DEEP LEARNING FOR IMAGE PROCESSING, APPLICATION TO DIABETIC MACULAR EDEMA (DME) DETECTION ON OPTICAL COHERENCE TOMOGRAPHY (OCT) IMAGES

by

GENEVIEVE CHAN CHEAU YANN

18430

Dissertation submitted in partial fulfilment of
the requirements for the
Bachelor of Engineering (Hons)
(Electrical & Electronic)

FYP II January 2017

PROF FABRICE MERIAUDEAU

Universiti Teknologi PETRONAS
32610 Bandar Seri Iskandar
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

DEEP LEARNING FOR IMAGE PROCESSING, APPLICATION TO DIABETIC MACULAR EDEMA (DME) DETECTION ON OPTICAL COHERENCE TOMOGRAPHY (OCT) IMAGES

by

GENEVIEVE CHAN CHEAU YANN

18430

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved by,

_____

Prof. Fabrice Meriaudeau
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

January 2017

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

_____

GENEVIEVE CHAN CHEAU YANN

ABSTRACT

Diabetic Macular Edema (DME) is a common eye disease which causes irreversible vision loss for diabetic patients, if left untreated. Health care and associated costs for treatment related to eye disease also increases with severity of case. Thus, early diagnosis of DME could help in early treatment and prevent blindness. Using a pretrained network of Convolutional Neural Network (CNN), this paper aims to create a framework based on deep learning for DME recognition on Spectral Domain Optical Coherence Tomography (SD-OCT) images through transfer learning, from a (limited) dataset retrieved from Singapore Eye Research Institute (SERI). The dataset consists of 16 volumes each for normal patients and DME patients with 128 images in each volume. AlexNet model is used for image classification. The images are pre-processed: noise removal using BM3D filtering; retinal boundary extraction; and image cropping, features are extracted using CNN and finally images are classified through cross-validation of multiclass linear SVM. The framework design utilizes MATLAB (2016-a) software and its drivers on image processing. The experiment shows that through noise removal, retinal boundary extraction and image cropping, it achieves better classification performance than other recent published works of 95.44% specificity (SP) and 93.89% sensitivity (SE) with a total accuracy of 94.66%.

ACKNOWLEGEMENT

Upon completion of my Final Year Project (FYP), I would like to express my sincere and heartfelt gratitude to so many who have patiently guided and helped me in this endeavour. These people have pushed me out of my confort zone to challenge myself to achieve greater things. Within 28 weeks, I have managed to complete my FYP project. In between, I have manage to find passion in what I learn, self-improvement and widened my scope of academic. Truly, I am learning eveyr day and each day is a new experienced gained. I would not have reached where I am right now without the help of many people whom I am deeply thankful for.

Special thanks to:

- Prof Fabrice Meriaudeau, for his coaching, patience, kindness and encouragement from the beginning. Prof Fabrice has made me feel comfortable working under him with the proper distribution of workload within the time frame.
- Mr Awais, for his guidance and constant motivation whom have trusted me on my capabilities and competencies on handling this project towards completion.
- Dr Norashikin bt Yahya, for her support and guidance in preparation to complete the FYP course.
- My family, fellow coursemates and friends who have been constantly motivating me to persevere and finish this semester.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1.Background of Study

Diabetic Macular Oedema (DME) is one the many eye diseases which is commonly found in diabetic patients, typically Type 2 diabetes. A review in 2012 shows that almost 7% of diabetic patients may have DME [1]. DME is a result of the accumulation of fluid in the macula due to Diabetic Retinopathy (DR) i.e. the damage of blood vessels in the retina of the eye. Consequently, an increase of retinal thickness within one disk diameter of the fovea center with or without hard exudates and sometimes associated with cysts [2]. Evidently, DME will only happen when DR is left untreated. When DME is unattended, it may cause blurry vision and if the worst comes to worst, an irreversible vision loss. Health care and associated costs related to eye disease is also very high when case becomes severe.

Of course, there is also an early detection process for such diseases e.g. fundus images and Optical Coherence Tomography (OCT) to avoid the severe effects of DME. These methods basically capture images of the retinal thickness. An irregular thickness of the retina indicates the presence of DME. Increasingly, Spectral Domain OCT (SD-OCT) images is preferable over fundus images as it can capture images with higher resolution and thus better quality for analysis [3].

Up to our knowledge, some researchers managed to create a system to detect DME using feature detection from SD-OCT images. An automated detection of DME from SD-OCT images brings us one step towards artificial intelligence technology. This study aims to formulate a framework based on deep learning for DME identification from SD-OCT images through transfer learning [4], given limited data resources, to provide a solution for better reliance in DME detection process.

1.2.Problem Statement

DME has been a common eye disease associated with Type 2 diabetes. Yet, there has been no improvement in diagnosing this disease although there has been a

widespread of technology for numerous systemic diseases i.e. lung disease clarification, cerebral internal bleeding and haemorrhage detection using fundus images. The lack of outputs of SD-OCT images (since SD-OCT is a new imaging tool) as a database leads to lack of datasets to be used to compare results of study which results in very few works, although there is, published with deep learning architectures for DME detection or segmentation.

1.3. Objectives and Scope of Study

1.3.1. Objective

The objective of this project is:

- To introduce deep learning architecture for DME detection.
- To investigate transfer learning or fine tuning of pre-trained networks on sets of SD-OCT images.
- To aid in the advancement of the medical technology of DME detection for the Ophthalmologist experts.

1.3.2. Scope of Study

The given project title is *"Deep learning for Image Processing, Application to DME Detection on OCT Images"*. This project was conducted to determine the suitable methods to design a framework based on deep learning for DME recognition on OCT images, typically SD-OCT using the MATLAB R2016-a software. This project hopes to create a method on implementing deep learning for DME Diagnosis. By doing so, it helps the medical industry and hopefully, thereafter helps to improve the technology and can be commercialized. This project is targeted to be conducted throughout the Final Year for a period of 28 weeks i.e. 2 semesters from 12th September 2016 to 8th May 2017 (tentatively).

The aspects covered for the project includes:

- Studying the architecture of Convolutional Neural Networks (CNN).
- Understanding the ideology of backpropagation in CNN.
- Determine the relationship between machine learning, deep learning and artificial intelligence.
- Review the fundamentals and methodology of OCT images.

- Lessons on designing MATLAB code through courseera.org online web courses.
- Construct code using the drivers available in the MATLAB R2016A.
- Review code for validation and quality process.

1.3.3. Relevance and Feasibility

At the end of this project, student is expected to deliver a full report and the MATLAB software code which encapsulates the algorithm of the deep learning network for DME detection through image processing of SD-OCT images using fine tuning technique. Due to the limited period, a careful revision was made to ensure that student can achieve her objectives by the end of the Final Year. The scheduled activities along with the timeline activities are listed in the Gantt Chart shown in Chapter 3: Methodology. Student will only focus on using a pretrained network for other applications and to apply it to the DME detection through transfer learning.

CHAPTER 2
LITERATURE REVIEW

This section presents a summary of recent researches conducted to review on deep learning for image processing and using OCT images for DME detection.

Over the last twenty years, much commentary has been circulating when Artificial Intelligence (AI) i.e. a machine capable of behaving like a human e.g. the ability to feel and make decisions, was first introduced in 1950 [5] i.e. the desire to create something that does something intelligently. The success of AI remains unanswered until 1980, when machine learning begins to flourish [5], and increasingly, deep learning after a historic breakthrough in the ImageNet Computer Vision competition in 2012, where deep convolutional networks trained on a dataset of a million images of one thousand classes produced significant decrease of error rates i.e. half of the previous competing approach [6].

2.1.Deep Learning

Machine-learning technology has progressively contributed to the modernization of the society e.g. object recognition, speech recognition and selection of relevant results while searching the web. It is widely available and has already been used in our everyday lives e.g. smartphones, cameras, and web searches. The basic idea behind these technologies is called representation learning, which automatically identifies the representations of a given raw input by extracting features needed, using a feature extractor, to train a classifier to detect or classify patterns in the input. Consequently, these applications' systems evolve into deep learning e.g. the social network, Facebook's photo tagging feature is a face recognition based on a deep network called *DeepFace* [7].

There are many types of machine learning algorithms [6, 8]:

- Supervised learning – generates a function which classify inputs to desired outputs based on their assigned labels. Then, a decision boundary is drawn

as a threshold to classify each input to desired output. This allows the machine to accurately map new examples. The drawback of supervised learning is that it requires labelled dataset to generate the function which may not be easy to obtain compared to unlabelled datasets.

- Unsupervised learning – looks for patterns native to a dataset (input) and models it into clusters e.g. K-means clustering. Note that each input is not labelled with its category which is relatively easier to collect.
- Semi-supervised learning – A bit of both of supervised and unsupervised learning. It uses both labelled and unlabelled data to train a network to generate a better classifier.
- Deep learning - Deep learning can be described as a type of self-learning tool whereby it creates an algorithm by learning the patterns of many sets of data samples extracted from multiple processing layers.

Conventional machine learning has its limitation in data interpretation as the supervised learning requires "handcrafted" features to properly categorise inputs. This can be avoided through deep learning by using general-purpose learning procedure. In deep learning, each layer is composed of representation-learning methods, producing multiple layers of representations, which processes each layer to a higher and more abstract level. Through deep learning, very complex functions can be learned.

2.2.Backpropagation

The goal of deep learning had been to replace "handcrafted" features network with a self-learning multilayer network. Later in mid 1980s, it is found that multilayer representations can be calculated using Stochastic Gradient Descent (SGD) i.e. a tool for optimization [6]. Increasingly, deep learning is accomplished using backpropagation procedure to compute SGD to train the multilayers architecture. Figure 1 and 2 illustrate how backpropagation works in a simple multilayer neural network with 1 input layer, 2 hidden layers and 1 output layer. First, feedforward pass is computed by getting the weighted sum of product of the *j*th neurons of *l*th layer and the weights to obtain the total inputs to each *j*th neuron of the next *(l+1)*th layer, as shown in Figure 1. Usually, the weighted sum is then added to the bias term, but for now it is omitted for simplicity. Then, this value is inserted into an activation function to obtain the output unit of the *j*th neuron of the next layer. Note

that the activation function may vary. At present, Rectified Linear Unit (ReLU) is the most commonly used non-linear function because it learns faster in networks with many layers, thus provides leverage for deep learning networks.

$$y_{j+k}^{l+i} = f\left(\sum_{i=0}\sum_{k}\sum_{n} z_{j+k}^{l+i}\right), z_{j+k}^{l+i} = w_{(j+k)(j+n)}^{l+i} y_{j+n}^{l} \quad -(1)$$

$where,$

$w_{(j+k)(j+n)}^{l+i} = Synaptic\ weight$

$y_{j+n}^{l} = Input\ value\ of\ the\ lth\ layer$

$y_{j+k}^{l+i} = Output\ value\ of\ the\ (j+k)th\ neuron\ in\ (l+i)th\ layer$

$z_{j+k}^{l+i} = Weighted\ sum\ of\ the\ outputs\ of\ the\ neurons\ in\ the\ lth\ layer$

The process repeats again for the next layer, until the output of the final layer is obtained.

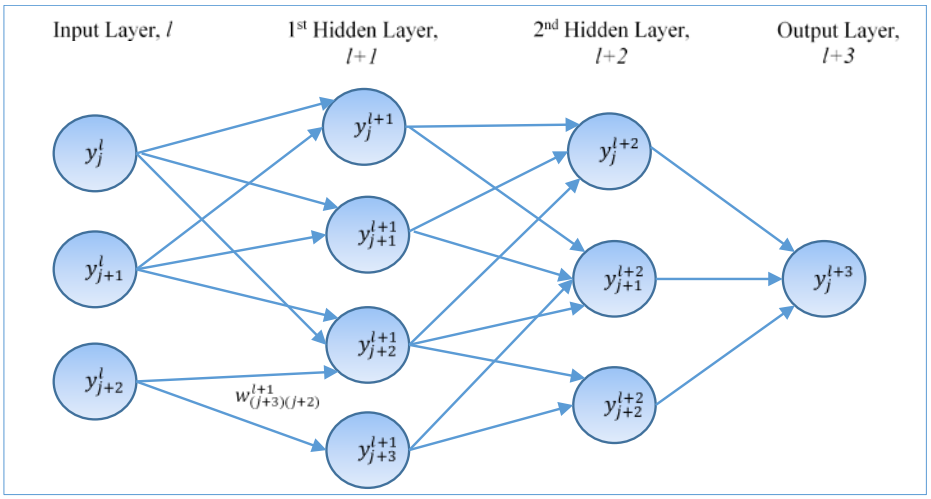

FIGURE 1 Multilayer neural network and backpropagation to compute the forward pass. Adapted from "Deep Learning" by LeCun, Y. et al. (2015) [6].

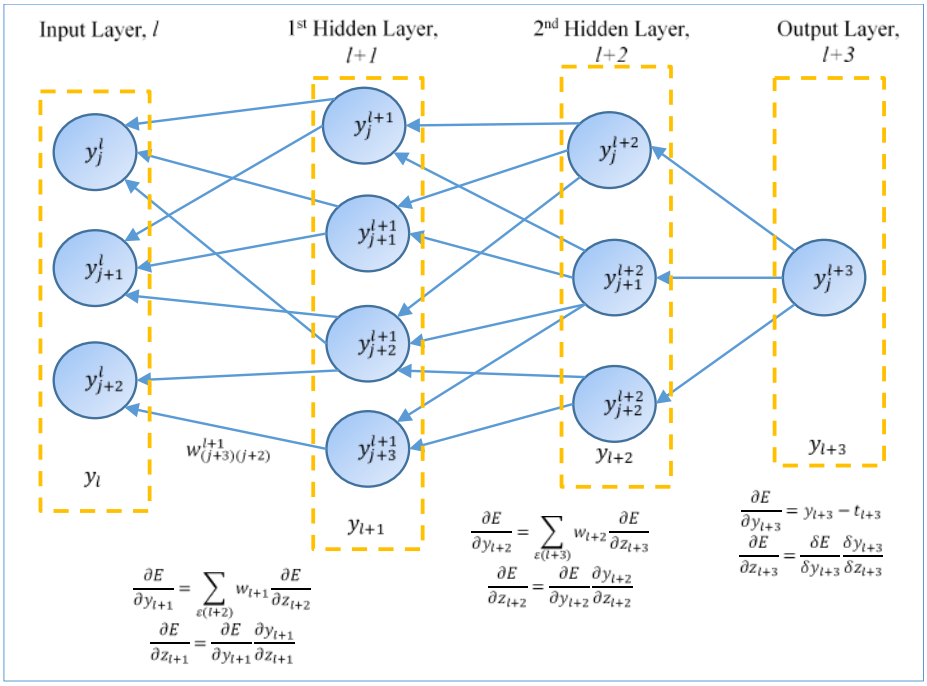

FIGURE 2 Multilayer neural network and backpropagation to compute the backward pass to optimize the weights. Adapted from "Deep Learning" by LeCun, Y. et al. (2015) [6].

The backpropagation algorithm typically requires 3 items: training set – a collection of input-output data to train the network, testing set – a collection of input-output data to test the network performance and learning rate – the parameters to adjust the network accuracy to improve the performance which is the calculated weights. The key aspect of backpropagation is that the derivative of the output of the neuron at each layer with respect to the input of the neuron at the same layer can be computing backwards. Starting from the output layer going backwards to the input layer, the error derivatives are computed with respect of the output of each neuron, which was calculated during the forward pass. Following chain derivative rule, the error derivative with respect to input can be obtained. This result is compared with the values of error derivatives with respect to input at the output layer, which is calculated by taking the differentiation of the cost function. At the end of the last layer when the gradient equation is obtained, the weights for each neuron of each layer are optimized and forward pass is computed again [6].

2.3. Convolutional Neural Network (CNN)

CNN is a deep, feedforward network with higher generalisation efficiency than other fully connected network. The architecture of CNN is in the form of a series of stages. There are typically 2 types of layers:

- Convolutional layer – All units in this layer are arranged in a feature map and connected to the weights. Over here, the weights are known as filter bank. The weighted sum is inserted to the ReLU. The operation of convolutional layer involves discrete convolution and it is used to calculate the representation junction from the previous layer.
- Pooling layer – This layer pulls similar representations calculated from convolutional layer together. It needs to ensure good pooling method as the position of each representation motif may vary.

In a typical CNN, there will be two to three stages of convolutional layers, activation function and pooling layers stacked together, followed by more convolutional and fully connected layers. Then, the weights in all the filter banks are trained and computed using backpropagation procedures of gradients. For example, the input of an image is fed to the network and the learned features in the first convolutional layer detect the presence of blobs and edges. Then, the second layer will detect motifs. Following, the third layer combines all motifs and form parts of an object and consequently detect objects with these features at the final layer [6].

Overall, CNN allows all the weights in all the filter banks to be trained in each layer using backpropagation procedure. The whole process of CNN involves sequential adaptation of the weights and bias. The CNN architecture that will be studied for this research is the AlexNet, who won the 2012 ImageNet competition halving the error rates of the previous competing method [9].

2.4.AlexNet Architecture

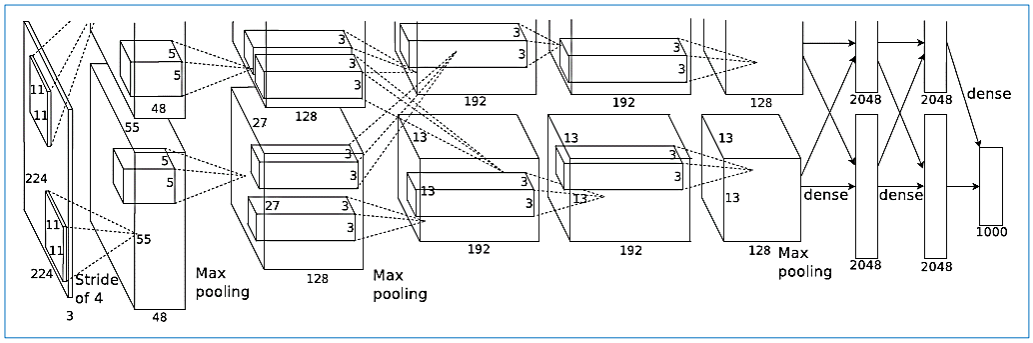

FIGURE 3 An illustration of the AlexNet architecture using CNN alongwith the filter banks value at each layer. Adapted from "ImageNet Classification with Deep Convolutional Neural Networks" by Krizhevsky A. et al. (2012) [9].

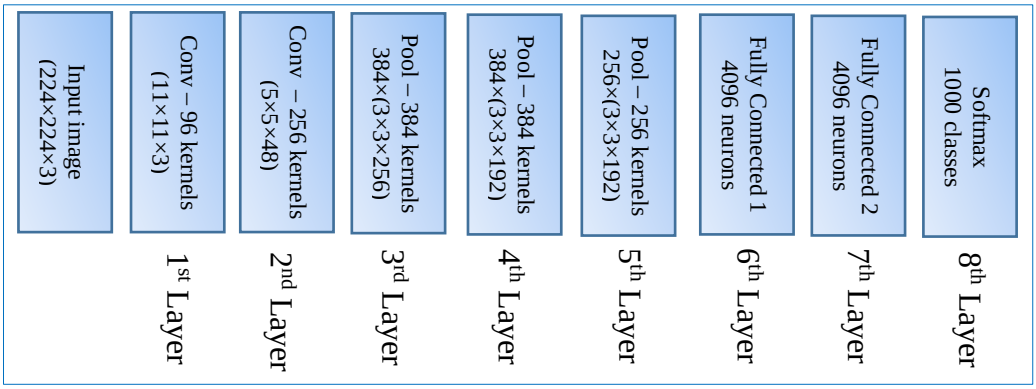

FIGURE 4 Description of each layer in AlexNet. Adapted from "Feature Evaluation of Deep Convolutional Neural Networks for Object Recognition and Detection" by Kataoka, H. et al. (2015) [10].

Figure 3 shows the architecture of AlexNet by Krizhevsky *et al.* whilst Figure 4 describes the type of layers for all 8 layers: 5 convolutional layers and 3 fully connected layers. The AlexNet receives input of size 224×224×3 RGB images. This research uses grayscale images dataset. Therefore, the images need to be concatenated thrice. The first convolutional layer will filter the input image and forms 96×55×55=290400 neurons. Then each neuron has 11×11×3=363 weights and 1 bias. Thus, the number of features on the first layer of AlexNet is 290400×363=105705600. ReLU is applied to the output of every layer. The feature vectors are further filtered in each layer until 1000 classes are obtained at the end using Softmax as the classifier.

2.5.Process of Image Classification

A review was recently conducted by Massich, Rastgoo, Lemaitre, Cheong, Wong, Sidibe and Meriaudeau (2016) to compare state of the art research methods on classifying DME and Normal patients using the SD-OCT volumes. There are six researches reviewed and it is found that each research follows the same process flow, but with different implementation methods in each stage. Note that in this research, input images do not go through feature detection and mapping, features representation is done by CNN and classification is involving Support Vector Machine (SVM) classifier. Therefore, the comparative study will only focus on these areas.

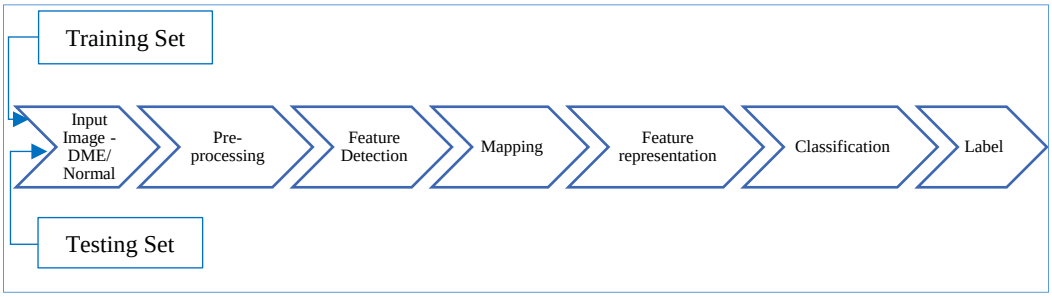

FIGURE 5 Common process flow to classify DME vs normal patients. Adapted from "Classifying DME vs Normal SD-OCT volumes: A review" by Massich et al. (2016) [11]

Image Pre-Processing

All input images go through *pre-processing* to improve the quality of images e.g. noise removal, flattening and cropping. A comparison study conducted on five noise removal methods (i.e. Block Matching 3D (BM3D), Bayesian Least Squares-Gaussian Scale Mixture (BLS-GSM), Complex Wavelet based Dictionary Learning methods (CWDL), Nonlinear Complex Diffusion Filter (NCDF), and Non-Local Means (NLM)) found that BM3D method performs better compared to others [12]. Out of 6 researches, 2 researches (i.e. Srinivasan *et al.* and Alsaih *et al.*) use BM3D filtering while Lemaitre *et al*. and Sankar *et al.* use NLM. The remaining two researches i.e. Venhuizen *et al.* and Liu *et al*. did not do denoising. Aside from that, 4 researches (i.e. Srinivasan *et al*., Alsaih *et al.*, Liu *et al.* and Sankar *et al.*) did image flattening and cropping (excluding Liu *et al.*'s).

Feature Detection

This process checks for the presence of certain features e.g. edges, blobs, and ridges at every point of the denoised images.

Mapping

There are two types of *mapping*: global – the image is considered as whole or local – the image is broken down into many parts

Feature Representation

Feature Representation is a process where features are learned by the particular feature extractor e.g. BoW, Histogram of Oriented Gradients (HOG) and Speeded-Up Robust Features (SURF).

Classification

This process determines the classes of each sample using a classifier. Srinivasan *et al.*, Alsaih *et al.* and Liu *et al.* use linear-SVM as the classifier while Venhuizen *et al*. and Lemaitre *et al.* use Random Forest (RF) as the image classification. Findings shown that SVM (83.02% accuracy) has higher performance compared to RF (82.63% accuracy) [13].

2.6.OCT

Sonka and Abramoff have conducted a review on retinal OCT quantitative analysis [3]. OCT capture images through the refraction of light as it passes through each tissue layer all with different refractive index, which also known as backscatters phenomenon. It is found that through 3-D OCT retinal imaging, retinal lesions can be identified thus, improving retinal diseases diagnosis as it allows viewing of higher resolutions of retinal layers. Figure 6 below shows an example of OCT volume with the segmentation of all retinal layers of the eye. Layer segmentation is crucial as it makes it easier to observe thickening of individual layers for disease diagnosis.

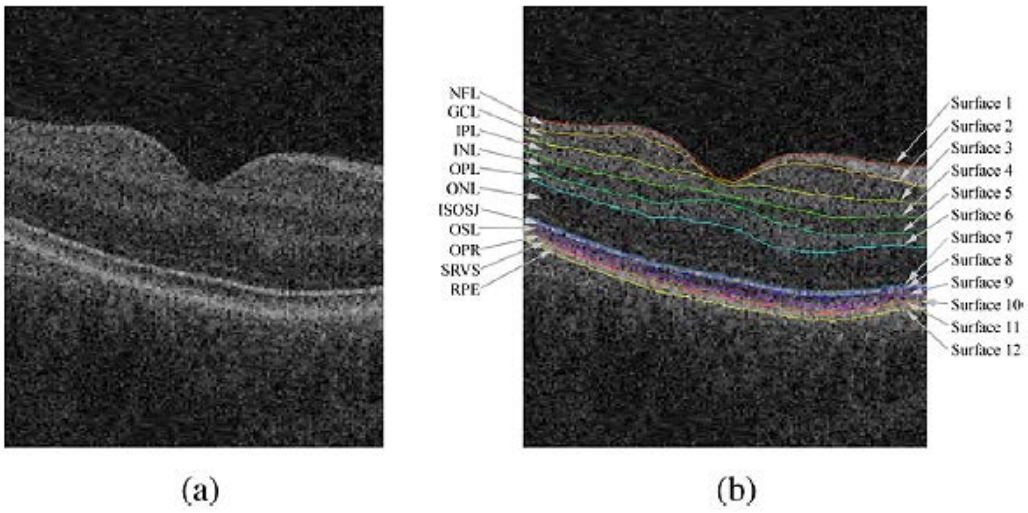

FIGURE 6 *Segmentation of 11 retinal layers. (a) X-Z image of the OCT volume. (b) Segmentation result of the retinal layers. Adapted from Quantitative Analysis of Retinal OCT by Sonka, M., & Abràmoff, M. D. (2016) [3].*

Using this approach, the borderlines can be used to crop out unnecessary pixels during pre-processing stage. Also, it is easier to identify the layers after segmentation. Similarly, Fu *et al.* has also done layer segmentation for Retinal Pigment Epithelium (RPE), Inner Segment Outer Segment Junction (ISOSJ) and Inner Limiting Membrane (ILM) which is above Nerve Fiber Layer (NFL). The figure below shows an example of layer segmentation of the retina [12].

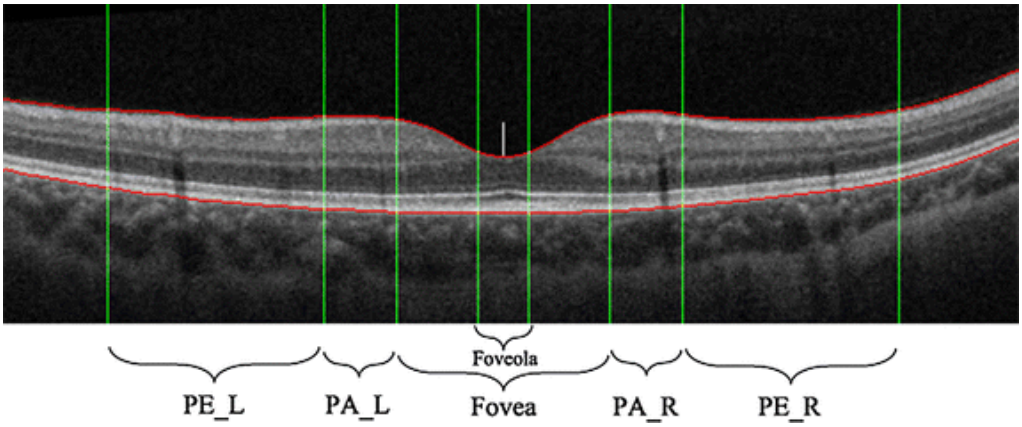

FIGURE 7 Layer segmentation of the ILM and RPE shown in red lines.The green lines indicate the medical macular segmentation: PE_L – Perifovea; PA_L – Parafovea; Fovea and Foveola. The white short line indicate the center position of the foveola. Adapted from "Retinal status analysis method based on feature extraction and quantitative grading in OCT images" by Fu, D. et al. (2016) [12].

2.7. DME

DME is a type of eye disease whereby the damage of blood vessels in the retina i.e. DR, when left untreated will cause the build-up of liquid in the macula leading to a swollen area on the retinal layer and consequently irreversible eye blindness.

A review was conducted by Trichonas and Kaiser (2014) to examine the patterns of DME on OCT volumes conducted by other researches [14]. It is found that there are 5 patterns of structural changes in DME: sponge-like retinal swelling which also known as Diffuse Retinal Thickening (DRT), Cystoid Macular Edema (CMD) and Serious Retinal Detachment (SRD), Posterior Hyaloidal Traction (PHT) without Tractional Retinal Detachment (TRD) and PHT with TRD.

A patient with DRT will have spongy like retina layer as a result of increased in retinal thickness. Figure 8 shows the OCT appearance of DRT.

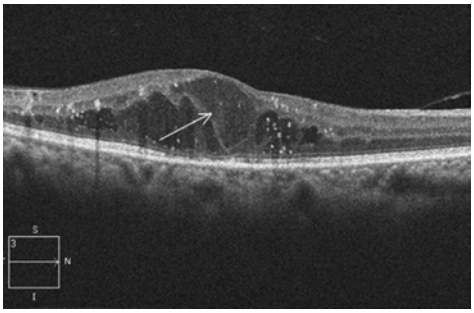

FIGURE 8 OCT appearance of DRT in a DME patient. Adapted from "Optical coherence tomography imaging of macular oedema" by Trichonas, G. & Kaiser, P.K. (2014) [14].

The next pattern change is CME which is caused by the accumulation of intraretinal fluid in well-defined spaces. Consequently, cystoid spaces are formed near the NFL. The figure below shows the OCT appearance of CME in a DME patient.

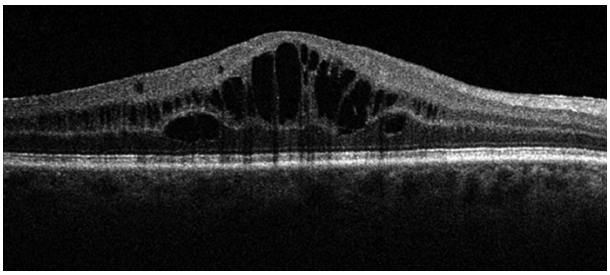

FIGURE 9 *OCT Scan of Retinal Layer with CME. Adapted from "OCT imaging of ME" by Trichonas, G., & Kaiser, P. K. (2014)* [14].

Serous detachment of the retina is usually due to chronic oedema. In these patients, the sensory retina is elevated and the cystic cavities often merge.

In this paper, we aim to apply deep learning onto Optical coherence tomography images of macular oedema. The method is based on transfer learning of a pre-trained network, AlexNet which has 1000 object categories [9]. Features are learned using CNN to describe the texture of OCT images and multiclass SVM classifier is used for

feature representation [15]. With the application of CNN, this project hopes to design an automated system for the diagnosis of DME and compare the results obtained from the image classification with the present researches.

CHAPTER 3
METHODOLOGY

3.1.Formal description to the solution

32 volumes of OCT images ($\boldsymbol{A} \in \mathbb{R}^{128\times1024\times512}$) and corresponding labels ($\boldsymbol{l} \subset \{normal, dme\}$) are used as the input for the network. The AlexNet downloaded from MatConvNet (link: http://www.vlfeat.org/matconvnet/pretrained/) is designed for 227×227 images with three channels (RGB images), thus all OCT images are first extracted from their volumes ($\boldsymbol{A} \in \mathbb{R}^{1024\times512}$), pre-processed ($\boldsymbol{A} \in \mathbb{R}^{227\times227}$) and contatenated thrice ($\boldsymbol{A} \in \mathbb{R}^{3\times227\times227}$). Through transfer learning, features are extracted with CNN using the pre-trained weights trained on ImageNet dataset [9]. Finally, the features extracted are used to train the multiclass SVM classifier, whereby 30% of images from each volume is training data whilst the remaining 70% is used for validation. The specificity (SP) and Sensitivity (SE) are evaluated at the end of each fully connected layer to evaluate the performance of the results. MATLAB R2016-a software will be used for image classifications with reference of algorithms from MATLAB [15].

3.2.Dataset

The dataset of retinal images is obtained by the Singapore Eye Research Institute (SERI). The dataset used in this study is obtained using CIRRUS TM (carl Zeiss Meditec, Inc., Dublin, CA) SD-OCT device. It contains 32 OCT volumes i.e. 16 volumes with DME diagnosed and 16 volumes normal cases. Each volume is captured with 128 B-scans with resolution of 1024px × 512px. All SD-OCT volumes are read and assessed by trained graders and identified as normal or DME cases based on evaluation of retinal thickening, hard exudates, intraretinal cystoid space formation and subretinal fluid [16]. Within the DME sub-set, many lesions have been selected to create a rather complete and diverse DME dataset. This dataset is currently being enriched through a collaboration with various clinics in Asia and Europe through the established network of Prof Meriaudeau.

3.3.Project Activities

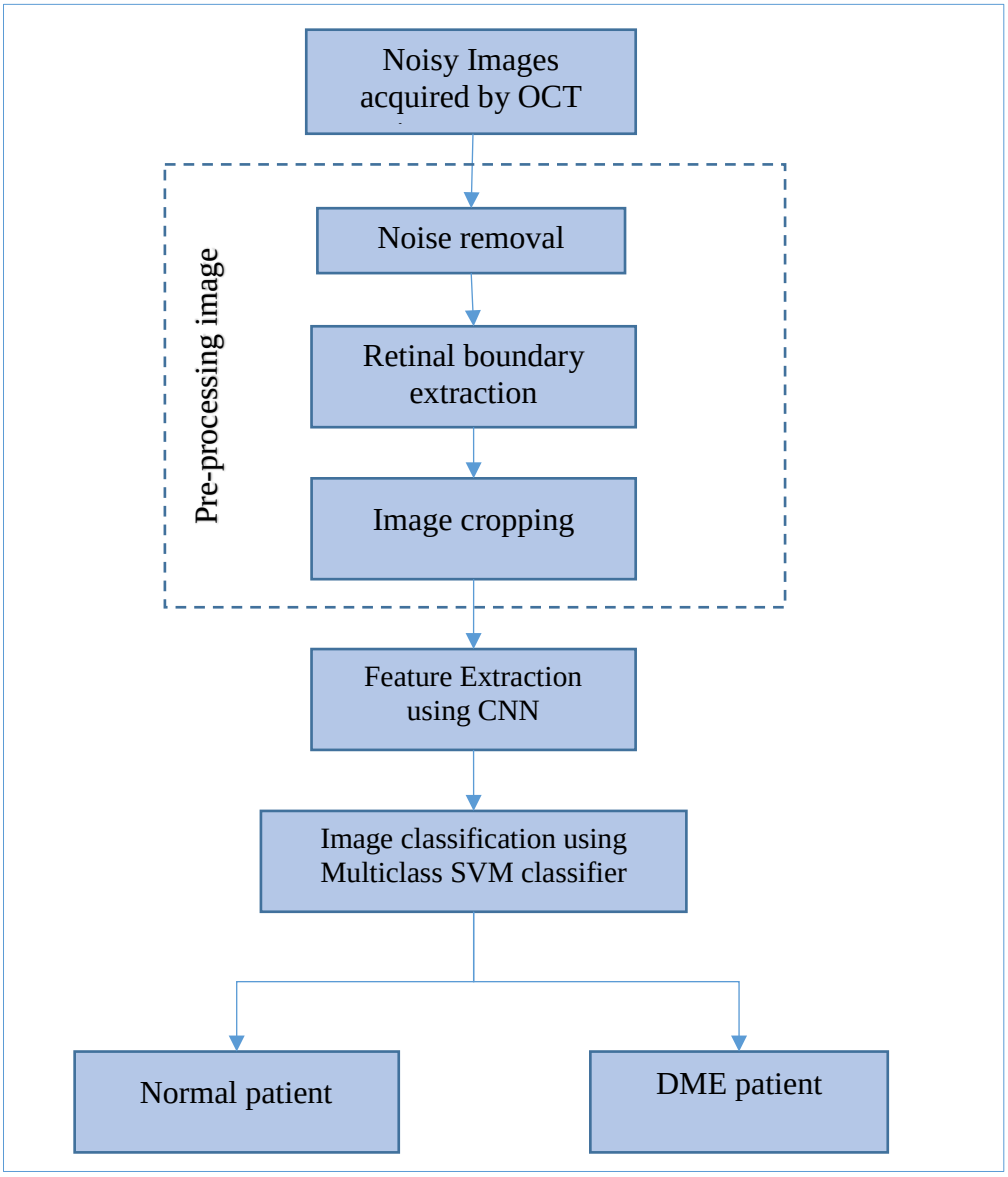

FIGURE 10 Process flow identification and implementation for image classification of DME

3.3.1. Noise removal using BM3D filtering

The dataset of the retinal OCT images is stored in volumes, whereby 16 volumes for normal patients and 16 volumes for patients with DME disease. In each volume, there are 128 images of size 512×1024 pixels. These images are extracted out from its volume. Then the images are resized, as the adopted AlexNet is designed for 227×227 images with three channels (for RGB images). Then, the images are concatenated thrice after BM3D filtering [17].

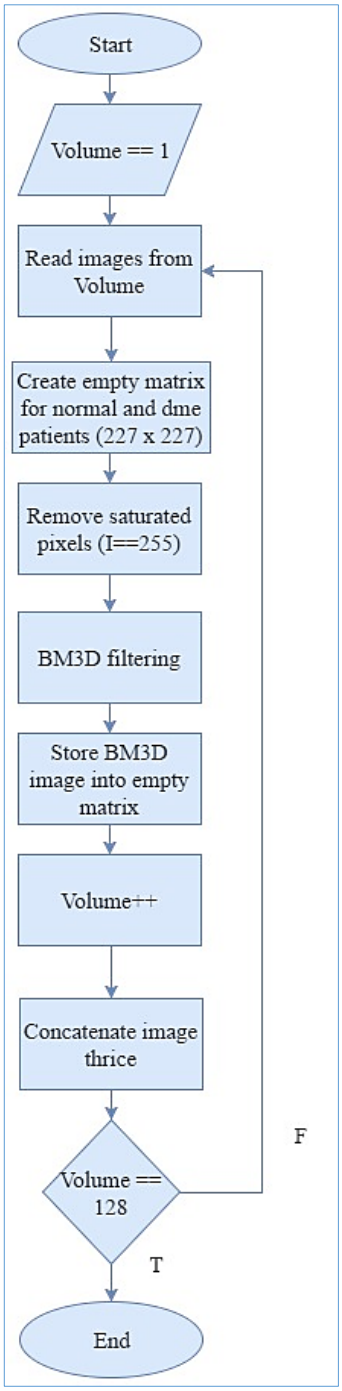

FIGURE 11 *Flowchart for BM3D filtering*

Figure 11 shows the flowchart for BM3D filtering. The images are read using the function “read_oct_volume”. Then each volume of 128 images are extracted, removed saturated pixels (255), resized, smoothened, and filtered. Images are concatenated three times, then stored into the image folder. This is because the images are all in grayscale. So, it is concatenated three times and each result is treated as channel information. The final images are of 227×227 ×3 images. Figure 12 shows the output of denoising.

(a) (b)

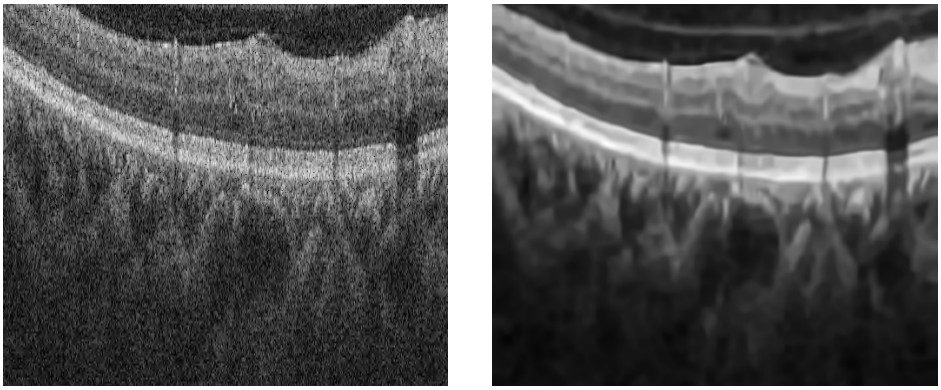

FIGURE 12 BM3D Filtering. (a) Noisy image. (b) BM3D Filtered

3.3.2. Image boundary extraction

Figure 14 shows the flowchart of image boundary extraction. Under this process, an image boundary line is drawn to indicate the Internal Limiting Membrane (ILM) and Retinal Pigment Epithelium (RPE) lower boundary to build the retinal reference model [12]. Figure 13 shows the output of the image boundary extraction.

(a) (b)

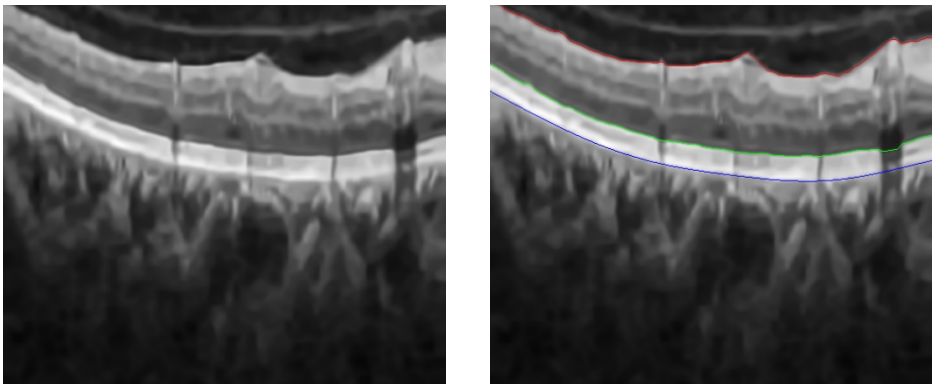

FIGURE 13 Image boundary extraction. (a) BM3D Image without boundary lines. (b) BM3D Image with boundary lines: ILM – Red Line; ISOS – Green Line; RPE – Blue Line.

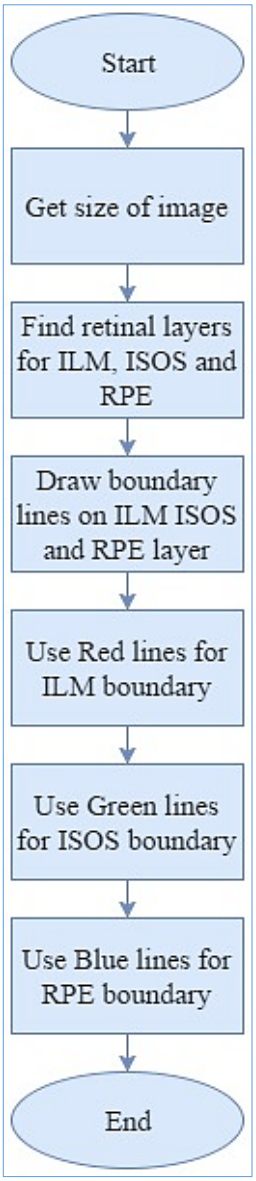

FIGURE 14 Flow chart for image boundary extraction

3.3.3. Image cropping

Note that the feature which distinguishes the difference between DME and Normal patients are the swelling of the retinal layer at the fovea. Consequently, the pixels which are not specific for normal or DME detection are therefore not relevant and should be removed. Using the boundary lines drawn during image boundary extraction stage, the image is cropped and stored. Figure 15 shows the output of the result of image cropping whilst Figure 16 describe the process of image cropping.

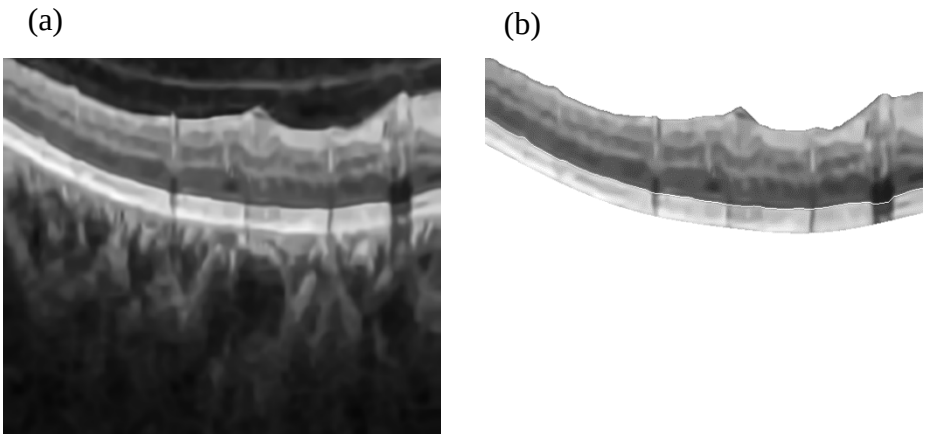

FIGURE 15 Image Cropping. (a) BM3D Image without cropping. (b) BM3D image after cropping

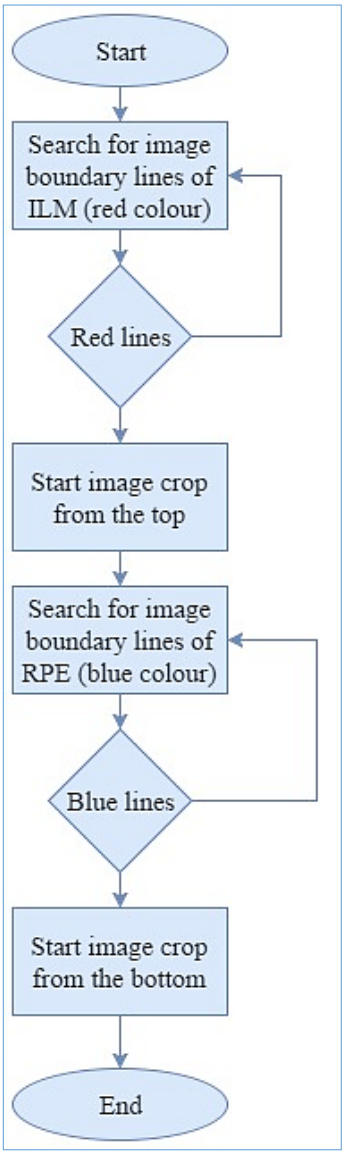

FIGURE 16 Flowchart for Image Cropping

3.3.4. Feature extraction

The AlexNet model is downloaded. The dataset is divided into training set and test set. Then, the training features are extracted using CNN. Each layer of a CNN produces a response, or activation to an input image. However, there are only a few layers within a CNN that are suitable for image feature extraction. The layers at the beginning of the network capture basic image features, such as edges and blobs [9]. Figure 17 shows the pre-trained weights of the first convolutional layer.

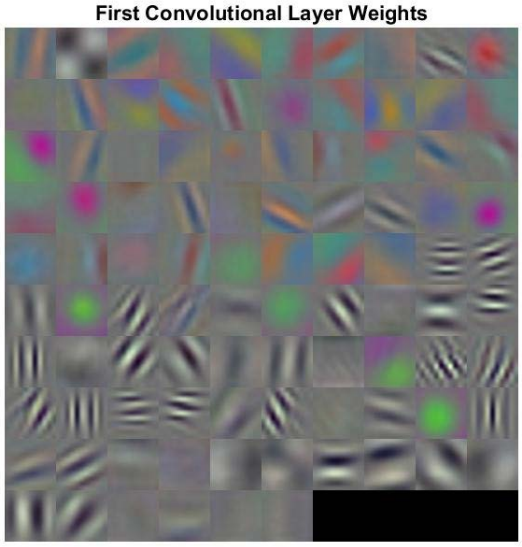

FIGURE 17 First convolutional layer of pre-trained weights

3.3.5. Image Classification using SVM classifier

After all features are extracted, they are used to train a classifier to perform image classification. The classifier used is a Support Vector Machine classifier. “Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. In this algorithm, we plot each data item as a point in n-dimensional space with the value of each feature being the value of a coordinate. Then, classification is performed by finding the hyper-plane that differentiate the two classes very well [18].

3.4. Gantt Chart and Project Key Milestone

TABLE 1 Gantt Chart and Project Key Milestone for FYP II

Course	ECB 4034 Final Year Project II														
Year	2017														
Details/Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1 Preparation															
1.1.Review FYPI	■														
1.2.Samples gathering		■	■												
2 Project Implementation															
2.1.Noise Removal using BM3D			■	■											
2.2.Retinal Boundaries Extraction				■	■	■	■								
2.3.Progress Report								◆							
2.4.Image cropping							■	■							
2.5.Feature extraction using CNN								■	■	■					
2.6.Image classification								■	■	■					
3 Documentation															
3.1.Pre-SEDEX Poster Presentation										◆					
3.2.Draft Final Report													◆		
3.3.Final Report (Dissertation)														◆	
3.4.Technical Paper														◆	
3.5.Project viva															◆

Labels:
◆ - Milestone
□ - Gantt Chart

CHAPTER 4
RESULTS AND DISCUSSION

4.1. Results

Four experiments are conducted on different types of datasets. Note that there are a total of 16 volumes each for DME and normal patients. Each volume has 128 images. Thus, there should be 2048 images for DME patients and 2048 images for normal patients. However, 13 images from volume 12 failed to undergo image cropping, for both noise removed and noisy images. Thus all dataset are normalized into 2035 images for all cases to prevent inconsistency results. For evaluation purposes, all the results are expressed in terms of Sensitivity (SE) and Specificity (SP) using 70% of data as testing and 30% of data as training for validation.

- Sensitivity (SE) – The ability of a test to correctly identify those with DME disease.
- Specificity (SP) – The ability of a test to correctly identify those without DME disease.

4.1.1. Experiment #1

Experiment #1 is carried out on raw datasets with no noise removal and image cropping. The figure below shows the example of input image for DME and normal patient. The results are reported in Table 2

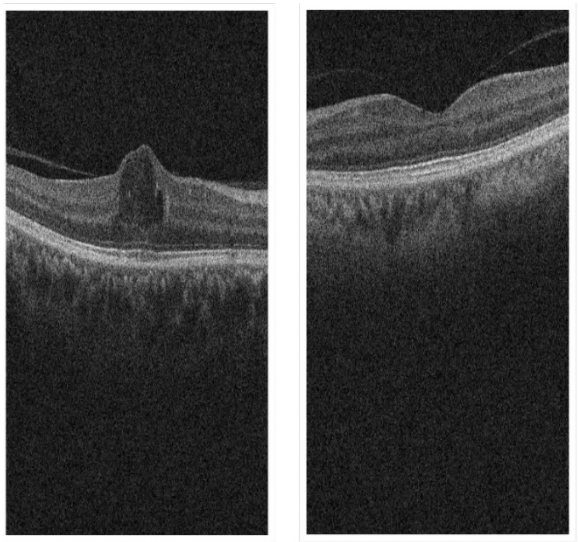

FIGURE 18 Dataset with no noise removal and image cropping. (a) DME Patient (b) Normal Patient

4.1.2. Experiment #2

Experiment #2 is carried out on datasets with noise removal but without image cropping. The figure below shows the example of input image for DME and normal patient. The results are reported in Table 2

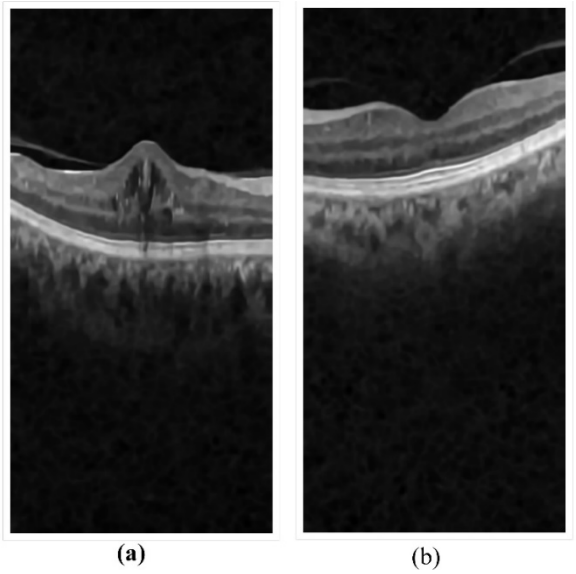

FIGURE 19 Dataset with noise removal but without image cropping. (a) DME patient (b) Normal Patient

4.1.3. Experiment #3

Experiment #3 is carried out on raw datasets with no noise removal but with image cropping. The figure below shows the example of input image for DME and normal patient. The results are reported in Table 2

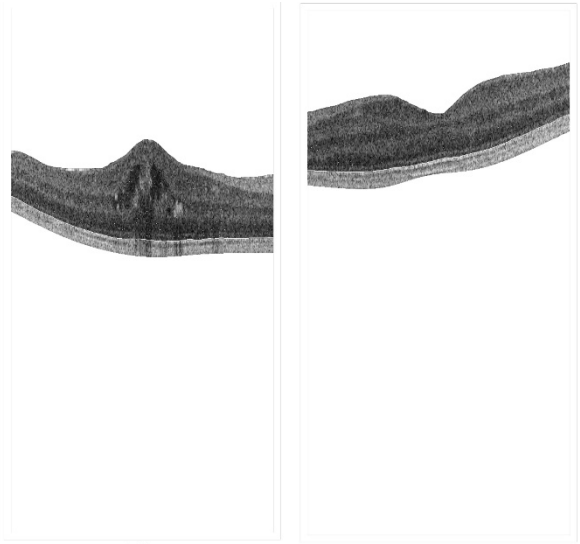

FIGURE 20 Dataset with no noise removal but with image cropping. (a) DME patient (b) Normal Patient

4.1.4. Experiment #4

Experiment #4 is carried out on datasets with noise removal and image cropping. The figure below shows the example of input image for DME and normal patient. It is seen that all the harsh edges are smoothened and a clearer image of the retinal layer is seen. Moreover, the irrelevant parts are also excluded from feature extraction. The results are reported in Table 2.

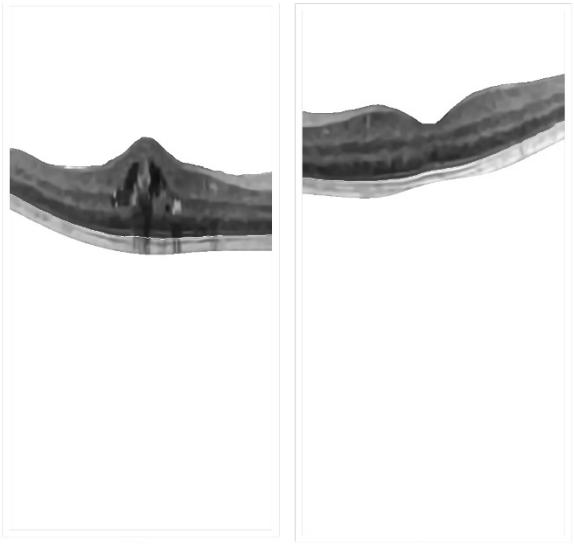

FIGURE 21 Dataset with noise removal and image cropping. (a) DME patient (b) Normal Patient

4.1.5. Retinal Boundary Extraction

Figure 22 shows the result of the retinal layer boundaries extraction for images without noise removal whilst Figure 23 is for images with noise removal. The red label indicates the internal limiting membrane (ILM) while the blue label indicates the outer Retinal Pigment Epithelium (RPE) layer.

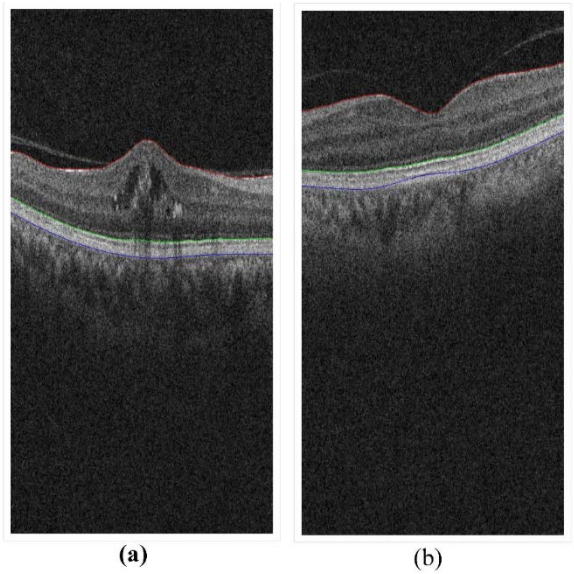

FIGURE 22 Retinal boundary extraction of image without noise removal. (a) DME Patient (b) Normal Patient

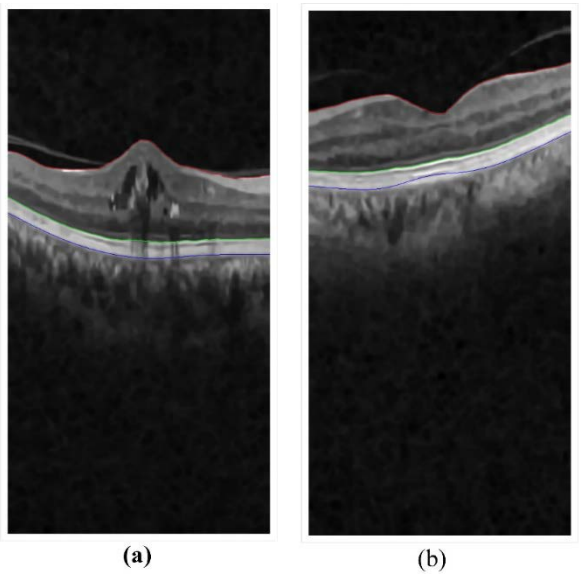

FIGURE 23 Retinal boundary extraction of image with noise removal. (a) DME Patient (b) Normal Patient

4.2. Discussion

Table 2 presents a summary of the classification performance for four different datasets in terms of SP and SE. It can be seen that the last fully connected layer, for noise removal and image cropping dataset, has the highest SP and SE compared to other dataset. Thus, it supports the claim that image pre-processing improves the classification performance. However, there are certain inconsistency of SP and SE, whereby, lower FC layer have higher SP and SE compared to higher FC layer. This is because the weights used are pre-trained from another dataset. Thus, for future recommendation, fine-tuning needs to be done to readjust the weight filters to obtain higher classification performance. The formulae used to calculate the SP and SE are:

$$SP = \frac{TN}{FP + TN} \qquad -(2)$$

$$where,$$

$$TN = True\ Negative$$

$$FP = False\ Positive$$

$$SE = \frac{TP}{TP + FN} \qquad -(3)$$

$$where,$$

$$TP = True\ Positive$$

$$FN = False\ Negative$$

TABLE 2 Summary of the classification performance in terms of accuracy, specificity (SP) and sensitivity (SE) in (%)[15]

Pre-processing		Fully Connected Layer	Accuracy	SP	SE
Noise removal	**Image Cropping**				
No	No	FC6	82.97	99.65	66.25
		FC7	90.06	80.75	98.95
		FC8	89.15	95.01	83.29
Yes	No	FC6	96.59	94.52	98.67
		FC7	94.94	97.00	92.89
		FC8	93.15	89.96	96.35
No	Yes	FC6	92.98	88.48	97.47
		FC7	90.14	81.32	98.95
		FC8	87.25	98.74	75.77
Yes	Yes	FC6	96.59	94.52	98.67
		FC7	81.07	99.93	62.20
		FC8	94.66	93.89	95.44

A comparison of classification performance between other researches is conducted:

TABLE 3 Comparison of Classification Performance between other researches in terms of SP and SE in (%) [11]

Ref	**Srinivasan *et al.***	**Alsaih *et al.***	**AlexNet**
Pre-processing	Noise Removal Flattening Cropping	Noise Removal Flattening Cropping	Noise Removal Cropping
SE	68.8	75.0	95.44
SP	93.8	87.5	93.89

Srinivasan *et al*. and Alsaih *et al.* researches are chosen for comparison because of similar pre-processing methods i.e. BM3D filtering and image classifier i.e. SVM. Flattening was not done on this paper as it is just a process of combining all layers into one layer. Both Srinivasan *et al.* and Alsaih *et al.* classify three categories from the OCT volumes i.e. DME, Age-related Macular Degeneration (AMD) and normal patients. Srinivasan *et al*. uses HOG feature extractor. In fact, Alsaih *et al.* is an extension of Srinivasan *et al*. research and they added Local Binary Patterns (LBP) to HOG to extract more features and PCA as feature representation to reduce the number of dimension [11]. The results were clearly promising as the SE and SP of Alsaih *et al.* is more consistent and relatively higher than the previous work of Srinivasan *et al.* However, when AlexNet was introduced, it can be seen that at the end of FC3, the SE and SP is more than 90%. The most significant improvement is the increase in SE performance of 38.72% from Srinivasan *et al.*'s

research and 27.25% from Alsaih *et al.*'s research. The increase in SP performance, meanwhile are between 1-8%. Nevertheless, using deep learning model and CNN as the feature extractor, the classification performance increases tremendously. Moreover, AlexNet model is the first deep learning model. This means that by using more advanced model with more convolutional layers should give higher performance. Therefore, for future works, we can experiment other deep learning model e.g. GoogleNet for further improvements of the image classification of DME using the results of AlexNet as a threshold.

CHAPTER 5
CONCLUSION AND RECOMMENDATIONS

In conclusion, the development of OCT which provides high resolution of retinal images for DME detection plus the adaptation of deep learning has proven to improve image classification with high performance of more than 90%. Deep learning application on DME detection using AlexNet has increase in SE performance of more than 20% compared to previous researches. This opens to a new, simple and effective method for early DME detection to aid ophthalmologists in biomedical technologies. The objectives of this project are accomplished.

For future works, we can explore image classification of other deep learning model to compare the performance on models with more convolutional layers. Other than that, one could also try using Leave-One-Patient Out Cross-Validation (LOPO-CV) strategy as validation to provide more promising results [16]. Moreover, fine-tuning technique should be explored to readjust the pre-trained weights to improve the classification performance of the current results as research findings have shown that using CNN as feature extractor does give almost 100% accuracy [15].

REFERENCES

[1] R. Lee, T. Y. Wong, and C. Sabanayagam, "Epidemiology of diabetic retinopathy, diabetic macular edema and related vision loss," *Eye and Vision*, vol. 2, p. 17, 09/3008/11/received09/01/accepted 2015.

[2] B. Cense, N. A. Nassif, T. C. Chen, M. C. Pierce, S.-H. Yun, B. H. Park, *et al.*, "Ultrahigh-resolution high-speed retinal imaging using spectral-domain optical coherence tomography," *Optics Express*, vol. 12, pp. 2435-2447, 2004/05/31 2004.

[3] M. Sonka and M. D. Abràmoff, "Quantitative analysis of retinal OCT," *Medical Image Analysis*, vol. 33, pp. 165-169, 2016.

[4] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, *et al.*, "Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning?," *IEEE Transactions on Medical Imaging*, vol. 35, pp. 1299-1312, 2016.

[5] M. Copeland. What's the Difference Between Artificial Intelligence, Machine Learning, and Deep Learning? [Online]. Available: https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436-444, 2015.

[7] Y. Taigman, M. Yang, M. A. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1701-1708.

[8] Saimadhu. Supervised and Unsupervised Learning [Online]. Available: http://dataaspirant.com/2014/09/19/supervised-and-unsupervised-learning/

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097-1105.

[10] H. Kataoka, K. Iwata, and Y. Satoh, "Feature Evaluation of Deep Convolutional Neural Networks for Object Recognition and Detection," *arXiv preprint arXiv:1509.07627*, 2015.

[11] J. Massich, M. Rastgoo, G. Lemaître, C. Cheung, T. Wong, D. Sidibé, *et al.*, "Classifying DME vs Normal SD-OCT volumes: A review," in *23rd International Conference on Pattern Recognition*, 2016.

[12] D. Fu, H. Tong, S. Zheng, L. Luo, F. Gao, and J. Minar, "Retinal status analysis method based on feature extraction and quantitative grading in OCT images," *BioMedical Engineering OnLine*, vol. 15, p. 87, 2016.

[13] P. Thamilselvan and J. Sathiaseelan, "A Comparative Study of SVM, RF and CART Algorithms for Image Classification," *Emerging Trends in Advanced Computing*, 2015.

[14] G. Trichonas and P. K. Kaiser, "Optical coherence tomography imaging of macular oedema," *British Journal of Ophthalmology*, vol. 98, pp. ii24-ii29, 2014.

[15] Image Category Classification Using Deep Learning [Online]. Available: https://www.mathworks.com/help/vision/examples/image-category-classification-using-deep-learning.html

[16] G. Lemaître, M. Rastgoo, J. Massich, S. Sankar, F. Mériaudeau, and D. Sidibé, "Classification of sd-oct volumes with lbp: Application to dme detection," 2015.

[17] S. P. K. Karri, D. Chakraborty, and J. Chatterjee, "Transfer learning based classification of optical coherence tomography images with diabetic macular edema and dry age-related macular degeneration," *Biomedical Optics Express*, vol. 8, pp. 579-592, 2017/02/01 2017.

[18] Y. Tang, "Deep learning using linear support vector machines," *arXiv preprint arXiv:1306.0239*, 2013.

APPENDICES

APPENDIX A – Noise Removal using BM3D Filtering

```
%create folder to store images
 for temp = 1:2
     for temp2 = 1:16
     mkdir(['BM3DImages/Train/',num2str(temp),'/',num2str(temp2)])
     mkdir(['BM3DImages/Test/',num2str(temp),'/',num2str(temp2)])
     end
 end
d=pwd;
%reading OCT volume
%Number of images = 128
%Dimension of image: X=512 Z=1024
s1 = 1024; %Googlenet=224/Alexnet=227
s2 = 512; %Googlenet=224/Alexnet=227
tic
for sub=1:16
    fprintf('Open volume #%d\n',sub);
    imagepath=[d,'\Deep_Learning_DME_FYP_Gen\Data\Dataset\data-seri'];
    normalvol =
read_oct_volume([imagepath,'\normal\',num2str(sub),'.img'],512,128,1024);
    dmevol =
read_oct_volume([imagepath,'\dme\',num2str(sub),'.img'],512,128,1024);
    k=0;
    normalimages=zeros(s1,s2,128);
    dmeimages=zeros(s1,s2,128);
    for i=1:128
        firstnormal = normalvol(:,:,i);
        firstdme = dmevol(:,:,i);
        %Error checking for same image file
        for i2=1:128
            checkdme = dmevol(:,:,i2);
            if (firstnormal==checkdme)
                error('ERROR: DME and Normal Same Image file');
            end
        end
        firstnormal(firstnormal==255)=10;
        firstdme(firstdme==255)=10;
        firstnormal = imresize(firstnormal,[s1,s2]);
        firstdme = imresize(firstdme,[s1,s2]);
        fprintf('INFO: Starting BM3D Denoising\n');
        [a,b]=BM3D(1,firstnormal,'np',1);
        fprintf('INFO: #%d Success NORMAL BM3D Denoising\n',i);
        [c,e]=BM3D(1,firstdme,'np',1);
        fprintf('INFO: #%d Success DME BM3D Denoising\n',i);
        normalimages(:,:,i)=(b);
        dmeimages(:,:,i)=(e);
    end
    %store images into files
    k=0;
     for temp =1:size(normalimages,3)
         k=k+1;

imwrite(cat(3,mat2gray(normalimages(:,:,temp)),mat2gray(normalimages(:,:,temp)),m
at2gray(normalimages(:,:,temp))),['BM3DImages/Train/1/',num2str(sub),'/',num2str(
temp),'.png'],'png')
         fprintf('INFO: #%d Success store NORMAL images\n',temp);
     end
    for temp =1:size(dmeimages,3)

imwrite(cat(3,mat2gray(dmeimages(:,:,temp)),mat2gray(dmeimages(:,:,temp)),mat2gra
y(dmeimages(:,:,temp))),['BM3DImages/Train/2/',num2str(sub),'/',num2str(temp),'.p
ng'],'png')
        fprintf('INFO: #%d Success store DME images\n',temp);

checkimage=imread(['BM3DImages/Train/2/',num2str(sub),'/',num2str(temp),'.png']);
    end
end
toc
```

APPENDIX B – Image Boundary Extraction

```
clear all;
close all;
for temp = 1:2
    for temp2 = 1:16
    mkdir(['RetinalBoundaries\Train\',num2str(temp),'\',num2str(temp2)])
    mkdir(['RetinalBoundaries\Test\',num2str(temp),'\',num2str(temp2)])
    end
end

d=pwd;
outputpath=[d,'\RetinalBoundaries\Train'];

for disease=1%:2
    for vol=12%:12
        for j=128%:128
            I_BM3D =
imread([d,'\Data\Vol_BM3DImages\Train\',num2str(disease),'\',num2str(vol),'\',num
2str(j),'.png']);
            src =
imread([d,'\Data\Vol_NoisyImage\Train\',num2str(disease),'\',num2str(vol),'\',num
2str(j),'.png']);
            if (I_BM3D(:,:,1)==src(:,:,1))
                error('ERROR: SAME IMAGE FILE');
            end
            I_gray = rgb2gray(I_BM3D);
            I_double = im2double(I_gray);
            [m,n]=size(I_double);
            %I_BM3D = double(I_BM3D);
            [retinalLayers, params] = getRetinalLayers(I_double);
            ILM = zeros(512,1); ISOS = zeros(512,1);RPE = zeros(512,1);

            for i = 1:3
                if isequal(retinalLayers(i).name,'ilm')
                    start = find(retinalLayers(i).pathY == 2);
                    ILM = retinalLayers(i).pathX(start:start+511);
                elseif isequal(retinalLayers(i).name,'isos')
                    start = find(retinalLayers(i).pathY == 2);
                    ISOS = retinalLayers(i).pathX(start:start+511);
                elseif isequal(retinalLayers(i).name,'rpe')
                    start = find(retinalLayers(i).pathY == 2);
                    RPE = retinalLayers(i).pathX(start:start+511);
                end
            end

            for i=1:n
                I_BM3D(ILM(i),i,1) = 255; I_BM3D(ILM(i),i,2) = 0;
I_BM3D(ILM(i),i,3) = 0;
                I_BM3D(ISOS(i),i,1) = 0; I_BM3D(ISOS(i),i,2) = 255;
I_BM3D(ISOS(i),i,3) = 0;
                I_BM3D(RPE(i),i,1) = 0; I_BM3D(RPE(i),i,2) = 0;
I_BM3D(RPE(i),i,3) = 255;
            end

imwrite(I_BM3D,[d,'\RetinalBoundaries\Train\',num2str(disease),'\',num2str(vol),'
\',num2str(j),'.png'],'png');
            fprintf('Success Disease#%d, Vol#%d, Image#%d\n',disease,vol,j);
        end
    end
end
```

APPENDIX C – Image Cropping

```
for temp = 1:2
    for temp2 = 1:16
    mkdir(['BM3DCropImage/',num2str(temp),'/',num2str(temp2)])
    end
end
d=pwd;
outputpath=[d,'\BM3DCropImage'];
fontSize = 20;
for disease=2
    %counter=1;
    for vol=1:12
        for imagenumber=1:128
            C =
imread([d,'\Data\Vol_NoisyImageRetinalBoundaries\Train\',num2str(disease),'\',num
2str(vol),'\',num2str(imagenumber),'.png']);
            [m n z]=size(C);
            red=zeros(n,2);
            blue=zeros(n,2);
            k=1;
            for row=1:m
                for col=1:n
                    if C(row,col,1)==255 && C(row,col,2)==0 && C(row,col,3)==0
                        red(k,:)=[row col];
                        k=k+1;
                    end
                end
            end
            k=1;
            for backrow=1:m
                for col=1:n
                    if C(backrow,col,1)==0 && C(backrow,col,2)==0 &&
C(backrow,col,3)==255
                        blue(k,:)=[backrow col];
                        k=k+1;
                    end
                end
            end
            crop=zeros(m,n,3);
            for i=1:n
                if(red(i,:)==0) | (blue(i,:)==0)
                    break;
                else
                crop(red(i),red(i,end),1)=255;
                crop(blue(i),blue(i,end),3)=255;
                end
            end
            for i=1:n
                for j=1:n
                    if (red(i,2)==blue(j,2))
                        startpoint=red(i,1);
                        endpoint=blue(j,1);
                        for imagepoint=startpoint:endpoint
                            crop(imagepoint,red(i,2),1)=C(imagepoint,red(i,2),1);
                            crop(imagepoint,red(i,2),2)=C(imagepoint,red(i,2),2);
                            crop(imagepoint,red(i,2),3)=C(imagepoint,red(i,2),3);
                        end
                    end
                end
            end
            crop(crop==0)=255;
            crop=uint8(crop);

imwrite(crop,[d,'\BM3DCropImage\',num2str(disease),'\',num2str(vol),'\',num2str(i
magenumber),'.png'],'png');
            fprintf('Success Disease#%d, Vol#%d,
Image#%d\n',disease,vol,imagenumber);
            %counter=counter+1;
        end
    end
end
```

APPENDIX D – Image Classification

```
function CaltechPretunedCnn
%Get GPU device information
deviceInfo = gpuDevice;

%Check the GPU compute compatibility
Compatibility = str2double(deviceInfo.ComputeCapability);
assert(Compatibility > 3.0, 'GPU capability lower than 3.0');
%Create folder for image files
RootFolder = fullfile('C:','Users','User','Documents','fyp');
Categories = {'dme','normal'};

%store image data once read
imds = imageDatastore(fullfile(RootFolder, 'Preprocess_Images', 'BM3DImageGeneral',...
    Categories), 'LabelSource','foldernames')
LabelCount = countEachLabel(imds)

%balance out the label count values with the min values
MinimumLabelCount = 2035;%min(LabelCount{:,2});

%trim the data count for each label following min value
imds = splitEachLabel(imds, MinimumLabelCount, 'randomize');
LabelCount = countEachLabel(imds)

%display one of the image for each label
dme = find(imds.Labels == 'dme',1);
normal = find(imds.Labels == 'normal',1);

%Load pretrained CNN
CnnMatFile = fullfile(RootFolder, 'imagenet-caffe-alex.mat');
ConvNet = helperImportMatConvNet(CnnMatFile)
%view CNN architecture
ConvNetArc = ConvNet.Layers

%Identify number of class names for ImageNet classification task
ClassNameNo = numel(ConvNet.Layers(end).ClassNames)
%Create a read function imds to avoid re-saving all images to
%RGB format
imds.ReadFcn = @(filename)ReadAndPreprocessImage(filename);
    function Iout = ReadAndPreprocessImage(filename)
        I = imread(filename);
        %some images are grayscale. Replicate the image 3 times to create
        %an RGB image.
        if ismatrix(I)
            I=cat(3,I,I,I);
        end

        %resize the image as required for the CNN
        Iout = imresize(I,[227,227]);
    end
%prepare training set and test image sets
[TrainingSet, TestSet] = splitEachLabel(imds, 0.3, 'randomize');
%extract training features using CNN
%get the network weights for the second convolutional layer
Weight1 = ConvNet.Layers(2).Weights;
%Scale and resize the weights for visualization
Weight1 = mat2gray(Weight1);
Weight1 = imresize(Weight1,5);

%extract features using activation method at the layer before
%classification layer
FeatureLayer = 'fc7';
TrainingFeatures = activations(ConvNet, TrainingSet, FeatureLayer, 'MiniBatchSize', 32,
'OutputAs', 'Columns');

%get training labels from TrainingSet
TrainingLabels=TrainingSet.Labels;

%Train multiclass SVM classifier using a fast linear solver and set
%'ObservationsIn' to 'columns' to match the arrangement used for training
%features.
Classifier = fitcecoc(TrainingFeatures, TrainingLabels, 'Learners', 'Linear', 'Coding',
'onevsall', 'ObservationsIn', 'columns');

%Evaluate classifier
%extract test features using CNN
TestFeatures = activations(ConvNet, TestSet, FeatureLayer, 'MiniBatchSize', 32);
%pass CNN image features into trained classifier
PredictedLabels = predict(Classifier, TestFeatures);
%get the known labels
TestLabels = TestSet.Labels;

%tabulate the result using a confustion matrix
ConfMat = confusionmat(TestLabels, PredictedLabels);

%convert confusion matrix into percentage form
ConfMat = bsxfun(@rdivide, ConfMat, sum(ConfMat,2))

%display the mean accuracy
AvgAccuracy = mean(diag(ConfMat))

end
```