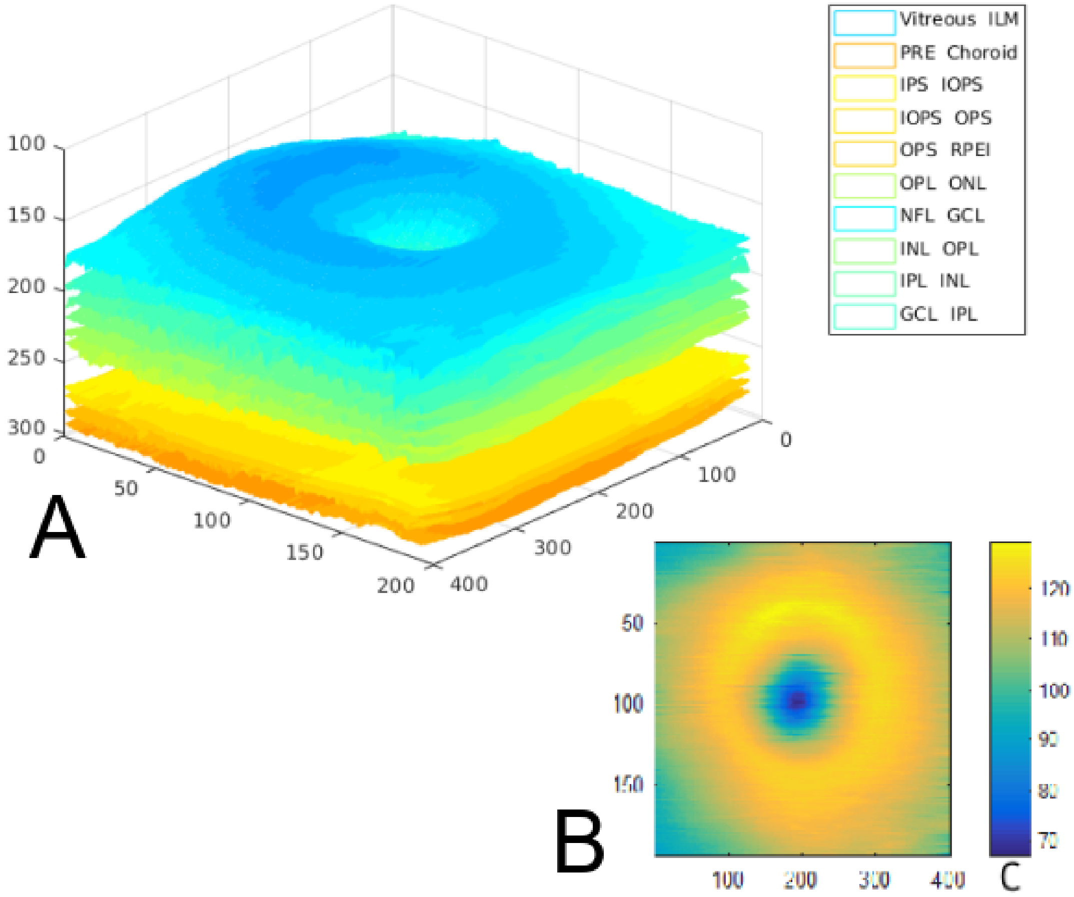


FIGURE 5.21: The segmentation result of 10 boundaries in the 3D volume. (A): The 3D surface model of segmented boundaries. (B): The thickness map of the NFL layer calculated from the volume segmentation.

Figure 5.21 provides a 3D view of the segmented OCT volume. In this volume, ten layers were segmented and coloured. Figure 5.21 (B) shows a thickness map of this foveal region, where the central blue region corresponds to the fovea point, and the circle is due to the normal fovea uplift. It is seen in the feature map that the left region was thinner than the right side. It is due to the NFL layer thickening. Specifically, in the healthy eye, the NFL layer is slightly thicker on one side of the fovea, which connects to the optical nerve head.

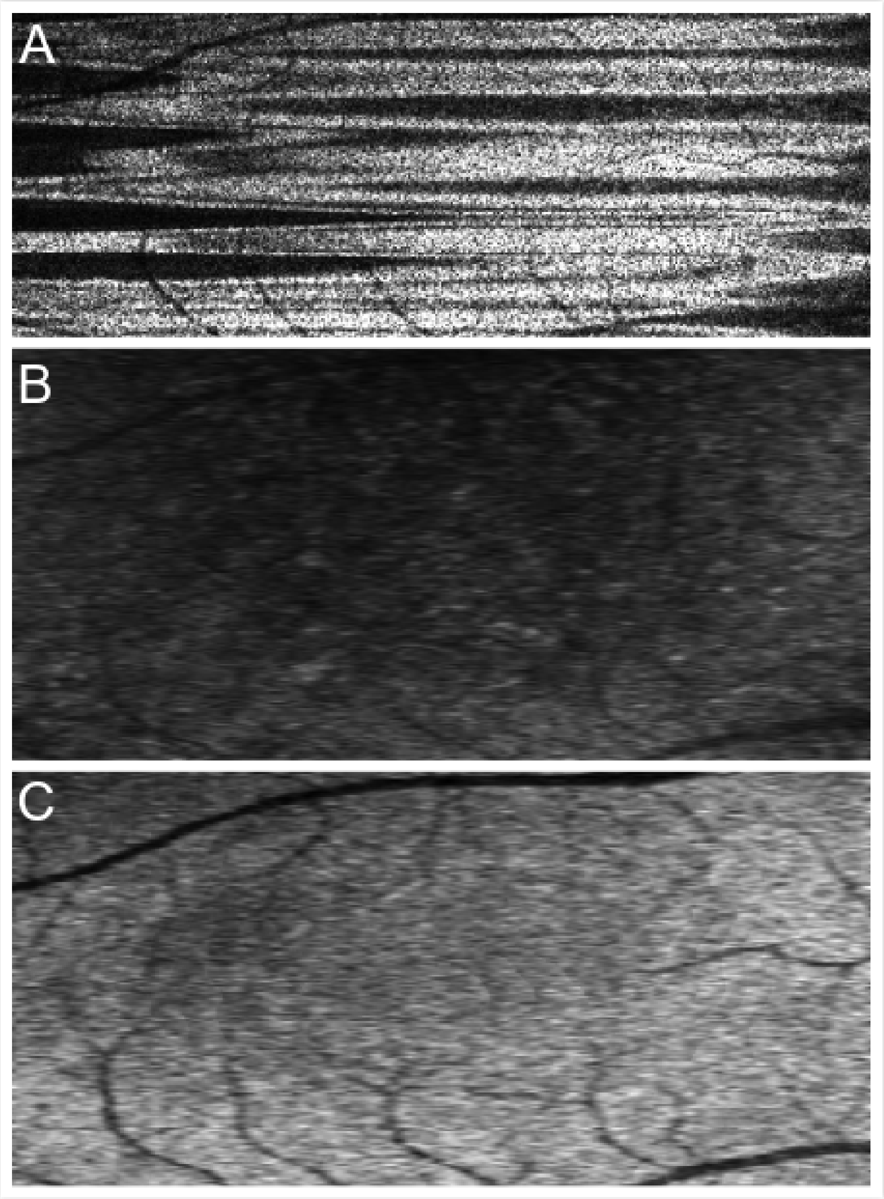


Figure 5.22: En-face image (C-scan) of the IS-OS layer. Top: C-scan view of the raw volumetric data. Middle: C-scan view after the B-scan frame alignment. Bottom: C-scan at the same position along the IS-OS layer boundary. The tissue intensity is consistent, and the blood vessel shadow is clearly shown.

Figure 5.22 shows the en-face image of the IS-OS layer, which was presented at three processing stages. These images were generated using three methods. Figure 5.22 (A) shows the C-scan sliced from the raw OCT system. It is seen that the shadow projection of the vessels is shown in the image. However, the frontal view was not completely presented since the spontaneous eye movement resulted in poor registration during scanning.

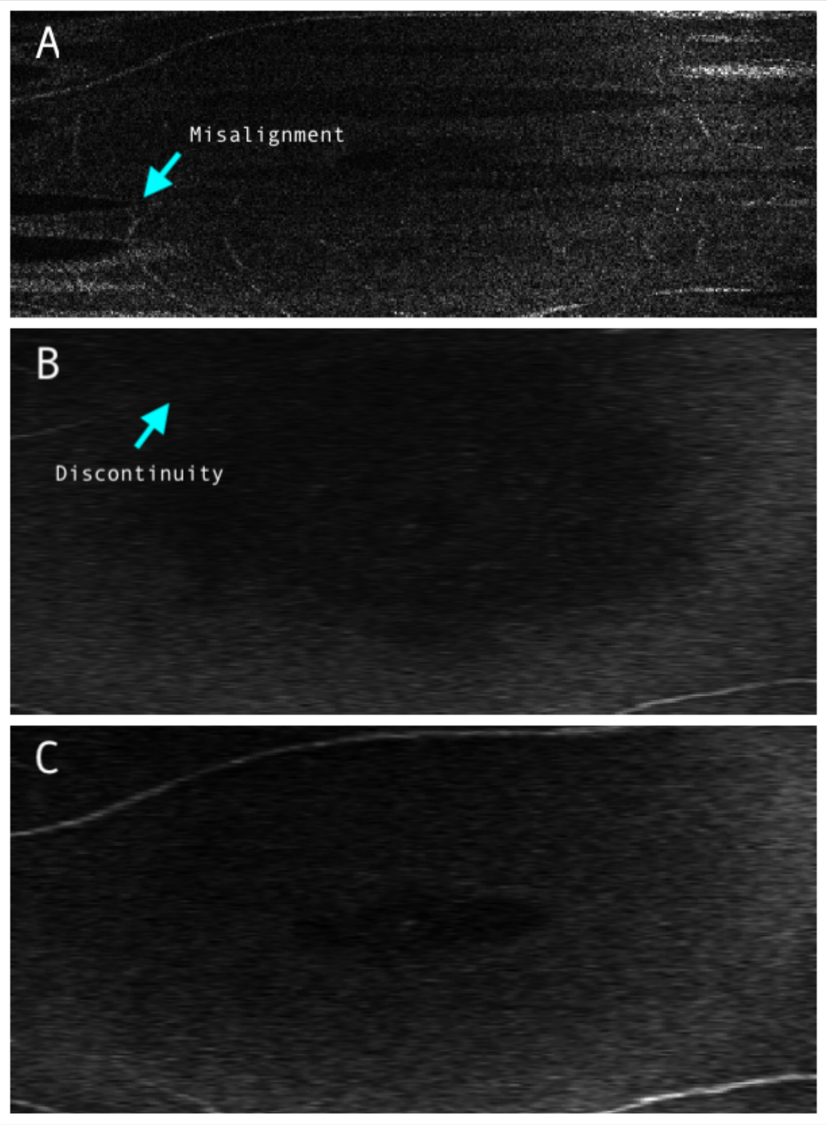


Figure 5.23: En-face image (C-scan) of the GCL layer. Top: C-scan view of the raw volumetric data. Middle: C-scan view after the B-scan frame alignment. Bottom: C-scan at the same position along the IS-OS layer boundary. The blood vessel embedded at this layer is shown with more consistent intensity.

The IS-OS layer was one of the thinnest layers. Its simple C-scan is shown in Figure 5.22 (A). It is seen that the C-scan does not represent the full view of that layer. Figure 5.22 (B) shows the same layer by slicing a registered OCT volume. It is seen the image quality can be improved by correcting the misalignment of the B-scan due to eye movement. However, the image was not displayed with clear content, i.e. the left side is slightly dimmer than the right side. This is because the layers were slightly tilted with respect to the C-scan plane, and there cannot do capture precisely the whole scan region. Figure 5.22 (C) provided the S-can that cut the layer along the boundaries, which were pre-segmented using our framework. It is seen that the tissue was displayed with vessel shadows and a smooth grayscale profile, making analysis of the vascular structure straight forward.

Lower retinal layers may exhibit the shadow of blood vessels that are located in the layers above. The en-face image can investigate vessels of each layer individually. Figure 5.23 shows the vessel profiles of the GCL layer (the second top retinal layer) at three different C-scan types. It is seen that the unregistered profile in Figure 5.23 (A) present vessels in different layers and the misalignment effect at the left edge. In Figure 5.23 (B), the C-scan with conventional registration removed the misalignment effect but still cannot show the correct vessel profile. In contrast, the full view of the vessels was visible in Figure 5.23 (C) thanks to our accurate boundary segmentation.

To summarise, our multi-boundary segmentation framework facilitates 3D retinal visualisation by providing a comprehensive frontal view of individual retinal layers and accurate thickness measurement of the retinal structure.

5.4 Discussion and conclusion

In this chapter, we described an approach to segmenting the retinal layer boundaries in the B-scan images and 3D OCT volumes. The framework consists of several detection algorithms that work in sequence: 1) the retinal region localisation using the adaptive curve, 2) internal layer segmentation using superpixel clustering, and active contour 3) boundary detection in adjacent B-scan using a Kalman filter. Our framework demonstrates an improvement of visualisation of en-face C-scan along the layer boundary in terms of stability and completeness.

The effectiveness of the adaptive curve algorithm is attributed to the region detection in only a few selected A-scans, rather than the entire B-scan. This method yielded a processing time of 0.010 seconds, which can satisfy the real-time requirement of the OCT scanning rate of less than 100Hz. Therefore, it can be utilised for fast processing in real-time retinal content stabilisation such that the OCT scanning process can display the retinal content stably on screen. In addition, the interpolation methods of the adaptive curve algorithm are carefully selected with respect to the features in the raw OCT image. It avoided overfitting while preserving the shape of the target layer.

The A-scan superpixel clustering and our customised active contour act as two key components of the framework. With the aim of ideally adapting the superpixel formation in our case, we conclude two crucial steps for the successes, i.e. the region limitation (range) and superpixel initialisation procedure for detecting each boundary. The number of the superpixels were initialised according to the remaining layers in the limited search region. These two points worked in conjunction during the whole process for all the internal boundaries. The further boundary refinement using the active contour, which followed the clustering of superpixels, modifies the coarse boundary shape such that the smoothness and accuracy were improved. Its accurate boundary adherence is based on the customisation made in energy functions, evolution direction, and windows sizes. By using these functions, the contour treated different boundaries separately according to their appearance and the coarse boundary position was enhanced significantly.

The generalisation of our framework was also an important factor. The performance of retinal layer segmentation is mainly evaluated on our own collected dataset. Our testing datasets were generated using two SD-OCT systems with different specifications. On these datasets, we achieved an accurate segmentation of 12 boundaries on the high-resolution OCT volume and successful boundary segmentation in the conditions of low boundary contrast, high-reflection vessels and vessel shadow projection. It is worth noting that the changes to the OCT system and retinal conditions may affect the overall performance. Different OCT systems may vary on de-noising process and contrast. As we demonstrated in this Chapter, our model can give an accurate performance on the low-contrast dataset. However, in other commercial OCT systems such as Topcon, the low-contrast OCT images were heavily affected by the speckle noise. In this case, our active contour model may be affected as it mainly relies on the image gradient.

In terms of the retinal conditions, we think the overall retinal structure is a key factor that can affect the layer segmentation accuracy. It was described that our framework was partially based on prior knowledge of the retinal structure. In this chapter, we evaluated the performance in a healthy retina. We think the framework is still applicable when the retinal pathology doesn't affect the retinal structure, i.e. the retinal tissues were slightly deformed but the overall structure was still maintained. This includes conditions such as intra-retinal thickening and age-related macular degeneration, where only the thickness of the affected retinal layer was changed. Our framework can give a good indication of those changes in thickness. However, we can foresee the limitations of our framework when dealing with the server pathological conditions, such as oedema or detachment presented in the retina.

The experimental results presented in this chapter indicate that the boundary detection framework can perform accurate layer segmentation on both B-scan images and volume datasets. The number of segmented boundaries is more than the previous method since the advanced feature of the framework allows it to segment more thin layers while being resistant to noise. However, the retinal layer analysis only provides partial information for retina examinations. It was discussed in Section 4.4 that the quantification analysis of macular oedema can provide more useful information for clinical diagnosis. Therefore, a reliable method for detecting macular oedema associated with various medical conditions is required. This stimulated our interest to develop a robust approach to segment macular oedema, which will be described in the next chapter.

Chapter 6

Cystoid macular oedema segmentation using transfer learning

6.1 Introduction

Cystoid macular oedema (CMO) is a symptom of diabetes and is a major cause of visual impairment. This ocular disorder is becoming a common health affecting people worldwide [100]. Other pathological conditions causing CMO related visual degradation include: AMD, vein occlusion and epiretinal membrane traction. [101]. OCT is the tool favoured by clinicians for non-invasive detection of CMO in patients with diabetic retinopathy. [102, 103]. OCT scanning simplifies the assessment procedure by showing the CMO in a cross-section view, which provides more details than fundus photography (although the latter modality is sometimes used to complement OCT). Tissue affected by CMO has different optical reflectivities which allows it to be distinguished from the healthy tissue using OCT. This variability in pixel texture makes it possible to measure accurately CMO size and monitor its change over time [104–106], which is essential for the clinical diagnosis.

The segmentation of CMO in images is always the most important step of a clinical analysis. This problem has previously been approached using B-scan images and a wide range of image processing methods which are summarised in

Section 4.4.2. However, this is a challenging task due to the complex morphological characteristics of CMO which can affect segmentation accuracy. In OCT images, the various symptoms caused by the CMO morphology often result in a high diffusion of the boundaries between the healthy and diseased tissue. This diffusion increases the risk of inconsistent segmentation results. Furthermore, some ad-hoc rules, which were built into the segmentation procedures, were designed based on the specific nature of the target object, such as oedema edges or selected texture features. When the character of the target object varies or becomes complex, more pathological conditions should be considered to maintain segmentation accuracy. This results in feature-driven methods becoming complicated and less efficient.

In an attempt to overcome these issues, deep learning based methods have been employed. The convolutional-based deep neural network has been developed to do image classification and segmentation by extracting 2D features from the training images. This data-driven approach demonstrates good performance when dealing with the appearance variation in image data. A fully convolutional neural network (FCN network) [28] is an end-to-end trainable network. It extends the image classification capability of deep neural networks to the task of semantic segmentation. Specifically, the image classification network, such as VGG net [107] and GoogleLeNet [108] is implemented by using a set of convolutional layers and fully-connected layers in sequence. The fully-connected layers are often placed at the end of the network to do the class prediction. A general FCN network is often constructed by replacing the fully-connected layers of the image classification network with convolutional layers, which extends the network's output from 1D to 2D. By making this change, the network uses a sequence of convolutional layers to extract local and global features and then predicts the class of each pixel contained within the input image. At the same time, the trained weights in the unchanged part, the original convolutional layers, of the network was still utilised. Therefore, this approach is able to exploit the learned convolutional kernels of a classification network for use in segmentation tasks. This approach is an example of transfer learning.

Transfer learning is a mechanism for efficient network training that utilises a pre-trained network. In the remainder of this chapter, we present an approach to transfer learning of the FCN-based framework for segmenting semantic area in retinal OCT B-scan images, exhibiting polymorphous CMO. Our framework consists of an FCN network and fully connected conditional random fields (dense CRFs). In the first step, the framework trained an FCN model to segment the CMO regions by extracting features from retinal images. Instead of training from scratch, the adopted FCN model was previously trained on natural image segmentation tasks [109]. A binary classification layer was added at the end of the original network structure, and the transfer learning was applied to further train the model with the aim of CMO segmentation. By doing the transfer learning, the natural-image derived convolutional kernels were fine-tuned, and the model was transferred to adapt to the retinal OCT images. In the second step, the dense CRFs were used as a post-process to refine the segmentation by utilising the local appearance of oedema.

In this chapter, we introduced the characteristics of the CMO condition contained within our database, followed by a description of our network framework. Then, the model training strategy was described in Section 6.3. In Section 6.4.2, we present the segmentation results on our testing dataset and discuss the model's performance. We conclude the chapter with some suggestions for modifications that could improve the accuracy of our work.

6.2 CMO characteristics and segmentation framework

In this section, we describe the characteristics of the CMO presented in the OCT dataset of diabetic retinopathy. Moreover, we explain the inconsistent data ‘ground truth annotations from multiple clinicians. After that, we describe details of two components used in the framework: the FCN network and the dense CRFs.

6.2.1 CMO characteristics of the diabetic retinopathy dataset

The target dataset of diabetic retinopathy includes the OCT B-scans that were collected from diabetic patients having two oedema types: 1) intraretinal cystoid spaces and 2) diffuse retinal thickening. The intraretinal cystoid spaces contain a fluid substance that forming low-reflectivity cavities that vary in shape. Therefore these regions generally appear dark in the OCT images and are surrounded by layers with higher reflectivity. Conversely, the thickening regions are formed by denser tissue and show high reflectivity. Often, thickening regions are not easily distinguishable from healthy tissue in OCT images because the boundary between healthy and diseased tissue is not clearly shown. An example of the OCT retinal image from the dataset is shown in Figure 6.1, exhibiting both two conditions of oedema. It is seen that the retinal layers at the fovea region are deformed and separated by large cystoid spaces within the intraretinal layers. They are surrounded by the diffused retinal thickening in the extended areas.

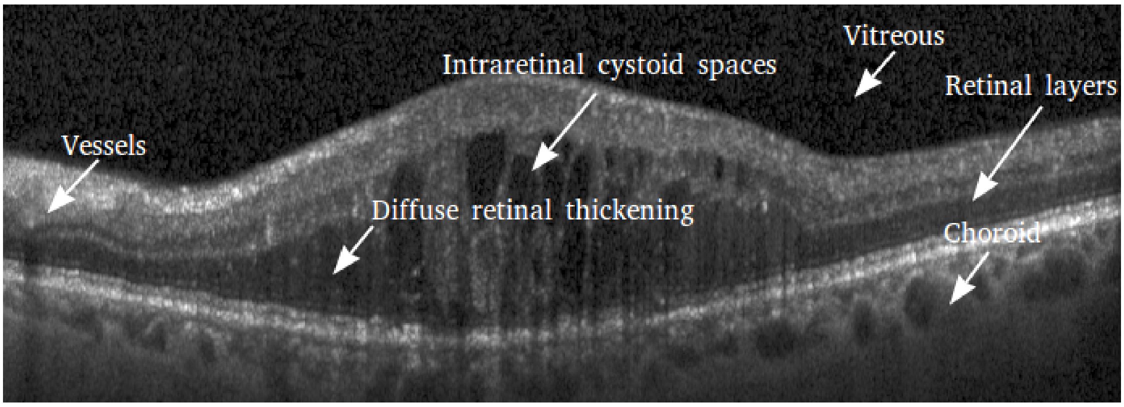


FIGURE 6.1: A retinal OCT image from our experimental dataset with two DMO symptoms in the foveal region: 1) intraretinal cystoid spaces and 2) diffuse retinal thickening. The retinal layers are located between the vitreous and choroid layers. In the fovea point, the retinal layers are deformed by the cystoids oedema (the dark empty spaces) and thickening tissue (at two sides of the fovea point). The highly reflective dots with shadows underneath are blood vessels, which also cause a small retinal layer deformation.

The appearance of the two symptoms may introduce bias to clinicians when doing ground-truth annotation. Conventionally, medical professionals segment the diffuse retinal thickening by subjectively estimating the thickness and delineating the boundary at a reasonable position based on their experience. Therefore, different professionals may produce inconsistent results. The manual annotation of the training and test dataset was implemented by two ophthalmologists. Their labelling work was done independently. The annotation of the retinal oedema was performed by freehand delineation using a custom software tool [110]. The cystoid regions were segmented individually according to their boundaries, while the retinal thickening was segmented according to the estimated location of oedema with respect to the retinal layers. Figure 6.2 depicts annotations obtained from two ophthalmologists (both are fellowship-trained medical retina specialists). It is seen that the cystoid oedemas, which have clear boundaries, were consistently identified by both graders. However, the segmentation of the retinal thickening regions by different ophthalmologists was affected by their subjective bias.

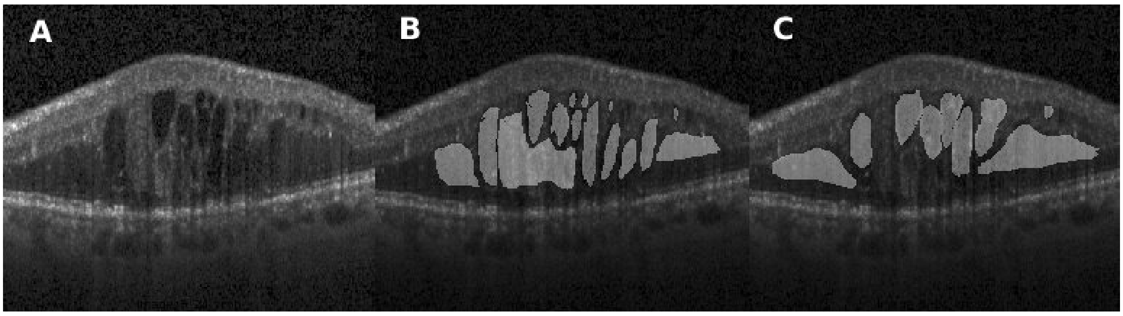


Figure 6.2: Ground-truth segmentations of DMO in OCT retinal image, carried out by two ophthalmologists separately: (A) An original OCT image of the dataset; (B) and (C) are manual segmentation results performed by two ophthalmologists (Expert A and B). There is a significant difference between the two manually segmented masks. The central region, where the appearance is closer to a retinal thickening, was annotated by expert A (B). But it was identified as deformation of normal retinal tissue by the expert B (C).

6.2.2 Problem statement

Given a retinal OCT image $I_{M\times N}$, the goal is to assign each pixel $\{x_{i,j}, i \in M, j \in N\}$ to a class c where c is in the range of the class space $C = \{c\} = \{1, \cdots, K\}$ for K classes. In our study, we consider a 2-class problem that treats the CMO regions as the target class and others as the background class.

6.2.3 FCN network model

The architecture diagram for FCN is shown in Figure 6.3. In an overall view, the FCN network structure was formed by using a sequence of convolutional layers, pooling layers, and transposed convolutional layers. These layers were applied to extract image features at different image scales. Moreover, a set of skip connections [111], which is the key component of the network, was used to improve the segmentation accuracy. These links provided an identical feature mapping operation that combined small-scale feature maps of the pooling layers and the large-scale feature maps of the transposed convolutional layer.

The network is based on a sequence of convolution and activation layers, forming a feature encoder that aims to recognise the spatial characteristics of the

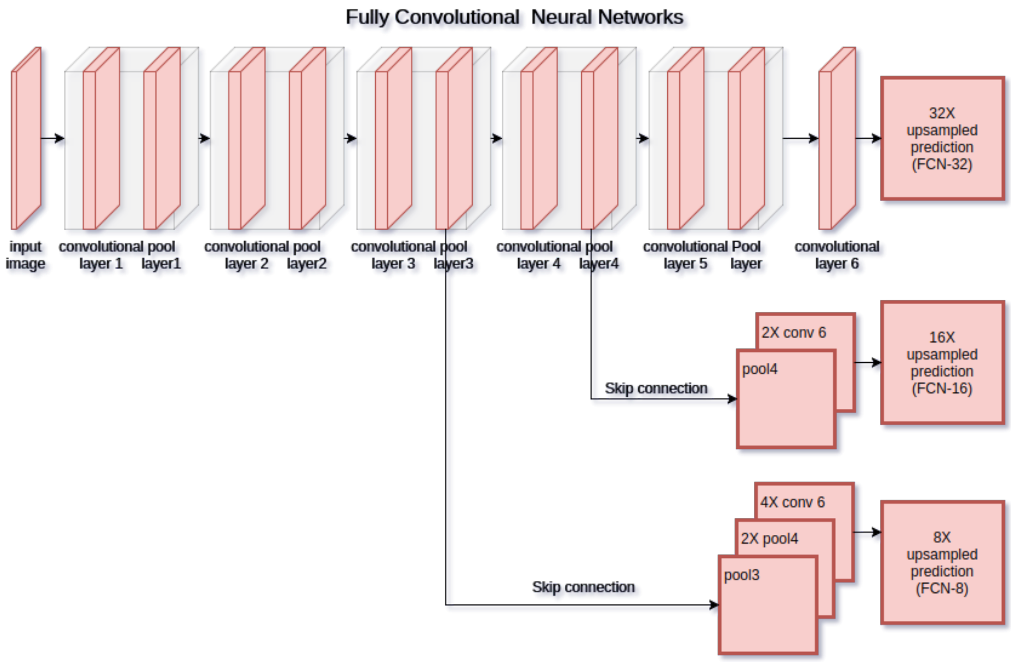


FIGURE 6.3: A schematic of the FCN network with different skip connection schemes. The convolutional layers with different filter sizes carry out the convolution operations to the initial image input or the output of the previous layer. Each of the convolutional layers is followed by a pooling layer that down-samples the spatial dimension of data while mitigating the computational overhead. The skip connection concatenates the feature maps of earlier pooling layers to the feature maps of the later convolutional layer. It combines the information of both deep and shallow layers, which refines the spatial segmentation of the image.

DMO regions. The retinal content is presented in a single-channel OCT B-scan image. Given an OCT image I, of size $M \times N \times 1$, the first convolutional layer implemented the data convolution with spatial kernels, $\{F_1\}$, where each kernel is in size of $m_1 \times n_1 \times 1$. The convolution for each kernel resulted in a feature map, whose size depends on image padding and stride size. To ensure the kernel can cover all the areas, zero paddings and single stride size are applied to the convolution. At the output of the first convolutional layer, the feature maps of $\{F_1\}$, were stacked as an input to the next layer. The process of successive convolutional layers extracted diverse features from the image data by applying the trained kernel weights.

For deeper convolutional layers, the intermediate feature maps had high dimensional feature channels, and therefore the convolutional operations of the feature maps involved intensive computations. To mitigate the high computational cost, the network applied a pooling layer after each set of convolutional layers. The pooling layer reduced the spatial dimensions of the feature maps by averaging local spatial information within the pooling window [112] and output a down-sampled version of each feature map. After this operation, the convolutions carried out in the deeper layers were calculated on smaller feature maps and therefore were more efficient. This improvement in efficiency is achieved at the expense of lost spatial resolution in the feature information.

The loss of resolution due to the pooling operation can be compensated by using skip connections, which have been shown to improve the prediction accuracy in previous work [113]. By doing the average pooling and feature activation after the intermediate convolutional layers, the feature maps have a high response to the target regions that we aim to segment. To ensure the segmentation output of the network has the same size as the input image, the network applied an up-sampling layer at the end to perform a transposed convolution (or deconvolution [114]) to its input feature maps. The up-sampled feature maps were then channel-wise averaged by the final binary classification layers to generate an output mask image that indicates the location of the target object. However, this predicted segmentation had coarse resolution resulting in poor boundary detection of the target object. This is due to the upscaling being based on the smallest feature maps, which discards most of the spatial information. The segment accuracy was improved by using the skip connections that concatenated the feature maps of the previous pooling layers to the outputs of the final convolution layer. For instance, in Figure 6.3, the feature maps of *Conv* 6 was upsampled by using a 2× upsampling layer and then concatenated to the feature maps of *pool layer* 4. The up-sampling layer was used to align the spatial resolution. Thereafter, the network performed the final binary mask prediction.

6.2.4 Fully connected CRFs classification

After the prediction of the network, a fully connected CRFs model [115] was applied as a post-processing step to refine the segmentation results based on the OCT image content. For a given image and its corresponding label prediction, which is the output feature maps produced by the FCN model, the fully connected CRFs model made fine adjustments to the class labels for each pixel. To this end, a set of feature vectors determined by the input image and feature maps (the vector contains pixel location and value in different channels) were modelled. Given an image I of size $N \times N$ and the corresponding random field X over K possible pixel-level labels $L = \{l_1, l_2, \cdots, l_k\}$, a conditional random field (I, X) is characterised by a Gibbs distribution.

$$P(X \mid I) = \frac{1}{Z(I))} \exp(-\sum_{c \in C_G} \psi_c(X \mid I)) \tag{6.1}$$

where $G = (V, E)$ is a graph on X in which each clique c in a set of cliques C_G induces a potential ψ_c. In the fully connected CRFs model, G is a fully connected graph of X and C_G is the set of all unary and pairwise cliques which forms the Gibbs energy function of labelling $x \in L^N$ in Equation 6.2,

$$\begin{aligned} E(X) &= \sum_{c \in C_G} \psi_c(x_c) \\ &= \sum_i \psi_u(x_i) + \sum_{i<j} \psi_p(x_i, x_j) \end{aligned} \tag{6.2}$$

where i and j range from 1 to N. The unary potential $\psi_u(x_i)$ is calculated for each pixel by the FCN model, producing a heat map with label assignment x_i. The pairwise potential is described by:

$$\psi_p(x_i, x_j) = \mu(x_i, x_j) \sum_{m=1}^{K} \omega^{(m)} k^{(m)}(f_i, f_j) \tag{6.3}$$

where each $k^{(m)}$ is a set of Gaussian kernel functions for the feature vector f of pixel i and j separately, $\omega^{(m)}$ describes the linear combination weights, and μ is a label compatibility function which introduces a penalty for neighbouring similar pixels which were assigned different labels. The pairwise function classifies pixels with similar feature vectors into the same class so that different classes are prone to be divided at content boundaries. Then the fully connected CRFs model infers the final label of each pixel by using mean-field approximation [115].

6.3 Fine-tuning strategy

During the network training process, the weights for convolutional layers can be learnt from a randomly initialised state. However, this strategy has a risk of overfitting the model when a small training dataset is used, e.g. 800 training samples in our case. To avoid the overfitting issue, we applied the transfer learning to fine-tune the model from a set of pre-trained weights to learn features in the new problem. In this strategy, the pre-trained weights in the shallow convolutional layers were previously optimised to extract the basic local features of the image content from a relatively large-scale dataset. We continued training the pre-trained weights and optimise these for the new dataset. To adapt the network for our segmentation task, the final classification layer was modified such that the number of prediction classes was aligned to the number of target objects (the type of macular oedemas) in our dataset. This new layer was initialised by random weights, which were trained together with the pre-trained weights.

6.3.1 Dataset and data augmentation

To validate the effectiveness of our approach, we trained and tested the fine-tuning framework on a publicly available OCT dataset, which was used to test the segmentation performance in previous work [3]. The training data was pre-processed with denoising and data augmentation. We randomly took 25% from it as our test samples in our four-fold cross-validation experiment.

The dataset comprised 110 labelled SD-OCT B-scan images. The dataset were collected from 10 anonymous patients with diabetic retinopathy from moderate to severe condition. For each patient, eleven images were selected from 3D volume data, where the spatial size of each B-scan is 512 × 740. These images were equally distributed at the fovea point and two branches of the foveal region, i.e. 1 B-scan at the central fovea and 5 B-scans at 2, 5, 10, 15 and 20 slices away from the central region on each side. For each image, the cystoid CMO regions, which had a dark-cavity appearance, were firstly annotated by clinicians using the software. Then they were reviewed and manually corrected to refine the segmentation. Thereafter, the clinician performed the annotation for the retinal thickening regions by comparison of the thickness of the entire retinal layer.

Two image processing steps were applied to the original images before they were used in training. First, the image data was denoised to suppress the speckle noise present in the images. Instead of denoising each individual image, our pre-processing involves 3D denoising. Specifically, we applied the BM3D algorithm [116] on each 3D OCT volume. This algorithm can generate high-contrast denoised

OCT images by using a block-matching denoising method. From our experimental results, the denoising process suppressed OCT speckle noise while sharpening the boundaries in the image and ultimately improved the presentation of CMO regions. Next, the retinal region was cropped from the OCT volume by removing unnecessary areas above the vitreous and under the choroidal layer since the content in those areas were almost blank and irrelevant to the retinal structures.

After the denoising, we applied data augmentation to the denoised images - a method for increasing the number of training examples when using small training sets. It is noticed that the number of patients was very limited compared to the size of data sets generally used in deep learning. The original data set, which contained 110 OCT images, was relatively small, and there is a high risk of the model overfitting to the training set. To overcome this issue, we augmented the data by applying spatial translations to each image, which involved random translation, rotation, flipping and cropping. Therefore, some of the augmented images contained partial retinal structures, which gave diverse samples during the model training. After the augmentation, the total amount of training images was extended by a factor of three in this manner. From the augmented data, we separated 20% of the total amount to be used for validation during the training.

6.3.2 Framework settings

Our framework used an FCN-8 model that was adapted from the VGG-16 network [107]. The layer construction of the network is shown in Table 6.1. This architecture concatenated the output of the final convolutional layer with the output from the last two pooling layers to make an 8-pixel stride prediction net, i.e. the output prediction mask of the network was 8-times smaller than the input images. On top of the based-line architecture, an additional classification layer was added after the original output layer to reduce the final mask to binary prediction. In our experiments, the FCN-8 model weights were pre-trained on the semantic boundaries detection in the real-word images [117], since the original task was similar to OCT region segmentation in terms of of of the boundary localisation. With these weights, the fine-tuning was performed for all weights to learn new features of macular oedema in retinal OCT images. The fine-tuning hyperparameters were set as follows: the number of training epochs was 40, the batch size was set to 10, the base learning rate was 1.0e^{-4} and reduced by order of magnitude after every ten epochs. During the training, the input batch used the original pixel values rather than the mean subtracted values. The grayscale OCT image data was reformatted to RGB with channel conversion as required by the network input channel. The experiments were run on a desktop computer with an Intel CPU, 16 GB of RAM, and an NVIDIA 1080Ti GPU with 10GB VRAM.

For the post-processing after the network prediction, the parameter of the fully connected CRFs model was modified empirically to perform the best results on the OCT image. The original model [115] was set for RGB image inference. In order to adapt the model for grayscale images, the parameters of the kernel were changed in the pairwise function. The standard deviation is 2 for the colour-independent function and 0.01 for the colour dependent term. We set the ground truth certainty to 0.6 in order to obtain the best results.

During the training, the network was jointly fine-tuned with a multinomial logistic loss function and a Dice loss function, which compared the ground truth label and the prediction masks. Specifically, the multinomial logistic loss with a softmax activation, formulated as Equation 6.4, provided a probabilistic similarity at the pixel level.

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta\left(x^{(i)}\right) + \left(1 - y^{(i)}\right)\log\left(1 - h_\theta\left(x^{(i)}\right)\right)\right] \tag{6.4}$$

$$\sigma_i(z) = \frac{\exp\left(z_i\right)}{\sum_{j=1}^{m} \exp\left(z_j\right)}, i = 1, 2 \ldots, m \tag{6.5}$$

The Dice loss [118], which is described in Section 6.4.1, was used to compute the spatial overlap between prediction and ground truth. During the training process, we used stochastic gradient descent (SGD) as an optimiser to reduce the loss function.

Table 6.1: The architecture of the FCN-8 model used in the experiment. FS: filter size. NF: number of filters. NB: number of blocks. Link: skip connection.

Name	Type	FS	Stride	NF	Padding	NB	Link
Block 1	Convolution	3	1	64	100	× 2	
	ReLU	-	-	-	-		
Block 2	Max Pool	2	2	-	0	× 1	
Block 3	Convolution	3	1	128	1	× 2	
	ReLU	-	-	-	-		
Block 4	Max Pool	2	2	-	0	× 1	
Block 5	Convolution	3	1	256	1	× 3	
	ReLU	-	-	-	-		
Block 6	Max Pool	2	2	-	0	× 1	
Block 7	Convolution	3	1	512	1	× 3	
	ReLU	-	-	-	-		
Block 8	Max Pool	2	2	-	0	× 1	
Block 9	Convolution	3	1	512	1	× 3	
	ReLU	-	-	-	-		
Block 10	Max Pool	2	2	-	0	× 1	
Block 11	Convolution	7	1	4096	0	× 2	
	ReLU	-	-	-	-		
Block 12	Convolution	1	1	21	0	× 1	
Block 13	Deconvolution	4	2	21	-		
Block 14	Convolution	1	1	21	0	× 1	×2 fusion
	Crop	-	-	-	-		
	Element-wise Fuse	-	-	-	-		
	Deconvolution	4	2	21	-		
Block 15	Convolution	1	1	21	0	× 1	×4 fusion
	Crop	-	-	-	-		
	Element-wise Fuse	-	-	-	-		
	Deconvolution	16	8	21	-		
Block 16	Crop	-	-	-	-	× 1	
Block 17	Convolution	1	1	2	0	× 1	
Block 17	Dice	-	-	-	-	× 1	
Block 19	Softmax	-	-	-	-	× 1	

6.4 Experimental results

6.4.1 Evaluation metrics

To evaluate segmentation accuracy, the segmentation results of the fine-tuned model was accessed by using a Dice coefficient, which was also used in previous work [3]. The Dice coefficient, which is defined in Equation 6.6, calculates the index of overlay area between auto-segmented results X_{auto} and manual annotation X_{manual}. We calculated the segmentation overlap between the predicted output and the ground-truth by the primary grader and calculated the mean Dice coefficient for all patients.

$$\text{Dice} = \frac{2\,|X_{\text{pred}} \cap X_{\text{manual}}|}{|X_{\text{pred}}| + |X_{\text{manual}}|} \tag{6.6}$$

6.4.2 Segmentation results

After training the FCN network, the framework was tested with a separate test set which was previously unseen by the network. The randomly selected data comprises a set of OCT images exhibiting various morphology of CMO symptoms. This section presents the segmentation results of the framework for different CMO conditions and compared them to manual segmentation results. Moreover, we discuss the segmentation bias of our framework and Chiu's method [3] on the same test images, and compare them by using the Dice index to show an improvement resulting from our framework in terms of segmentation accuracy. Lastly, We discussed some prediction results on difficult data samples, where the automatic segmentation was performed with lower accuracy when compared with the manual annotations.

We found that our fine-tuned model demonstrated a generalisation capability that it had less segmentation bias in comparison with manual segmentation. At the same time, the model had a good performance on cystoid segmentation. The appearance of cystoid in OCT images is often shown with a higher contrast against surrounding tissue. Figure 6.4 (A) depicts a foveal region affected by cystoid spaces. Dominant oedema detaches retinal layers, and small oedema regions occur surrounding it. Figure 6.4 (B) and (C) shows the manual segmentation by

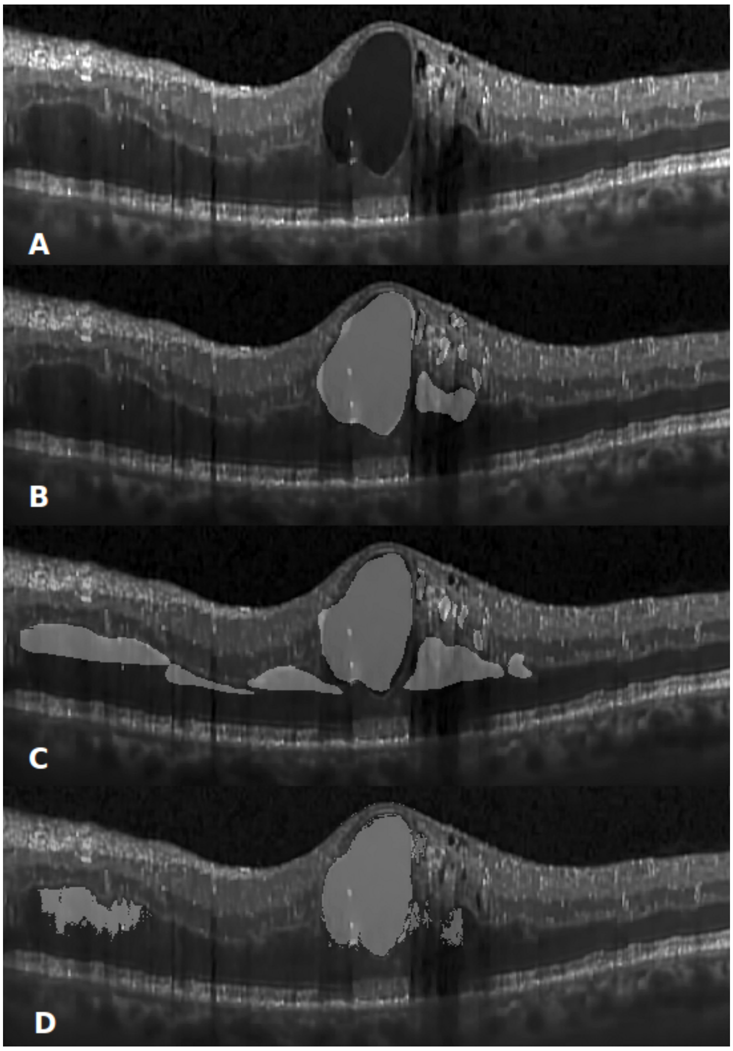


Figure 6.4: The segmentation results of macular oedema by two clinicians and our framework. (A): the original image of the macular oedema region. (B) and (C): segmented results by primary and secondary clinicians. The manual segmentation of the cystoid spaces is in good agreement. However, the region on the left branch was not identified as DMO by the primary grader. (D): segmented results obtained with the framework. The cystoid spaces are segmented with a good boundary alignment. A partial region of retinal thickening is also detected on the left branch, which indicates a balance of bias between two clinicians.

the primary and secondary clinicians. It is seen that there is significant inconsistency between two clinicians on the left branch of the fovea. Figure 6.4 (D) shows our results. It is observed that our model, which was trained using ground-truth provided by the primary clinician, was also able to predict partial regions that were labelled by the second clinician. This indicates that the model was trained to generalise the DMO features from the training data and therefore reduced the bias due to subjective experience. Moreover, we found that our predicted boundaries of the dominant oedema were segmented more consistently along the actual edge compared with the manual segmentation. It implies that our framework shows good performance in cystoid segmentation in terms of boundary detection.

Figure 6.5 demonstrates the model generalisation on predicting the retinal thickening. The raw image in Figure 6.5 (A) shows a retinal thickening within retinal layers on the right branch of the fovea region. Compared with the cystoid, the boundary of this oedema is not visually distinct from other tissues in the image due to similar optical reflectivity within the retinal layers. Although, the presence of oedema causes a deformation of the retinal layers, making them noticeably thicker than the layers on the left branch. It is observed in Figure 6.5 (D) that the segmented results obtained with our approach are consistent with the manually segmented results. It is worth noting that the unlabelled region near the right edge, which is out of their manual segmentation range, was also successfully segmented.

Apart from the simple-type DMO condition, the automatic segmentation on mixed-severity DMO condition also shows a good agreement with the manual segmentation. The retinal layer in Figure 6.6 (A) was deformed by a large section of oedema with a mixture of cystoid spaces and retinal thickenings of various shapes. The central section is affected by the cystoid oedema, and the thickening parts are located on both branch of the fovea. It is seen that both two sections were successfully segmented by our approach. It accurately captured the appearance of the pathology and covered the entire region. Several small sections were also segmented successfully. In Figure 6.7, we present more results from severe conditions that exhibit complex structure.

The evaluation results indicate that our framework yields better segmentation results in comparison with the previously proposed method [3]. Figure 6.7 shows a visual comparison of segmentation results for severe DMO, which were generated by two manual annotations in (A) and (B), the compared method (C), and our

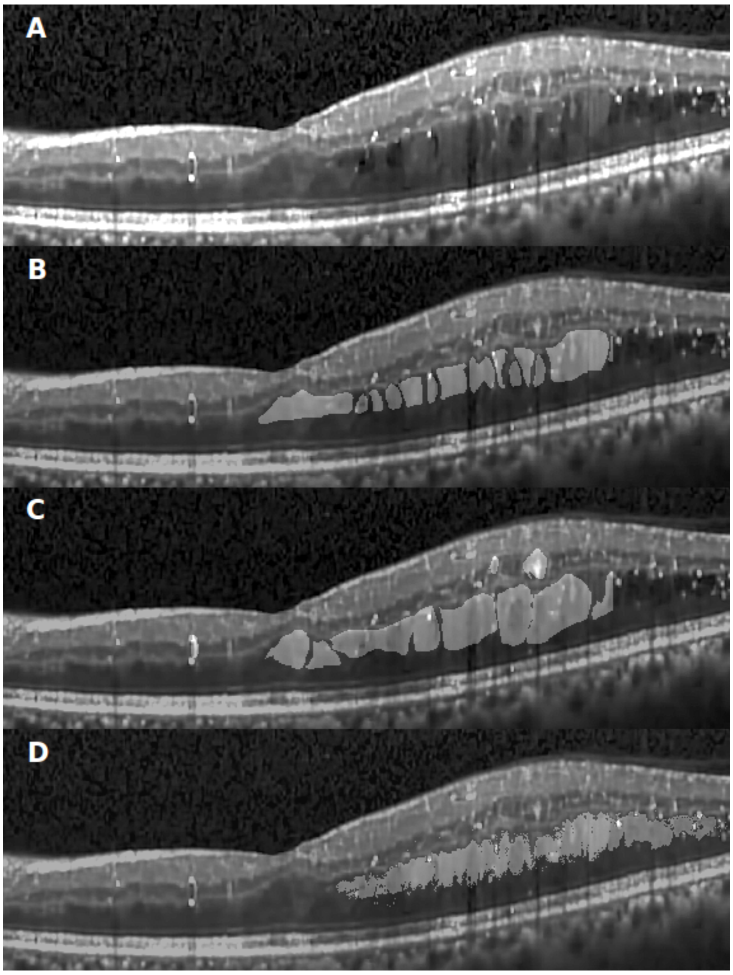


FIGURE 6.5: The segmentation results of retinal thickening by two clinicians and our framework. (A): original image. (B) and (C): segmented results by primary and secondary clinicians. The segmentation results by two clinicians show good consistency. The region near the right edge is out of the manual segmentation range. (D): segmented results by our framework. On the right branch, the area is segmented continuously.

framework (D). It is noticed that the segmentation by the compared method covers more tissues than the ground-truth annotation near the subretinal area (at the bottom of the retina). It is because the defect region has a very similar feature to the healthy tissue, which is due to the low image contrast and blurry boundary. Therefore the feature-based segmentation method wrongly identified the healthy tissues as the DMO defect. In contrast, the results from our method are more close to the primary segmentation results, and the predicted boundaries are more consistent with the ground truth.

We found that our method shows more accurate segmentation than the compared method on areas of retinal thickening. Figure 6.8, shows three samples that

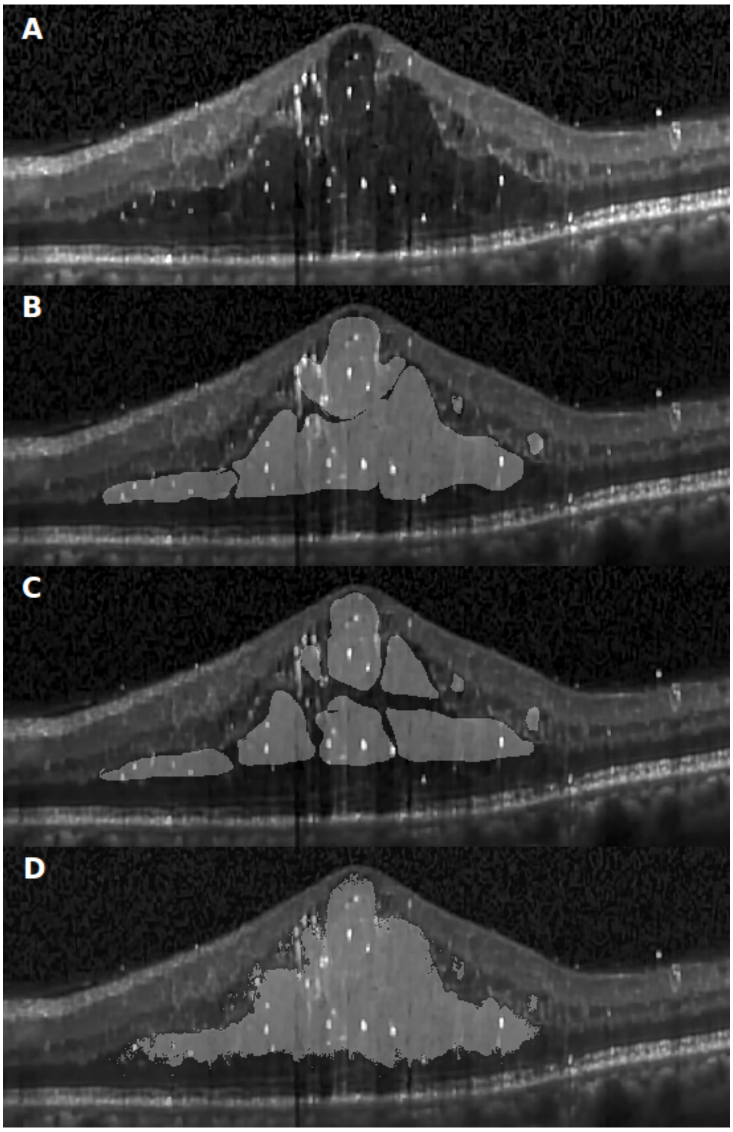


FIGURE 6.6: Segmentation results for more severe oedema as carried out by clinicians and the framework. (A): original image. (B) and (C): segmented results by two clinicians. (D): segmented results by the framework. It is seen that both the thickening cystoid spaces exhibiting complex appearances are segmented.

depict the segmentation results on retinal thickening areas. Retinal thickening structures are close in appearance to normal tissue, which can result in incorrect segmentation. This is clearly observed in the results of the compared method (Figure 6.8 (C)). In comparison, the results obtained by our method (Figure 6.8 (D)) show better agreement with the manual segmentation.

A quantitative assessment of segmentation accuracy was obtained using the Dice overlap coefficient. The Dice coefficient was calculated across all ten patients using the method previously described in [3]. Specifically, the Dice coefficient

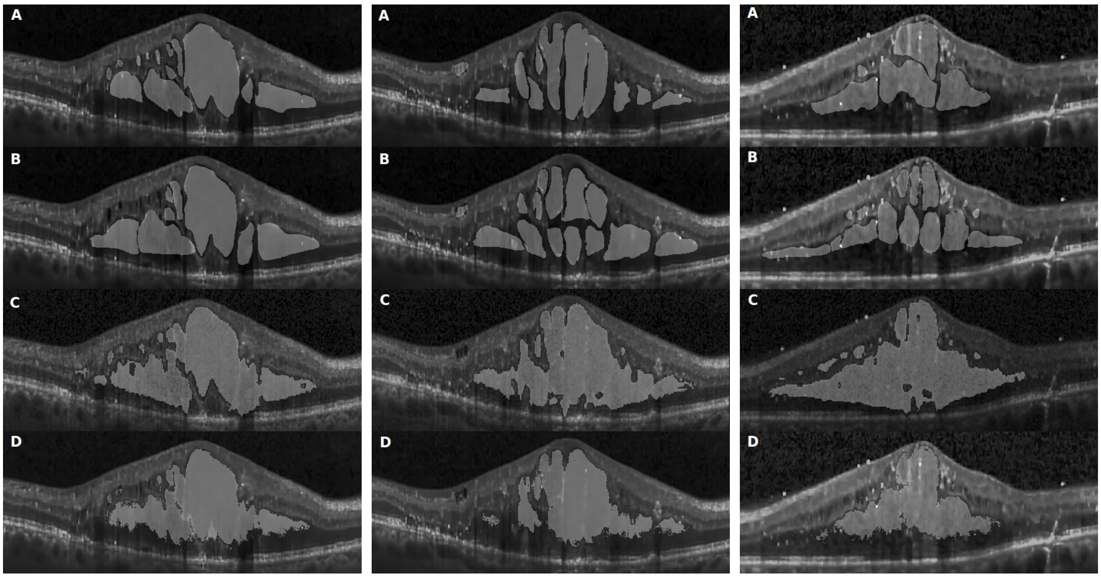


Figure 6.7: A visual comparison of manual and automatic segmentation. (A) and (B): annotations by two clinicians. (C): segmentation results from Chiu's work. The figure is presented with re-usage permission from [2]. (D): our results. From the comparison, it can be seen that the results at third row is slightly over-segmented due to the feature similarity. In contrast, our results are between two manual segmentation.

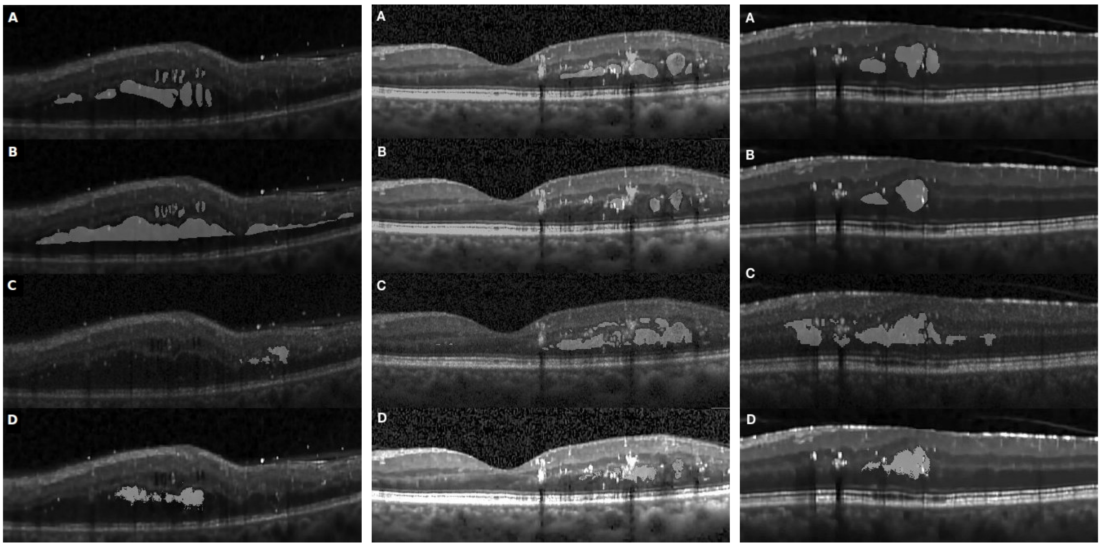


Figure 6.8: A comparison between Chiu's work and our results. (A) and (B): annotations by two clinicians. (C): segmentation results from Chiu's work. The figure is presented with re-usage permission from [2]. (D): our results. From the comparison, it can be seen that our results have a higher agreement with manual annotation. The compared method performed worse due to the similar feature between the defects and normal tissue.

was calculated based on all test images. It was found that the Dice coefficient for our approach is 0.60 ± 0.26 standing for mean and standard deviation, which outperforms 0.51±0.34 reported in [3] (a high mean and small standard deviation is a good indicator of the performance). Our Dice value is comparable to 0.58 ± 0.32, calculated for manual segmentation similarity between the two clinicians' (where a coefficient value of 1 indicates perfect agreement).

Apart from the segmentation of large areas, there are still some cases where the segmentation of the target region was performed with low boundary recall. One such event occurred when small target regions were not segmented out, as shown in Figure 6.4 (D). In this case, the small regions were affected by speckle noise and smoothing effects caused by the pre-processing and resulting in the small regions being undetected by the framework. This is due to their small initial size leading to them being smoothed out after several pooling layers. In another case, some target regions were only partially segmented, as shown in Figure 6.5 and 6.9.

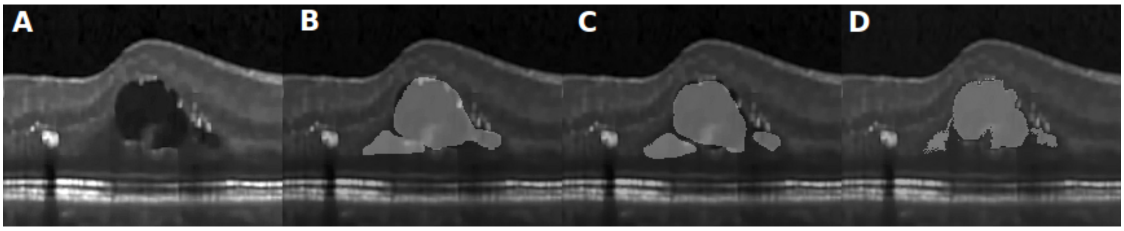


FIGURE 6.9: The segmentation result from inaccurate prediction. (A): original image. (B) and (C): segmented results by two clinicians. (D): segmented results by the framework. It can be seen that part of the retinal thickening is not completely segmented where it appears close to the surrounding tissue.

6.5 Conclusions and future work

In this chapter, we presented a framework for segmenting CMO regions in retinal OCT images. It comprised a segmentation model built with a fully convolutional neural network (FCN network) and a dense conditional random field (CRF) model for post-processing label correction.

The medical image analysis with a deep-learning model is often data-driven, which is usually critical to the success of the model. To counter this challenge, we applied several strategies. We used an FCN network as a base model that takes advantage of the convolutional layers to predict tissue classification at the pixel level. Our transfer learning strategy fine-tuned the network using pre-trained weights, and we applied data augmentation to compensate for the requirement of large amounts of training data. The network was trained based on a joint of cross-entropy loss and Dice index loss, which took into account the overlap between the predicted segmentation and ground-truth annotation during the training process.

The most important advantage of the FCN model is its skipping link. The skipping link allows the output from deeper convolutional layers to be concatenated with the output from shallower convolutional layers. This feature compensates for the shortage of the coarse prediction caused by down-sampling in the pooling layer and produces more refined results.

We emphasise that the key advantage of this method is its capability to recognise a variety of OCT features. The morphological properties of certain retinopathy can be diverse. Conventional segmentation methods for OCT retinal image segmentation were often tailored for a specific task, and their performance often varies depending on the retinopathy condition. Conversely, convolutional neural networks can be trained to perform well on various retinopathy conditions, thus avoiding the requirement for a multitude of ad-hoc rules. The segmentation results obtained by our framework demonstrated the capability of end-to-end convolutional neural networks for OCT retinal image segmentation tasks. They demonstrated that our learning-based approach can be adapted for more complex cases and is more flexible than fixed mathematical model-based approaches, and is, therefore, a step towards a universal OCT segmentation approach.

The work described in this chapter focused on the segmentation of cystoid macular oedema. The results of our method indicate that it could be extended to a broader range of ophthalmic conditions, such as deformed retinal layer segmentation caused by age-related macular degeneration, epiretinal membrane, macular hole and retinal detachment. Although the network architecture involves the skip link to fuse the appearance information of shallow layers with the deeper layer outputs to make a more precise prediction, the structural character still presents limitations in extracting accurate locations of small target regions. Those limitations encourage us to investigate the possibility of using more efficient network architectures to improve segmentation accuracy. Therefore, in the next chapter, we will present further research on a metric learning-based neural network that provides a more accurate segmentation of retinal lesions.

Chapter 7

Image retrieval-based few-shot medical image segmentation

7.1 Introduction

In the previous chapter, we trained a model to detect and segment retinal thickening and cystoid oedema from OCT B-scan images. In terms of guiding retinal image analysis, the neural network-based segmentation algorithms, described in the previous chapter, demonstrated good performance when dealing with the diverse appearance of macular oedema. In this chapter, we extend the work to image retrieval-based segmentation that the model can mark the disease area using an existing reference image of a similar symptom. This approach fits closer to the real-world diagnosis scenario in the hospital.

Medical imaging has been generally seen as the best diagnostic as it can be used instead of surgery or other invasive procedures. Along with the rapid development of OCT described in Chapter 4, various imaging modalities have also been applied widely in hospitals, including computed tomography (CT), magnetic resonance imaging (MRI), ultrasound imaging, and dermoscopy. The growth of medical imaging techniques has led to the development of a wide range of medical image analysis techniques.

Medical image segmentation is the fundamental objective of medical image analysis that assists in clinical diagnosis. It is applied to data acquired from various imaging modalities, such as optic disc segmentation in fundus images (ophthalmoscopy), retinal layer segmentation in the OCT images, melanoma segmentation in skin surface images (dermoscopy), and organ segmentation in radiation body scans (CT, MRI, and ultrasound imaging).

In the task of medical image segmentation, a common situation for implementing a CNN-based segmentation method is that the network training is usually data-hungry and task-specific. This is because training a model that has good generalisation capability for a particular segmentation task usually needs a large number of training images, which enables the network to learn diverse feature while avoiding over-fitting to the training dataset. However, developing a medical image dataset with a large number of manually annotated/labelled samples is challenging. Collecting large numbers of images for various pathological conditions is usually time-consuming and resource-intensive, and depends on the availability of clinical cases for a specific task [119].

To solve this problem, some researchers have proposed various strategies that reduced the risk of over-fitting when using small training dataset [120–122]. Our neural network-based segmentation algorithms, described in Chapter 6, also demonstrated an example of using a transfer learning strategy for model training with a small dataset. Using the trained model, we segmented the tissue lesion of the retinal thickening and cystoid oedema from the OCT B-scan images. Since the model training was object-oriented, i.e. training the model to only differentiate one type of retinal lesion from healthy tissue, the results indicated good performance when dealing with the variation of the macular oedema appearance. However, the model's generalisation still depends on the quantity and diversity of the available training samples. Therefore, we expected a method with more capability of model generalisation and feature representation learning. To this end, we developed a new segmentation method based on one-shot learning and content-based medical image retrieval (CBMIR), which are explained in the following content.

One-shot segmentation is a subarea task in one-shot learning. It is based on the principle that the neural network model is set to segment the object in a query image by providing one support image of an object, and the object belongs to the same class as the query object (thus the term one-shot). The one-shot approach allows the model to learn task-related segmentation, i.e. the model was trained to do a segmentation task given different objects. In this task, the segmentation is the objective, rather than doing segmentation for a specific object, where the object is the objective. In contrast to our previous object-oriented method, this one-shot segmentation network has the capability of doing cross-class segmentation when there are only a few training data of each class, enabling to train the network with fewer training examples. This naturally leads itself to the data-shortage conditions in which the medical images segmentation problem is set.

The CBMIR system retrieves medical images of similar content from image databases by using automatic feature extraction approaches. The development and wide use of medical imaging have contributed to the establishment of image databases, for which images are collected from different patients. This type of database offers us an opportunity to collect medical images that exhibit diseases at different degrees of severity. For the purpose of evidence-based analysis and diagnosis, the CBMIR aims to provide an efficient method to search the database and retrieve the images that have the most similar characteristics to the case of interest. These characteristics include visual features represented by characteristic colour, shape, and texture. The CBMIR has become a popular technique, and performs well when applied to data of multi-modality medical imaging [123].

Motivated by the aforementioned limitations of previous CNN methods, and inspired by the one-shot learning and CBMIR, we propose a new network-based approach, which we call MetaSegNet, to the medical image segmentation task. The segmentation procedure of the proposed approach is summarised as follows: Given a query image containing an area of interest that needs to be segmented, the network searches an existing database of multi-class medical images and retrieves a characteristics-matched support image. Then the network uses the support image for guidance to do the segmentation of the object(s) in the query image. This approach is designed to provide an automatic segmentation method under a situation where the medical image database has a collection of images in various symptoms (pathological morphology) while each specific morphology only happens in a small number of medical samples. For example, in the context of retinal OCT images, the database contains various morphological features of macular oedema at different severity conditions, each of which is observed in a small portion of the patients. In our work, we test our proposed method on the task of medical image segmentation, applied to data obtained from two imaging modalities: 1) macular oedema segmentation of OCT images, 2) skin cancer segmentation in dermoscopic images. With the OCT image dataset, we demonstrate that our method is able to handle different classes of macular oedemas in varying severity. With the skin cancer dataset, we present the generalisation of our network to different medical image contents.

In the rest of this chapter, we first illustrate the general problem set up of the one-shot segmentation. Next, we describe the details of the framework in Section 7.2, including the network structure (support branch and query branch), image matching strategy and network training method. In Section 7.3, we report the segmentation results for the two imaging modalities, followed by an analysis of the network's performance. At the end, we draw our conclusion in Section 7.5.

7.2 Proposed method

In this section, we described the overall working pipeline of the network. First, we give the definition of the one-shot learning problem in the context of medical image segmentation. Then we explain the details of the proposed network structure, which includes a query branch and a support branch. After that, we describe our embedding matching strategy that is applied to the trained network to find the most suitable support image in a database, thereby guiding the segmentation in the query branch. Lastly, we described the training strategy and the loss functions used for training.

7.2.1 Problem setup for one-shot segmentation

In the context of the one-shot segmentation, there are two types of image dataset involved: the query and support set. The query set comprises the new patient images which need to be segmented. The support set contains images and their associated ground-truth labels, which have been assessed by clinicians, and are used to guide the segmentation task on the query image. Accordingly, the segmentation network is functionally separated into two branches, and the query set and support set are the data inputs to the query branch and support branch, respectively.

In the general case of few-shot learning, the definition of the x-shot segmentation problem depends on the number of support images used and the number of ground-truth classes in each support image. For example, given that the support set contains s support images and each image has k distinct ground-truth classes, it is termed as k-way, s-shot segmentation. Specifically, our network only uses a single support image. The support image may contain a different number of ground-truth classes (three classes for the OCT dataset and a single class for the skin cancer dataset). Therefore, our network models a multi-way one-shot segmentation problem.

We define a group of data at one step of the training as an episode. Each episode contains a number of image-mask pairs $\{\mathcal{I}^i_{\text{train}}, \mathcal{Y}^i_{\text{train}}\}^N_{i=1}$, where there are N pairs of input image $\mathcal{I}_{\text{train}}$ and its corresponding label mask $\mathcal{Y}_{\text{train}}$. The training images $\mathcal{I}_{\text{train}}$ contains N pairs of support and query images and is defined as

$$\begin{aligned} \mathcal{I}_{\text{train}} &= \{\{(x_1, L_1, \overline{x}_1), \dots (x_N, L_N, \overline{x}_N)\}\,; x, \overline{x} \in \mathcal{D}_{\text{train}}\} \\ L &= \{l \in \{1, \dots, K\}; l \in \mathcal{L}_{train}\} \end{aligned} \tag{7.1}$$

where x is the support image, $\overline{x}$ is query image, and L is the ground-truth class label group associated with x and $\overline{x}$. The label group L contains labels l of K classes, which are sampled from the training classes group $\mathcal{L}_{\text{train}}$. After the label sampling, the support and query images are randomly sampled from the training set $\mathcal{D}_{\text{train}}$ and each support and query image contains objects of the sampled label L . The ground-truth mask $\mathcal{Y}_{\text{train}}$ is defined as

$$\mathcal{Y} = (y_1, \dots, y_N)\,, \quad y_i = \{(p_j, l_j) : l \in L_{\overline{x}_i}\} \tag{7.2}$$

where N is the number of masks. Each query image $\overline{x}$ has a corresponding mask y, in which the pixel value p_j corresponds to the class l_j. The l_j belongs to one of $L_{\overline{x}}$, namely the objects classes in the query image. The aim is to train a model, $\mathcal{F}(\,\cdot\,)$, with $\mathcal{D}_{\text{train}}$ such that the segmentation mask $\mathcal{Y}_{\text{train}}$ of query image $\overline{x}$ is predicted.

7.2.2 Network architecture

In accordance with the definition of one-shot segmentation, the proposed network architecture consists of two CNN encoders which act as the support and query branches. Figure 7.1 shows a schematic of the network structure that we use in the training and testing stage. To provide an overview, we first outline the functionality of these two branches as follows and explain the implementation details of each branch in Section 7.2.3 and 7.2.4, followed by a further description of the embedding matching strategies.

The support branch, taking a support image and its ground-truth mask as inputs, is set to learn an encoded representation of the labelled object. This representation is obtained by the support branch's encoder during the training process, where it extracts the features of the labelled object and encodes the features into an embedding vector. The extracted feature of each labelled object makes its corresponding embedding vector unique in a high-dimensional metric space, where the embeddings of objects of the same class are closer than the embeddings for different classes. Therefore the embedding vectors of different target objects or the image background are distinguishable in the metric space. At the model testing stage, the learned embedding vectors of database's images are used to guide the query branch to predict the mask of the same object from a query image.

The query branch takes a query image as input and outputs the mask prediction for the object of interest. To this end, the query branch is trained to extract the texture features of the query image content and predict the region of pixels where its feature embedding matches the support embedding vector.

The connection between two branches is an attention unit. It is used to generate an attention mask by using both the support and query embedding vectors. The attention mask contains the information of the similarity information between the query image content and the target objects in the support image. This attention mask is used by the query branch to distinguish the region of interest from the image background.

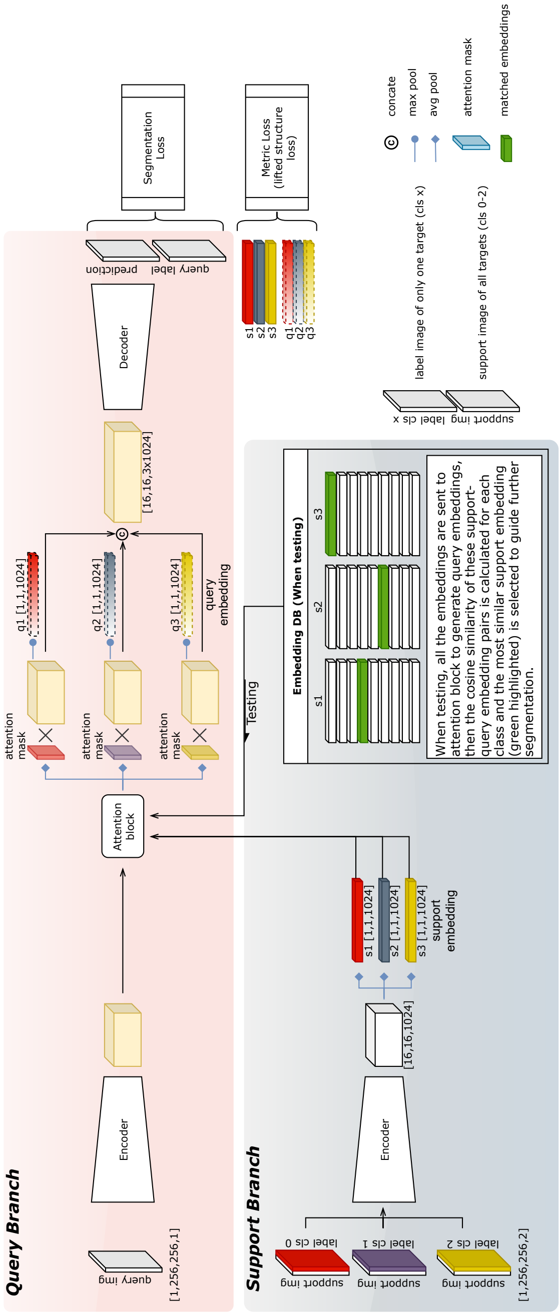

Query Branch
query img
[1,256,256,1]
Encoder
Attention block
attention mask
q1 [1,1,1024]
q2 [1,1,1024]
q3 [1,1,1024]
query embedding
[16,16,3x1024]
Decoder
prediction
query label
Segmentation Loss
Metric Loss (lifted structure loss)
s1
s2
s3
q1
q2
q3
Testing
Support Branch
support img
label cls 0
label cls 1
label cls 2
[1,256,256,2]
[16,16,1024]
s1 [1,1,1024]
s2 [1,1,1024]
s3 [1,1,1024]
support embedding
Embedding DB (When testing)
When testing, all the embeddings are sent to attention block to generate query embeddings, then the cosine similarity of these support-query embedding pairs is calculated for each class and the most similar support embedding (green highlighted) is selected to guide further segmentation.
label cls x
support img
label image of only one target (cls x)
support image of all targets (cls 0-2)
concate
max pool
avg pool
attention mask
matched embeddings

Figure 7.1: The overall structure of the SegMetaNet network. The encoder of the support branch encodes both the input image and label mask as embedding vectors. The encoder of the query branch encode the query image and use support embedding vectors to predicted the label mask of the query image by the query decoder. When doing testing, we can use the pre-generated support embeddings in the database.

7.2.3 Support branch

We start the description of the support branch with its input and output. The support branch requires the input data to be in an image-mask-concatenated format. Given a support image x, its ground-truth mask M contains K number of labels indicating different target regions. The mask M is first converted to a set of binary masks m_k such that each of them only has pixel labels of a single class other than the background. By doing this, it forms K pairs of the support image and ground-truth mask. Then, they are sent to the support encoder, which generates a support embedding vector for each class separately. In addition, it also generates an embedding vector for the background. As a result, the outputs of the support branch are a set of embedding vectors, denoted as $(v_1, \ldots, v_{K+1}) \in \mathcal{V}$, for K classes and one background.

In our work, the input data format of our network should be practical for different medical image types. The network was designed to be able to take both grayscale images (CT scan) and RGB images (fundus photography). To accommodate the two different image types, the support branch applied a universal four-channel input. The first three inputs correspond to three RGB image channels (for grayscale images, we replicated the single-channel by three). The fourth channel is for the label mask. It is worth noting that another possible approach to present the image content is to filter out the background content from the input image by using the ground-truth label and only keep the object region. In principle, this approach can enable the network to learn a better embedding vector of the target object without the presence of the background content. However, in practice, this background suppression operation is not suitable for medical image data. This is because the biomedical objects imaged by many imaging modalities have homogeneous texture inside the object. The structural information in the global context is more important than the texture feature to differentiating the target object from other image content. This typically happens for the tomography images, which only present image content in grayscale. For instance, macular oedemas of different types in OCT images have similar texture appearance (like black holes), but they are located at different retinal layers. Therefore, the spatial context of the background content of the image is necessary for medical imaging applications.

The support branch structure is formed by four encoding blocks and an embedding block, as seen in Figure 7.2. The four encoders are used to extract the

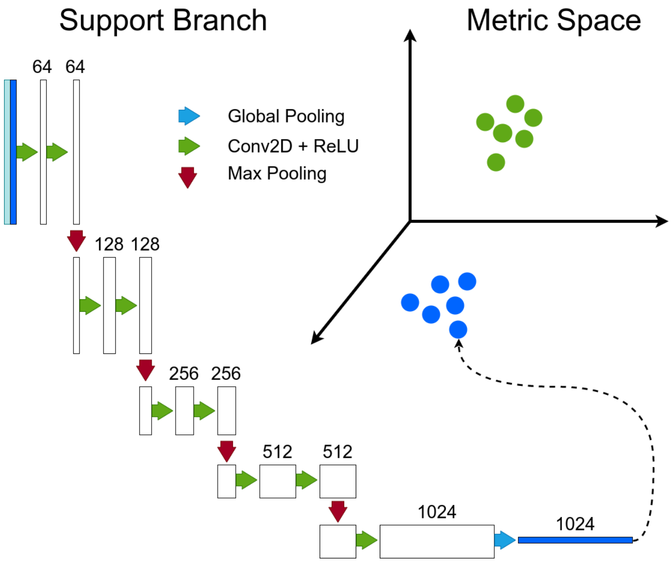


FIGURE 7.2: The support branch of the MetaSegNet. This branch takes the support images and the ground-truth label as inputs and outputs the corresponding embeddings. The embeddings in a dimension of 1024 are clustered in the metric space. The right side of the figure shows an illustration of the metric space having the target embedding in blue and background embeddings in green.

features at different spatial scales. A single encoder block consists of two convolutional layers, each of which is followed by a parametric ReLU activation and a batch normalisation layer. We set all convolutional layers with a kernel size of 3×3 and stride of 1. The reason for using small kernel and step consistently is because it can capture features of small target objects in medical images, such as early-stage cystoid oedemas. In terms of the kernels of the encoder, we set 64 kernels for the initial convolutional layers and doubled the number of kernels after each max-pooling layer to expand the dimensionality of the feature embedding. The output feature maps of the last support encoder have a channel-wise dimension of 512. After each encoder, we applied a max-pooling layer, with a pooling kernel of 2×2 and stride of 2, to reduce the spatial dimension of the feature maps. Consequently, the spatial dimension of the feature maps after four encoders are 16 times smaller than the original images. By doing so, the feature maps extract the image information that is important for the identification of the ground-truth object while eliminating the superfluous spatial information. In our network, it was found that the max-pooling made the model perform better than the average-pooling in terms of object boundary detection accuracy. This is because the average-pooling may include features located in the background when doing the averaging. As a result, the final embedding vector encodes features of both the target objects and background, making the guidance at attention block inaccurate.

At the end of the support branch, we used an embedding block to convert the feature maps to the final embedding vectors. The embedding block consists of a convolutional layer and a global average pooling layer. It averages each feature map to a single-pixel element, and all the element forms the embedding vector and entirely discards the object's spatial information. In our network, the embedding block converts the feature maps into an embedding vector with channel dimension 1024. The reason for this operation is that the target object may appear at different locations in the query and support image. By doing the global average pooling, the resulting embedding vector can be used for matching objects at multiple spatial locations of the query feature maps.

7.2.4 Query branch

The goal of the query branch is to predict the segmentation mask for the object(s) of interest in a given query image using the feature embeddings of a support image for guidance. To that end, the query branch first uses the encoder to generate a feature map of the query image. Then it uses an attention block to generate a group of query embeddings by masking target object(s) in the feature map using the support embeddings in the database. Then, we use an embedding matching strategy to find a best-matched support-query embedding pair by comparing the distance between the support and query embeddings. Finally, the decoder of the query branch uses the best-matched support embedding to guide the mask prediction on the query image.

The structure of the query branch is developed based on a fully convolutional auto-encoder structure. It is inspired by U-Net [120], which uses skip links between the encoder and decoder and performs well on image segmentation tasks. The network structure is illustrated in Figure 7.3. However, since the generic U-Net structure is a task-specific model, it is not suitable for one-shot segmentation. We keep the skip link operations and make several modifications to let the query branch work with the support branch.

First, we make the encoder structure the same as that of the support branch, except for the input layer since it has three input channels that do not include a

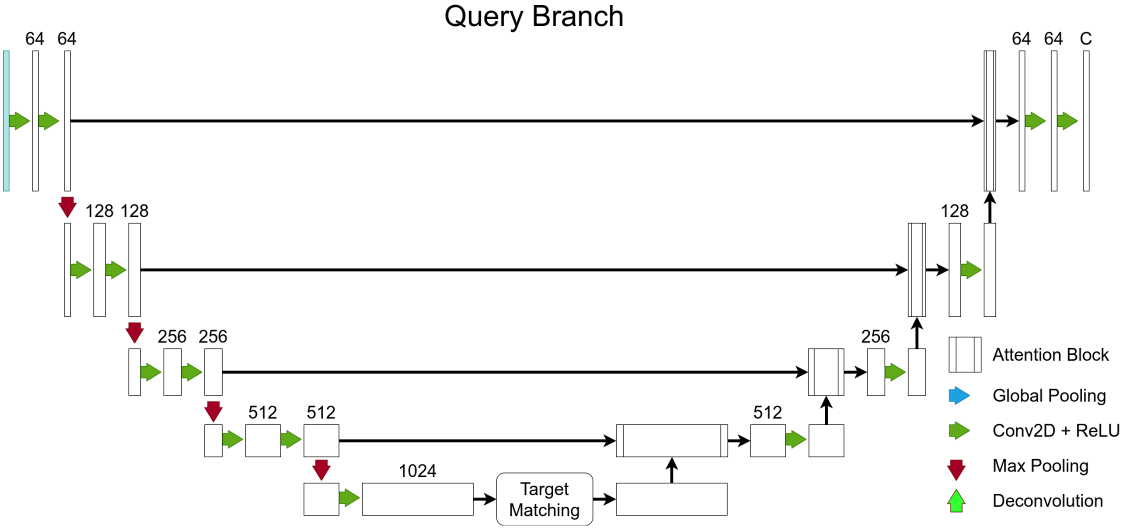


FIGURE 7.3: The query branch of the MetaSegNet. This branch is based on a U-net structure. The input is a query image and the output is segmentation masks of C number of classes. In addition, we added a target matching block and attention blocks. They are applied to better active the regions of target objects by leveraging the attention guide of the support embedding.

ground-truth channel. The encoders of the query and support branch share weights at each layer, forming a symmetric structure. We found that sharing weights is important to the embedding matching process because the shared-weight kernels allow the support and query branch generate similar embeddings from the features of the same object, which then can be close to each other in the metric space.

Second, we add an attention unit before the decoder blocks at different scales. The original U-Net had skip links that directly concatenate the encoded feature maps and input them to the decoder block with the same spatial dimension, i.e. at the same scale. Conversely, in our network, the encoded feature maps are firstly processed by an attention unit using the feature maps generated by the decoder block at a lower level, and then concatenated and passed to the current level decoder block. The attention unit uses the lower-scale feature maps to suppress the unrelated background content in the current encoded level feature map such that only the target regions are activated by an attention mask. By doing so, it encourages the network to focus on the features which are highly correlated to the support embedding.

The last modification is applied to the bottom encoder of the query branch. We add a feature matching block that connects the query branch with the support branch. The structure is shown in Figure 7.4. In the matching block, the support

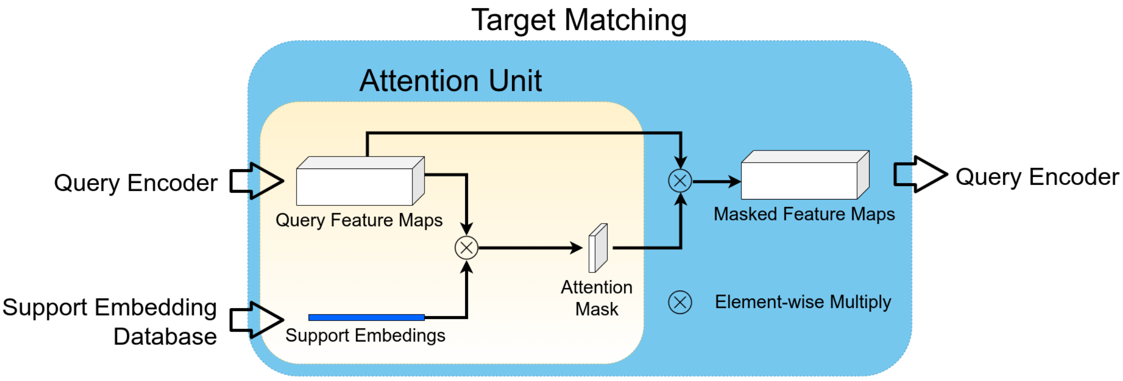


FIGURE 7.4: The target matching mechanism with the attention unit inside. The inputs of the target matching block are query feature maps and support embeddings. The support embedding correlates with the feature map and generates an attention mask of the high correlation regions. This mask is then used to weight the feature map such that the features of the target object(s) will be kept in the decoding process.

embedding is multiplied with the query feature maps at each pixel location to generate an attention mask. For each pixel of the attention mask, the query feature vectors which are highly correlated with the support embeddings result in a big attention response, i.e. a high scalar weight in the attention mask. The masks are multiplied with the query feature maps, resulting in weighted feature maps. Then these feature maps are concatenated and passed to the decoder.

7.2.5 Embedding matching strategy

The embedding matching strategy in our CBMIR system aims to find the best-matched image from a database such that its embedding can guide the segmentation in a given query image. To implement this matching process, we first build an embedding database for all support images using our trained model. Each support image was supplied as an input to only the support branches of the network to generated its embeddings. These embeddings are then stored in a database, which we called support embeddings database, shown as embedding DB in Figure 7.1.

When doing matching at the inference stage, we used the support embeddings from a pre-generated database. Since the support embeddings are directly retrieved from the database, rather than the support image, we leave the support branch untouched. First, the network takes a query image as input to the query branch to generate its corresponding query feature maps. Then, the support embeddings are inserted iteratively into the merging block to generate the query embeddings. Having the query embeddings, we use the cosine similarity as our matching metric to search the highest-score support-query embedding pair. It is worth noting that our network is designed for up to three-way one-shot segmentation. Therefore, the query image can contain objects belonging to up-to three classes. Our matching strategy finds the matched support embeddings for each class individually throughout the database. This indicates that the matched support embeddings may originate from different images.

7.2.6 Loss function

To train the network in an end-to-end manner, we formed the loss function with two objectives, the embedding matching accuracy and the segmentation accuracy. The total loss function is formulated as:

$$\mathcal{L}(\mathcal{Y}, x, \overline{x}, W) = L_{\text{emd}} + L_{\text{seg}} + \alpha L_{\text{reg}} \tag{7.3}$$

where L_{emb} is embedding loss, L_{seg} is segmentation loss and L_{reg} is the network regularisation loss weighted by the factor α.

The objective of the embedding loss is to build a metric space where the distance between embeddings of the same class is small while the distance of embeddings from different classes is large. We use a lifted structure loss function [124] to learn the embedding between pairs of examples in the training batch.

The objective of the segmentation loss is to increase the overlap between the ground-truth object mask and the predicted mask. To this end, we compute the multi-class cross-entropy loss between the output of the query branch and the ground-truth mask. In addition, we compute the dice loss for each class and combined this with the entropy loss. The final segmentation loss is then formed as

$$L_{\text{seg}} = -\gamma \sum_{k=1}^{n} \sum_{i=1}^{C} t_{ki} \log\left(y_{ki}\right) + \beta L_{\text{dice}} \tag{7.4}$$

where the first term is the cross-entropy loss and the second term is the dice loss. γ and β are balance weights to control the influence of each term.

7.2.7 Training strategy

The training data for each training step is grouped as an episode. This is illustrated in Figure 7.5. Specifically, the network is trained with the training data D_{train} that contains the query and support images, which will be described in Section 7.3.1. To build an episode during the training procedure, we first randomly sampled the classes of the D_{train} and select a small subset of classes. Then we randomly select a query image that contains objects belonging to the sampled classes. Lastly, we randomly select one support image for each of the sampled classes. This procedure makes sure that the objects of each class in the query image can be guided by a support image.

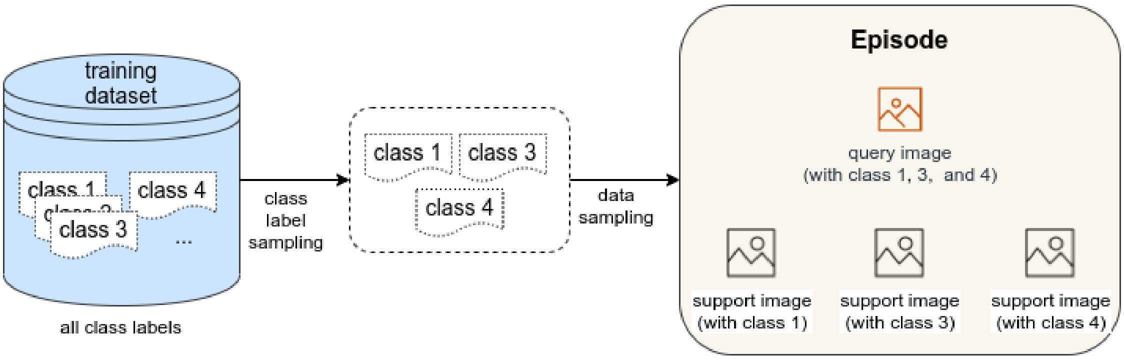


FIGURE 7.5: The diagram of forming an training episode. We first randomly sample a subset of all classes in the dataset. Then we randomly select a query image and a set of support images that contain the object(s) of those chosen classes.

We implemented two data pre-processing procedures: the training image augmentation and the ground-truth mask weighting. The training images were augmented by applying brightness and contrast changes, scaling of intensity, and horizontal and vertical flipping, which simulate images generated under different imaging conditions or by different imaging systems. The ground-truth mask weighting was done when calculating loss. Specifically, the ground-truth mask was multiplied by a weights map that applies a larger weight to the object boundaries, forcing the network to put more focus on the boundary recall.

During the training, we encourage the network to use different combinations of support embeddings to guide the segmentation. To do so, we switch support embeddings randomly such that the combined support embeddings are from different images. For a query image, having objects in three classes, we need three support embeddings (one for each class). In a single episode, these three support embeddings are extracted from the same support image. We randomly select three embeddings from different episodes such that they are extracted from different support images. By doing so, we can increase the number of training samples, and the network can be trained using different embedding combinations rather than the combinations only from a single image.

The trainable weights of the network are optimised using the Adam Optimisation algorithm. We set the learning rate as $5e^{-5}$ with an exponential decay rate of 0.75. The batch size is 4. The network was trained using the Nvidia GTX 1080Ti On Ubuntu 16.04.

7.3 Test results

We evaluated the segmentation performance of our trained model on two test datasets: 1) a retinal OCT b-scan dataset and 2) a skin cancer image dataset. These two datasets contain different targets, thus are suitable to evaluate the generalisation of our model. The retinal OCT dataset consists of grayscale images of three object classes, including intraretinal oedema, subretinal oedema, and pigment detachment. The skin cancer dataset consists of RGB images of one object class, melanoma. In these two datasets, the same disease usually shows various morphological features between patients. In the remainder of this section, we first introduce the datasets and then present the segmentation results. Then we discuss the key components of the network with the visualisation of the metric space and attention mask.

7.3.1 Test datasets

The OCT retinal dataset was collected from real patients specifically for this project by clinicians from the ophthalmology department at Kent & Canterbury Hospital. It contains the 1800 images (BRVO and DMO pathology, each has 900 images) that have three types of oedema: the intraretinal fluid, the subretinal fluid, and the pigment epithelial detachment. In OCT images, these three types of lesions are often represented as dark cavities, which are similar to each other in texture but can be differentiated according to their locations within the retina. The intraretinal fluids are located near the surface of the retina. They are formed by a large region of fluid-filled tissue that shows many separate low-reflectivity cystoid blocks in the OCT image. The subretinal fluids are located underneath the intraretinal fluids. They are usually between the outer retinal layers and the retinal pigment epithelium. Their appearance is often presented as a clear liquid swell. The pigment epithelial detachment is located at the bottom of the retinal tissue, near the Bruch membrane. It is often shown as a fluid build-up, which makes the retinal pigment epithelium detached from the Bruch membrane. The Figures 7.6 shows an example image that contains all these three types of oedema.

[REDACTED]

FIGURE 7.6: An example of a retinal OCT B-scan image showing all three types of oedemas in the retinal tissue.

To make our network more robust to the practical clinical setting, we trained our model with OCT B-scan images acquired from three mainstream systems with different signal-to-noise ratios. These systems were: Cirrus, Spectralis, and Topcon. These three devices generated the images with their default settings and therefore produced retinal images with different noise levels. Figure 7.7 shows healthy retinal tissue captured by these three devices. It can be seen that the Topcon image (bottom) quality is poorer than the other two images as several parts of the low-contrast retinal layers are no longer recognisable due to the noise.

[REDACTED]

FIGURE 7.7: The appearance of normal retinal tissue, captured by three different OCT scanning devices. The scale bar denotes 100 μm. Top: Cirrus (resolution: 6 μm); middle: Spectralis (resolution: 3.9 μm); bottom: Topcon (resolution: 6 μm).

In addition to the OCT dataset, we used a skin cancer dataset to evaluate the model performance. The dataset consists of 900 training images of melanoma captured using dermoscopy [125]. The melanoma appears as pigmented lesions on the skin surface and is often imaged when performing a clinical visual inspection. Compared with standard photography, the dermoscopy can present a clearer visualisation of the melanoma by eliminating the light reflection from the skin's surface, which can be seen in Figure 7.8. In the test set, a collection of 350 melanoma images were collected with various melanoma appearances. Each image contains a single melanoma spot, which is surrounded by normal skin. The specialist clinician created the ground-truth masks using either a semi-automated process or a manual process.

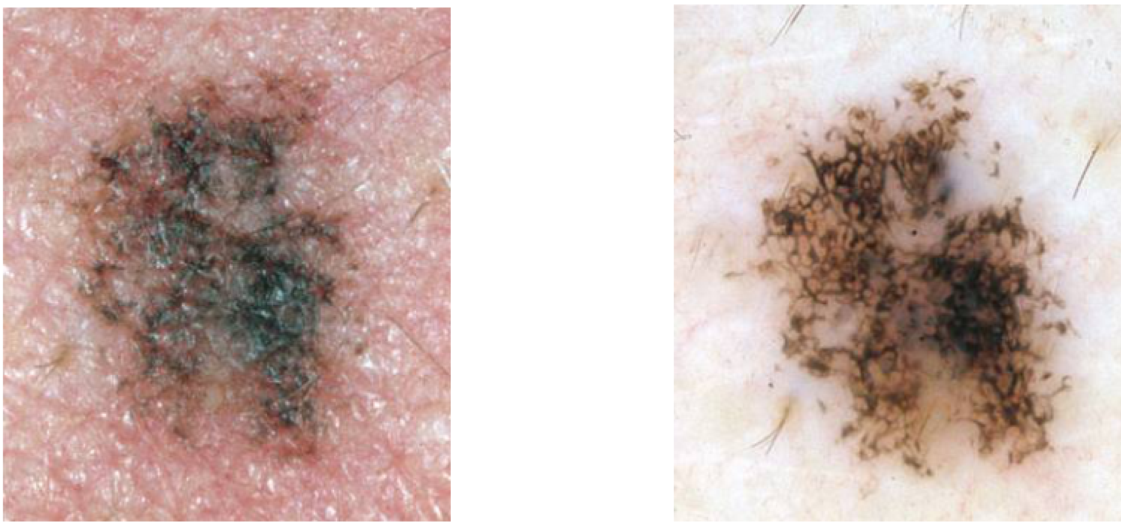


FIGURE 7.8: The images of the same melanoma tissue, captured by a conventional camera (left) and a dermoscopy (right).

7.3.2 Evaluation results

In our evaluation stage, the test results of the oedema segmentation task indicate the trained model has a good performance in terms of segmentation accuracy. Quantitatively, our model achieved a high Dice coefficient of 83% (100% indicates an exact same segmentation as the ground-truth mask). The segmentation results of the retinal OCT images are presented in Figure 7.9. The first column presents the original images, containing oedema at different severity levels. The predicted oedema regions are shown in the second column using grayscale masks. For each image, we used different grayscale values to differentiate the oedema types. The third column presents the ground-truth masks, which are used for comparison. From the figures, it is seen that the model predicted the areas of all three types of oedema accurately, which are comparable with the ground truth.

It is worth noting that there are two important image features that are difficult to segment but play an important role in classification. We show that our model is capable of achieving this.

[REDACTED]

Figure 7.9: The segmentation results of retinal OCT images. First row: all three types of oedema; second row: only the intraretinal fluid; third row: the intraretinal and subretinal oedemas; last row: subretinal and pigment epithelial detachment.

The first of these is the boundary between subretinal oedema, and pigment detachment is clearly segmented. For the subretinal fluid and pigment epithelial detachment, the segmentation between their boundary is important since these two target regions are often located very close and often caused incorrect recognition. From the results, it is seen that the network segmented their boundaries clearly, and no overlapping between two regions happens.

Second, the small intraretinal oedemas, which are distributed intensively at the upper part of the retina, are also segmented. This segmentation accuracy can be attributed to the co-action of the attention units and weighted ground-truth mask. In the query branch, the attention unit created a binary mask for the target region at different scales. The basic features, represented in the large feature maps, were used by the attention unit to achieve the detection of thin boundaries between different oedemas. Meanwhile, the large weights of the ground-truth mask gave a greater penalty at the target object boundaries when calculating the loss function. Because of the high loss penalty, the pixels of the object's perimeter tended to be predicted as background.

At the same time, it is noticed that the model had a segmentation bias on different appearances of the intraretinal fluids. Specifically, it is observed that the segmentation of the large fluid section is consistent, while the segmentation of small spots varies in accuracy. This is due to the feature complexity. Because the large blocks have a clear boundary and homogeneous fluid inside, their appearance is less variable. This helps the network to predict a good mask that is consistent with the ground truth. In contrast, successful segmentation of the scattered spots is limited by speckle noise and spot density. One reason is that inconsistent labelling introduced confusion to the network training. When doing the labelling, the clinician sometimes does not differentiate between individual spots but instead labels a region of high spot density as a single region. The consequence of this is the region is labelled with thin membranes in between. As a result, the segmentation of the small spots was slightly inconsistent, and it is seen that the model didn't fully segment some small regions but can detect some other regions which are missed labelled in the ground-truth masks. Although our model failed to label spots in regions of high spot density, it was able to segment other regions that were not identified in the ground truth images, such as unlabelled thin fluid sections.

The test results of the melanoma segmentation task are shown in Figure 7.10. It presents the original images in the first column, followed by the predicted segmentation and the ground-truth mask in the second and third columns, respectively. In this dataset, our model yielded a very good Dice coefficient of 89%. From the results, it is seen that the regions of the melanoma lesion were predicted successfully and agree with the manual annotations. Furthermore, it is also seen from the boundary segmentation results that the model was trained to balance two different ground-truth labelling styles. It is seen that the melanoma lesion normally has an ambiguous boundary in the original images. Because of this vague

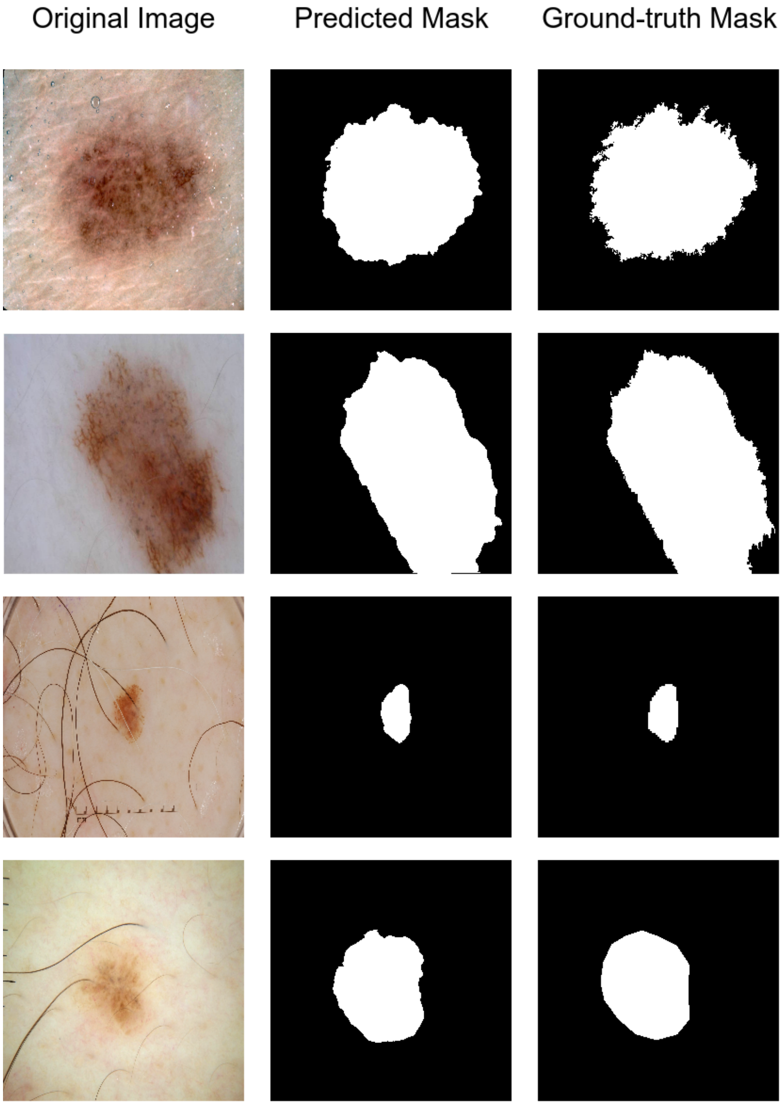


Figure 7.10: The segmentation results of melanoma lesion in the dermoscopy images. We show the segment results of melanoma in four different shapes and textures. There are two types of ground-truth masks: the masks in the top two rows are annotated using semi-automatic labelling tools; the masks in the bottom two rows are annotated using hand-drawing tools.

boundary position, the ground-truth masks generated in a different way result in different boundary styles. The ground truth masks with smooth boundaries (in the two bottom rows, third column) are generated by manual segmentation, while the masks with coarse boundaries (in the two top rows, third column) (burr edge) are generated using a semi-automatic segmentation tool. Since the model relies on the local texture difference, the predicted masks have region boundaries that balance manual and tool labelling style, i.e. the edges tend to be smoother and closer to the shape of the actual melanoma boundary.

7.4 Key components analysis

One essential task of our one-shot segmentation network is to utilise the features of the support image to provide segmentation guidance on the query image. When doing the object segmentation in the query images, the network provides good guidance for the region prediction based on two key facts: the support image matching and attention masking. For the support image matching, the network should retrieve a support image that is a good match with the query image in terms of target content. For the attention masking, the network should generate correct attention masks using the support image and use these to predict the target region in the query image. In the following subsection, we breakdown our approach into three aspects: embedding distribution, support image retrieval, and attention masks.

7.4.1 Embedding distribution

Good distribution of all support embeddings in their metric space is the foundation of accurate image retrieval. In this metric space, the support embeddings of the same class should be close to each other and embeddings of different object classes should be separated and have low cosine similarity. When the network does the segmentation, our embedding matching strategy uses the cosine similarity metric to find the best-matched embeddings.

To establish such a metric space for OCT images, we trained the network to encode oedema features such that the normalised embeddings of the same oedema class have small L2 distances to each other, while the embeddings of different oedema should have at least a larger pre-defined margin distance. Since the retinal OCT image is shown in grayscale, the training of the embedding representation was focused on the morphological and texture features. When the model was used for matching support images, the embeddings containing similar features should have larger cosine similarity and thus will be matched.

We used t-SNE to visualise the embeddings in the metric space. The t-SNE is a non-linear algorithm used to map the embedding vectors in high-dimensional space to a set of estimated representations of those points in a lower-dimensional space, such as 3D space in our case. Meanwhile, the low-dimensional points keep

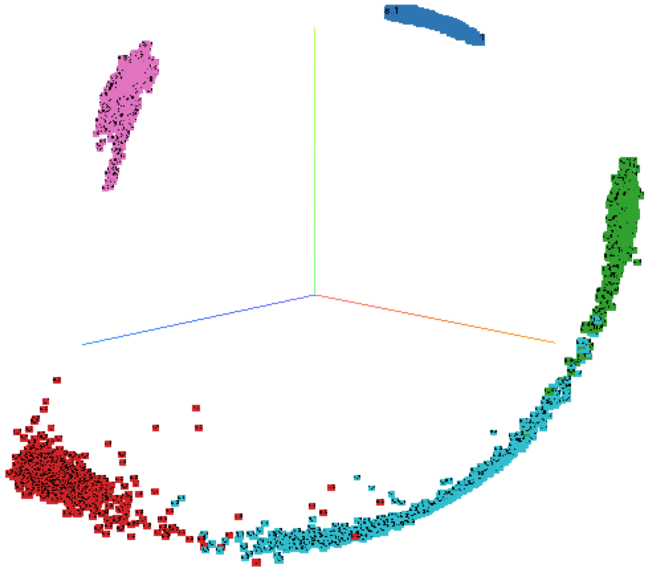


FIGURE 7.11: The embedding visualisation of all support OCT images. We applied the t-SNE to reduce the dimension of our embedding vectors from 1024d to 3d. As expected, these embeddings are categorised into five classes: non-existent disease, background retina tissue, intraretinal oedema, subretinal oedema, and pigment detachment. It is seen in the metric space that the clusters are separated from each other, although the embedding clusters of three disease are distributed closely due to their similar appearance.

a similar distribution as that in the original high-dimensional space. Therefore, it is often used for visualising the relations of the embeddings. We convert the original 1024 dimension embeddings into 3D space using t-SNE. By doing this, we see that the embeddings of retinal OCT images and the dermoscopy images are grouped according to their class in their own metric space, respectively. Figure 7.11 shows embeddings of the retinal OCT images of the support database, which was used for the embedding matching. In the metric space, the embeddings are represented as data points, and their class was labelled in different colours. It is seen that the embeddings generated by the support branch are distributed as five groups, and embeddings belonging to the same class are well clustered. These five classes include three oedema classes, one background class, and one non-exist class. The non-exist class denotes particular oedema that does not exist in the image, although other types of oedema might be present. For example, a support image, which has intraretinal and subretinal fluid but not pigment epithelial detachment, will generate a non-exist embedding for the pigment epithelial detachment class. The distribution of the embedding groups indicates that the features of different oedema types were learnt well by the model, and therefore an effective matching can be achieved. This is demonstrated in the next section. Figure 7.12 shows the embeddings of the dermoscopy images. In this case, we have two object classes: the lesion region and the background skin. It is seen that the embeddings of melanoma lesion are clustered, and they are separated from the embeddings group of the background skin.

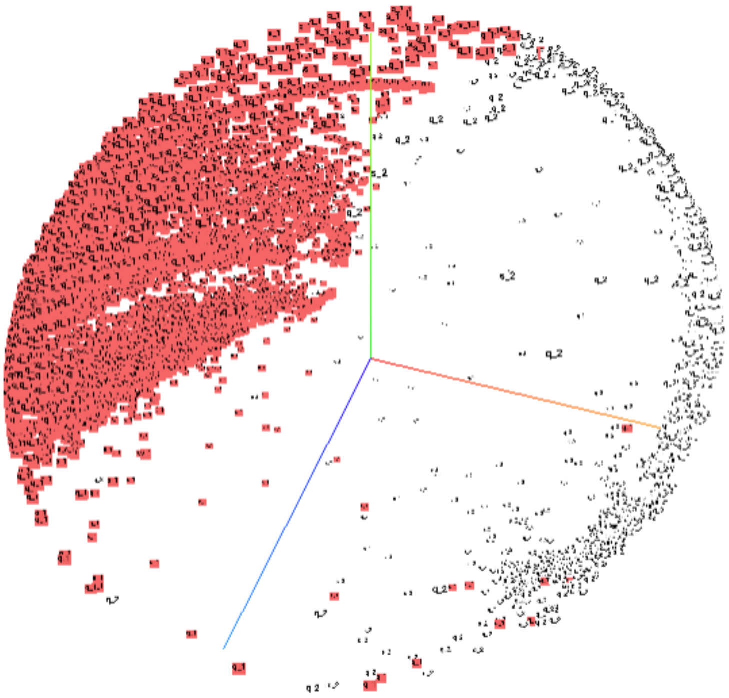


FIGURE 7.12: The embedding visualisation of the all support dermoscopy images. We applied the t-SNE to reduce the dimension of our embedding vectors from 1024d to 3d. We can see that the two classes' embedding clusters (the background and melanoma) are well separated in our built metric space.

7.4.2 Image retrieval

Based on the good embedding distribution in the metric space, our image retrieval module can accurately pick the matched embedding to provide guidance for segmenting the target object in the query image. At the same time, the support image of the matched embedding can also be retrieved from the database to give clinicians a reference example for diagnosis purposes. This enables the clinician to make a treatment plan based on the patient history or cases of other patients who have similar lesions.

[REDACTED]

Figure 7.13: The retrieved OCT images based on the query image content. It is seen that for each query image, the retrieved image has the same classes of oedema(s) that are similar in terms of the structure and disease severity level.

For the OCT images, our image retrieval module is expected to obtain a reference image that is similar to the query image in terms of oedema severity. The various oedema appearance indicates the different severity levels. Given a query image, the embedding matching aims to find the support image in which oedema has similar lesions. In Figure 7.13, we show the support images that were retrieved by using our embedding matching strategy. The query image is shown in the first column, followed by the retrieved support image. It is seen that our embedding matching strategy can successfully retrieve a support image that has similar lesion severity in the query image, i.e. the oedema shape and texture in the support images are similar to the oedema appearance in the query images.

For the dermoscopy images, our image retrieval module is expected to obtain a reference image such that the similar appearance of the melanoma lesion in the obtained image helps the clinician to diagnose the progression of cancer through different stages. The melanoma lesion has irregular shape, texture, and colour, which are distinct from the appearance of normal skin. Figure 7.14 shows the retrieved support image. It is seen that the melanoma lesion on the retrieved image contains similar features to the query image. This indicates that our embedding matching can work on both the morphological and the colour information.

7.4.3 Attention masks

The attention mask is the intermediate feature of the attention unit. It indicates the regions of query feature maps where the network places attention by using the matched support embedding. To implement this guidance, the attention unit takes the support embeddings and query feature maps as inputs. For each support embedding, it rates the correlation between each pixel location of the query feature maps and the support embeddings. Consequentially, the high-correlation pixels were activated by the ReLU activation layer, generating the foreground of the attention mask. The attention unit helps the network to activate the target region while suppressing the background. To better show the attention mechanism, we visualised the attention masks to illustrate the attention unit's functionality.

The attention mask for different classes may vary based on their appearance. Figure 7.15 shows the attention masks for four object classes in the retinal OCT image. In each row, the query image is shown in the first column, followed by its attention mask of the background class (the second column) and three oedema attention (third column). By comparing with the query image and the masks, it is seen that the attention mask of grayscale image places attention on the structural information. The background attention masks put greater attention on the healthy retinal tissue and vitreous which is distinct from the appearance of the oedemas. The mask of the oedema classes gives additional focus on the location of the lesions.

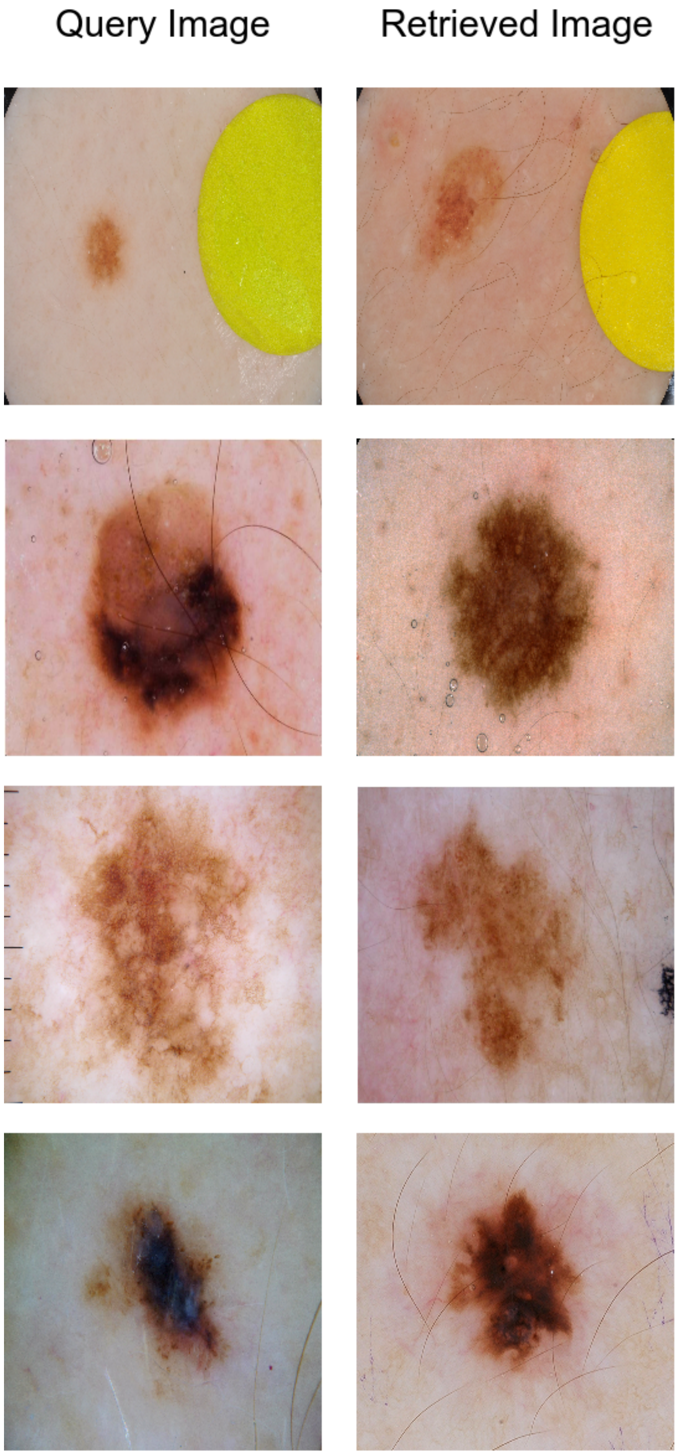


Figure 7.14: The melanoma objects of the retrieved images. The melanoma in the query image at each row is distinct in terms of colour, texture and shape. It is seen in four examples that the melanoma in the retrieved images has a high-similarity appearance to the query melanoma.

[REDACTED]

Figure 7.15: The attention masks generated for the OCT images. From left to right: raw OCT image, background attension mask, oedema attention mask.

For the dermoscopy image, the visualisation of the attention masks is more straightforward since the attention unit was trained to distinguish the foreground object from the background, i.e. a binary segmentation. Since the dermoscopy image contains colour information, the attention unit used both the colour and texture features to generate the masks. Figure 7.16 shows the attention for the background and the melanoma lesion. It is clear to see that the attention mask focuses on the lesion region, which has a distinct appearance compared with the surrounding healthy skin.

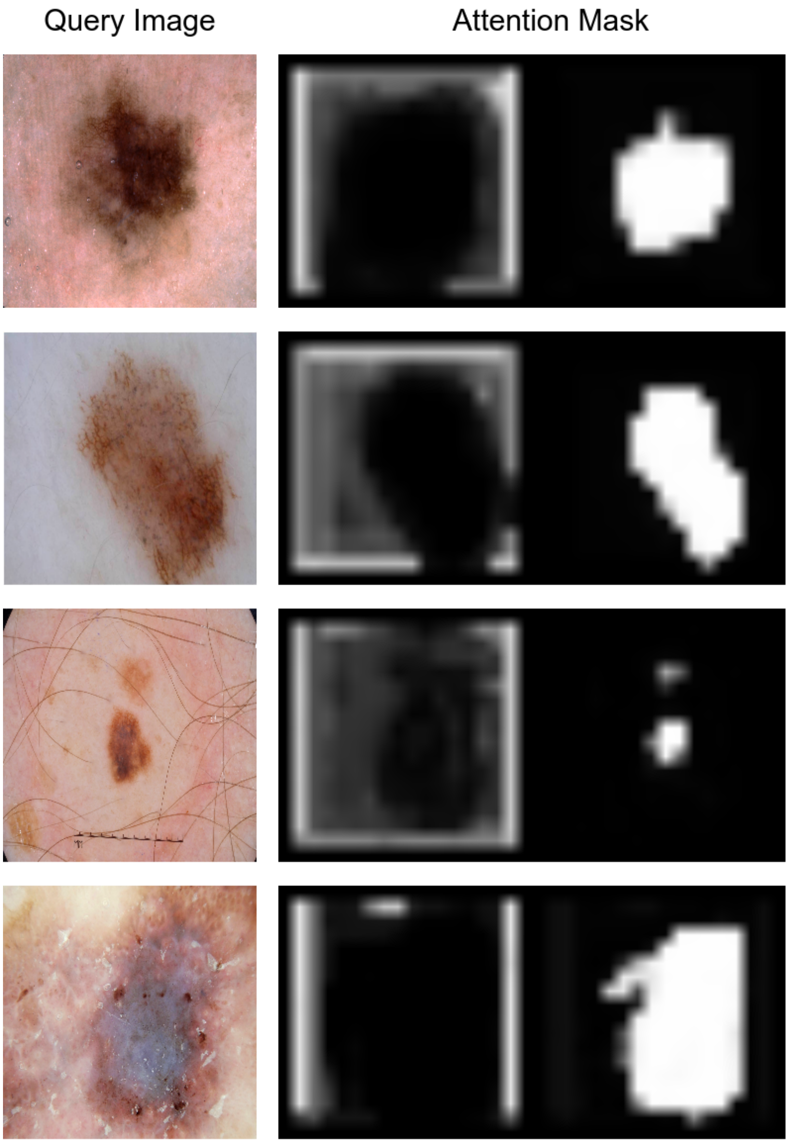


Figure 7.16: The attention masks of the dermoscopy images. There are two masks classes, i.e. the background (the second column) and melanoma (the second column). It is seen that the attention unit generated a mask where the support embedding has a high correlation with the melanoma region.

7.5 Conclusion

In this chapter, we presented an approach to medical image segmentation that used an image retrieval-based neural network model. Given a query image that we wish to segment, the model searched through a medical image database and selected the most suitable support image. Then the network used the embedding of this support image, pre-generated using our model, to segment the target object in the query image.

Our method focuses on a common problem associated with object-oriented segmentation approaches in medical image scenarios, whereby the number of target objects in images is not always large enough to satisfy the network training. Therefore, we introduced the task-oriented segmentation framework that combines *one-shot segmentation* and *CBMIR* techniques and our proposed novel neural network architecture to solve the problem.

We attributed the good performance of our model to three key design properties of the network structure and the training strategy. First, the input data format of the support branch was designed to learn the representation of objects in the grayscale images of our dataset. For typical images of this type, the content of target objects with no background encourages the network to learn object features without interference from unrelated contents. However, in the case of medical images, especially for the grayscale images, lesion appearance depends both on its own features and the surrounding healthy tissue. Therefore, keeping the background information helps the network to learn more reliable features.

Second, the embedding switch strategy enhanced the model generalisation during the training. The network used embeddings of the sampled classes from different images and concatenated them to form a long feature embedding. This concatenated feature embedding was subsequently used to segment the objects of the sampled classes in the same image (query image). This forced the network to learn an embedding representation that is image-independent and therefore increases the generality of our proposed method.

Lastly, the good performance of the CBMIR-based support image guidance is based on the combination of embedding matching strategy and attention masking, which are the core modules, to implement. When using the trained model, the embedding matching module found the most similar support image by matching the query embeddings with the support embeddings in the database. Once the support embedding was obtained, the attention masking used the support embedding to recognise the target objects in the query image and predict their segmentation masks.

From the test experiments, we saw that the network yielded good accuracy on the retinal OCT images and dermoscopy images segmentation tasks. Before this approach was proposed, there was no similar method presented in the literate to implement the segmentation works in a combination of image retrieval. Our method demonstrated the feasibility of combining the advantages of CBMIR and few-shot segmentation. Our proposed segmentation method is used with image retrieval from an existing database. Therefore, its application is not restricted to the images demonstrated in the chapter but can be extended to other medical databases. In real clinical situations, such databases can be built using different medical imaging modalities, such as CT, MRI, and ultrasound imaging.

Chapter 8

Conclusion

8.1 Summary

In this thesis, we presented our recent research that focused on two subjects. The first subject is compressive signal sampling and reconstruction. We systematically investigated compressive sensing-based algorithms that recovered a) the compressively sampled time-stretched OCT spectra and b) images acquired by a single-pixel camera architecture. Furthermore, we developed a deep-learning-based algorithm that performed compressive sampling and reconstruction of single-pixel camera images. The second subject is medical image analysis. We developed three algorithms for the automatic segmentation of retinal structures or lesions in the OCT image. A summary for each subject is presented in the remainder of this section.

The work for compressive signal sampling and reconstruction is present as the first part of the thesis. We developed a flexible single-pixel camera to carry out our image reconstruction task. We also evaluated and optimised the hardware setup and control software for a time-stretch OCT system to produce the signal sampling and reconstruction. We evaluated the performance of the reconstruction algorithms for these two systems. Motivated by an opportunity to further improve image quality and measurement ratio, we developed a state-of-the-art image reconstruction model using a convolutional neural network, specially optimised for use with single-pixel camera architectures.

In Chapter 2, we first introduced the principles of image compressive sensing. We described the development details of single-pixel camera hardware and reconstruction software, followed by an evaluation of our system.

We demonstrated a systematic evaluation of the signal reconstruction algorithms with the aim of optimising the time-stretch OCT system in terms of reconstruction accuracy and reconstruction time. Specifically, the accuracy was evaluated with respect to recovering spectral frequency components and accurate component intensity. The time efficiency was evaluated based on signal reconstruction at different length data streams.

As an alternative to compressive sampling image reconstruction, we presented a deep-learning-based approach in Chapter 3. This proposed network was designed to be a practical solution for single-pixel camera hardware. To this end, we developed the network to use both binary and float-point weights. The binary weights were learnt during the training stage and deployed as the sampling patterns on the single-pixel camera to take the measurement. The floating-point weights were trained to reconstruct the images from the compressively sampled measurements. Moreover, we presented our LSHR network scheme (low-resolution sampling, high-resolution reconstruction) to recover a better-quality image. With this scheme, the network used low-resolution patterns to improve the sampling efficiency and used recursive residual blocks to reconstruct the image with enhanced image quality. By using this scheme, the network model achieved a better image quality while reducing the model size. Lastly, we presented the reconstruction results and demonstrated the successful integration of the trained model with a single-pixel camera architecture.

For the medical image analysis part of this thesis, we presented three frameworks for retinal OCT image analysis. The aim of the first framework was to segment retinal layer boundaries, and the other two were for segmenting macular oedemas. We evaluated the framework on the real OCT image datasets. Comparing the segmentation results with methods in previous work, showed that our frameworks achieved the highest segmentation accuracy.

This part of the thesis began with an introduction and literature review on retinal OCT imaging in Chapter 4. This included a description of imaging modality, and a presentation of the human retinal structure, and the common retinal lesions observed in OCT images. We also reviewed the existing mainstream OCT image segmentation techniques for retinal layers and macular oedemas separately. The key components of their framework and their segmentation results were presented, followed by a discussion on the limitations of those techniques.

We developed a framework for segmenting the retinal layers from OCT volume data, described in Chapter 5. It consists of a pipeline that combines several algorithms in a particular sequence, to segment boundaries in cross-section retinal images (B-scans). These key algorithms include adaptive-curve, superpixel clustering algorithm, active-contour and Kalman filter prediction algorithms. Specifically, the adaptive-curve algorithm was used to define the retinal region. The superpixel clustering and active contour algorithm were used to segment the internal boundaries within the retinal region. The Kalman filter function predicted the boundary location in the adjacent OCT B-scans. By using the framework, the retinal boundary in 3D data volume was segmented and visualised, enabling the calculation of the thickness map at the regional region.

In Chapter 6, we proposed a transfer learning framework that segmented macular oedema from OCT B-scans using a convolutional neural network and conditional random fields. We first introduced the target region that consisted of diabetic macular oedema in two conditions: retinal thickening and cystoid oedemas. Then, we illustrated the application of the fully convolutional neural network and conditional random fields to the problem of oedema boundary detection. Specifically, we applied a transfer learning strategy to fine-tune a network model to adapt the pre-trained convolutional kernels trained on general-object images, to make it applicable to retinal images. In the last part of the chapter, the framework's segmentation results are presented and discussed, which indicates the fine-tuned model achieved a better accuracy compared with previous related work.

A meta learning-based network was then presented that is suitable for small-scale datasets in Chapter 7. Based on the concept of meta-learning, the network was designed to be task-oriented. In our work, the task was segmentation. Using this approach, the network captures the features of the target object from a reference image and automatically segments the corresponding object in the query image, in which the target segmentation needs to be done. This idea is different from our previous object-oriented network described in Chapter 6 that only learns the appearance of oedema from the training dataset, which is usually a large-scale dataset to ensure the model generalisation. The meta-learning principle for the segmentation task was described first, followed by the proposed network architecture and the training strategy. Then we presented the results and discussed the network attention scheme in the rest of that chapter.

8.2 Achievements

This thesis covered work that put the focus on the application of OCT imaging and made several contributions to two research fields: image reconstruction from compressed sensing and image analysis from the medical imaging device.

Efficiency is often seen as a crucial factor when evaluating an image reconstruction approach. We have noticed several bottlenecks of the existing approaches in terms of efficiency. Therefore, our work aims at improving imaging efficiency by applying compressed sensing and deep learning. In our fundamental work, we developed general sampling hardware and control software for the single-pixel camera. This single-pixel camera prototype can be used for easy layout deployment and measurements sampling, and we made this open-source so others can benefit from our implementation. Based on this work, we made an important contribution by proposing a novel image reconstruction algorithm (LSHR) based on a neural network. This network utilises sets of optimised binary sampling patterns and they are ideally suited to be used by the single-pixel camera imaging hardware architectures. Our network improved the imaging efficiency by using the LSHR sampling scheme and the one-off feedforward reconstruction computation. As a result, it yielded better image reconstruction quality at highly compressed sensing measurements while using less reconstruction time. For the OCT system, we evaluated a time-stretch OCT hardware and our work focused on using compressed sensing to improve the sampling efficiency, while solving the bottleneck of the system's sampling speed. The evaluation of the signal reconstruction algorithms provided a systematic reference on how to choose the best algorithm for the reconstruction of the time-stretched signal. We determined that the L_1 Magic package was the optimal approach for our time-stretch OCT in terms of reconstruction time and data scalability.

In the scope of the medical image analysis, the detection of biomedical tissue and lesion is often treated as a fundamental work together with disease diagnosis. We proposed three approaches aiming at object detection in retinal OCT image analysis. Starting from rule-based algorithms, we developed a novel framework for retinal OCT boundary detection. The framework achieved full retinal layer segmentation and it improves the visualisation of en-face retinal images and, therefore, enables more accurate clinical assessments. However, due to the variety of biomedical features limited access to large datasets, the rule-based algorithms might not satisfy all the conditions. We, therefore, stepped further to developing a deep neural network-based model. We proposed two approaches, based on transfer learning and meta-learning, to increase the model performance and generalisation, without the need for ad-hoc rules and manual intervention. Our model is trained with the task of segmenting retinal oedema in the OCT images. Both networks performed an excellent segmentation on retinal images representing different stages of macular oedema. The transfer learning-based model achieved an overall dice coefficient of 0.60 ± 0.26; the meta-learning-based model achieved 0.83 on the OCT dataset and 0.89 on an additional dermoscopy dataset. Both are better than the performance quoted in previous work.

The work that we detailed in this thesis demonstrated several achievements to solve the existing issues and improve the performance. It is also worth mentioning that our deep neural network-based approach has shown a more advanced capability in terms of the final performance. It should be emphasised that, when we design the network, the data distribution of the samples in the training dataset should be as close as possible to the real-world distribution. Therefore, in this thesis, we think of the model generalisation in two different aspects. For the LSHR network, we selected a very large-scale dataset to train the model or fine-tuning the model which has been trained in a large-scale dataset; for the meta-learning-based model, we aim to distil the general knowledge from the limited dataset, rather than taking care of the distribution of a large dataset. Although both approaches have presented good performance, we should always rethink the advantage of them when dealing with new problems.

8.3 Future work

Beyond the achievements of the developed approaches, there are several improvements that can be further explored. For the single-pixel camera, one important factor that limits the measurement accuracy is the analogue-to-digital conversion resolution. This can be solved by adopting a higher bandwidth conversion range, resulting in more accurate image reconstruction.

For the medical image processing work, the retinal layer segmentation algorithms also have several aspects worth further exploration. In the future, we can extend our framework to wider ophthalmology assessment, including deformed retinal layer segmentation caused by age-related macular degeneration, epiretinal membrane, macular hole and retinal detachment. In addition, our approach can be extended to 3D segmentation by using voxel clustering algorithms and recurrent neural networks in post-processing to enhance inference efficiency. This improvement can be more practical for quantitative analysis and clinical diagnosis. Furthermore, the application of 3D convolutional kernels for volume segmentation is more efficient for volume scanning modalities such as OCT volume scanning and MRI scanning. Our meta-learning feature extraction approach has generated interest from clinicians in a number of different fields and the medical technology industry. It is the subject of continued collaboration with the East Kent Hospital Trust.

Bibliography

[1] Ahmet Murat Bagci, Mahnaz Shahidi, Rashid Ansari, Michael Blair, Norman Paul Blair, and Ruth Zelkha. Thickness profiles of retinal layers by optical coherence tomography image segmentation. *American journal of ophthalmology*, 146(5):679–687, 2008. xiv, 85, 87, 88

[2] Stephanie J Chiu, Xiao T Li, Peter Nicholas, Cynthia A Toth, Joseph A Izatt, and Sina Farsiu. Automatic segmentation of seven retinal layers in sdoct images congruent with expert manual segmentation. *Optics express*, 18(18):19413–19428, 2010. xiv, 84, 88, 89, 90, 91, 92, 97, 98, 99, 158

[3] Stephanie J Chiu, Michael J Allingham, Priyatham S Mettu, Scott W Cousins, Joseph A Izatt, and Sina Farsiu. Kernel regression based segmentation of optical coherence tomography images with diabetic macular edema. *Biomedical optics express*, 6(4):1172–1194, 2015. xv, 84, 97, 98, 149, 153, 155, 157, 159

[4] James Fujimoto and Eric Swanson. The development, commercialization, and impact of optical coherence tomography. *Investigative ophthalmology & visual science*, 57(9):OCT1–OCT13, 2016. 2

[5] Anders Eklund, Paul Dufort, Daniel Forsberg, and Stephen M LaConte. Medical image processing on the gpu–past, present and future. *Medical image analysis*, 17(8):1073–1094, 2013. 2

[6] Vishal M Patel, Glenn R Easley, Dennis M Healy Jr, and Rama Chellappa. Compressed synthetic aperture radar. *IEEE Journal of selected topics in signal processing*, 4(2):244–254, 2010. 3

[7] Xilin Liu, Hongjie Zhu, Milin Zhang, Andrew G Richardson, Timothy H Lucas, and Jan Van der Spiegel. Design of a low-noise, high power efficiency neural recording front-end with an integrated real-time compressed sensing unit. In *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2996–2999. IEEE, 2015. 3

[8] Georgios N Lilis, Daniele Angelosante, and Georgios B Giannakis. Sound field reproduction using the lasso. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(8):1902–1912, 2010. 3

[9] Mark A Davenport, Chinmay Hegde, Marco F Duarte, and Richard G Baraniuk. Joint manifolds for data fusion. *IEEE Transactions on Image Processing*, 19(10):2580–2594, 2010. 3

[10] Chaitanya K Mididoddi, Fangliang Bai, Guoqing Wang, Jinchao Liu, Stuart Gibson, and Chao Wang. High-throughput photonic time-stretch optical coherence tomography with data compression. *IEEE Photonics Journal*, 9(4):1–15, 2017. 4

[11] Fangliang Bai, Jinchao Liu, Xiaojuan Liu, Margarita Osadchy, Chao Wang, and Stuart J Gibson. Lshr-net: A hardware-friendly solution for high-resolution computational imaging using a mixed-weights neural network. *Neurocomputing*, 406:169–181, 2020. 4

[12] Fangliang Bai, Stuart J Gibson, Manuel J Marques, and Adrian Podoleanu. Superpixel guided active contour segmentation of retinal layers in oct volumes. In *2nd Canterbury Conference on OCT with Emphasis on Broadband Optical Sources*, volume 10591, page 1059106. International Society for Optics and Photonics, 2018. 4

[13] Gang Huang, Hong Jiang, Kim Matthews, and Paul Wilford. Lensless imaging by compressive sensing. In *2013 IEEE International Conference on Image Processing*, pages 2101–2105. IEEE, 2013. 6, 7

[14] Michael B Wakin, Jason N Laska, Marco F Duarte, Dror Baron, Shriram Sarvotham, Dharmpal Takhar, Kevin F Kelly, and Richard G Baraniuk. An architecture for compressive imaging. In *2006 international conference on image processing*, pages 1273–1276. IEEE, 2006. 6

[15] Claude Elwood Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949. 7

[16] Emmanuel Candes and Justin Romberg. l1-magic: Recovery of sparse signals via convex programming. *URL: www. acm. caltech. edu/l1magic/downloads/l1magic. pdf*, 4:14, 2005. 16

[17] Xuan Liu and Jin U Kang. Compressive sd-oct: the application of compressed sensing in spectral domain optical coherence tomography. *Optics express*, 18(21):22010–22019, 2010. 22

[18] Ning Zhang, Tiancheng Huo, Chengming Wang, Tianyuan Chen, Jinggao Zheng, and Ping Xue. Compressed sensing with linear-in-wavenumber sampling in spectral-domain optical coherence tomography. *Optics letters*, 37(15):3075–3077, 2012.

[19] Daguang Xu, Yong Huang, and Jin U Kang. Real-time compressive sensing spectral domain optical coherence tomography. *Optics letters*, 39(1):76–79, 2014. 22

[20] Bahram Jalali and Mohammad H Asghari. The anamorphic stretch transform: Putting the squeeze on "big data". *Optics and Photonics News*, 25(2):24–31, 2014. 24

[21] Stephen Boyd, Neal Parikh, and Eric Chu. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011. 26

[22] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010. 26

[23] Stephen Becker, Jérôme Bobin, and Emmanuel J Candès. Nesta: A fast and accurate first-order method for sparse recovery. *SIAM Journal on Imaging Sciences*, 4(1):1–39, 2011. 26

[24] Yuye Ling, William Meiniel, Rajinder Singh-Moon, Elsa Angelini, Jean-Christophe Olivo-Marin, and Christine P Hendon. Compressed sensing-enabled phase-sensitive swept-source optical coherence tomography. *Optics express*, 27(2):855–871, 2019. 35

[25] Hao Chi and Zhijing Zhu. Analytical model for photonic compressive sensing with pulse stretch and compression. *IEEE Photonics Journal*, 11(1):1–10, 2018. 35

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. 38, 53

[27] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[28] 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*, pages 3431–3440, 2015. 38, 142

[29] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE international conference on computer vision*, pages 4539–4547, 2017. 38

[30] A. Mousavi, A. B. Patel, and R. G. Baraniuk. A deep learning approach to structured signal recovery. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1336–1343, Sept 2015. 38, 40

[31] Amir Adler, David Boublil, Michael Elad, and Michael Zibulevsky. A deep learning approach to block-based compressed sensing of images. *CoRR*, abs/1606.01519, 2016. 42

[32] A. Mousavi and R. G. Baraniuk. Learning to invert: Signal recovery via deep convolutional networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2272–2276, March 2017. 41

[33] A. Mousavi, G. Dasarathy, and R. G. Baraniuk. DeepCodec: Adaptive Sensing and Recovery via Deep Convolutional Neural Networks. *ArXiv e-prints*, July 2017. 42

[34] Kuldeep Kulkarni, Suhas Lohit, Pavan Turaga, Ronan Kerviche, and Amit Ashok. Reconnet: Non-iterative reconstruction of images from compressively sensed measurements. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 449–458, June 2016. 40, 55, 56

[35] Hantao Yao, Feng Dai, Dongming Zhang, Yike Ma, Shiliang Zhang, and Yongdong Zhang. Dr^2-net: Deep residual reconstruction network for image compressive sensing. *CoRR*, abs/1702.05743, 2017. 41, 53, 56

[36] Xuemei Xie, Yuxiang Wang, Guangming Shi, Chenye Wang, Jiang Du, and Xiao Han. Adaptive measurement network for cs image reconstruction. In Jinfeng Yang, Qinghua Hu, Ming-Ming Cheng, Liang Wang, Qingshan Liu, Xiang Bai, and Deyu Meng, editors, *Computer Vision*, pages 407–417, Singapore, 2017. Springer Singapore. 42, 56

[37] Jiang Du, Xuemei Xie, Chenye Wang, Guangming Shi, Xun Xu, and Yuxiang Wang. Fully convolutional measurement network for compressive sensing image reconstruction. *Neurocomputing*, 328:105 – 112, 2019. Chinese Conference on Computer Vision 2017. 38, 42, 55, 56, 67

[38] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, Aug 2007. 41

[39] J. Haupt, R. Nowak, and R. Castro. Adaptive sensing for sparse signal recovery. In *2009 IEEE 13th Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop*, pages 702–707, Jan 2009. 41

[40] J. D. Haupt, R. G. Baraniuk, R. M. Castro, and R. D. Nowak. Compressive distilled sensing: Sparse recovery using adaptivity in compressive measurements. In *2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers*, pages 1551–1555, Nov 2009. 41

[41] Michael Iliadis, Leonidas Spinoulas, and Aggelos K. Katsaggelos. Deep fully-connected networks for video compressive sensing. *Digital Signal Processing*, 72:9 – 18, 2018. 42

[42] Michael Iliadis, Leonidas Spinoulas, and Aggelos K. Katsaggelos. Deepbinarymask: Learning a binary mask for video compressive sensing. *CoRR*, abs/1607.03343, 2016. 43

[43] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016. 49

[44] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 5449–5458, Cham, 2016. Springer International Publishing. 51

[45] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 1026–1034, Washington, DC, USA, 2015. IEEE Computer Society. 51

[46] Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1874–1883, June 2016. 51

[47] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng. End-to-end text recognition with convolutional neural networks. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 3304–3308, Nov 2012. 53

[48] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba. End to End Learning for Self-Driving Cars. *ArXiv e-prints*, April 2016. 53

[49] E. Agustsson and R. Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1122–1131, July 2017. 55

[50] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 56

[51] David Taubman and Michael Marcellin. *JPEG2000 image compression fundamentals, standards and practice: image compression fundamentals, standards and practice*, volume 642. Springer Science & Business Media, 2012. 56

[52] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on computational imaging*, 3(1):47–57, 2016. 65

[53] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 70

[54] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. 71

[55] Ou Tan, Vikas Chopra, Ake Tzu-Hui Lu, Joel S Schuman, Hiroshi Ishikawa, Gadi Wollstein, Rohit Varma, and David Huang. Detection of macular ganglion cell loss in glaucoma by fourier-domain optical coherence tomography. *Ophthalmology*, 116(12):2305–2314, 2009. 76

[56] Julio E DeLeón-Ortega, Stella N Arthur, Gerald McGwin, Aiyuan Xie, Blythe E Monheit, and Christopher A Girkin. Discrimination between glaucomatous and nonglaucomatous eyes using quantitative imaging devices and subjective optic nerve head assessment. *Investigative ophthalmology & visual science*, 47(8):3374–3380, 2006. 76

[57] Takahiro Horii, Tomoaki Murakami, Kazuaki Nishijima, Tadamichi Akagi, Akihito Uji, Naoko Arakawa, Yuki Muraoka, and Nagahisa Yoshimura. Relationship between fluorescein pooling and optical coherence tomographic reflectivity of cystoid spaces in diabetic macular edema. *Ophthalmology*, 119(5):1047–1055, 2012. 76, 81

[58] 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. *science*, 254(5035):1178–1181, 1991. 76, 77

[59] Lavinia Ferrante di Ruffano, Jacqueline Dinnes, Jonathan J Deeks, Naomi Chuchu, Susan E Bayliss, Clare Davenport, Yemisi Takwoingi, Kathie Godfrey, Colette O'Sullivan, Rubeta N Matin, et al. Optical coherence tomography for diagnosing skin cancer in adults. *Cochrane Database of Systematic Reviews*, 1(12), 2018. 76

[60] Alan C Sull, Laurel N Vuong, Lori Lyn Price, Vivek J Srinivasan, Iwona Gorczynska, James G Fujimoto, Joel S Schuman, and Jay S Duker. Comparison of spectral/fourier domain optical coherence tomography instruments for assessment of normal macular thickness. *Retina (Philadelphia, Pa.)*, 30(2):235, 2010. 77, 78

[61] Johannes F De Boer, Barry Cense, B Hyle Park, Mark C Pierce, Guillermo J Tearney, and Brett E Bouma. Improved signal-to-noise ratio in spectral-domain compared with time-domain optical coherence tomography. *Optics letters*, 28(21):2067–2069, 2003. 78

[62] R Leitgeb, CK Hitzenberger, and Adolf F Fercher. Performance of fourier domain vs. time domain optical coherence tomography. *Optics express*, 11(8):889–894, 2003. 78

[63] MN Menke, P Knecht, V Sturm, S Dabov, and J Funk. Reproducibility of nerve fiber layer thickness measurements using 3d fourier-domain oct (topcon 3d-oct1000). *Investigative Ophthalmology & Visual Science*, 49(13):4259–4259, 2008. 78

[64] John E Legarreta, Giovanni Gregori, Robert W Knighton, Omar S Punjabi, Geeta A Lalwani, and Carmen A Puliafito. Three-dimensional spectral-domain optical coherence tomography images of the retina in the presence of epiretinal membranes. *American journal of ophthalmology*, 145(6):1023–1030, 2008. 78

[65] Wolfgang Drexler, Harald Sattmann, Boris Hermann, Tony H Ko, Michael Stur, Angelika Unterhuber, Christoph Scholda, Oliver Findl, Matthias Wirtitsch, James G Fujimoto, et al. Enhanced visualization of macular pathology with the use of ultrahigh-resolution optical coherence tomography. *Archives of ophthalmology*, 121(5):695–706, 2003. 78

[66] Tony H Ko, James G Fujimoto, Jay S Duker, Lelia A Paunescu, Wolfgang Drexler, Caroline R Baumal, Carmen A Puliafito, Elias Reichel, Adam H Rogers, and Joel S Schuman. Comparison of ultrahigh-and standard-resolution optical coherence tomography for imaging macular hole pathology and repair. *Ophthalmology*, 111(11):2033–2043, 2004. 79

[67] Erdem Ergun, Boris Hermann, Matthias Wirtitsch, Angelika Unterhuber, Tony H Ko, Harald Sattmann, Christoph Scholda, James G Fujimoto, Michael Stur, and Wolfgang Drexler. Assessment of central visual function in stargardt's disease/fundus flavimaculatus with ultrahigh-resolution optical coherence tomography. *Investigative ophthalmology & visual science*, 46(1):310–316, 2005.

[68] Gadi Wollstein, Leila A Paunescu, Tony H Ko, James G Fujimoto, Andrew Kowalevicz, Ingmar Hartl, Siobahn Beaton, Hiroshi Ishikawa, Cynthia Mattox, Omah Singh, et al. Ultrahigh-resolution optical coherence tomography in glaucoma. *Ophthalmology*, 112(2):229–237, 2005. 79

[69] Tony H Ko, James G Fujimoto, Joel S Schuman, Lelia A Paunescu, Andrew M Kowalevicz, Ingmar Hartl, Wolfgang Drexler, Gadi Wollstein, Hiroshi Ishikawa, and Jay S Duker. Comparison of ultrahigh-and standard-resolution optical coherence tomography for imaging macular pathology. *Ophthalmology*, 112(11):1922–e1, 2005.

[70] Matthias G Wirtitsch, Erdem Ergun, Boris Hermann, Angelika Unterhuber, Michael Stur, Christoph Scholda, Harald Sattmann, Tony H Ko, James G Fujimoto, and Wolfgang Drexler. Ultrahigh resolution optical coherence tomography in macular dystrophy. *American journal of ophthalmology*, 140(6):976–983, 2005. 79

[71] Lelia A Paunescu, Tony H Ko, Jay S Duker, Annie Chan, Wolfgang Drexler, Joel S Schuman, and James G Fujimoto. Idiopathic juxtafoveal retinal telangiectasis: new findings by ultrahigh-resolution optical coherence tomography. *Ophthalmology*, 113(1):48–57, 2006.

[72] Tony H Ko, Andre J Witkin, James G Fujimoto, Annie Chan, Adam H Rogers, Caroline R Baumal, Joel S Schuman, Wolfgang Drexler, Elias Reichel, and Jay S Duker. Ultrahigh-resolution optical coherence tomography of surgically closed macular holes. *Archives of Ophthalmology*, 124(6):827–836, 2006.

[73] Christoph Scholda, Matthias Wirtitsch, Boris Hermann, Angelika Unterhuber, Erdem Ergun, Harald Sattmann, Tony H Ko, James G Fujimoto, Adolf F Fercher, Michael Stur, et al. Ultrahigh resolution optical coherence tomography of macular holes. *Retina*, 26(9):1034–1041, 2006.

[74] Andre J Witkin, Tony H Ko, James G Fujimoto, Annie Chan, Wolfgang Drexler, Joel S Schuman, Elias Reichel, and Jay S Duker. Ultra-high resolution optical coherence tomography assessment of photoreceptors in retinitis pigmentosa and related diseases. *American journal of ophthalmology*, 142(6):945–952, 2006. 78

[75] Emily Huynh, Erandi Chandrasekera, Danuta Bukowska, Samuel McLenachan, David A Mackey, and Fred K Chen. Past, present, and future concepts of the choroidal scleral interface morphology on optical coherence tomography. *The Asia-Pacific Journal of Ophthalmology*, 6(1):94–103, 2017. 80

[76] 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. *Ophthalmology*, 121(8):1572–1578, 2014.

[77] Richard F Spaide and Christine A Curcio. Anatomical correlates to the bands seen in the outer retina by optical coherence tomography: literature review and model. *Retina (Philadelphia, Pa.)*, 31(8):1609, 2011. 80

[78] FG Holz and RF Spaide. Essentials in ophthalmology-medical retina, 2005. 82

[79] Takahiro Horii, Tomoaki Murakami, Tadamichi Akagi, Akihito Uji, Naoko Ueda-Arakawa, Kazuaki Nishijima, and Nagahisa Yoshimura. Optical coherence tomographic reflectivity of cystoid spaces is related to recurrent diabetic macular edema after triamcinolone. *Retina*, 35(2):264–271, 2015. 82

[80] Glenn J Jaffe and Joseph Caprioli. Optical coherence tomography to detect and manage retinal disease and glaucoma. *American journal of ophthalmology*, 137(1):156–169, 2004. 82

[81] Richard F Spaide. Enhanced depth imaging optical coherence tomography of retinal pigment epithelial detachment in age-related macular degeneration. *American journal of ophthalmology*, 147(4):644–652, 2009. 83

[82] Fernando M Penha, Philip J Rosenfeld, Giovanni Gregori, Manuel Falcão, Zohar Yehoshua, Fenghua Wang, and William J Feuer. Quantitative imaging of retinal pigment epithelial detachments using spectral-domain optical coherence tomography. *American journal of ophthalmology*, 153(3):515–523, 2012. 83

[83] Pascal A Dufour, Lala Ceklic, Hannan Abdillahi, Simon Schroder, Sandro De Dzanet, Ute Wolf-Schnurrbusch, and Jens Kowal. Graph-based multi-surface segmentation of oct data using trained hard and soft constraints. *IEEE transactions on medical imaging*, 32(3):531–543, 2012. 84

[84] Raheleh Kafieh, Hossein Rabbani, Michael D Abramoff, and Milan Sonka. Intra-retinal layer segmentation of 3d optical coherence tomography using coarse grained diffusion map. *Medical image analysis*, 17(8):907–928, 2013. 84

[85] Rabia Gürses-Özden, Christopher Teng, Roberto Vessani, Samiah Zafar, Jeffrey M Liebmann, and Robert Ritch. Macular and retinal nerve fiber layer thickness measurement reproducibility using optical coherence tomography (oct-3). *Journal of glaucoma*, 13(3):238–244, 2004. 85

[86] Dara Koozekanani, Kim Boyer, and Cynthia Roberts. Retinal thickness measurements from optical coherence tomography using a markov boundary model. *IEEE transactions on medical imaging*, 20(9):900–916, 2001. 85

[87] Annie Chan, Jay S Duker, Hiroshi Ishikawa, Tony H Ko, Joel S Schuman, and James G Fujimoto. Quantification of photoreceptor layer thickness in normal eyes using optical coherence tomography. *Retina (Philadelphia, Pa.)*, 26(6):655, 2006. 85

[88] Ahmet M Bagci, Rashid Ansari, and William D Reynolds. Low-complexity implementation of non-subsampled directional filter banks using polyphase representations and generalized separable processing. In *2007 IEEE International Conference on Electro/Information Technology*, pages 422–427. IEEE, 2007. 86

[89] Edsger W Dijkstra et al. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959. 89

[90] Tianqiao Zhang, Zhangjun Song, Xiaogang Wang, Huimin Zheng, Fucang Jia, Jianhuang Wu, Guanglin Li, and Qingmao Hu. Fast retinal layer segmentation of spectral domain optical coherence tomography images. *Journal of biomedical optics*, 20(9):096014, 2015. 92, 94, 95, 103

[91] Abraham Savitzky and Marcel JE Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964. 94

[92] Delia Cabrera Fernandez. Delineating fluid-filled region boundaries in optical coherence tomography images of the retina. *IEEE transactions on medical imaging*, 24(8):929–945, 2005. 96

[93] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern analysis and machine intelligence*, 12(7):629–639, 1990. 96

[94] Lynne Pezzullo, Jared Streatfeild, Philippa Simkiss, and Darren Shickle. The economic impact of sight loss and blindness in the uk adult population. *BMC health services research*, 18(1):63, 2018. 103

[95] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012. 112

[96] Philippe Thevenaz, Urs E Ruttimann, and Michael Unser. A pyramid approach to subpixel registration based on intensity. *IEEE transactions on image processing*, 7(1):27–41, 1998. 127

[97] Rachid Tahiri Joutei Hassani, Hong Liang, Mohamed El Sanharawi, Emmanuelle Brasnu, Sofiene Kallel, Antoine Labbe, and Christophe Baudouin. En-face optical coherence tomography as a novel tool for exploring the ocular surface: a pilot comparative study to conventional b-scans and in vivo confocal microscopy. *The ocular surface*, 12(4):285–306, 2014. 132

[98] Tarkan Mumcuoglu, Gadi Wollstein, Maciej Wojtkowski, Larry Kagemann, Hiroshi Ishikawa, Michelle L Gabriele, Vivek Srinivasan, James G Fujimoto, Jay S Duker, and Joel S Schuman. Improved visualization of glaucomatous retinal damage using high-speed ultrahigh-resolution optical coherence tomography. *Ophthalmology*, 115(5):782–789, 2008.

[99] Mehreen Adhi and Jay S Duker. Optical coherence tomography–current and future applications. *Current opinion in ophthalmology*, 24(3):213, 2013. 132

[100] Ronald Klein, Michael D Knudtson, Kristine E Lee, Ronald Gangnon, and Barbara EK Klein. The wisconsin epidemiologic study of diabetic retinopathy xxii: the twenty-five-year progression of retinopathy in persons with type 1 diabetes. *Ophthalmology*, 115(11):1859–1868, 2008. 141

[101] Mark W Johnson. Etiology and treatment of macular edema. *American journal of ophthalmology*, 147(1):11–21, 2009. 141

[102] Michael R Hee, Carmen A Puliafito, Carlton Wong, Jay S Duker, Elias Reichel, Bryan Rutledge, Joel S Schuman, Eric A Swanson, and James G Fujimoto. Quantitative assessment of macular edema with optical coherence tomography. *Archives of ophthalmology*, 113(8):1019–1029, 1995. 141

[103] Igor Kozak, Victoria L Morrison, Thomas M Clark, Dirk-Uwe Bartsch, Byung Ro Lee, Iryna Falkenstein, Ajay M Tammewar, Francesca Mojana, and William R Freeman. Discrepancy between fluorescein angiography and optical coherence tomography in detection of macular disease. *Retina (Philadelphia, Pa.)*, 28(4):538, 2008. 141

[104] John C BuAbbud, Motasem M Al-latayfeh, and Jennifer K Sun. Optical coherence tomography imaging for diabetic retinopathy and macular edema. *Current diabetes reports*, 10(4):264–269, 2010. 141

[105] Pedro Romero-Aroca. Managing diabetic macular edema: the leading cause of diabetes blindness. *World journal of diabetes*, 2(6):98, 2011.

[106] Colin Siang Hui Tan, Milton Cher Yong Chew, Louis Wei Yi Lim, and Srinivas R Sadda. Advances in retinal imaging for diabetic retinopathy and diabetic macular edema. *Indian journal of ophthalmology*, 64(1):76, 2016. 141

[107] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 142, 150

[108] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 142

[109] D Martin, C Fowlkes, D Tal, and J Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 416–423. IEEE, 2001. 143

[110] Joo Yong Lee, Stephanie J Chiu, Pratul P Srinivasan, Joseph A Izatt, Cynthia A Toth, Sina Farsiu, and Glenn J Jaffe. Fully automatic software for retinal thickness in eyes with diabetic macular edema from images acquired by cirrus and spectralis systems. *Investigative ophthalmology & visual science*, 54(12):7595–7602, 2013. 144

[111] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006. 145

[112] Paweł Liskowski and Krzysztof Krawiec. Segmenting retinal blood vessels with deep neural networks. *IEEE transactions on medical imaging*, 35(11):2369–2380, 2016. 147

[113] Michal Drozdzal, Eugene Vorontsov, Gabriel Chartrand, Samuel Kadoury, and Chris Pal. The importance of skip connections in biomedical image segmentation. In *Deep Learning and Data Labeling for Medical Applications*, pages 179–187. Springer, 2016. 147

[114] 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 *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014. 147

[115] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011. 147, 148, 151

[116] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. BM3D Image Denoising with Shape-Adaptive Principal Component Analysis. In Rémi Gribonval, editor, *SPARS'09 - Signal Processing with Adaptive Sparse Structured Representations*, Saint Malo, France, April 2009. Inria Rennes - Bretagne Atlantique. 149

[117] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *2011 International Conference on Computer Vision*, pages 991–998. IEEE, 2011. 150

[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)*, pages 565–571. IEEE, 2016. 151

[119] Choong Ho Lee and Hyung-Jin Yoon. Medical big data: promise and challenges. *Kidney Research and Clinical Practice*, 36(1):3–11, mar 2017. 162

[120] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Lecture Notes in Computer Science*, pages 234–241. Springer International Publishing, 2015. 162, 170

[121] Mohammad Havaei, Axel Davy, David Warde-Farley, Antoine Biard, Aaron Courville, Yoshua Bengio, Chris Pal, Pierre-Marc Jodoin, and Hugo Larochelle. Brain tumor segmentation with deep neural networks. *Medical Image Analysis*, 35:18–31, jan 2017.

[122] Eli Gibson, Wenqi Li, Carole Sudre, Lucas Fidon, Dzhoshkun I. Shakir, Guotai Wang, Zach Eaton-Rosen, Robert Gray, Tom Doel, Yipeng Hu, Tom Whyntie, Parashkev Nachev, Marc Modat, Dean C. Barratt, Sébastien Ourselin, M. Jorge Cardoso, and Tom Vercauteren. NiftyNet: a deep-learning platform for medical imaging. *Computer Methods and Programs in Biomedicine*, 158:113–122, may 2018. 162

[123] Ashnil Kumar, Jinman Kim, Weidong Cai, Michael Fulham, and Dagan Feng. Content-based medical image retrieval: A survey of applications to multidimensional and multimodality data. *Journal of Digital Imaging*, 26(6):1025–1039, jul 2013. 163

[124] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016. 173

[125] David Gutman, Noel CF Codella, Emre Celebi, Brian Helba, Michael Marchetti, Nabin Mishra, and Allan Halpern. Skin lesion analysis toward melanoma detection: A challenge at the international symposium on biomedical imaging (isbi) 2016, hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1605.01397*, 2016. 177