Developing algorithms for the analysis of retinal Optical Coherence Tomography images

by

Peyman Gholami

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Science
in
Vision Science and System Design Engineering

Waterloo, Ontario, Canada, 2018

© Peyman Gholami 2018

Examining Committee Membership

The following served on the Examining Committee for this thesis.

Supervisors:	Dr. Vasudevan Lakshminarayanan Professor, School of Optometry & Vision Science University of Waterloo
	Dr. John Zelek Associate Professor, System Design Engineering Department University of Waterloo
Committee Members:	Dr. Daphne McCulloch Professor, School of Optometry & Vision Science University of Waterloo
	Dr. Alexander Wong Associate Professor, System Design Engineering Department University of Waterloo

Author Deceleration

I hereby declare that this thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

The following papers have resulted from the work presented in this thesis, and are published:

1. *Peyman Gholami, Priyanka Roy, Mohana Kuppuswamy Parthasarathy, Abbas Ommani, John Zelek, and Vasudevan Lakshminaraynan, "Intra-retinal segmentation of optical coherence tomography images using active contours with a dynamic programming initialization", in Proc.SPIE vol. 10483, (2018), pp. 10483 - 10483 - 6.*

2. *Peyman Gholami, Mohsen Sheikh Hassani, Mohana KuppuswamyParthasarathy, John S. Zelek, and Vasudevan Lakshminarayanan, "Classification of optical coherence tomography images for diagnosing different ocular diseases", in Proc.SPIE vol. 10487, (2018), pp. 10487 - 10487 - 6.*

In addition, the following publications are under submission:

3. *Peyman Gholami, John S. Zelek, and Vasudevan Lakshminarayanan, Transfer-learning-based classification of Optical Coherence Tomography images, under submission to Journal of Imaging*

and

4. *Peyman Gholami, Amor Gueddana, John S. Zelek, and Vasudevan Lakshminarayanan, Fully automated identification of Ocular diseases using Optical Coherence Tomography images, under submission to Medical Image Analysis Journal)*

Abstract

Vision loss, with a prevalence loss greater than 42 million in the United States is one of the major challenges of today's health-care industry and medical science. Early detection of different retinal-related diseases will dramatically reduce the risk of vision loss. Optical Coherence Tomography (OCT) is a relatively new imaging technique which is of great importance in the identification of ocular and especially retinal diseases. Thus, the efficient analysis of OCT images provides several advantages. In this thesis, we propose a series of image processing and machine learning techniques for the automated analysis of OCT images. The proposed methodology in chapter 2 localizes different retinal layers using a modified version of active contour models. In chapter 3, we propose a method which classifies OCT images based on different pathological conditions using novel methods, e.g., transfer learning and new texture detection techniques. The proposed methods along with the clinically meaningful extracted characteristics provide numbers of applications and benefits, e.g., saving a considerable amount of time and providing more-efficient and -accurate indices for the diagnosis and treatment of different ocular diseases to ophthalmologists and finally reducing the overall risk of vision loss.

Acknowledgements

I would like to express my deepest gratitude and appreciation to my supervisors, Dr. Vasudevan Lakminarayanan and Dr. John Zelek for giving me this valuable opportunity to learn and explore a field of study that continuously intrigues me, and for their unwavering support, encouragement and mentorship. Dr. Lakshminarayanan provided all kinds of support for me throughout this journey and he means more than just a supervisor to me, but like one of my best friends. Dr. Zelek also played a complementary role in my professional development with his valuable ideas and continuous supports. Furthermore, I would like to thank them both for giving me complete freedom in research, while still guiding me towards a fruitful path to completing my degree. I believe that this research would have not been possible without the active and loving support of my supervisors.

I would also like to express my special thank to my committee members Dr. Daphne McCulloch and Dr. Alexander Wong, for their valued knowledge, advice and time invested in guiding me throughout my degree.

A heartfelt thank to all of my colleagues at the theoretical and experimental epistemology lab (TEEL), Priyanka, Asmaa, Abbas, and Henry. A very special thank to my friend Mrs. Mohana Kuppuswamy Parthasarathy for sharing her clinical knowledge with me.

This study was supported by an NSERC discovery grant to Dr. Lakminarayanan. I would also like to thank Dr. Bhende and Ms. Girija from SN hospital, Chennai, India for their assistance in the process of collecting data.

Lastly but certainly not least, I am grateful to my beloved parents for their constant support and encouragement.

Dedication

To my beloved parents, Parvin and Hossein

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AMD	Age related Macular Degeneration
AUC	Area Under the Curve
CNN	Convolutional neural networks
DR	Diabetic Retinopathy
DS	Dice similarity coefficient
DSIFT	Dense scale-Invariant Feature Transfor
DTD	Describable texture data-set
ELM	External limiting membrane
EZ	Ellipsoid zone
FN	False Negative
FP	False Positive
FV	Fisher Vector
GCL	Ganglion cell layer
HMM	Hidden Marcov Model
INL	Inner nuclear layer
IS	Inner segment
IZ	Interdigitation zone

JI	Jaccard index
LBP	Local Binary Pattern
MH	Macular Hole
MSVM	Multi-phase Support Vector Machine
NO	Normal
OCT	Optical Coherence Tomography
OCTID	Optical Coherence Tomography Image Databas
ONL	Outer nuclear layer
OPL	Outer plexiform layer
OS	Outer segment
RNFL	Retinal nerve fiber layer
ROC	Receiver Operating Characteristic
RPE	Retinal Pigment Epithelium
RPE	Retinal pigmented epithelium
SIFT	Scale-Invariant Feature Transfor
SVM	Support Vector Machine
TN	True Negative
TP	True Positive

Chapter 1

Introduction

1.1 Background

Vision loss and visual impairment, with a loss prevalence greater than 42 million in the United States [2], is one of the main challenges of today's health care system. In addition to the economic burden of vision loss, equivalent to $139 billion [3], studies [2] have shown that a problem free vision plays an important role in the overall quality of life. A major part of ocular disease involves the retina.

1.1.1 The Retina

The retina, is the photo-sensitive tissue covering the back of the eye which processes light and transmits the information to brain through optic nerves. It contains different cells and photo-receptors which form different layers. Histology and anatomy of the eye books [4] localize up to 13 different layers for the retina. Starting from the vitreous body to choroid, the nine most outstanding retinal layers include retinal nerve fiber layer and ganglion cell layer (RNFL+GCL), inner nuclear layer (INL), outer plexiform layer (OPL), outer nuclear layer (ONL), external limiting membrane and inner segment (ELM+IS), ellipsoid zone (EZ), outer segment (OS), and retinal pigmented epithelium and interdigitation zone (RPE+IZ). The shape and the thickness of these layers are one of the main indicators of a healthy vision. Figure 1.3 shows the structure of retinal layers.

A common way to capture retinal structure and analyze its health is by using fundus

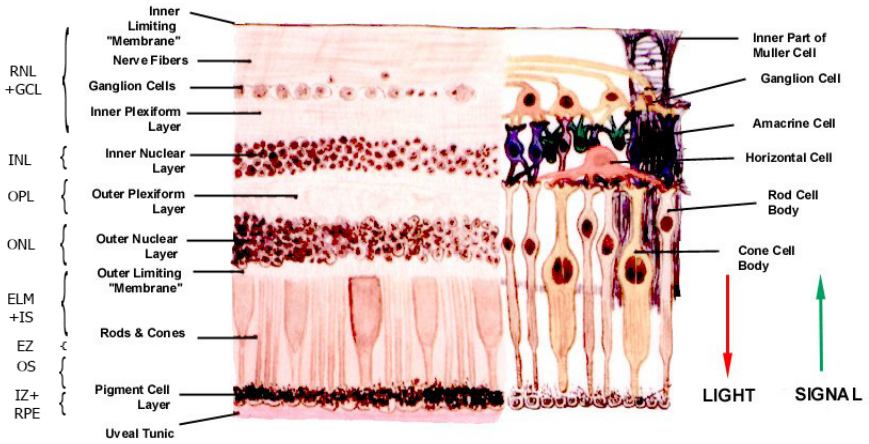


Figure 1.1: The retina and the structure of retinal layers [1]

photography. The fundus is the inner-lining of the retina. Fundus images visualize macula, fovea, optic disc and retinal blood vessels. Figure 1.2 shows a sample fundus image.

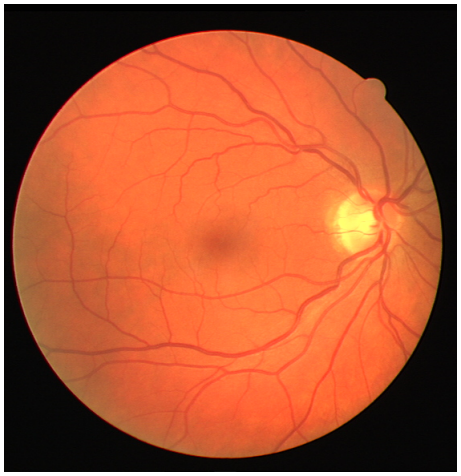


Figure 1.2: A sample fundus image

While providing much useful information, fundus images cannot visualize the morphology and shape of each of the individual retinal layers. Therefore, for a more accurate diagnosis and to track the treatment progress, clinicians need to have a visualization of the changes in the morphology of each retinal layer. Thanks to the advancement in medical imaging technologies, nowadays, it is possible to capture the form of these layers using

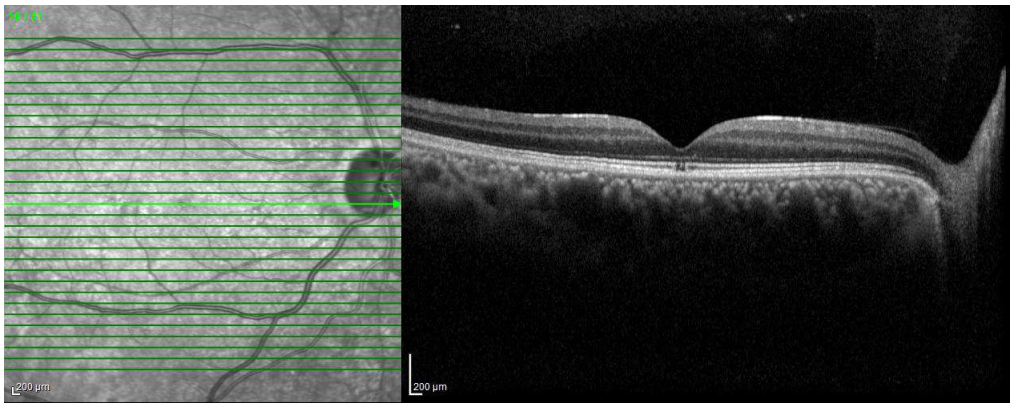


Figure 1.3: Sample retinal fundus image with the visualization of each of the B-Scans on it (left) with the corresponding OCT image centered at the fovea (right). Each green line represents a 2D B-Scan and together, set of B-Scans form a 3D OCT volume.

Optical Coherence Tomography (OCT) imaging.

1.1.2 Optical Coherence Tomography

OCT in a non-invasive imaging modality which uses coherent light to capture micro resolution tomographic (cross-sectional) images from within an optical scattering media, e.g., biological tissues. OCT devices measure the echo time delay and intensity of back-reflected and back-scattered light [5]. OCT images can be captured from within any optical scattering media. OCT images are created by measuring the time delay and the strength of reflected light from a tissue. In clinical ophthalmology, OCT images have become one of the main standard procedures for anterior and posterior eye segment imaging and for the diagnosis of eye disease. Using Fourier transform and broad range of wavelengths, Spectral Domain (SD) OCT results in a significant reduction in the acquisition time and improvement in image resolution [5].

In addition to the volumetric and cross sectional visualization of retinal layers, OCT images can provide several clinically meaningful indices, qualitative information, and quantitative measurements, e.g., retinal layer thicknesses, which can be used for the diagnosis and classification of various eye diseases and help clinicians obtain more accurate and efficient treatments. Figure 1.3 shows a sample retinal fundus image on the left, with the corresponding OCT image for one of the cross-sections on the right. The image was taken using a Spectralis OCT Heidelberg Engineering, GmbH (Heidelberg, Germany).

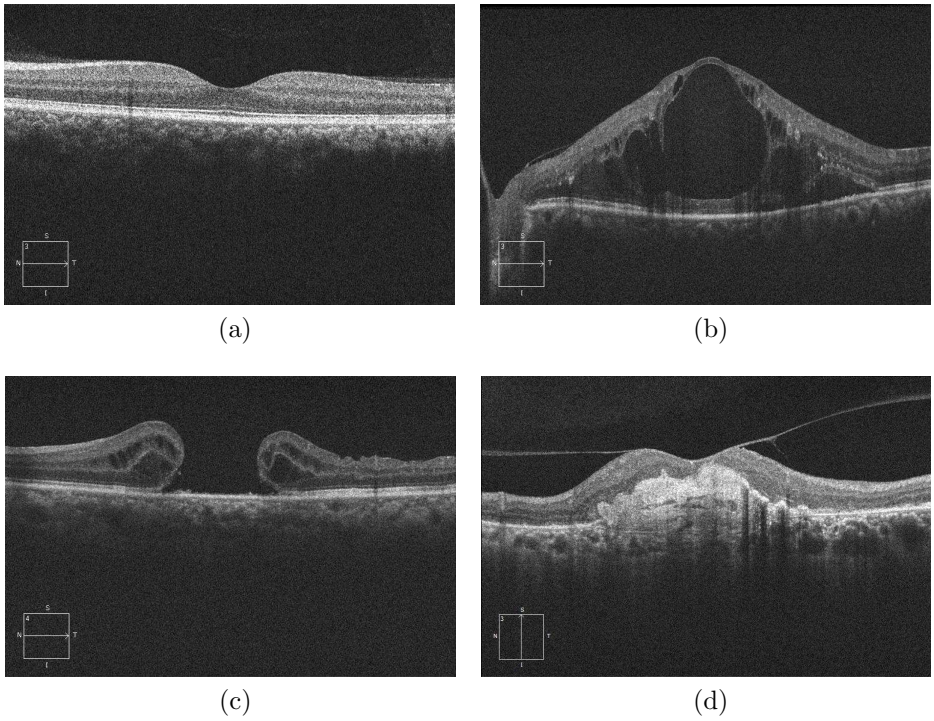
1.1.3 Retinal related diseases

In the following chapters, we mainly focus on three common retina related diseases, i.e., Age related Macular Degeneration (AMD), Diabetic Retinopathy (DR), and Macular Hole (MH). These diseases account for over 25% of severe visual impairment causes [6] worldwide. In what follows, we present some general information about each disease and their symptoms:

- Diabetic retinopathy (DR), is one of the most important causes of blindness in adults over the age 65 in the United States [5]. Due to diabetes, retinal blood vessels can break down, leak or become blocked. Diabetic macular edema can occur in any stage of DR and cause several changes to retinal layer shapes, including increased retinal thickness. This is due to intra-retinal fluid accumulation and is indicated by reduced intra-retinal reflectivity. Other visible changes in DR OCT images include abnormal foveal contour, changes in the central macular thickness, etc. [5]. There are two major types of DR, nonproliferative diabetic retinopathy (NPDR) and proliferative diabetic retinopathy (PDR). NPDR's clinical features include microaneurysms, hard exudates, nerve fiber layer infarcts or cotton-wool exudates, and intraretinal microvascular abnormalities; while PDR shows proliferating new vessels on the optic nerve head, retina, or iris in addition to the NPDR features [5].

- Age-related Macular Degeneration (AMD) is the leading cause for sever visual impairments and vision loss in Western countries. AMD affects the part of the retina responsible for sharp central vision, namely the macula. The OCT image for AMD usually appears with thickened, irregular, and disorganized retinal layers with the thickening of Retinal Pigment Epithelium (RPE) layer which results in the elevation of the photoreceptor layer. Also, it usually causes fluid accumulation beneath the fovea and in some cases the detachment of RPE layer can occur [5]. AMD has two different stages known as known as dry (nonneovascular) and wet (neovascular). Wet AMD happens in the late stages of AMD and occurs by the formation of choroidal neovascularization (CNV) under the macula or the formation of intraretinal neovascularization or a combination of both processes [5]. Wet AMD, while being less prevelance, causes most of the sever vision loss in AMD.

- Macular Hole (MH) is also one of the other common age-related eye diseases and causes a decrease in visual acuity. MH retinal defects can vary from small hyproflective breaks to hyproflective intra-retinal spaces in the OCT images. Other signs include thickened edges with cavities of reduced reflectivity, macular edema, and



(a) (b)

(c) (d)

Figure 1.4: Sample OCT images representing each of the four categories: (a) Normal (b) Diabetic Retinopathy (c) Macular Hole (d) Age-related Macular Degeneration

fluid accumulation [5]. Furthermore, using OCT images one can understand on the etiology of the MH disease.

For more information and detailed explanation about common retinal-related diseases please see [7].

1.1.4 Computer aided diagnosis systems

The aforementioned ocular diseases will lead to a change on the local texture and the overall morphology of OCT images. Segmentation of the retinal layers is one of the most common ways to use this information. There are two ways of segmenting OCT images: manual and automated methods. Manual segmentation is subjective and a very time-consuming procedure, especially in 3D cross sectional images [8]. On the other hand, most of the default image segmentation software that are provided with OCT devices and are being

used by clinicians require high amount of supervision and are plagued with limitations and problems. The main limitation is that most of them are designed for the normal and healthy retina. However, in practice, when dealing with pathological retina and different ocular diseases, the retinal layers are detached from their place and their structures are changed. Other limitations include high computational cost, manual initialization, etc. [9]

The segmentation of retinal layers in Spectral Domain Optical Coherence Tomography (SD-OCT) images has led to several applications and benefits. By quantifying the characteristics of retinal layers in different OCT images, clinicians can have more-efficient and -accurate indices for the diagnosis and treatment of different ocular diseases [10]. Also, by this means, the relative assessment error-rate is reduced dramatically. Some applications include the early detection of different diseases, e.g., glaucoma, Age-related Macular Degeneration (AMD), etc., and also the measurement of retinal layer thickness [9]. Furthermore, the analysis of different ocular diseases from OCT images is a time-consuming and difficult task, especially in the early stages of some diseases such as AMD [11]. However, the information extracted from segmentation can be used as features to detect different ocular diseases and improve OCT image classification results while saving a considerable amount of time and effort.

On the other hand, several studies suggest that more than 75% of the ocular diseases are preventable if we provide the platform to detect them in their early stages [12]. Thus, by designing a reliable computer aided analysis of OCT images and correlating the structural changes to changes in visual acuity, clinicians can have a better understanding of the mechanism of vision loss associated with those diseases, as well as understanding the mechanism of visual acuity benefits attributed to various treatments [5].

1.2 Objectives and major contributions

In this thesis, we propose a set of image processing and machine learning algorithms in order to facilitate the analysis of OCT images and obtain more effective diagnoses. The ultimate goal of this study is to propose a fully automated technique which can finally be used for the early diagnosis of several ocular diseases and reduce the risk of vision loss. We propose an active contour-based image segmentation technique for determining the boundary and location of different retinal layers. Also, we classify OCT images based on different diseases. We use a new feature extraction method and combine it with a new structure of a conventional classifier. We also use the state-of-the-art methods, including transfer learning for this purpose. One of the main contribution of work is on the formulation of an

innovative segmentation method. Furthermore, we are the first to uses the state-of-the-art texture feature extraction methods for OCT image classification and then compare it with transfer learning methods.

Specific aims of the research are as follows:

- ⊙ To quantify the characteristics and measure the thickness of each of the different retinal layers
- ⊙ To nvestigate the performance of different pre-trainted networks for the classification of OCT images
- ⊙ To study the effect of common methods for increasing the performance of deep neural networks, e.g., data augmentation techniques
- ⊙ To collect and setg up an open-source SD-OCT database containing images from different pathological conditions along with the ground truth delineations
- ⊙ To Design and implement a semi-automatic GUI for determining the ground truth delineations by clinicians
- ⊙ To propose a noise-removal technique for SD-OCT images which preserves details and information contained in the images
- ⊙ To compare the proposed segmentaiton method with other common methods currently described in the literature

1.3 Organization of the thesis

In this chapter, we presented an introduction on the importance of developing an efficient computer-aided system for the analysis of OCT images which assists clinicians. In the following chapters we will delve into the details of each of the proposed methods and solutions. In each chapter, we will first present a literature review on each topic, elaborating on the methods, and finally at the end of each chapter, we will compare our results with the most outstanding works in the literature. Finally, we will provide a discussion on each of the proposed methods in each chapter.

In Chapter 2, we propose a fully automated segmentation method for localizing each of the retinal layers and their thicknesses. We also elaborate on the denoising method and

the manual segmentation in this chapter. In Chapter 3 we build a new fully automated classification method based on OCT image textures which can be used for the identification of different ocular diseases. Chapter 4 concludes the thesis with some future directions.

Chapter 2

Intra-retinal Segmentation of Optical Coherence Tomography images

(Based on: *Gholami, P., Roy, P., Parthasarathy, M. K., Ommani, A., Zelek, J., & Lakshminarayanan, V. (2018). Intra-retinal segmentation of optical coherence tomography images using active contours with a dynamic programming initialization and an adaptive weighting strategy. In Optical Coherence Tomography and Coherence Domain Optical Methods in Biomedicine XXII (Vol. 10483, p. 104832M). International Society for Optics and Photonics*)

2.1 Background

Several studies have addressed OCT image segmentation. In general, the current state of the art OCT image segmentation techniques can be divided into two categories: fixed mathematical-model-based methods and machine-learning-based methods[13].

Mathematical model-based methods include A-scan, B-scan, active contours, graph theoretical methods, etc., among which active contours have shown to be very effective [14, 15, 16, 17]. Active contours are deformable curves that are widely used in image segmentation. By initializing the curve and minimizing the energy function, thorough an iterative process, the curve evolves and matches the region of interest. The energy function includes both internal forces which correspond to the specifications of the curve and keep the energy function toward the initial states; and external energy, which considers the features from the image and pulls the curve toward the object. Generally, there are two

different types of deformable models: parametric deformable models and non-parametric deformable models. Each of these methods can adapt by altering either the formulation of the energy function, the initialization, or the optimization technique.

Niu et al. [14] obtained a mean overlap ratio of 75.93% for two data sets using region-based active contour models via local similarity factor. Mishra et al. [16] used traditional active contours with a dynamic programming initialization which is very similar to the method that we are using in this chapter for initialization. Ghorbel et al. [17] segmented eight retinal layers using active contours and Markov random fields with Kalman filters and obtained a similarity index greater than 0.7 for all layers, 0.87 for the RNFL layer, and 0.97 for OPL+ONL layer. While presenting good results, most of the mathematical models require precise initialization and a robust noise-removal technique.

In recent years, some deep-learning-based methods have presented good results. Fang et al. [13] obtained mean difference between the segmented image and gold standard of 1.26 pixels for total retina by combining convolutional neural networks (CNN) and graph search methods. However, due to the high amount of time, data, and computational cost in deep learning methods, they are still not being widely used, especially in clinical practices. For a more detailed review of OCT image segmentation studies see Usman et al. [18]

In OCT images, as shown in Figure 1.3, each of the retinal layers contain homogeneous regions, i.e., similar gray-scale intensities. Therefore, one strategy to localize retinal layers is to use a region-growing technique inside each of these regions. On the other hand, the boundary of each of the retinal layers contains distinct edges which can be used as a feature for segmentation. Therefore, we have developed a technique which is able to use both of these information efficiently. In what follows, we describe a new method which uses a combination of these two features in the formulation of an active contour model.

2.2 Methodology

In this chapter we propose a new and efficient image segmentation algorithm for delineating the location and the thickness of different intra-retinal layers in SD-OCT images which is consistent across noisy images and different pathological conditions. We will use a modified active contour model, based on the active contour without edges method [19]. A prior term is then added to the energy function so that we can specify the active contour for OCT images and the method can use both edge and region information contained on the image.

2.2.1 Data-set structure

The data-set consisted of 25 spectral-domain OCT volumes from healthy human retina. The images were taken using a Heidelberg Spectralis device (Heidelberg Engineering, Heidelberg, Germany) with a 6mm scan length, containing 1,536 pixels. A fovea-centered image was then chosen on each volume by an optometrist.

2.2.2 Pre-processing of the images

Generally, the analysis of OCT images is divided into the pre-processing, e.g., denoising, cropping, etc., and the image-segmentation parts. According to the nature of the procedure, OCT images contain large amounts of noise, namely Speckle noise. Speckle significantly reduces the image quality and makes the image-segmentation and -processing more challenging. One of the major effects of most denoising techniques is a decrease in the image resolution [9]. Therefore, designing and implementing an efficient noise-removal technique for OCT images is indispensable. Some examples of the noise removal methods that are used in the literature include mean filters [15], and diffusion filters [10, 17, 20].

In recent years, the wavelet-based noise-removal techniques have appeared to be very effective in reducing the speckle noise as well as preserving the image quality [9]. We used a wavelet decomposition denoising method using a biorthogonal spline wavelet (Bior 3.7) with 17 levels of decomposition. The denoising threshold parameters were obtained using the Birgé-Massart method [21].

2.2.3 Initialization

In order to estimate the initial location of the active contour curves, a dynamic programming approach, that used by Mishra et al. [16], was used. The process is started by choosing three pixels manually around the location of each layer. The initial pixels need not to be very accurate and are only used for building a sparse trellis for further search around them. By assuming a sparse trellis $V = v_1, v_2, \ldots, v_{m^n}$ on an OCT image which contains m lines and n nodes on each line, the nodes on the trellis which satisfy the following equation will be the location of the initial curve:

$$v_i = \underset{v \notin v_k (k<i)}{\operatorname{argmin}} (s_j \psi_j) \tag{2.1}$$

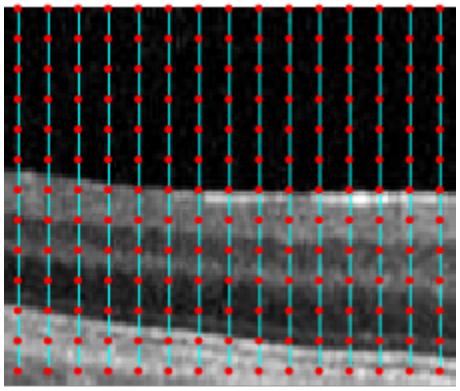


Figure 2.1: Demonstration of the sparse trellis on a sample image.

in which ψ is a strictly decreasing function defined as:[16].

$$\psi_i = \frac{1}{1 + k\,|\nabla(v_i)|^2} \tag{2.2}$$

k in 2.2 is set to 4 empirically. Also, $s_j = \|v_{j-1} - v_j\|^2$is the Euclidean norm of two consecutive vertices on the trellis.

2.2.3.1 Viterbi dynamic programming

We model the initialization problem as a Hidden Marcov Model (HMM) and then use the Viterbi dynamic programming algorithm [22] to find the optimal solution, i.e., the strongest local boundaries lying on the trellis. Each point on the trellis contains a probability associated with a strong point on local boundaries. The Viterbi algorithm finds a sequence of nodes which maximizes the overall hypothesis under the first order Markovian assumption [23]. Figure 2.1 presents the demonstration of a sparse trellis on a sample image.

Finally, a binary mask is created over the coordinates between each of the two consecutive curves and specifies the pixels in between to the region of interest. The binary mask is used in next step as input for the next step in energy function.

2.2.4 Energy Function

After doing a Viterbi search and estimating the initial mask, we delve into the main procedure for active contour evolution. We used a Chan-Vese-based [19] energy-minimizing active contour without edges model. Consider an evolving curve represented by ϕ, and c_1 and c_2 as constant average intensities of inside and outside of ϕ in the image I. The energy function is defined as:

$$E(\phi, c_1, c_2) = \mu \int_\Omega \delta(\phi)|\nabla\phi| dxdy + \lambda_1 \int_\Omega H(\phi)|I - c_1|^2 dxdy + \lambda_2 \int_\Omega (1 - H(\phi))|I - c_2|^2 dxdy + \lambda_b \int_\Omega (D(\phi))|\nabla I(\phi)| dxdy \tag{2.3}$$

where $H(\phi)$ is a regularized Heaviside function and μ, λ_1, λ_2 and λ_b are positive weighting parameters. These parameters are set using an adaptive approach as follows:

$$\lambda_{1,2}(n) = \lambda_{1,2}(1) + \frac{n[\lambda_{1,2}(N) - \lambda_{1,2}(1)]}{N} \tag{2.4}$$

$$\lambda_e(n) = \lambda_e(1) + \frac{\lambda_e(N) - \lambda_e(1)}{\cosh[8(\frac{n}{N} - 1)]} \tag{2.5}$$

In the primary iterations, higher values are assigned to the region term in order to let the curve evolve towards boundaries, whereas the boundary term weighting is increased as the algorithm approaches to the final iterations and helps the curve to converge to the layer edges precisely [24, 25]. Figure 2.2 presents a plot of the weightings for 100 iterations.

The first term in the energy function ensures smooth boundaries while the last two terms fit the level set to the image pixel data averaged over all RGB channels. The energy function is minimized using an iterative two-step algorithm in which c_1, c_2 are first computed while ϕ is considered fixed, then ϕ is updated.

The method will localize intra-retinal boundaries for eight retinal layers, i.e., retinal nerve fiber layer and ganglion cell layer (RNFL+GCL), inner nuclear layer (INL), outer plexiform layer (OPL), outer nuclear layer (ONL), external limiting membrane and inner segment (ELM+IS), ellipsoid zone (EZ), outer segment (OS), and retinal pigmented epithelium and interdigitation zone (RPE+IZ).

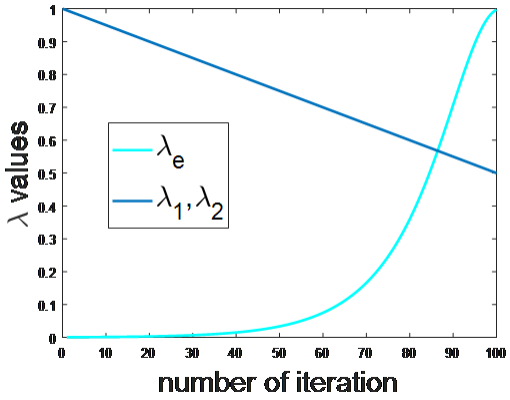


Figure 2.2: Plot of weightings for 100 iterations.

2.3 Evaluation

By intersecting a segmented image with the corresponding ground truth, which is determined by a clinician, four outcome relative states, i.e., True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN), were obtained. Figure 2.3 shows a sample demonstration of the four regions obtained after the intersecting a segmented image with its corresponding ground truth.

To evaluate the performance of the proposed methods, based on the aforementioned regions we created several indexes, namely Accuracy, Precision, Sensitivity, and Specificity which are defined by:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Specificity = \frac{TN}{TN + FP}$$

$$Sensitivity = \frac{TP}{TP + FN}$$

in which TP is the true positive, TN is the true negative, FN is the false negative and FP is the false positive.

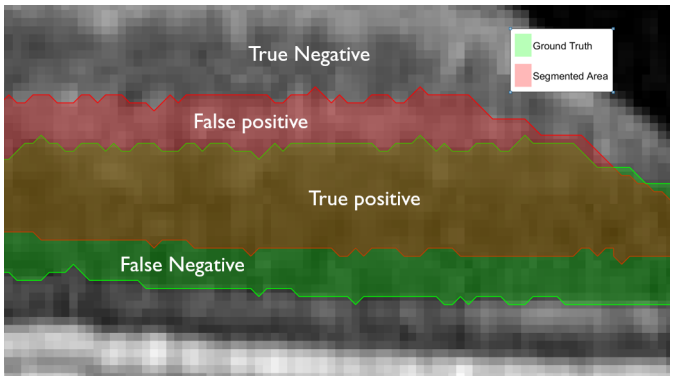


Figure 2.3: A demonstration of ground truth and segmented region relative states. The red curve corresponds to a segmented region, while the green curve represents the corresponding ground truth. True Positive region which is the intersection of the two curves is shown in yellow.

In addition, we also use the Jaccard index (JI) [26], and the Dice similarity coefficient (DS)[27]. The results were compared to the ground truth delineations, determined by an optometrist through a combination of manual and semi-automatic segmentation procedure, in which the Livewire algorithm [28] (vide infra) is used as a semi-automatic segmentation with supervision and B-spline for fully manual segmentation.

2.3.1 Ground truth delineation

2.3.1.1 Semi-automatic segmentation

In order to facilitate the process of determining ground truth locations for optometrists, we propose a semi-automatic method.

Livewire algorithm

The Livewire algorithm, also known as Intelligent Scissors [29], is a powerful semi-automatic method which enables users to discard the outlier edges. In this method, first a seed point

is chosen on a wound border. The objective is to determine a path from the seed to the mouse controlled pointer that follows the object boundaries. When the user moves the pointer, the boundaries of the object of interest are extracted. The main local cost function is defined as:

$$l(p,q) = w_Z f_Z(q) + w_D f_D(p,q) + w_G f_G(q) \tag{2.6}$$

where f_Z, f_D and f_G are Laplacian zero-crossings, gradient direction and gradient magnitude cost terms, respectively, representing different aspects of edge features. The weights of the corresponding features were chosen empirically as follows: $w_Z = 0.2, w_D = 0.3, w_G = 0.8$. The image edges were obtained using a Canny edge detector. The minimum cost function path was obtained using the Dijkstra algorithm [29].

Thus, using the livewire algorithm, the user clicks on the boundary of each layer and chooses a seed point. Afterward, as he/she moves the cursor, the software follows the boundary of each layer. By choosing a few seed points and moving the cursor, all of the layer boundary will be extracted. Figure 2.4 shows the interface of the GUI designed for the semi-automatic segmentation. The algorithm returns the binary mask for each of the layers as output.

The users had the option to switch between the manual and semi-automatic methods. Using this method, the average time for localizing the boundary of each layer was reduced from 47 seconds to 18 seconds, as measured in MATLAB 2017a (Microsoft Windows 10, Intel Core i7, CPU 3.41 GHz, 16.00 GB of RAM) by the average time consumed by a clinician user over 25 normal OCT images.

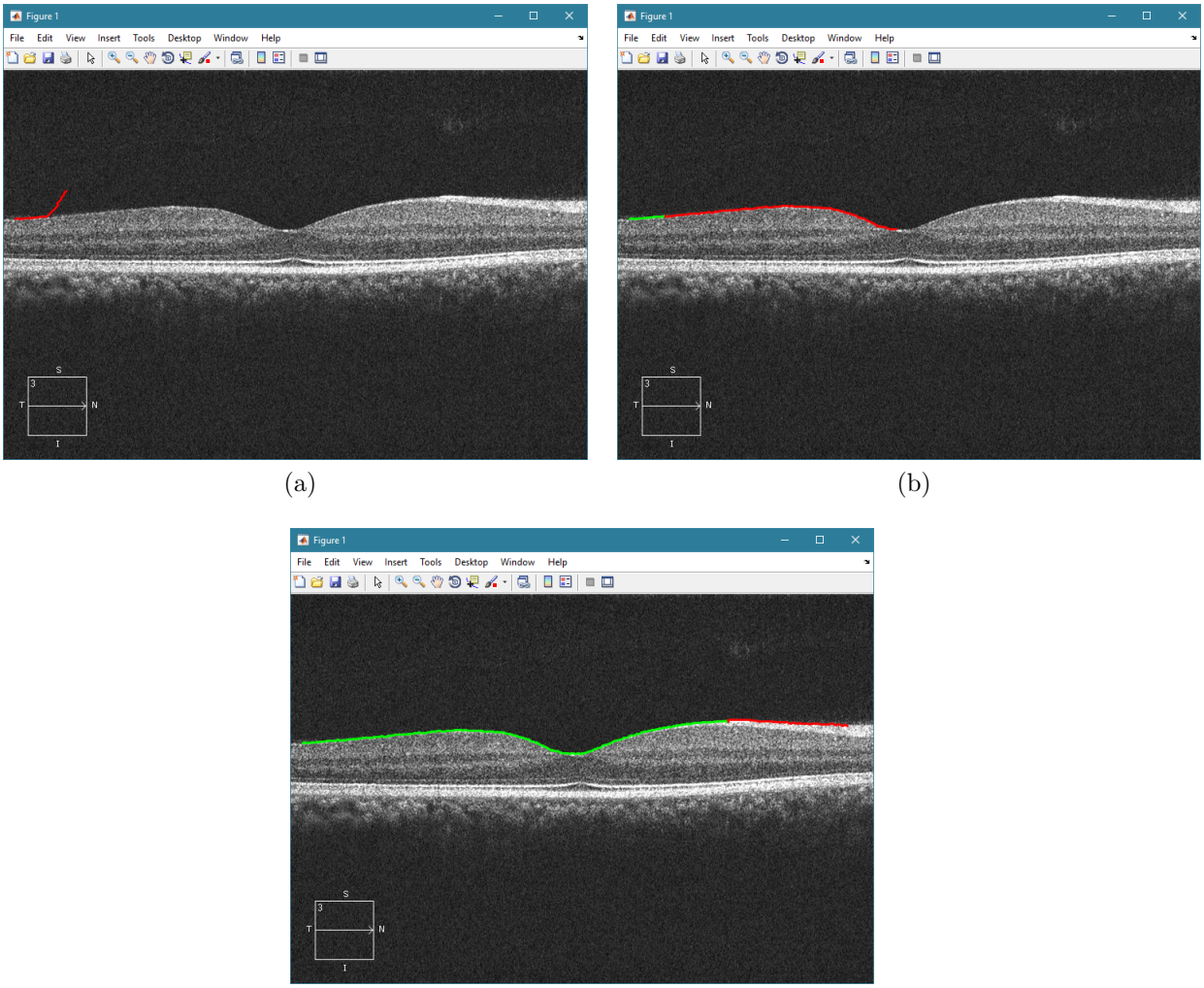


Figure 2.4: semi-automatic segmentation of OCT images for ground truth delineation

2.4 Results

Our method localizes intra-retinal boundaries for eight retinal layers. Figure 2.5 demonstrates the segmentation results on a sample image. Table 2.1 presents the results of different performance measures. We report the results for 25 SD-OCT images, after ap-

plying the denoising procedure, using MATLAB 2017a, and tested on a personal computer (Microsoft Windows 10, Intel Core i7, CPU 3.41 GHz, 16.00 GB of RAM).

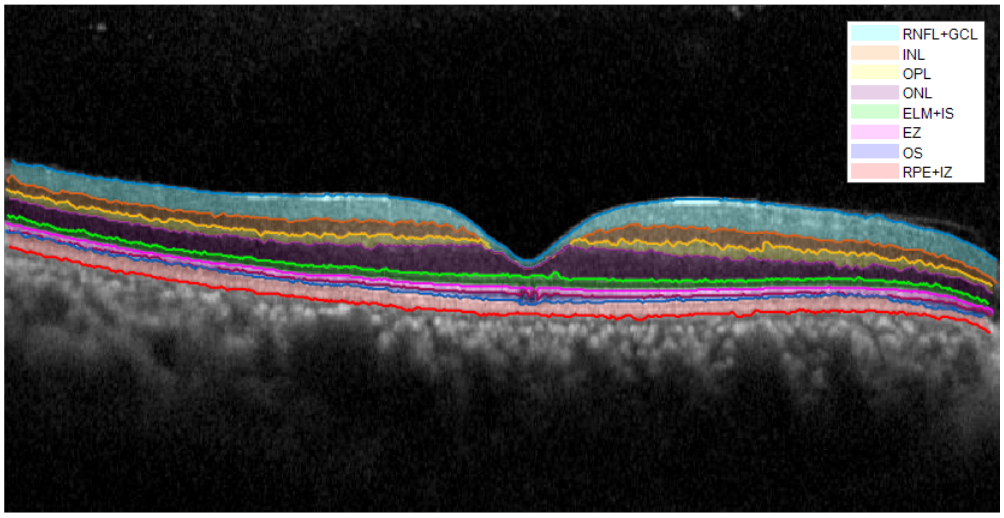


Figure 2.5: Segmentation results of the proposed method on a sample SD-OCT image.

2.5 Discussion

As discussed in 2.1, retinal OCT images contain distinct edges as well as homogeneous regions. Our method uses a dynamic programming approach for initialization and an adaptive weighting strategy to use both region and edge information in the image. Using such strategy is critical in OCT image segmentation because the presence of the speckle noise does not let us to use only gradient information on the image.

One of the major problems of active contours which only rely on the image gradient, e.g., geodesic active contours is being trapped in the local minima, i.e., converge to the local minimum point of energy functional [30]. Using the Chan-Vese model's basis in the

		Retinal layers			
		RNFL +GCL	**INL**	**OPL**	**ONL**
Performance measures	**Accuracy**	97.45±0.82	93.52±0.90	92.41±1.38	95.81±0.93
	Sensitivity	96.95±0.74	93.26±1.01	94.12±1.29	90.91±1.22
	Specificity	97.69±0.99	93.64±0.94	91.74±1.08	97.11±1.13
	Precision	95.13±1.12	86.97±0.69	81.63±1.42	89.29±0.81
	Jaccard index	92.36±1.03	81.82±0.74	77.67±2.01	81.96±0.96
	Dice similarity	96.03±0.99	90.00±0.72	87.43±1.54	90.09±0.82

		Retinal layers				Average
		ELM +IS	**EZ**	**OS**	**RPE +IZ**	
Performance measures	**Accuracy**	96.98±0.90	96.03±0.98	93.12±1.25	97.30±0.53	95.29±0.83
	Sensitivity	96.15±0.88	93.75±0.75	78.95±2.61	98.21±0.55	92.78±0.94
	Specificity	97.40±0.64	96.61±1.10	95.69±1.95	97.01±0.69	95.86±0.67
	Precision	94.94±0.94	87.38±1.20	76.92±2.78	91.24±0.61	87.93±1.05
	Jaccard index	91.46±1.13	82.56±0.89	63.82±3.09	89.74±0.70	82.67±1.22
	Dice similarity	95.54±1.09	90.45±0.77	77.92±2.06	94.59±0.88	90.25±0.99

Table 2.1: Performance measures of the segmentation method (The reported values are mean ± standard deviation)

formulation of the proposed method is very important for preventing contours from being trapped in a local minimum. Furthermore, using this model in the early iterations helps the algorithm to be less sensitive to initialization.

While the proposed method is consistent with initialization to some extent, we used the dynamic programming initialization to make it even more efficient. Using such an initialization prevents the region-based active contour from any possible leakage to the other regions, i.e., other retinal layers.

As shown in Table 2.1, the experimental results indicate that the proposed method combined with a wavelet-based denoising algorithm, which preserves important information on the image, is able to efficiently determine the location of different retinal layers and their thicknesses. The overall accuracy of 95.29% indicates a better performance of the proposed method when compared to results found in the literature. [14, 15, 16, 17]. The proposed method obtained the best performance for the segmentation of RNFL+GCL and RPE+IZ (first and last) layers. However, due to the fact the edges are less visible and distinguished in OS layer, the algorithm has obtained a lower performance.

Chapter 3

Classification of Optical Coherence Tomography images

(Based on: ***Gholami, P., Hassani, M. S., Parthasarathy, M. K., Zelek, J. S., & Lakshminarayanan, V. (2018, March). Classification of optical coherence tomography images for diagnosing different ocular diseases. In Multimodal Biomedical Imaging XIII (Vol. 10487, p. 1048705). International Society for Optics and Photonics.***

and

Transfer-learning-based classification of Optical Coherence Tomography images, Gholami, P., Zelek, J. S., & Lakshminarayanan, V., under submission to Journal of Imaging

and

Fully automated identification of Ocular diseases using Optical Coherence Tomography images, Gholami, P., Gueddana, A., Zelek, J. S., & Lakshminarayanan, V., under submission to Medical Image Analysis Journal)

3.1 Background

As discussed in section 1.1.4, a crucial part of analyzing OCT images is classifying them in order to facilitate the detection of different ocular diseases. Generally there are two

methods for the classification of OCT diseases. The conventional method which is still widely used in clinical practice is direct diagnosis by visual inspection by clinicians. However, this method is very time consuming, demanding, and not practical for large data-sets, especially for 3D cross-sectional volumes and OCT images.

Recent studies have addressed the automated classification of OCT images by comparing some certain clinically significant indices between the normal and diseased retina. These indices are usually obtained using a direct measurement of indicators, e.g., retinal layer thickness [31, 32], or the presence or absence of a certain structure, attribute, or material in a certain disease, e.g., the presence of fluids in the retina [10]. The measurement of retinal layer thicknesses requires precise segmentation of OCT images which can be done either manually or automatically. Manual segmentation is subjective and not accurate. Automated segmentation can be done using mathematical-based model methods (e.g., graph theoretical [33], active contour [34], etc.) or machine learning-based methods, e.g., convolutional neural networks [13]. OCT image segmentation is a time consuming and difficult task and also, different types of diseases require different strategies especially in their early stages. Furthermore, while providing useful information, these quantitative measures, such as the retinal layer thicknesses, do not provide a specific information regarding the existence or non-existence of a certain disease and are not sufficient on their own. On the other hand, for some of pathological conditions, the thickness of the layers might be the same while the overall shape and morphology of the retina is changed. Few studies have addressed fully automated classification of OCT images and identification of ocular diseases based on such information. These studies have used common feature extraction and machine learning algorithms, e.g., Support Vector Machine (SVM), decision trees, etc. A complete and detailed review on ophthalmic disease diagnostic studies is presented in [18].

Koprowski et al. [35], in one of the first works on OCT image classification, extracted features using morphological and texture information and obtained 75% of accuracy. One of the texture features that has shown to be very effective and is widely used in the literature, on 2D OCT images or 3D volumes [36], is local binary pattern (LBP) features. Liu et al. [37] used LBP and multi-scale spatial pyramids. Even though they reported an average area under the ROC curve (AUC) of 93%, their method contained limitations and needed a series of alignments and flattening of the images. In a recent study, (published as Gholami et al. [38]) we used simple LBP with a sampling point of 8 and radius of 1 and an improved version of SVM and obtained 94.4% of accuracy.

In the past decade, with the introduction of new fast processors, the trend is to use raw data to rather than using some features which rely on prior knowledge and need

many pre-processing stages. This has led to the development of new techniques known as deep learning, which have revolutionized many fields including artificial intelligence, image processing, etc. However, the main assumption in such methods is that we have ample data in order to train a deep neural network on them. Lack of sufficient data will result in over-fitting, which causes a negative effect on the performance of a network [39]. These problems, along with the high computational cost, has led to the emergence of techniques known as transfer learning, in which one uses a pre-trained network in order to build a model on a relative task for which ample data are available. Karri et al. [39] used transfer learning on a pre-trained convolutional neural network (CNN), GoogLeNet, for classification of OCT images and obtained 99%, 89%, and 86% accuracy respectively for normal, AMD, and diabetic macular edema patient groups.

While presenting good results, most of these studies require manual inputs, their applications are limited, require high computational cost, and have only focused on few pathological conditions, e.g., AMD or normal. Therefore, developing a reliable and low cost platform which can be widely used in the clinic is important.

In this chapter we propose a new fully automated classification method based on OCT image textures. Also, we propose a new multi-phase image classifier and evaluate its performance by using some common feature extraction methods. We then compare the results with some of the state of the art work in OCT image classification. We will implement our algorithm on the diseases mentioned in chapter 1 along-with normal retinal OCT images.

3.2 Methodology

In this chapter, we compare four methods, including two SVM-based classifiers with texture descriptors and two convolutional neural network-based methods. We will also use the data-set on three pre-trained networks of AlexNet [40], GoogleNnet [41], and ResNet [42], and classify OCT images using transfer learning. One can split the automated ocular disease detection task into four main steps: collecting the data set, extracting features, training a classifier based on the features and testing new images with classifier.

3.2.1 Optical Coherence Tomography Image Database (OCTID)

We have created an open access OCT image database and it is available to the public at https://doi.org/10.5683/SP/UIOXXK [43]. The database contains different data-sets

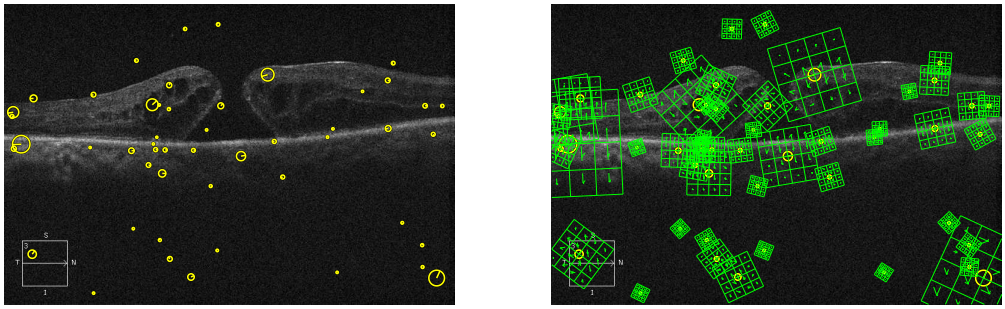
categorized into different diseases and includes jpeg images that can be downloaded as zip files. The database consists of more than 500 spectral-domain OCT volumetric scans, including four categories of Normal (NO), Macular Hole (MH), Age related Macular Degeneration (AMD), and Diabetic Retinopathy (DR). The images were from a raster scan protocol with a 2mm scan length, containing 512x1024 pixels, captured using a Cirrus HD-OCT machine (Carl Zeiss Meditec, Inc., Dublin, CA) at Sankara Nethralya (SN) eye hospital, Chennai, India. In each volumetric scan, a fovea-centered image was selected by an optometrist. The images were then resized to 500x750 pixels. Our database consisted of 102 MH, 55 AMD, 107 DR, and 206 NO retinal images. The pathologies were detected by clinicians and image class labeling was done based on the diagnosis of clinical experts at SN hospital.

3.2.2 SVM and texture descriptors

In this method, the main focus is on retinal layer textures, since different eye diseases can cause a change in the local texture of the retinal layers and can be detected using texture descriptors in OCT images. Texture descriptors, based on their application include different types and formulations. We used a texture classification method based on the "describable texture data-set" (DTD) method presented in [44]. Using some unconventional object recognition techniques, Cimpoi et al. obtained the state-of-the-art for texture analysis tasks. In this method, textures are grouped in accordance with the corresponding word in English they can be described, e.g., grid, line, paisley, etc. First, they extract key points and textures by using SIFT (scale invariant feature transform) descriptors. Then using two methods of Fisher Vectors and Deep Convolutional Activation Features, they encode those features. In [44] they show that using a simple Support Vector Machine (SVM) classifier, they can easily obtain good results. Finally, their method distinguishes between 47 texture categories. In an improved version of their work [45], they use convolutional neural networks (CNN) in order to build a deep filter bank for texture classification. In this chapter we will follow the former method, i.e., FV, since our data-set is relatively small to train a CNN.

DSIFT Features

We used Dense SIFT (DSIFT) local features which is an extension to Scale-Invariant Feature Transform (SIFT) descriptors [46]. SIFT features contain a selected image region, called SIFT key-points which can be extracted using SIFT detectors, and SIFT descriptors,



(a) (b)

Figure 3.1: Demonstration of DSIFT descriptors on a sample OCT image(a) visualization of a random selection of 50 features (b) overlaying SIFT descriptors

characterizing the appearance of those local key-points. A SIFT descriptor in a given pixel consists of a three-dimensional histogram of the gradient formed by the pixel location and the gradient orientation. SIFT descriptors have shown to be very effective in many machine learning and computer vision problems, including image classification. DSIFT [47], rather than using key-points, calculates SIFT descriptors densely over a given grid and reduces the complexity from $O(Q^2R^2)$ to $O(Q^2R)$ [48]. DSIFT descriptors have the capability of being robust among possible illumination changes, which occur commonly in the morphology of the retinal layers in different pathological conditions, e.g., DR.

Figure 3.1 presents the demonstration of DSIFT descriptors on a sample OCT image.

Fisher Vectors

A Fisher Vector (FV) [49] is a function measuring the similarity of objects which is used to encode local descriptors in an image. FVs are widely used in image classification. Once DSIFT descriptors are extracted, the FV formulation uses them through soft-quantizing the features using Gaussian Mixture Models (GMM) in order to model those descriptors and construct a visual word dictionary, which is a collection of DSIFT local image features. The GMM connects each SIFT descriptor x_i by fitting its distribution to a mode k in the mixture with the probability of q_{ik} which is defined as [49]:

$$q_{ik} = \frac{\exp[-0.5(x_i - \mu_k)^T \sum_k^{-1}(x_i - \mu_k)]}{\sum_{t=1}^{k} \exp[-0.5(x_i - \mu_t)^T \sum_k^{-1}(x_i - \mu_t)]} \tag{3.1}$$

in which μ_k, Σ_k, and π_k are the parameters of a Gaussian, and u and v are mean and co-variance deviation vectors, defined for each mode k as:

$$u_{jk} = \frac{1}{N\sqrt{\pi k}} \sum_{i=1}^{N} q_{ik} \frac{x_{ji} - \mu_{jk}}{\sigma jk} \tag{3.2}$$

$$v_{jk} = \frac{1}{N\sqrt{2\pi k}} \sum_{i=1}^{N} q_{ik} [(\frac{x_{ji} - \mu_{jk}}{\sigma_{jk}})^2 - 1] \tag{3.3}$$

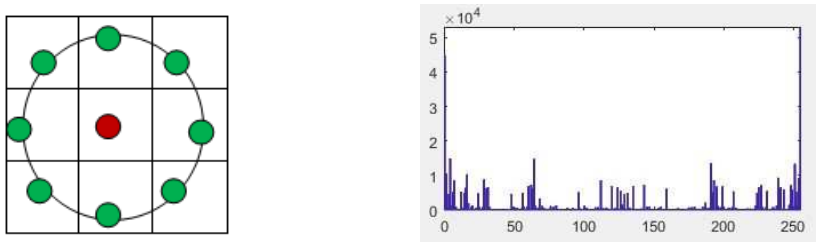
Finally, the FV is created by concatenating each of the vectors u_k and then vectors v_k for each of the modes in GMM. Since FV formulation contains GMM, they have the advantages of generative models and can handle variable data lengths. Moreover, they provide the advantages of discriminative models as they provide the ability to be used directly into a classifier as an input [50]. Following [44] we use the improved version of FV (IFV) which is presented in [51]. IFV descriptors boost the accuracy of the FV kernel and add the capability for classifying images in larger scales by using L2 normalization and a non-linear additive kernel, i.e. adding sign-square-rooting to the FV formulation [51].

SVM Classifier

SVM classifiers are simple and fast. In addition, a SVM classifier is able to use linear kernels rather than using a non-linear kernel which saves a considerable amount of computational cost. On the other hand, IFV creates a vector which can be used directly as a linear kernel in a SVM classifier. Therefore, we used a SVM to determine the feature vector describing the classes of the image, i.e., NO, DR, MH, or AMD.

3.2.3 MSVM and Local Binary Pattern

This method is based on the work presented in [38] with the same data-set. It uses Local Binary Pattern (LBP) as features and then, using a series of methods, including pre-processing, Meta learning, and active learning, improves the performance of the classifier.



(a) (b)

Figure 3.2: Local binary pattern. (a) A 3x3 neighborhood used to define texture and calculate LBP (b) a sample LBP histogram for a normal OCT image. X axis shows the LBP histogram bins and Y axis shows the corresponding count of each bin

3.2.3.1 Local Binary Pattern (LBP) Features

LBP is a non-parametric texture descriptor which encodes the local information in an image based on the relationship between adjacent pixels. Because of the high discriminative capability the LBP is one of the most common techniques used for feature extraction in image classification, especially in medical image analysis problems. Using a thresholding, LBP encodes the relationship between neighboring pixels into a set of binary values [52].

Following [52] we used a rotation-invariant descriptor as:

$$LBP_{P,R} = \sum_{i=1}^{p-1} S(g_i - g_0)2^i \quad , \quad S(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{3.4}$$

where g_0 is the intensity of a given center pixel and g_i the intensity of a given neighbor of g_0 , P is the sampling point and R is the radius.

The LBP descriptor compares the center pixel g_0 in a 3x3 neighborhood with its eight neighbors $g_1 - g_8$ and formulates an eight bit binary code. A single feature vector is then constructed through concatenating the histogram (from 0-255) of the obtained binary codes circularly. Figure 3.2a shows a sample pixel with its 8 neighbors and Figure 3.2b demonstrates a sample LBP histogram obtained from a normal OCT image.

3.2.3.2 Multi-phase Support Vector Machine (MSVM)

We introduce a multi-phase approach for designing the classifier. The method is based on the work published in [38]. We then compare the results of the MSVM with a simple SVM, and a random forest classifier using both of the LBP features and texture descriptors.

Data pre-processing and feature selection

As the first step, missing values in the data-set were removed. These could be due to any of the images not having one of the features recorded. For this purpose, we cleansed the numeric data from values that were too small or too big by marking blank and zero values as missing and then we applied a filter to replace the missing values. This filter collects all the values flagged as missing and replaces them with the mean or median value of that feature.

After the data were pre-processed, we reduced the number of features used in classification by omitting the less informative ones in an effort to lower computation time. Amongst the different feature selection methods, we used Correlation-based Feature Subset Selection [53]. This method looks at the individual predictive ability of each attribute, and chooses the attributes that are most informative. It also evaluates the degree of redundancy between each of the values. This selection method results in features with low inter-correlation and high class correlation. LBP feature extraction resulted in 375,000 features for each image, which was too large to be handled at once using this algorithm. Therefore, the features were split into 20 groups, and Correlation-based Feature Subset Selection was performed on each group separately. The features obtained from each group were then added into a new group of features, and the algorithm was run once again. This procedure resulted in a final feature set consisting of 16,383 of the most informative features.

Structure of the classifier

We used a random forest classifier for this experiment. A random forest classifier randomly selects a subset of the data and features, and trains individual decision trees and repeats this process. The individual trees are trained on the training data and the result of the classifier will be the class predicted by the majority of the trees for each specific test data. The main parameters of a random forest classifier are the number of trees, number

of features in each tree and the depth of each tree. We chose these parameters through experimental testing to give optimal performance. In terms of testing and training data, because of the limited number of images we had, we used 70% of the data for training the classifier, 15% for validation and 15% for testing.

Setting up the random forest classifier

By nature, random forests are several individual decision trees. Therefore, we experimented to obtain the optimal parameters which result in high precision for a strong performing single individual decision tree. All the parameters of a tree were kept constant except one that was changed, in order to obtain the optimal value for that parameter; this was repeated for the other parameters. We first chose the percentage of data to use for each tree. We tested using 70%, 80%, 90% and 100% of the training data on a single decision tree using the default values for the other parameters. Considering the limited number of images, we expected the results to show the necessity for a higher percentage of the data in each tree.

To find the maximum tree depth, we limited the maximum depth by incrementally increasing by 100 each time. No significant difference was observed in the result of the decision tree for these values and since it had little effect on computation time, we decided not to set a maximum depth for each tree. The final parameter was the number of features per tree. As commonly done in random forest classification, we started with the square root of number of features. This number of features however proved extremely time inefficient. Therefore, the number of features were reduced which essentially results in a trade-off between the performance and computation time. Finally, through these computational experiments we set 900 features for each tree. Determining the number of trees in the classifier required a different experimental procedure and needed testing on random forests rather than individual trees. We designed different random forests using previously obtained optimal parameters, and changed the number of trees used in the random forest each time. We tested with different number of trees ranging from 50 to 450, increment by 50 per test, and 250 trees was our optimal value. So overall in our classifier, from the 90% training data, 100% was used for each decision tree, with no maximum depth set for each tree, 900 randomly selected features in each tree and a total of 250 trees in the random forest. The approaches for meta-learning and active learning were then tested to try and improve the classifier.

Meta-learning

Random forest classifiers are classifiers based on decision trees, and can be considered as a meta-learning approach. Considering this fact, we did not expect significant changes in our results after meta-learning. We combined a K Nearest Neighbors (KNN) classifier with our classifier for meta-learning. The KNN classifier was tested three times with $K = 1, 3$ and 5. There was no increase in performance and we saw a decrease when K was set to 5, therefore we decided to use a different Meta-learning approach. This time we combined the random forest classifier we previously obtained with a Support vector machine. For the SVM, a linear kernel was used with the degree set to 3, cache size was set to 40 MB, the entire training set was used, and the data were not normalized. The tolerance of the termination criterion was set to 0.001 and gamma, i.e., the free parameter of the Gaussian radial basis function, was set to $1/max\,index$. We set the result of the Meta-learning approach to be a vote of the average of the probabilities of each of the separate classifiers.

Active learning

In order to further improve the performance of the classifier and make it more robust we aimed to use active learning to strengthen our decision boundary separating the different classes. Doing so, we first looked for the misclassified points from our classifier and obtained a ranking of the strength of prediction, thereby determining the misclassification error. In the first step we ran the data on our model to obtain a prediction in classification. Since the classifier showed high accuracy, we continued assuming that the ranking is satisfactory. Based on the obtained ranking, we chose 7 of the weakest classified images, having the lowest score in accuracy. This means these images would be on the decision boundary, i.e., they were just slightly allocated to one of the classes, and thus represent the hardest images to correctly classify. We checked the true class of these images, and since the model was not able to confidently classify those images by itself, by adding them to the data-set and retraining our model, we hypothesized it would learn that weakness and would classify more of those type of images correctly in the future.

3.2.4 Transfer learning

As mentioned in section 3.1, transfer learning is a method commonly used in the context of deep learning and image classification. In transfer learning, we use a pre-trained network to build a model on a relevant task for which ample data are available and then based on

our data-set (OCTID) model parameters are fine-tuned for the target task (OCT image classification). While several pre-trained networks are available, in this section, we will use three of the most well-known, i.e., AlexNet [40], GoogLeNet [41], and ResNet [42].

AlexNet

The structure of the layers in AlexNet is relatively simpler than most of the new sophisticated neural network architectures. AlexNet [40] contains 5 convolutional layers, maximum-pooling layers, dropout layers, and three fully connected layers. It is trained over 15 million images and has the ability to classify up to 1000 possible categories.

Since the image input layer in AlexNet requires a certain size for the images, all of the images were resized to 277x277x3. We then fine-tuned the last three layers,which were already configured for 1000 categories, for our four targeted classes. We replace the last three layers with a fully connected layer with the same size as the number of classes, i.e., four, a softmax layer and a classification output layer respectively. We set the initial learning rate for transferred layer to a small value near zero in order to slow down learning in the transferred layers and increased the learning rate factors (k=20) for the fully connected layer to speed up learning in the new final layers. Therefore, the combination of learning rates will result in fast learning only in the new layers and slower learning in the other layers.

GoogLeNet

GoogLeNet [41] contains 22 layers and uses more than 12 times fewer parameters than Alexnet. Therefore, it provides several benefits and advantages over many other available networks, e.g., being much faster, and reducing the chance of overfitting. Due to these characteristics, GoogLeNet is considered to be the state-of-the-art performance in the field of transfer learning.

For the fine-tuning of the GoogLeNet, we modified the last three layers, which contain information regarding how to combine the features that the network extracts into class probabilities and labels. The image input layer in GoogLeNet requires input images of size 224x224x3. After resizing the images in the data-set (same as in previous method) we replace the final layers in GoogLeNet which include 'loss3-classifier', 'prob', and 'output' layers. We also froze the initial parameters for the transferred layers so that it would not update the parameters of the frozen layers in order to speed up the computation and

prevent over-fitting.

ResNet

The Residual Neural Network, known as ResNet [42] introduced in 2015 and is the state-of-the-art in the field of deep learning. ResNet is known for its high representational ability and unlike other conventional deep neural networks makes it possible to train up to thousands of layers while the performance of the network is still preserved. He et al. [42] proposed a new method to tackle the vanishing gradient problem (which is nothing but the back-propagated gradient to earlier layers) in order to ease stacking deep layers together. Doing so, they used the so-called "identity shortcut connection" method which skips one or more layers.

We used ResNet-101 pre-trained model which has been trained on more than a million images and contains 347 layers in total, corresponding to a 101 layer residual network, and can classify images into 1000 object categories. Same as GoogLeNet, the image input layer requires 224x224x3. Therefore, all of the images are resized to be compatible with the network. In order to fine-tune the network, we extract three layers of 'fc1000', 'prob', and 'ClassificationLayer_predictions', and connect to the 'pool5' layer. Similar to the previous methods, for the purpose of speeding up the network and preventing from over-fitting, we freeze the initial parameters for the transferred layers.

Effect of Denoising

As discussed in 2.2.2, OCT images contain a large amount of speckle noise. This noise reduces the resolution of OCT images and subsequently affects the task of classifying images. Following the method in section 2.2.2, we used a wavelet decomposition denoising method using a biorthogonal spline wavelet (Bior 3.7 from MATLAB) with 17 levels of decomposition. The denoising threshold parameters were obtained using Birgé-Massart method [54]. Since the input layers in all of the images require an 3 dimensional (RGB) input image and the filtered images are gray-scale, an $a \times b \times 3$ images was created for each of the images by concatenating the image three times to itself, where a and b are the dimensions of the gray-scale image.

3.3 Results

3.3.1 Experimental Setup

For the two feature extraction methods, we divided the data into three sets of training (70%), testing (15%), and validation (15%). Also, in all of the transfer learning methods, i.e., AlexNet, GoogLeNet, and RestNet, we used 70% of the images in the data-set for training and 30% for validation. For LBP features, following [37] P, the sampling point and R, the radius were set to 1 and 8 respectively, which is the most common set of adjustment for LBPs.For both AlexNet and GoogLeNet, the mini-batch size was set to 10 and a maximum of 6 epochs was uses. A frequency of 3 was used for the validation on the entire training set. Following [44], DSIFT descriptors were sampled every 2 pixels, at scales of $2^{\frac{i}{3}}(i = 0; 1; 2...)$, and the spatial bins were set to have an extent of 6x6 pixels. Also, for FV, we used 256 visual words and 60 PCA dimensions [44]. All of the methods were implemented using MATLAB 2018a and tested on a personal computer (Microsoft Windows 10, Intel Core i7, CPU 3.41 GHz, 16.00 GB of RAM). The formulation of DSIFT and FV were implemented using the functions provided in the VLfeat toolbox [47].

3.3.2 Classification Results

Table 3.1 summarizes the accuracy, precision, and recall (sensitivity) for each of the methods. We also present the classification accuracy for each of the individual classes using different proposed methods.

		Texture (with svm)	MSVM (with lbp)	AlexNet	GoogLeNet	ResNet
Accuracy	NO	99.79	77.42	99.29	98.34	**100**
	DR	**98.30**	77.42	92.20	91.49	94.33
	AMD	**99.79**	98.72	98.58	94.33	99.29
	MH	**98.30**	75.00	92.91	93.62	95.04
	Total	**99.12**	79.30	96.21	93.84	97.55
Precision	NO	98.36	69.67	98.41	93.65	**100**
	DR	79.31	53.34	75.61	79.41	**92.86**
	AMD	**100**	95.01	**100**	68.18	94.12
	MH	93.94	35.76	95.65	**100**	85.29
	Total	93.26	63.13	92.81	88.81	**94.49**
Recall	NO	100	100	100	95.16	100
	DR	92.00	27.60	**96.88**	84.38	81.25
	AMD	94.74	95.03	87.50	93.75	**100**
	MH	83.78	17.98	70.97	70.97	**93.55**
	Total	94.04	68.17	91.53	87.29	**94.33**
Specificity	NO	99.76	53.33	98.73	94.94	**100**
	DR	**98.65**	92.63	90.83	93.58	98.17
	AMD	100	98.72	100	94.40	99.20
	MH	99.54	91.00	99.09	**100**	95.45
	Total	**99.49**	75.77	97.16	95.66	98.50
F-measure	NO	99.17	82.21	99.20	94.40	**100**
	DR	85.19	36.40	84.93	81.82	**86.67**
	AMD	**97.30**	95.01	93.33	78.95	96.97
	MH	88.57	23.83	81.48	83.02	**89.23**
	Total	93.47	62.97	91.42	87.26	**94.27**

Table 3.1: Classification performance percentage for each of the classes using different proposed methods

3.3.3 Performance of the MSVM classifier

In MSVM, before applying the active learning and meta-learning, as expected, the results showed a significant increase in performance each time we increased the percentage of data in use, prompting us to use the entire test set on each tree of the forest, since the changes

in computation time varied in the range of a few seconds. The time for 70%, 80%, 90% and full data-set were 22, 24, 28 and 32 seconds respectively (Microsoft Windows 10, Intel Core i7, CPU 2.8 GHz, 8.00 GB of RAM).

Applying the active learning method and retraining our model in this manner led to an increase in precision by 1.1%. Table 3.2 compares the performance of MSVM with simple SVM, and a random forest classifiers using LBP and texture features.

Classifier	feature	AMRD	DR	MH	NORMAL	Average
SVM	LBP	91.02	29.622	22.75	69.63	55.17
Random Forest		94.95	45.98	38.52	89.50	71.19
MSVM		95.02	47.55	39.14	90.11	71.98
SVM	Texture	73.97	66.54	71.89	94.10	81.12
Random Forest		57.21	57.56	66.10	97.68	77.41
MSVM		93.10	87.52	83.78	97.94	92.22

Table 3.2: Percentage of the area under the ROC curve for SVM, random forest, and MSVM classifiers using LBP and texture features

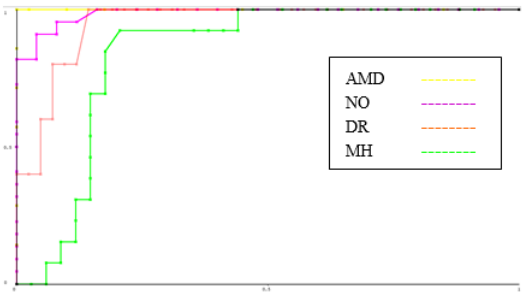
Figure shows ROC curves for each of the proposed algorithms and for each class separately.

3.3.4 Transfer learning progress results

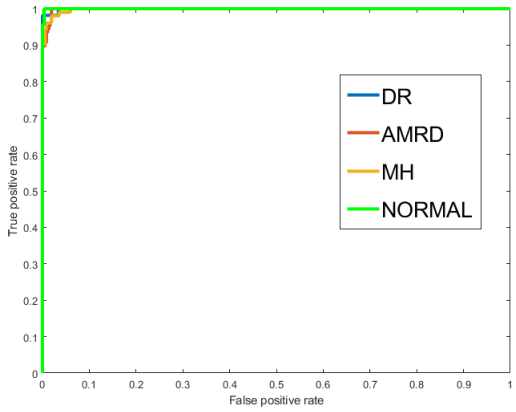
Figure 3.4 displays the training progress including accuracy and percentage of loss for AlexNet, GoogLeNet, and ResNet.

3.3.5 Transfer learning data augmentation

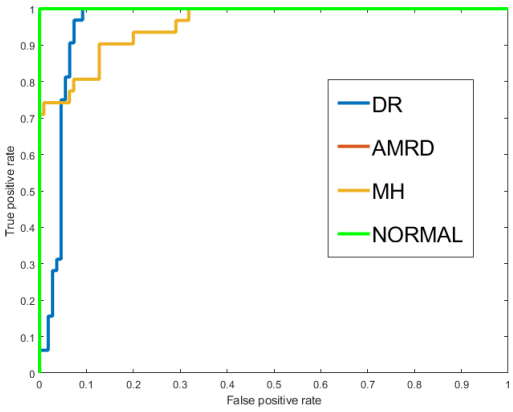
We also performed a data augmentation on the data-set in order to prevent the network from over-fitting and memorizing the exact details of the training images. Data augmentation is a common method used in deep learning where the amount of the number of data (images) is limited and obtaining more images is not possible. Data augmentation is established on the premise that convolutional neural networks can be invariant to transformations such as translation, flipping, size, etc. In this study, the data augmentation was



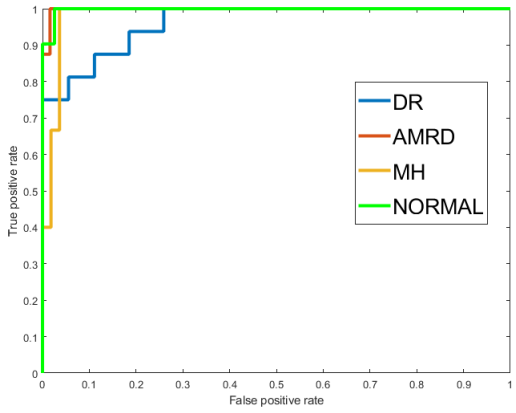
(a)



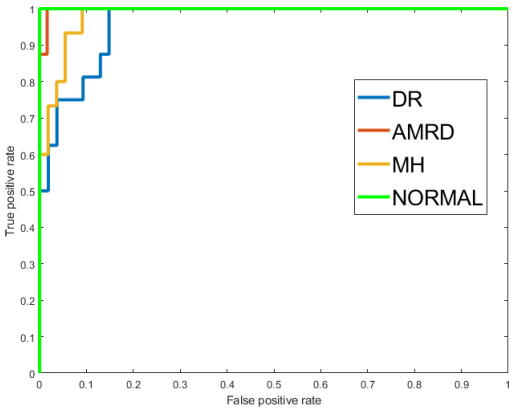
(b)



(c)

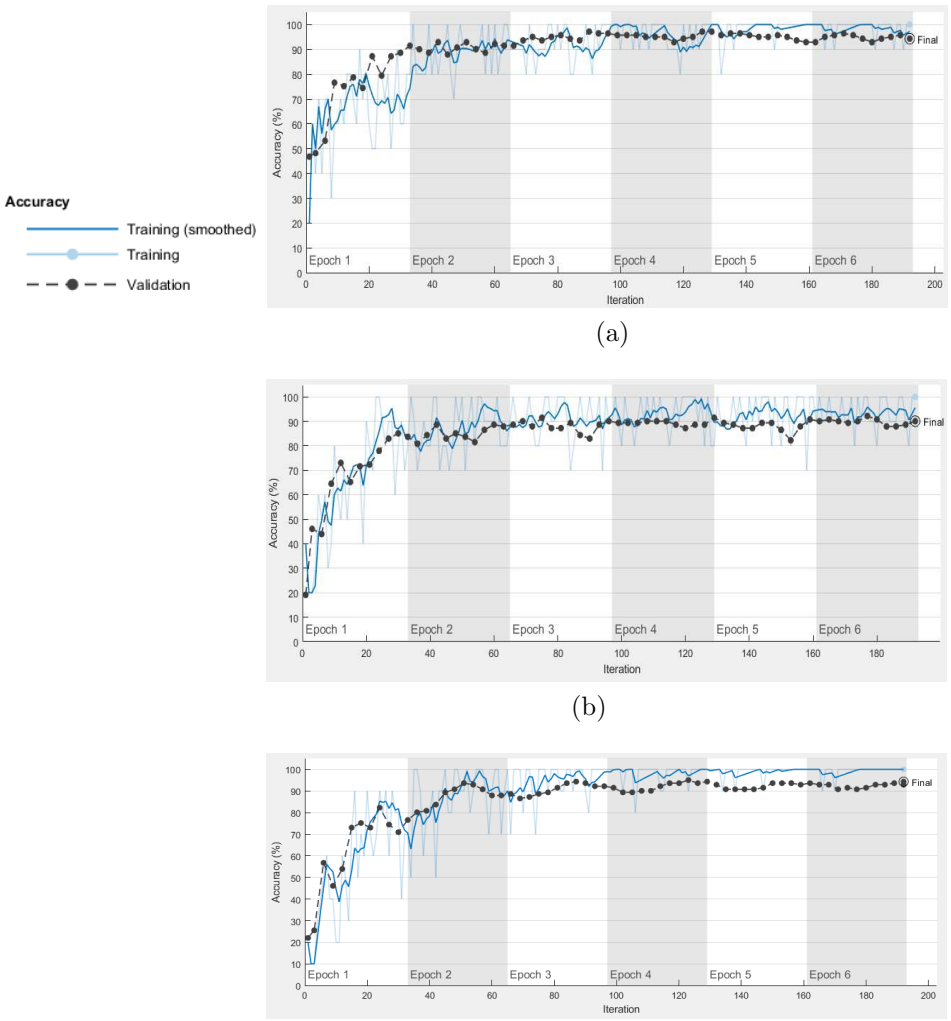


(d)



(e)

Figure 3.3: ROC curves for each of the proposed methods. Each figure demonstrate the ROC curves for each class separately. (a) LBP with MSVM (b) Texture with SVM (c) AlexNet (d) GoogleNet (e) ResNet.



(a)

(b)

(c)

Figure 3.4: Training progress for the transfer learning methods. (a) AlexNet accuracy (b) GoogLeNet accuracy (c) RestNet accuracy

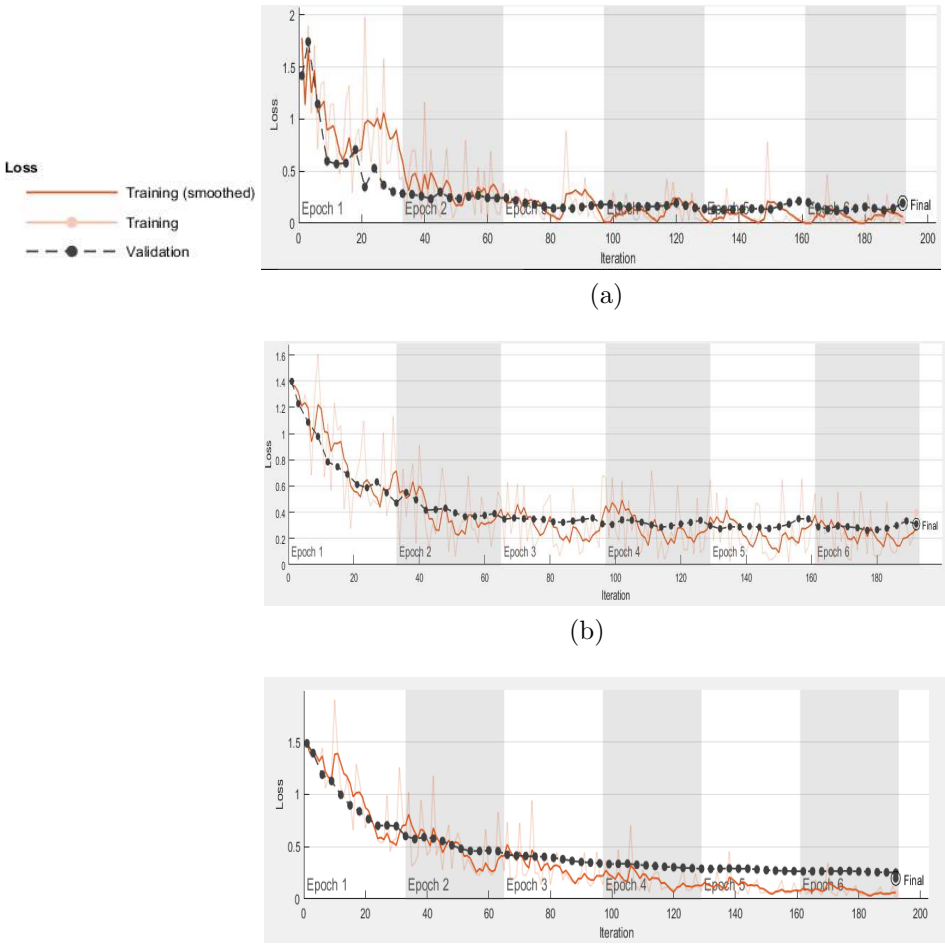


Figure 3.5: Training Loss for the transfer learning methods. (a) AlexNet loss (b) GoogLeNet loss (c) ResNet loss

done though randomly flipping the training images along the vertical axis, and randomly translate them up to 30 pixels horizontally and vertically.

Network	Accuracy	Precision	Recall	Specificity	f-measure
AlexNet	-2.37	-4.46	-3.75	-3.65	-4.35
GoogLeNet	-0.49	-2.25	-2.34	-1.05	-3.08
ResNet	-0.33	-1.54	-1.45	-0.20	-1.39

Table 3.3: Percentage of the change in the overall accuracy, precision, and recall after applying the augmentation for AlexNet, GoogLeNet, and ResNet

3.3.6 Effect of denoising

In this section we study the effect of denoising on the proposed image classification technique by implementing them on denoised OCT images. Table 3.4 shows the difference in performance indices after the wavelet decomposition denoising.

Classifier	Accuracy	Precision	Recall	Specificity	f-measure
SVM with texture	0.03	-0.05	-0.62	0.05	-0.16
MSVM with LBP	5.29	6.21	4.20	2.21	6.89
ResNet	-1.33	-2.52	-2.79	-0.86	-2.80
AlexNet	0.86	0.63	1.31	0.44	1.26
GoogLeNet	1.58	1.20	2.70	0.19	2.51

Table 3.4: Percentage of the change in the overall accuracy, precision, and recall after applying the denoising method

3.3.7 Computational cost

Table 3.5 shows the average computational cost for each of the methods. All methods were implemented in MATLAB 2017a (Microsoft Windows 10, Intel Core i7, Single CPU 3.41 GHz, 16.00 GB of RAM).

Method	Time spent
LBP with MSVM	2 min 34 sec
Texture with SVM	24 min 19 sec
AlexNet	4 min 6 sec
GoogleNet	6 min 26 sec
ResNet	31 min 39 sec

Table 3.5: Computational cost for each of the proposed methods

3.4 Discussion

The MSVM method was able to improve the performance both on the average and on each of the classes separately compared to the single classifiers. Using the method in 3.2.3.2, we were able to train the random forest classifier to yield a relatively good result. However, the addition of the SVM, which on its own performed inefficiently, increased the average area under ROC curve by 2%. The classifier was run once again, and this time 7 of the points which were weakly classified were added to the training set through active learning. This resulted in another increase in average performance by 1.1%, yielding a final average area under ROC curve of 92.2%. The different results for classification of different diseases can be due to the variety of the images in a certain class, i.e., the overall texture of an OCT image for a certain disease, is changed based on the different stages of that disease. As Table 3.2 indicates, using both of the features, the MSVM classifier obtained a better performance. Using texture features resulted in a 20.24% of increase in the performance of the MSVM classifier which indicates the superiority of this feature extraction method than LBP.

As Table 3.3 shows, after applying the data augmentation all of the performance measures are reduced in both of AlexNet and GoogLeNet. This can be due to the fact that data augmentation in OCT images require more sophisticated techniques. In other words, in a diseased condition, flipping the OCT images might produce the same features that one can find in another type of disease.

Based on the results of the Table 3.1 all of the proposed methods have obtained good results compared to other methods in the literature. While transfer learning methods have obtained higher accuracy, in the overall precision and recall, the proposed texture method obtained comparable and in some performance measures better results. LBP feature extraction showed a comparatively lower performance than the other methods, suggesting that while being low on computation cost, LBP is not the best feature extraction

method. Therefore, we decided to use a more efficient feature extraction technique and obtained much higher performance.

In Table 3.4, after implementing the denoising technique, AlexNet and GoogleNet obtained slightly higher performances, as expected. Res-Net, unlike the other two networks, obtained a slightly lower performance after denosing. This fact suggests that Res-Net is more sensitive to the local structures and resolution in the image. Furthermore, the result for the texture feature also did not change after applying the denoising and this classifier appeared to be insensitive noise. However, MSVM with LBP was the only classifier that obtained a much higher performance after applying the denoising technique. This result indicates that LBP features are greatly sensitive to speckle noise and in order to efficiently use them for classification we need to pre-process the images and apply a robust denoising method.

Chapter 4

Conclusion and future directions

Future studies will implement the proposed method on more comprehensive data sets with different diseases in order to obtain classification indices for different ocular diseases. The comparison between the performance and time consumption of the proposed method and the classic active contour without edges will be also studied later. Moreover, one of the other objectives will be calculating the average retinal layer thickness of the proposed method and the manual segmentation.

Creating the open source OCT image database (OCTID) 3.2.1 provides several benefits and applications. This is one the largest OCT databases that provides a comprehensive collection of high resolution images and is categorized based on different pathologies. The ultimate goal to create this database was to facilitate the public access to OCT images in order to help researchers world-wide who want to develop computer aided algorithms for the analysis of OCT images and might not have access to sufficient and good quality data. Therefore, by removing such a barrier, we hope that more efficient techniques can be created and aid in accurate diagnosis. Furthermore, since this database is comprehensive and well organized, it can be considered as a gold-standard.

In addition, we have also received a huge ocular database from UK Bio-bank, which contains more than 100,000 images . One of the future directions is to decipher this database and extract the images in order to create a bigger and more comprehensive set of OCT images. Therefore, by establishing such a huge database, we can build our novel neural networks and train it using these images. It is expected that by using such a large database, the performance of the neural network will increase dramatically and we can create a highly reliable identification technique for ocular diseases.

The semi-automatic segmentation has received a positive feedback from clinicians and also resulted in reducing a significant amount of time for manual segmentation. We will further work on building user friendly software for clinicians which contains both of the methods and gives them the freedom to choose between fully manual, semi-automatic, and fully automated segmentation that can be used for clinical diagnoses.

As discussed in section 2.1 the adaptive weighting strategy enables us to use both region information and image boundaries for segmentation. In many images and videos, there are various conditions, thus it is necessary to use both of these types of information; the presence of noise does not permit using image gradients alone. One of the future works will be on studying the performance of the proposed method on non-medical data, such as hand written numerals, etc. It is expected that using the proposed method will result in achieving a comparable result much faster and with less computational cost, along with the consistency across noisy images.

The formulation of the active contour makes it possible to add several other priors for different purposes. As discussed in 2.2.2, speckle noise is a major challenge in the analysis of OCT images. A possible way to address the problem of speckle denoising is to consider its effect while doing the segmentation and add it in the formulation of the active contour. Wang et al., [55] showed that in ultrasound images, speckle noise follows a Rayleigh distribution and added it to the formulation of active contour. OCT images are formed in a similar way to ultrasound images, therefore the presence of the speckle noise would also be expected to be similar to some extent (if we neglect some differences, e.g., the effect of using coherent laser light). Therefore, in future we will investigate statistical distributions of speckle noise and use the same strategy for OCT image noise removal. Specifically, one of the future directions is to embed the noise prior term to the formulation of the active contour and combine it with the adaptive weighting strategy.

We showed that texture features are very effective in the classification of OCT images. As discussed in chapter 3, we used FV method for modeling DSIFT features. However, instead of FV, we can use other more sophisticated feature encoding methods, e.g., deCAF [56], VLAD [57], etc. methods. One of the future directions will be studying the effect of these encoding techniques on the problem of classifying OCT images. Also, in future studies we plan to use the proposed texture method with more sophisticated and powerful classifiers and compare the results.

The MSVM classifier obtained a better performance compared to SVM and random forest using both of the feature extraction methods in chapter 3. In future studies, we will further investigate the effect of using the meta-learning strategy and also study other applications of such a classifier on non-medical images.

Clinicians need information about the severity and types of diseases for a precise diagnosis. One of the other future directions is to provide more diagnostic detail, i.e., different stages of each of the diseases, which will lead to improve training and classification.

References

[1] Igor O. Nasonkin. Biology of vision@online, 2018. (document), 1.1

[2] National Academies of Sciences Engineering and Medicine. *Making Eye Health a Population Health Imperative: Vision for Tomorrow*. The National Academies Press, Washington, DC, 2016. 1.1

[3] John S Wittenborn, Xinzhi Zhang, Charles W Feagan, Wesley L Crouse, Sundar Shrestha, Alex R Kemper, Thomas J Hoerger, and Jinan B Saaddine. The economic burden of vision loss and eye disorders among the united states population younger than 40 years. *Ophthalmology*, 120(9):1728–1735, 2013. 1.1

[4] Göran Darius Hildebrand and Alistair R Fielder. Anatomy and physiology of the retina. In *Pediatric retina*, pages 39–65. Springer, 2011. 1.1.1

[5] J.S. Schuman, C.A. Puliafito, J.G. Fujimoto, and J.S. Duker. *Optical Coherence Tomography of Ocular Diseases*. Online access: EBSCO eBook Clinical Collection. SLACK Incorporated, 2013. 1.1.2, 1.1.3, 1.1.4

[6] Rupert RA Bourne, Seth R Flaxman, Tasanee Braithwaite, Maria V Cicinelli, Aditi Das, Jost B Jonas, Jill Keeffe, John H Kempen, Janet Leasher, Hans Limburg, et al. Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: a systematic review and meta-analysis. *The Lancet Global Health*, 5(9):e888–e897, 2017. 1.1.3

[7] Jack J Kanski and Brad Bowling. *Clinical ophthalmology: a systematic approach*. Elsevier Health Sciences, 2011. 1.1.3

[8] 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, 2010. 1.1.4

[9] Raheleh Kafieh, Hossein Rabbani, and Saeed Kermani. A Review of Algorithms for Segmentation of Optical Coherence Tomography from Retina. *Journal of Medical Signals and Sensors*, 3(1):45–60, 2013. 1.1.4, 2.2.2

[10] Delia Cabrera Fernández, Harry M Salinas, and Carmen A Puliafito. Automated detection of retinal layer structures on optical coherence tomography images. *Optics Express*, 13(25):10200–10216, 2005. 1.1.4, 2.2.2, 3.1

[11] Sina Farsiu, Stephanie J. Chiu, Rachelle V. O'Connell, Francisco A. Folgar, Eric Yuan, Joseph A. Izatt, and Cynthia A. Toth. Quantitative classification of eyes with and without intermediate age-related macular degeneration using optical coherence tomography. *Ophthalmology*, 121(1):162–172, 2014. 1.1.4

[12] Donatella Pascolini and Silvio Paolo Mariotti. Global estimates of visual impairment: 2010. *British Journal of Ophthalmology*, 96(5):614–618, 2012. 1.1.4

[13] Leyuan Fang, David Cunefare, Chong Wang, Robyn H. Guymer, Shutao Li, and Sina Farsiu. Automatic segmentation of nine retinal layer boundaries in OCT images of non-exudative AMD patients using deep learning and graph search. *Biomedical Optics Express*, 8(5):2732, 2017. 2.1, 3.1

[14] Sijie Niu, Luis de Sisternes, Qiang Chen, Theodore Leng, and Daniel L. Rubin. Automated geographic atrophy segmentation for SD-OCT images using region-based C-V model via local similarity factor. *Biomedical Optics Express*, 7(2):581, 2016. 2.1, 2.5

[15] Mircea Mujat, Raymond C Chan, Barry Cense, B Hyle Park, Chulmin Joo, Taner Akkin, Teresa C Chen, and Johannes F de Boer. Retinal nerve fiber layer thickness map determined from optical coherence tomography images. *Optics Express*, 13(23):9480–9491, 2005. 2.1, 2.2.2, 2.5

[16] Akshaya Mishra, Alexander Wong, Kostadinka Bizheva, and David A Clausi. Intra-retinal layer segmentation in optical coherence tomography images. *Optics Express*, 17(26):23719–28, 2009. 2.1, 2.2.3, 2.2.3, 2.5

[17] Itebeddine Ghorbel, Florence Rossant, Isabelle Bloch, Sarah Tick, and Michel Paques. Automated segmentation of macular layers in OCT images and quantitative evaluation of performances. *Pattern Recognition*, 44(8):1590–1603, 2011. 2.1, 2.2.2, 2.5

[18] Muhammad Usman, Muhammad Moazam Fraz, and Sarah A. Barman. Computer Vision Techniques Applied for Diagnostic Analysis of Retinal OCT Images: A Review. *Archives of Computational Methods in Engineering*, 24(3):449–465, 2017. 2.1, 3.1

[19] Tony F. Chan and Luminita A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001. 2.2, 2.2.4

[20] Damber Thapa, Kaamran Raahemifar, and Vasudevan Lakshminarayanan. Reduction of speckle noise from optical coherence tomography images using multi-frame weighted nuclear norm minimization method. *Journal of Modern Optics*, 62(21):1856–1864, 2015. 2.2.2

[21] Lucien Birgé and Pascal Massart. From model selection to adaptive estimation. *Festschrift for Lucien Le Cam: Research papers in probability and statistics*, pages 55–87, 1997. 2.2.2

[22] Amir A Amini, Terry E Weymouth, and Ramesh C Jain. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on pattern analysis and machine intelligence*, 12(9):855–867, 1990. 2.2.3.1

[23] PhD thesis. 2.2.3.1

[24] Charnchai Pluempitiwiriyawej, José MF Moura, Yi-Jen Lin Wu, and Chien Ho. Stacs: New active contour scheme for cardiac mr image segmentation. *IEEE transactions on medical imaging*, 24(5):593–603, 2005. 2.2.4

[25] Azadeh Yazdanpanah, Ghassan Hamarneh, Benjamin R Smith, and Marinko V Sarunic. Segmentation of intra-retinal layers from optical coherence tomography images using an active contour approach. *IEEE transactions on medical imaging*, 30(2):484–496, 2011. 2.2.4

[26] Paul Jaccard. The distribution of the flora in the alpine zone. *New phytologist*, 11(2):37–50, 1912. 2.3

[27] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945. 2.3

[28] Eric N Mortensen and William A Barrett. Intelligent scissors for image composition. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 191–198. ACM, 1995. 2.3

[29] Eric N Mortensen and William A Barrett. Intelligent scissors for image composition. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 191–198. ACM, 1995. 2.3.1.1, 2.3.1.1

[30] Baoxiang Huang, Ge Chen, Xiaolei Zhang, and Huan Yang. A coastal zone segmentation variational model and its accelerated admm method. *Journal of Ocean University of China*, 16(6):1081–1089, 2017. 2.5

[31] Giovanni Gregori, Fenghua Wang, Philip J Rosenfeld, Zohar Yehoshua, Ninel Z Gregori, Brandon J Lujan, Carmen A Puliafito, and William J Feuer. Spectral domain optical coherence tomography imaging of drusen in nonexudative age-related macular degeneration. *Ophthalmology*, 118(7):1373–1379, 2011. 3.1

[32] Felipe A Medeiros, Linda M Zangwill, Christopher Bowd, Roberto M Vessani, Remo Susanna, and Robert N Weinreb. Evaluation of retinal nerve fiber layer, optic nerve head, and macular thickness measurements for glaucoma detection using optical coherence tomography. *American journal of ophthalmology*, 139(1):44–55, 2005. 3.1

[33] Priyanka Roy, Peyman Gholami, Mohana Kuppuswamy Parthasarathy, John Zelek, and Vasudevan Lakshminarayanan. Automated intraretinal layer segmentation of optical coherence tomography images using graph-theoretical methods. In *Optical Coherence Tomography and Coherence Domain Optical Methods in Biomedicine XXII*, volume 10483, page 104832U. International Society for Optics and Photonics, 2018. 3.1

[34] Peyman Gholami, Priyanka Roy, Mohana Kuppuswamy Parthasarathy, Abbas Ommani, John Zelek, and Vasudevan Lakshminarayanan. Intra-retinal segmentation of optical coherence tomography images using active contours with a dynamic programming initialization and an adaptive weighting strategy. In *Proc.SPIE*, volume 10483, pages 10483 – 10483 – 6, 2018. 3.1

[35] Robert Koprowski, Slawomir Teper, Zygmunt Wróbel, and Edward Wylegala. Automatic analysis of selected choroidal diseases in oct images of the eye fundus. *Biomedical engineering online*, 12(1):117, 2013. 3.1

[36] Guillaume Lemaître, Mojdeh Rastgoo, Joan Massich, Carol Y Cheung, Tien Y Wong, Ecosse Lamoureux, Dan Milea, Fabrice Mériaudeau, and Désiré Sidibé. Classification of sd-oct volumes using local binary patterns: experimental validation for dme detection. *Journal of ophthalmology*, 2016, 2016. 3.1

[37] Yu-Ying Liu, Hiroshi Ishikawa, Mei Chen, Gadi Wollstein, Jay S Duker, James G Fujimoto, Joel S Schuman, and James M Rehg. Computerized macular pathology diagnosis in spectral domain optical coherence tomography scans based on multiscale

texture and shape features. *Investigative ophthalmology & visual science*, 52(11):8316–8322, 2011. 3.1, 3.3.1

[38] Peyman Gholami, Mohsen Sheikh Hassani, Mohana KuppuswamyParthasarathy, John S. Zelek, and Vasudevan Lakshminarayanan. Classification of optical coherence tomography images for diagnosing different ocular diseases. In *Proc.SPIE*, volume 10487, pages 10487 – 10487 – 6, 2018. 3.1, 3.2.3, 3.2.3.2

[39] S P K Karri, Debjani Chakraborty, and Jyotirmoy Chatterjee. Transfer learning based classification of optical coherence tomography images with diabetic macular edema and dry age-related macular degeneration. *Biomedical optics express*, 8(2):579–592, 2017. 3.1

[40] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 3.2, 3.2.4, 3.2.4

[41] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, June 2015. 3.2, 3.2.4, 3.2.4

[42] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3.2, 3.2.4, 3.2.4

[43] Peyman Gholami, Priyanka Roy, and Vasudevan Lakshminarayanan. Optical coherence tomography image database (octid), 2018. 3.2.1

[44] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 3.2.2, 3.2.2, 3.3.1

[45] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, and Andrea Vedaldi. Deep filter banks for texture recognition, description, and segmentation. *International Journal of Computer Vision*, 118(1):65–94, 2016. 3.2.2

[46] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999. 3.2.2

[47] Andrea Vedaldi and Brian Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1469–1472. ACM, 2010. 3.2.2, 3.3.1

[48] Brian Fulkerson, Andrea Vedaldi, and Stefano Soatto. Localizing objects with smart dictionaries. In *European Conference on Computer Vision*, pages 179–192. Springer, 2008. 3.2.2

[49] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. 3.2.2

[50] David B Abou Chacra and John S Zelek. Fully automated road defect detection using street view images. In *Computer and Robot Vision (CRV), 2017 14th Conference on*, pages 353–360. IEEE, 2017. 3.2.2

[51] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *European conference on computer vision*, pages 143–156. Springer, 2010. 3.2.2

[52] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987, 2002. 3.2.3.1

[53] Mark Andrew Hall. Correlation-based feature selection for machine learning. 1999. 3.2.3.2

[54] Lucien Birgé and Pascal Massart. From model selection to adaptive estimation. In *Festschrift for lucien le cam*, pages 55–87. Springer, 1997. 3.2.4

[55] Guodong Wang, Qian Dong, Zhenkuan Pan, Ximei Zhao, Jinbao Yang, and Cunliang Liu. Active contour model for ultrasound images with rayleigh distribution. *Mathematical Problems in Engineering*, 2014, 2014. 4

[56] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655, 2014. 4

[57] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3304–3311. IEEE, 2010. 4

APPENDICES

Appendix A

In this appendix we present the MATLAB codes for the main functions of each of the proposed methods. All of the codes are run in MATLAB 2018a.

Transfer learning with GoogLeNet:

```
% GoogleNet transfer learning
clear
load imdata
[trainset,validationset] = splitEachLabel(imdata,0.7,'randomized');
net = googlenet;
layerg = layerGraph(net);
insize = net.Layers(1).insize;
layerg = removeLayers(layerg, {'loss3-classifier','prob','output'});
numClasses = numel(categories(trainset.Labels));
newLayers = [
fullyConnectedLayer(numClasses,'Name','fc','WeightLearnRateFactor',10,
'BiasLearnRateFactor',10)
softmaxLayer('Name','softmax')
classificationLayer('Name','classoutput')];
layerg = addLayers(layerg,newLayers);
layerg = connectLayers(layerg,'pool5-drop_7x7_s1','fc');
layers = layerg.Layers;
connections = layerg.Connections;
layers(1:110) = freezeWeights(layers(1:110));
layerg = createlayergUsingConnections(layers,connections);
```

```
% with data augmentation
pixelRange = [-30 30];
imageAugmenter = imageDataAugmenter( ...
    'RandXReflection',true, ...
    'RandXTranslation',pixelRange, ...
    'RandYTranslation',pixelRange);
augtrainset = augmentedImageDatastore(insize(1:2),trainset, ...
    'DataAugmentation',imageAugmenter);
% without data augmentation
% augtrainset = augmentedImageDatastore(insize(1:2),trainset)
augvalidationset = augmentedImageDatastore(insize(1:2),validationset);
options = trainingOptions('sgdm', ...
'MiniBatchSize',10, ...
'MaxEpochs',6, ...
'InitialLearnRate',1e-4, ...
'ValidationData',augvalidationset, ...
'ValidationFrequency',3, ...
'ValidationPatience',Inf, ...
'Verbose',false ,...
'Plots','training-progress');
transferednet = trainNetwork(augtrainset,layerg,options);
[prediction,scores] = classify(transferednet,augvalidationset);
EVAL1 = Evaluate2(validationset.Labels=='DR',prediction=='DR')
EVAL2 = Evaluate2(validationset.Labels=='ARMD',prediction=='ARMD')
EVAL3 = Evaluate2(validationset.Labels=='NORMAL',prediction=='NORMAL')
EVAL4 = Evaluate2(validationset.Labels=='MH',prediction=='MH')
FINAL=EVAL2*55/470+EVAL4*102/470+EVAL3*206/470+EVAL1*107/470
```

Transfer learning with ResNet:

```
% ResNet transfer learning
clear
load imdata
[imdataTrain,imdataValidation] = splitEachLabel(imdata,0.7,'randomized');
net = resnet101;
```

```
layerg = layerGraph(net);
figure('Units','normalized','Position',[0.1 0.1 0.8 0.8]);
plot(layerg)
insize = net.Layers(1).insize;
layerg = removeLayers(layerg, {'fc1000', 'prob','ClassificationLayer_predictions'});
numClasses = numel(categories(imdataTrain.Labels));
newLayers = [
fullyConnectedLayer(numClasses,'Name','fc','WeightLearnRateFactor'
10,'BiasLearnRateFactor',10)
softmaxLayer('Name','softmax')
classificationLayer('Name','classoutput')];
layerg = addLayers(layerg,newLayers);
layerg = connectLayers(layerg,'pool5','fc');
figure('Units','normalized','Position',[0.3 0.3 0.4 0.4]);
plot(layerg)
ylim([0,10])
layers = layerg.Layers;
connections = layerg.Connections;
layers(1:110) = freezeWeights(layers(1:110));
layerg = createlayergUsingConnections(layers,connections);
pixelRange = [-30 30];
%%
%imageAugmenter = imageDataAugmenter( ...
%    'RandXReflection',true, ...
%    'RandXTranslation',pixelRange, ...
 %   'RandYTranslation',pixelRange);
%augimdataTrain = augmentedImageDatastore(insize(1:2),imdataTrain, ...
%    'DataAugmentation',imageAugmenter);
%%
 augimdataTrain = augmentedImageDatastore(insize(1:2),imdataTrain)
augimdataValidation = augmentedImageDatastore(insize(1:2),imdataValidation);
options = trainingOptions('sgdm', ...
'MiniBatchSize',10, ...
'MaxEpochs',6, ...
'InitialLearnRate',1e-4, ...
'ValidationData',augimdataValidation, ...
'ValidationFrequency',3, ...
'ValidationPatience',Inf, ...
```

```
'Verbose',false ,...
'Plots','training-progress');
netTransfer = trainNetwork(augimdataTrain,layerg,options);
[prediction,scores] = classify(netTransfer,augimdataValidation);
EVAL1 = Evaluate2(imdataValidation.Labels=='DR',prediction=='DR')
EVAL2 = Evaluate2(imdataValidation.Labels=='ARMD',prediction=='ARMD')
EVAL3 = Evaluate2(imdataValidation.Labels=='NORMAL',prediction=='NORMAL')
EVAL4 = Evaluate2(imdataValidation.Labels=='MH',prediction=='MH')
FINAL=EVAL2*55/470+EVAL4*102/470+EVAL3*206/470+EVAL1*107/470
```

OCT image segmentation main fucntion

```
function [X, phiOut] = OCTseg(img, phi0, nu, mu, lam_1, lam_2, lam_3,...
    dt, epsilon, numIter, fun_n, imn, PAT, METH)
im = single(img);
rCount = fun_n+1;
imgSize = [size(im,1) size(im,2)];
Ivec = reshape(im, [prod(imgSize) 1]);
% E = zeros(numIter, fun_n);
phi = phi0;
H = Heaviside(reshape(phi0, [prod(imgSize) fun_n]),epsilon);
showCurveAndPhi(im,phi,1, fun_n, imn);
g = spatialWei(im,2, 1);
if (METH ==1)
    w_0 = spatialWei(im,1,1e-6);
    w_1 = spatialWei(im,1, 1e-9);
    %w_2 = spatialWei(im,1, 1e-3);
else %-- No spatial weight
    w_0 = 1;w_1 = 1; w_2 = 1;
end
% fx_f = Dx_forward(im);
% fy_f = Dy_forward(im);
% gamma = 1./( sqrt(fx_f.^2+ fy_f.^2)+epsilon );
[Ix Iy]= gradient(im);
gamma= 1./( Ix.^2+ Iy.^2+epsilon );
```

```
[gx,gy]=gradient(gamma);
for m=1:numIter
    for uC = 1:fun_n % Number of phi
        tmp_phi = phi0(:,:,uC);
        dirac  = Delta(tmp_phi,epsilon);
        differenceScheme=1;
        D = estimCircle(phi0, g);
    if (uC == 2)
            w =w_1;
        elseif(uC == 4 || uC == 3 )
            w =w_1;
        else
            w =w_0;
        end
        P = lam_3.*w.*( D(:,:,uC).^2 )+ nu+ gamma; %lam_3 *
D^2 + lam_2* gamma +alpha

        Curv = CURVATURE(tmp_phi,differenceScheme);
        [Q f_den]= inproPAPhi(tmp_phi,P);
        %      Htmp = Heaviside(reshape(phi0, [prod(imgSize) fun_n]),
        %        epsilon);
        [X, Xp] = datafit(fun_n, uC, dirac, H, imgSize);
        XVec = X';
        XpVec = Xp'; % reshape(Xp, [prod(imgSize) rCount])
        %-- force from image information
        [energy_term C]= energy( Ivec, X, rCount );
        fidTerm = -lam_1.*sum( energy_term.*Xp, 2); %
        tmp = sum(sum(abs(fidTerm)));
        %-- gradient descent
        if(METH ==3)
            shap_term = 10*Curv;
        else
            shap_term =10* Curv + P.* Curv + Q ;
        end
        curveTerm = Delta(tmp_phi, epsilon).* shap_term; % regularization term
        regTerm = mu*(4*del2(tmp_phi)-Curv);
        if(METH ==3) % update phi
            energy_term = reshape(fidTerm, [imgSize 1]);
```

```
            phi_update = energy_term./max(max( abs(energy_term) ) )
+ curveTerm + regTerm;
        elseif (METH == 4)
            %             energy_term = EVOLUTION_GAC(tmp_phi, gamma);
            %             phi_update = energy_term + curveTerm+ regTerm;
            phi_update = EVOLUTION_GAC(tmp_phi, gamma, gx,gy);%+ regTerm;
        else
            energy_term = reshape(fidTerm, [imgSize 1]);
            phi_update = energy_term + curveTerm + regTerm;
        end
        dt =10/(max(phi_update(:))+eps);
        tmp_phi = tmp_phi + dt*phi_update;
        phi0(:, :, uC) = tmp_phi;
        H(:, uC) = Heaviside( reshape(tmp_phi, [1 prod(imgSize)]), epsilon)';
    end
    %-- Compute weighted value lamda1 & lamda2 & lamda3
    %lam_1 = 1 - m/(numIter+eps) *(1 - 0.5);
    %lam_1 = 0.5*( 1+cos(i*pi/iteration) )+ 0.001
    %lam_2 = lam_1;
    if (METH ==2 || METH ==1)
        if (uC == 5)
            lam_3 =( 1 /( cosh( 8*( m/(numIter+eps)-1 ))) );
            lam_1 = 1 - m/(numIter+eps) *(1 - 0.5);
        elseif (uC == 1)
            lam_3 =( 1 /( cosh( 10*( m/(numIter+eps)-1 ))) );
            lam_1 = 2 - m/(numIter+eps) *(1 - 0.5);
        else
            lam_3 =( 1 /( cosh( 6*( m/(numIter+eps)-1 ))) );
            lam_1 = 1 - m/(numIter+eps) *(1 - 0.5);
        end
    else
        lam_3=1;
    end
    %     phi_d =(phi-phi0).^2;
    %     phi_error(m) = sqrt( sum(phi_d(:))/numel(phi));
    %     E = abs( E - E_update/prod(imgSize) );
    phi = phi0;
    if( mod(m,5) == 0 )
```

```
        showCurveAndPhi(im,phi,m, fun_n, imn);
    end
end
showcurve(im,phi,m, fun_n, imn);
phiOut =phi;
axis off; axis equal;
% return the characterstic
X = datafit(fun_n, 1, Delta(phi0(:, :, 1), epsilon), H, imgSize);
X = reshape(X, [imgSize rCount]);
```

Texture feature extraction

```
function OCTtexture()
%% This code is a modified version of the DTD avaiable
online at https://www.robots.ox.ac.uk/~vgg/data/dtd/
lite = false;
clear ex;
save_report = 0;
datasetList = {'oct'};
numSplits = cell(1, numel(datasetList));
defaultNumSplits = 1;
if lite
  defaultNumSplits = 1;
end
for ii = 1 : numel(datasetList)
  numSplits{ii} = defaultNumSplits;
end
featureType = 'dsift';
encoder = { 'fv'};
accuracy = zeros(numel(datasetList), numel(encoder));
pmAcc = zeros(numel(datasetList), numel(encoder));
for ii = 1 : numel(datasetList)
  for ee = 1 : numel(encoder)
    if (1 ~= numel(numSplits{ii}))
```

```
    nSplits = numSplits{ii};
  else
    nSplits = 1 : numSplits{ii};
  end
  resAcc = zeros(1, numel(nSplits));
  for jj = nSplits
    ex.trainOpts = {'C', 15} ;
    ex.prefix = sprintf('%s-%s-seed-%d', encoder{ee}, ...
      featureType, jj);
    switch  encoder{ee}
      case 'fv'
        ex.opts = {...
          'type', encoder{ee}, ...
          'numWords', 256, ...
          'layouts', {'1x1'}, ...
          'numPcaDimensions', 67 ...
          'extractorFn', @(x) getDenseSIFT(x, ...
                                           'step', 8, ...
                                          'scales', 2.^(1.5:-.5:-3))};
    end
    res = traintest(...
      'prefix', [tag '-' datasetList{ii} '-' ex.prefix], ...
      'seed', jj, ...
      'dataset', datasetList{ii}, ...
      'datasetDir', fullfile('data', datasetList{ii}), ...
      'lite', lite, ...
      'kernel', 'linear', ...
      'featureType', featureType, ...
       ex.trainOpts{:}, ...
      'encoderParams', ex.opts);
    resAcc(jj) = res.mAcc * 100;
  end
  if (1 ~= numel(numSplits{ii}))
    resAcc = resAcc(numSplits{ii});
  end
  accuracy(ii, ee) = mean(resAcc);
  pmAcc(ii, ee) = std(resAcc);
end
```

```
end
if 0
  imdb = load('imdb.mat');
  m_acc = mean(all_acc);
  for jj = 1 : numel(imdb.meta.classes)
    fprintf('%s\t%.2f\n', imdb.meta.classes{jj}, 100 * m_acc(jj));
  end
end
%% Print table
if (save_report)
    fid0 = fopen('reports.html', 'w');
    if (fid0 > 0)
      fprintf(fid0, '<table border="1">\n');
      fprintf(fid0, ['<tr><td rowspan="2">
Dataset</td><td colspan="5">SIFT</td>' ...
        '<td rowspan="2">DeCAF</td><td rowspan="2">
IFV + DeCAF</td></tr>\n']);
      fprintf(fid0, '<tr><td>IFV</td><td>BOVW</td><td>
VLAD</td><td>LLC</td><td>KCB</td></tr>\n');
      for ii = 1 : size(accuracy, 1)
        line = td(datasetList{ii});
        for jj = 1 : size(accuracy, 2)
          line = [line td(sprintf('%.2f +/- %.2f',
accuracy(ii, jj), pmAcc(ii, jj)))];
        end
        fprintf(fid0, ['<tr>' line '</tr>\n']);
      end
      fprintf(fid0, '</table>\n');
      fclose(fid0);
    end
end
end
function evaluate = evaluateuate2(gold,pred)
idx = (gold()==1);
p = length(gold(idx));
n = length(gold(~idx));
N = p+n;
tp = sum(gold(idx)==pred(idx));
```

```
tn = sum(gold(~idx)==pred(~idx));
fp = n-tn;
fn = p-tp;
tp_rate = tp/p;
tn_rate = tn/n;
accuracy = (tp+tn)/N;
sensitivity = tp_rate;
specificity = tn_rate;
precision = tp/(tp+fp);
recall = sensitivity;
f_measure = 2*((precision*recall)/(precision + recall));
gmean = sqrt(tp_rate*tn_rate);
evaluate = [accuracy sensitivity specificity
precision recall f_measure gmean];
function y = td(x)
  y = ['<td>' x '</td>'];
end
```