5-31-2020

Deep learning for quantitative motion tracking based on optical coherence tomography

Peter Abdelmalak
New Jersey Institute of Technology

Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

ABSTRACT

DEEP LEARNING FOR QUANTITATIVE MOTION TRACKING BASED ON OPTICAL COHERENCE TOMOGRAPHY

by
Peter Abdelmalak

Optical coherence tomography (OCT) is a cross-sectional imaging modality based on low coherence light interferometry. OCT has been widely used in diagnostic ophthalmology and has found applications in other biomedical fields such as cancer detection and surgical guidance.

In the Laboratory of Biophotonics Imaging and Sensing at New Jersey Institute of Technology, we developed a unique needle OCT imager based on a single fiber probe for breast cancer imaging. The needle OCT imager with sub-millimeter diameter can be inserted into tissue for minimally invasive *in situ* breast imaging. OCT imaging provides spatial resolution similar to histology and has the potential to become a device to perform virtual biopsy to fast and accurate breast cancer diagnosis, because abnormal breast tissue and normal breast tissue have different characteristics in OCT image. The morphological features of OCT image are related to the microscopic structure of the tissue and the speckle pattern in OCT image is related to cellular/subcellular optical properties of the tissue. In addition, depth attenuation of OCT signal depends on the scattering and absorption properties of the tissue. However, the above described OCT image features are at different spatial scales and it is challenging for human visualization to effectively recognize these features for tissue classification. Particularly, our needle OCT imager, given its simplicity and small form factor, does not have a mechanical scanner for beam steering and relies on

manual scan to generate 2D images. The nonconstant translation speed of the probe in manual scanning inevitably introduces distortion artifacts in OCT imaging, which further complicates the tissue characterization task.

OCT images of tissue samples provide comprehensive information about the morphology of normal and unhealthy tissue. Image analysis of tissue morphology can help cancer researchers develop a better understanding of cancer biology. Classification of tissue images and recovering distorted OCT images are two common tasks in tissue image analysis.

In this master thesis project, a novel deep learning approach is investigated to extract beam scanning speed from different samples. Furthermore, a novel technique is investigated and tested to recover distorted OCT images. The long-term goal of this study is to achieve robust tissue classification for breast cancer diagnosis, based on a simple single fiber OCT instrument.

The deep learning network utilized in this study depends on Convolutional Neural Network (CNN) and Naïve Bayes Classifier. For image retrieval, we used algorithms that extract, represent and match common features between images. The CNN network achieved accuracy of 97% in tissue type and scanning speed classification, while the image retrieval algorithms achieved very high-quality recovered image compared to the reference image.

DEEP LEARNING FOR QUANTITATIVE MOTION TRACKING BASED ON OPTICAL COHERENCE TOMOGRAPHY

by
Peter Abdelmalak

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Electrical Engineering

Helen and John C. Hartmann
Department of Electrical and Computer Engineering

May 2020

APPROVAL PAGE

DEEP LEARNING FOR QUANTITATIVE MOTION TRACKING BASED ON OPTICAL COHERENCE TOMOGRAPHY

Peter Abdelmalak

Dr. Xuan Liu, Thesis Advisor — Date
Assistant Professor of Electrical and Computer Engineering, NJIT

Dr. Ali Abdi, Committee Member — Date
Professor of Electrical and Computer Engineering, NJIT

Dr. Cong Wang, Committee Member — Date
Assistant Professor of Electrical and Computer Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Peter Abdelmalak

Degree: Master of Science

Date: May 2020

Undergraduate and Graduate Education:

- Master of Science in Electrical Engineering
  New Jersey Institute of Technology, Newark, NJ, 2020

- Bachelor of Science in Electronics and Communication Engineering
  Arab Academy for Science, Technology, and Maritime Transport
  Cairo, Egypt, 2015

Major: Electrical Engineering

- *To my beloved wife and son*

ACKNOWLEDGMENT

I would like to express the deepest gratitude and appreciation to my committee chair and research advisor, Professor Xuan Liu for giving me the opportunity to do research and providing invaluable guidance throughout this research.

I would like to thank my committee members, Professor Ali Abdi and Professor Cong Wang for their true guidance, encouragement and insightful comments.

I am extremely grateful to my family: I would like to thank my wife Marina Gerges and my parents Margeret Khala and George Abdelmalak for their love, support and prayers.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Goals and Overview

The experiments performed in this study have 2 main goals; The first goal is to build and test the accuracy of a CNN deep learning model in classifying the tissue type and its lateral scanning speed based on OCT. The second goal is to solve the artifact problem occur when using the Single Fiber OCT Imager, then classify its tissue type and lateral scanning speed, which would help in predicting the speed of any manually scanned OCT samples.

The CNN deep learning model wouldn't be able to achieve the second goal, to the best of our knowledge. We need powerful algorithms that can detect, extract and match features of 2 or more images using their descriptors, then find a transformation corresponding to their matching features. After doing the proper research, we find that 3 of computer vision existing algorithms [26-29] may help in achieving the second goal of this study.

This thesis is organized as follows. The rest of chapter 1 provides a brief introduction about Optical Coherence Tomography (OCT), deep learning and the architecture of convolutional neural networks. Chapter 2 discuss some of the related OCT studies using deep learning. Chapter 3 discusses the materials and methods used in this study. Experimental results and related discussions are presented in chapter 4, respectively. Chapter 5 has our concluding remarks.

1.2 Optical Coherence Tomography (OCT)

Optical coherence tomography (OCT) is a high-resolution optical imaging instrument used in medical fields [1]. For many clinical applications, a hand-held OCT system could be specifically useful; it would offer physicians more freedom to access imaging areas of interest. In a hand-held OCT system, it is advantageous to have a vigorous and lightweight probe which can image full anatomical structures with a large field-of-view [1].

1.2.1 Introduction

Optical coherence tomography (OCT) is a non-invasive imaging technique which generates cross-sectional images of tissue with high resolution. Therefore, it is especially valuable in organs, where traditional microscopic tissue diagnosis by means of biopsy is not available—such as the human eye [2].

Since OCT is completely noninvasive, it provides in vivo images without affecting the tissue that is imaged. High scanning rates and robust signal processing allows for image visualization in real time and at video rate. The resolution of OCT is much higher than that of other medical imaging methods like ultrasound or magnetic resonance imaging (MRI). It merges an axial resolution with a lateral resolution comparable to confocal scanning laser ophthalmoscopy. Typically, OCT systems have high resolution of 1-10 μm [3].

One of the major advantages of OCT is that it provides real-time non-invasive diagnostic feedback about tissue structure. This information, for example, is useful to physicians to assist them in making real-time decisions during time-sensitive

diagnostic and surgical procedures such as needle biopsy, minimally-invasive surgery or procedures, or the removal of tumor tissue. Despite the real-time, high-resolution imaging capabilities of OCT, the feasibility and success of implementing OCT in clinical and intraoperative conditions may largely be determined by the adaptability of OCT instrumentation and image acquisition techniques to make it more 'surgeon friendly'. While real-time portable OCT systems have been successfully demonstrated for clinical research over the last few years, the technology has yet to evolve towards providing an imaging capability that can be readily used by physicians under the diverse set of conditions encountered in an operating room [3].

1.2.2 Fundamental Idea of OCT

OCT is often compared to medical ultrasound because of the similar working principles. Both medical imaging techniques direct waves to the tissue under examination, where the waves echo off the tissue structure. The back reflected waves are analyzed and their delay is measured to reveal the depth in which the reflection occurred. OCT uses light in the near-infrared, which travels much faster than ultrasound. The delays of the back reflected waves cannot be measured directly, so a reference measurement is used. Using an interferometer, part of the light is directed to the sample and another part is sent to a reference arm of a known length [2].

The idea of low-coherence interferometry is the underlying concept for all OCT implementations. Temporal coherence is a characteristic of a light source and characterizes the temporal continuity of a wave pulses sent out by the source and measured at a given point in space. Wave pulses emerging from a light source of low temporal coherence maintain a fixed phase relation only over a very limited time interval corresponding to a limited travel range, the coherence length. A light source with a broad spectral bandwidth is composed of a range of wavelengths. Such a broadband source has low coherence, while monochromatic laser light has a narrow spectral line and features a coherence length of at least several meters. An interferometer splits light, coming from a source, into two separate paths and combines the light coming back from the two paths at the interferometer output. There, under certain conditions, interference can be observed: coherent waves superimpose and their electromagnetic field amplitudes add constructively (i.e. they reinforce each other) or destructively (i.e. they cancel out each other) or meet any condition in between. The associated light intensity can be measured as an electrical signal

using a photo detector. This signal is a function of the difference in optical path length between both arms. For a low coherent light source (like a pulsed laser source) interference is only possible if the optical paths are matched to be equal in length within the short coherence length of the source, which usually is in the order of micrometers [2].

1.2.3 OCT Technical Understanding

In the first implementation of OCT, the reference length was modulated for each depth scan and the record of the intensity of the combined light at the sensor gave the reflectance profile of the sample and the main setup is shown in figure.1.1.

As depicted, the light of a low-coherence source is guided to the interferometer, which in this example is a fiber-based implementation. In a system using bulk optics the fiber coupler is replaced by a beam splitter. The input beam is split into the sample beam and into the reference beam travelling to a mirror on a translational stage. The back-reflected light from each arm is combined and only interferes if the optical path lengths match and therefore the time travelled by the light is nearly equal in both arms. Modulations in intensity, also called interference fringe bursts, are detected by the photodiode. The amount of back-reflection or back-scattering from the sample is derived directly by the envelope of this signal (see figure 1.1, lower row).

For each sample point, the reference mirror is scanned in depth (z) direction and the light intensity is recorded on the photo detector. Thereby a complete depth profile of the sample reflectivity at the beam position is generated, which—in analogy to ultrasound imaging—is called A-scan (amplitude scan).

To create a cross-sectional image (or B-Scan), the sample beam is scanned laterally across the sample. This abbreviation originated in ultrasound imaging, where B-Scan means brightness scan.

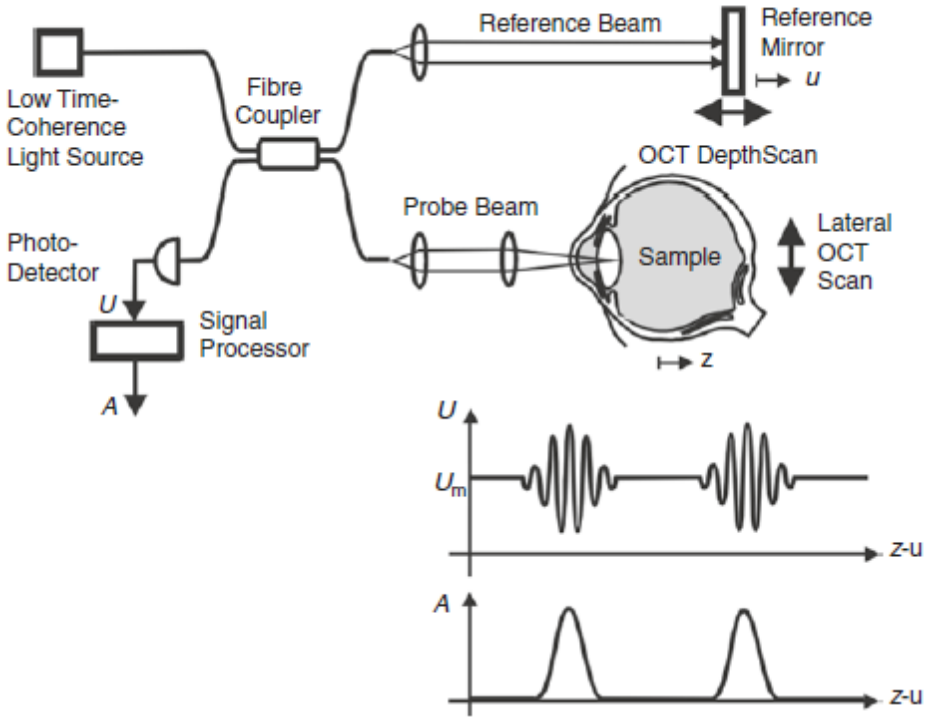


Figure 1.1 Time domain OCT: Light from the light source is split into the reference beam and the central beam. Back reflected light from both arms is combined again and recorded by the detector. To record one depth profile of the sample (A-scan) the reference arm needs to be scanned. This has to be repeated for each lateral scan position. Figure reprinted from [2].

1.3 Deep Learning

When computers were first conceived, people wondered whether such machines might become intelligent, over a hundred years before one was built. Today, artificial intelligence (AI) is a thriving field with many practical applications and active research topics. We look to smart software to replace routine labor, understand speech or images, make diagnoses in medicine and support basic scientific research.

Several artificial intelligence projects have sought to hard-code knowledge about the world in formal languages. A computer can reason about statements in these formal languages automatically using logical inference rules. This is known as the knowledge base approach to artificial intelligence. The difficulties faced by systems relying on hard-coded knowledge recommend that AI systems need the ability to obtain their own knowledge, by learning patterns from raw data. This capability is known as machine learning [4].

The introduction of machine learning allowed computers to address problems requiring knowledge of the real world and make decisions that appear subjective. A simple machine learning algorithm called logistic regression can determine whether to recommend cesarean delivery. A simple machine learning algorithm called Naive Bayes can separate legitimate e-mail from spam e-mail. We will use Naïve Bayes classifier in this study as we will see later in chapter 3

Many AI tasks can be solved by designing the right set of features to select for that task, then providing these features to a simple machine learning algorithm. For example, a useful feature for speaker identification from sound is an estimate of the size of speaker's vocal tract. It therefore gives a strong clue as to whether the speaker is a man, woman, or child [4].

However, for many tasks, it is difficult to determine what features should be selected. For example, suppose that we would like to write a program to detect cars in photographs. We know that cars have wheels, so we might like to use the presence of a wheel as a feature. Unfortunately, it is difficult to describe exactly what a wheel looks like in terms of pixel values. A wheel has a simple geometric shape, but its image may be complicated by shadows falling on the wheel, the sun glaring off the metal parts of the wheel, the fender of the car or an object in the foreground obscuring part of the wheel, and so on [4].

One solution to this problem is to use machine learning to discover the mapping from representation to output and the representation itself. This approach is known as representation learning. Learned representations often result in better performance than can be obtained with hand-designed representations. They also allow AI systems to quickly adjust to new tasks. A representation learning algorithm can discover a good set of features for a simple task in minutes, or a complex task in hours to months. Manually designing features for a complex task requires a great deal of human time and effort [4].

Deep learning solves this problem in representation learning by introducing representations that are expressed in terms of other easier representations. Deep learning let the computer decide and build complex concepts out of easier concepts. Figure 1.2 shows how a deep learning system can represent the concept of an image of a person by combining easier concepts, such as corners and contours, which are in turn defined in terms of edges [4].

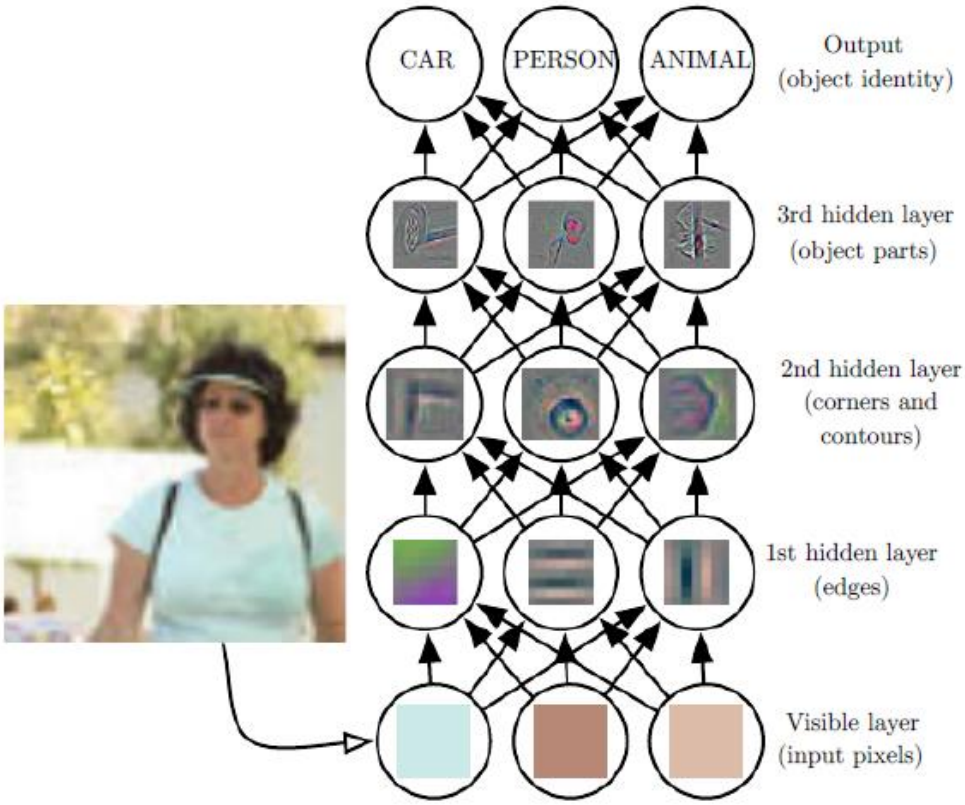


Figure 1.2 Illustration of a deep learning model. It is difficult for a computer to understand the meaning of raw sensory input data, such as this image represented as a collection of pixel values. Deep learning resolves this difficulty by breaking the desired complicated mapping into a series of nested simple mappings, each described by a different layer of the model. The input is presented at the **visible layer**. Then a series of **hidden layers** extracts features from the image. The images here are visualizations of the kind of feature represented by each hidden unit. Given the pixels, the first layer can easily identify edges, by comparing the brightness of neighboring pixels. Given the first hidden layer's description of the edges, the second hidden layer can easily search for corners and extended contours, which are collections of edges. Given the second hidden layer's description of the image in terms of corners and contours, the third hidden layer can detect entire parts of specific objects, by finding specific collections of contours and corners. Finally, this description of the image in terms of the object parts it contains can be used to recognize the objects present in the image. Figure reprinted from [4].

1.4 Convolutional Neural Networks

Deep learning can be defined as neural networks with a large number of parameters and layers in one of four fundamentals network architectures:

- Unsupervised Pre-trained Networks
- Recursive Neural Networks
- Recurrent Neural Networks
- Convolutional Neural Networks

In this thesis study, we are interested in the latter. Generally, convolution is an operation on two functions of a real valued argument. For the sake of simplification, we give the example below:

Suppose we are tracking the location of a spaceship with a laser sensor. Our laser sensor provides a single output $x(t)$, the position of the spaceship at time t. Both x and t are real-valued, i.e., we can get a different reading from the laser sensor at any instant in time.

Now suppose that the laser sensor is noisy. To obtain a less noisy estimate of the spaceship's location, we would like to average multiple measurements. Of course, more recent measurements are more relevant, so we want this to be a weighted average that gives more weight to recent measurements. We can do this with a weighting function $w(a)$, where a is the age of a measurement. If we apply such a weighted average operation at every moment, we obtain a new function s providing a smoothed estimate of the position of the spaceship:

$$s(t) = \int x(a)w(t-a)da \tag{1.1}$$

This operation is called **convolution**. The convolution operation is typically denoted with an asterisk:

$$s(t) = (x * w)(t) \tag{1.2}$$

In our example, *w* needs to be a valid probability density function, or the output is not a weighted average. Also, *w* needs to be 0 for all negative arguments, or it will predict the future, which is impossible [5].

In convolutional network terminology, the first argument (in this example, the function *x*) to the convolution is often referred to as the **input** and the second argument (in this example, the function *w*) as the **kernel**. The output is sometimes referred to as the **feature map**. In our example, the idea of a laser sensor that can provide measurements at every instant in time is not realistic. Usually, when we work with data on a computer, time will be discretized, and our sensor will provide data at regular intervals. In our example, it might be more realistic to assume that our laser provides a measurement once per second. The time index *t* can then take on only integer values. If we now assume that *x* and *w* are defined only on integer *t*, we can define the discrete convolution:

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a) \quad (1.3)$$

In machine learning applications, the input is usually a multidimensional array of data and the kernel is usually a multidimensional array of parameters that are adapted by the learning algorithm. We will refer to these multidimensional arrays as tensors. Because each element of the input and kernel must be explicitly stored separately, we usually assume that these functions are zero everywhere but the finite set of points for which we store the values. This means that in practice we can implement the infinite summation as a summation over a finite number of array elements [5].

Finally, we often use convolutions over more than one axis at a time. For example, if we use a two-dimensional image *I* as our input, we probably also want to use a two-dimensional kernel *K*:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n). \quad (1.4)$$

The fundamental difference between a densely connected layer and a CNN is this: Dense layers learn global patterns in their input feature space, whereas CNN learn local patterns

This key characteristic gives CNN two interesting properties [5]:

1. The patterns they learn are translation invariant. After learning a certain pattern in the lower-right corner of a picture, a CNN can recognize it anywhere: for example, in the upper-left corner. A densely connected network would have to learn the pattern anew if it appeared at a new location. This makes CNN data efficient when processing images (because the visual world is fundamentally translation

invariant): they need fewer training samples to learn representations that have generalization power.

2. They can learn spatial hierarchies of patterns (see figure 1.3). A first convolution layer will learn small local patterns such as edges, a second convolution layer will learn larger patterns made of the features of the first layers, and so on. This allows CNN to efficiently learn increasingly complex and abstract visual concepts

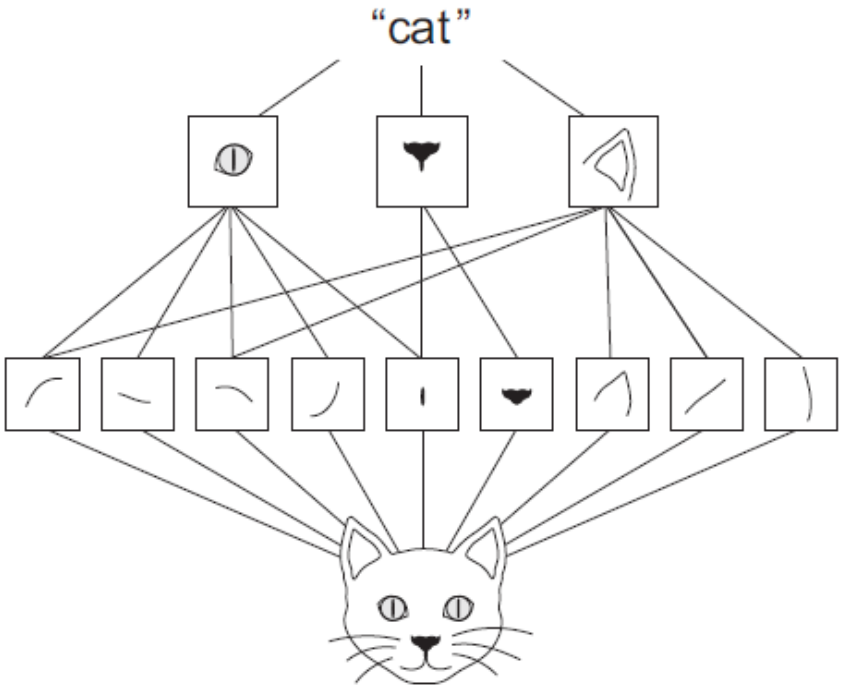


Figure 1.3 The visual world forms a spatial hierarchy of visual modules. Figure reprinted from [4].

1.4.1 Convolutional Layer

The most important building block of a CNN is the convolutional layer: neurons in the first convolutional layer are not linked to every single pixel in the input image but connected only to pixels in their receptive areas (see Figure 1.4). In turn, each neuron in the second convolutional layer is linked only to neurons located within a small rectangle in the first layer. This structure allows the network to focus on low-level features in the first hidden layer, then construct them into higher-level features in the next hidden layer, and so on. This hierarchical structure is common in real-world images, which is one of the reasons why CNNs work very well for image recognition [5].

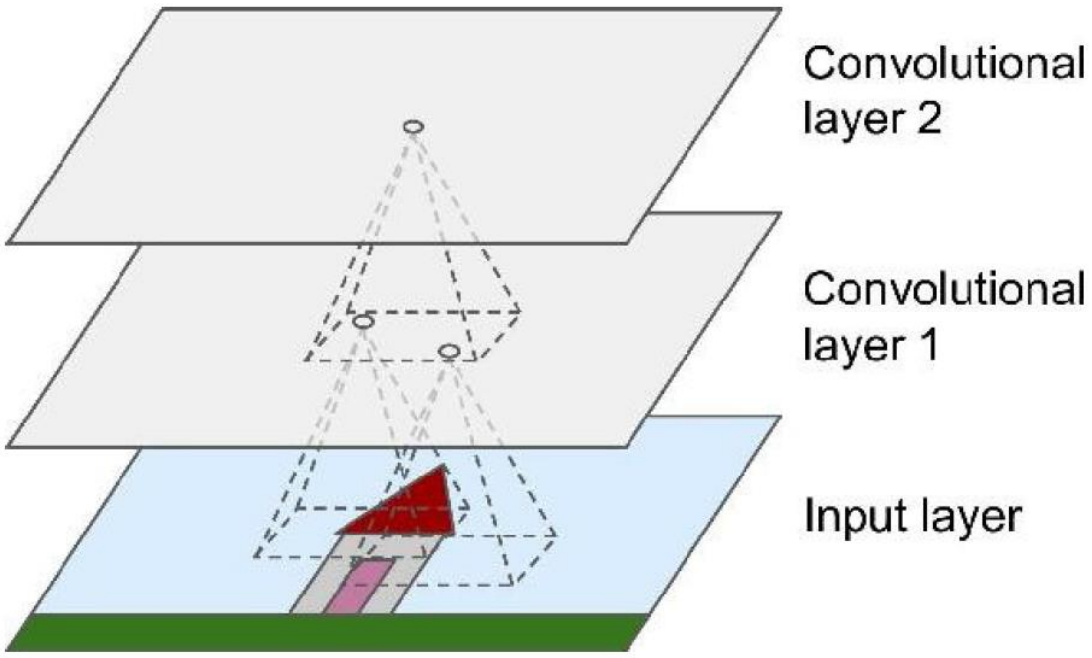


Figure 1.4 Convolutional layers with rectangular local receptive fields. Figure reprinted from [5].

A neuron located in row i, column j of a given layer is connected to the outputs of the neurons in the previous layer located in rows i to $i + f_h - 1$, columns j to $j + f_w - 1$, where f_h and f_w are the height and width of the receptive field (see Figure 1.5). For a layer to have the same height and width as the previous layer, it is common to add zeros around the inputs, as shown in the diagram. This is called **zero padding** [5]**.**

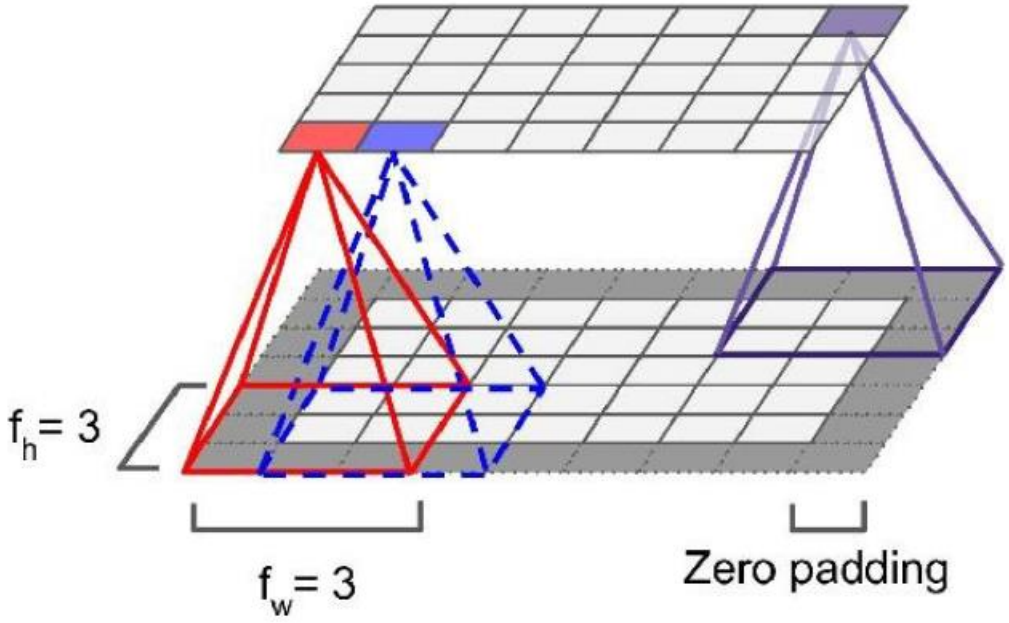


Figure 1.5 Connections between layers and zero padding. Figure reprinted from [5].

1.4.2 Filters

A neuron's weights can be represented as a small image the size of the receptive field. For example, Figure 1.6 shows two possible sets of weights, called **filters** (or **convolution kernels**). The first one is represented as a black square with a vertical white line in the middle (it is a matrix full of 0s except for the central column, which is full of 1s); neurons using these weights will ignore everything in their receptive field except for the central vertical line. The second filter is a black square with a horizontal white line in the middle. Again, neurons using these weights will ignore everything in their receptive field except for the central horizontal line [5].

Now if all neurons in a layer use the same vertical line filter, and you pass to the network the input image shown in Figure 1.6 (bottom image), the layer will output the top-left image. Notice that the vertical white lines get intensified while the rest gets blurred. Similarly, the upper-right image is the output if all neurons use the horizontal line filter; notice that the horizontal white lines get intensified while the rest is blurred. Thus, a layer full of neurons using the same filter gives you a **feature map**, which highlights the areas in an image that are most similar to the filter. During training, a CNN finds the most useful filters for its task, and it learns to combine them into more complex patterns [5].

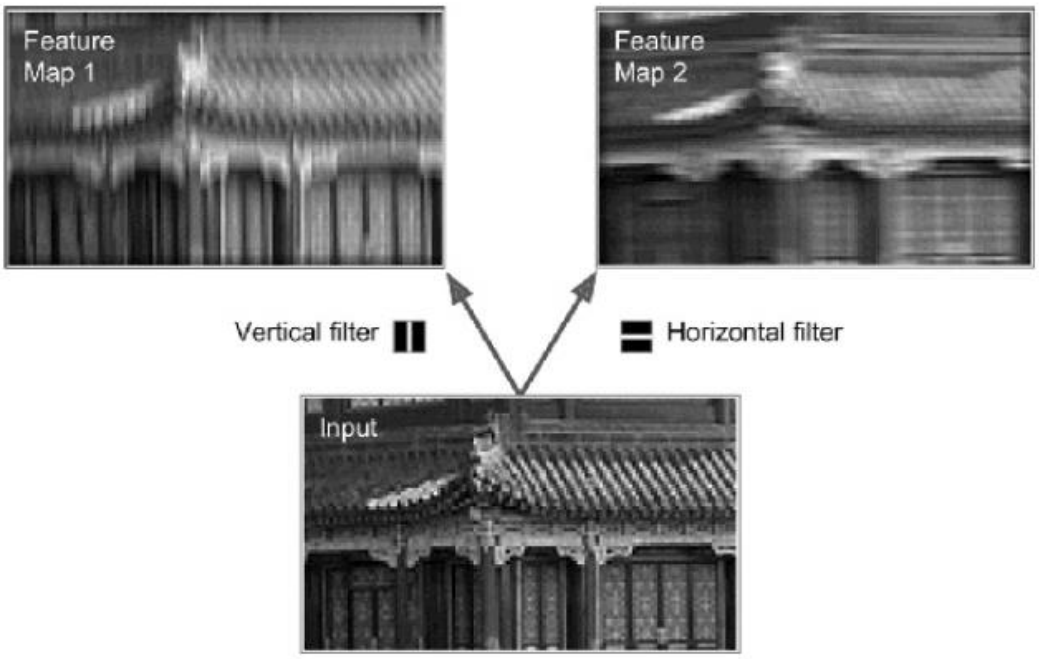


Figure 1.6 Applying two different filters to get two feature maps. Figure reprinted from [5].

1.4.3 Pooling Layer

The goal of the pooling layer is to downsample the input image to reduce the computational load and the number of parameters (thereby limiting the risk of overfitting).

Just like in convolutional layers, each neuron in a pooling layer is connected to the outputs of a limited number of neurons in the previous layer, located within a small rectangular receptive area. However, a pooling neuron has no weights; all it does is aggregate the inputs using an aggregation function such as the max or mean. Figure 1.7 shows a max pooling layer, which is the most common type of pooling layer. In this example, we use a 2 × 2 pooling kernel, a stride of 2, and no padding. Note that only the max input value in each kernel makes it to the next layer. The other inputs are discarded [5].

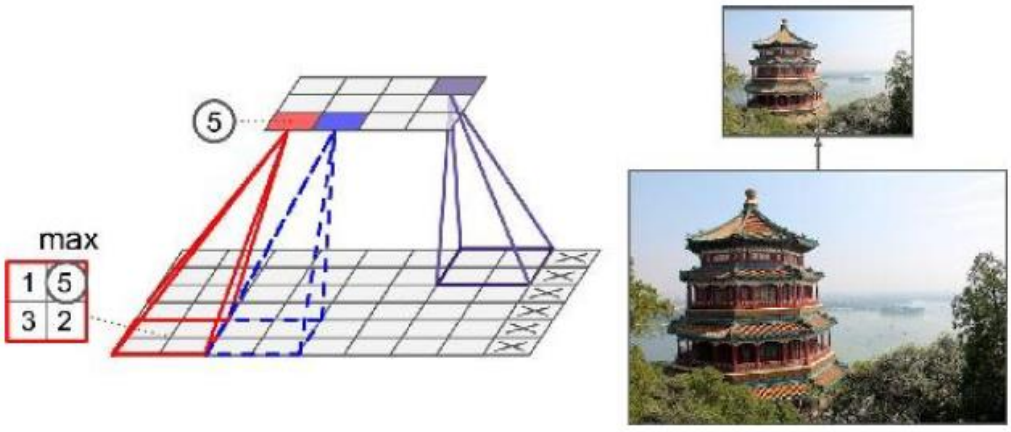


Figure 1.7 Max pooling layer (2 x 2pooling kernel, stride 2, no padding). Figure reprinted from [5].

This is obviously a very destructive kind of layer: even with a tiny 2 × 2 kernel and a stride of 2, the output will be two times smaller in both directions (so its area will be four times smaller), simply dropping 75% of the input values.

A pooling layer typically works on every input channel independently, so the output depth is the same as the input depth. We may alternatively pool over the depth dimension, in which case the image's spatial dimensions (height and width) remain the same, but the number of channels is reduced.

1.4.4 CNN Architectures

Typical CNN architectures consist of multiples of convolutional layers (each one generally followed by a ReLU layer), then a pooling layer, then another few convolutional layers (+ReLU), then another pooling layer, and so on. The image gets smaller and smaller as it progresses through the network, but it also typically gets deeper and deeper (see Figure 1.8). At the top of the stack, a feedforward neural network is added, composed of a few fully connected layers (+ReLUs), and the final layer outputs the prediction [5].

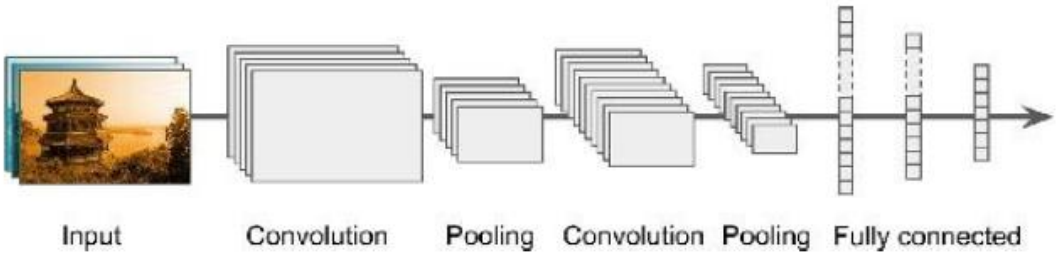


Figure 1.8 Typical CNN architecture. Figure reprinted from [5].

CHAPTER 2

RELATED WORK

In [7], Rong et al. have used CNNs and proposed a classification method of surrogate-assisted to classify retinal OCT images automatically. They have reduced the noise from the image and applied thresholding and morphological dilation for extraction. Consequently, they have generated surrogate images which are used for training. The accuracy is 97.83% for their local database, and 98.56% for the public database.

By using B-scans of OCT, a fully automated method to detect lesion activity of AMD is studied by Chakravarthy et al. in [8]. They have tested retinal specialist (RS) grading versus Notal OCT analyzer (NOA) for faster treatment purpose and achieved an accuracy of 91%.

Burlina et al. describe a technique in which they used a pre-trained DCNN to detect AMD in [9]. This approach has achieved quite an excellent preliminary result.

By using VGG-19 model, automated AMD detection combining OCT and fundus images are shown by Yoo et al. in [10] where five different models are used. An accuracy of 82.6% and 83.5% are achieved from only OCT image-based DL model and only fundus image-based DL model whereas, in case of a combined image model, the accuracy is 90.5%.

Convolutional Neural Networks (CNNs) have been demonstrated as very powerful techniques in broad range of tasks and in various fields of studies such as computer vision, language processing, image processing, and medical image analysis [11-14].

The recent applications of CNNs in medical image analysis include pancreas segmentation using CT images of the abdomen [15], classification of pulmonary peri-fussural nodules [16], and brain tumor segmentation [17].

CHAPTER 3

MATERIALS AND METHODS

3.1 Study Dataset

In this study, we train and test 4 different types of datasets on the same CNN model for robustness and accurate evaluation of the deep learning model we are using to classify and predict the tissue type and its scanning speed of the OCT images with accuracy reaches 97% for 2 of the datasets we are using and accuracy 90% for the third dataset.

70% of the data used for training and 30% for testing throughout this study. Phantom OCT images are used as data type 1, Dense tissue OCT images are data type 2, Adipose tissue OCT images are used as data type 3, then we use mixed data set of Dense tissue and Adipose tissue OCT images as data type 4.

Phantom OCT images obtained with 10 different lateral scanning speed, 5000 images are the size of the phantom dataset, 500 images acquired for each lateral scanning speed. Therefore, we have 10 classes of the lateral scanning speed for the phantom dataset.

For the adipose and dense tissue datasets, we have 5 classes of the lateral scanning speed. Each dataset contains 500 OCT images equally distributed to the 5 scanning speed classes. Therefore, each scanning speed class contains 100 images.

The CNN and algorithms we are using in this study is considered robust and extremely accurate for two reasons as follow:

1. The data used in this study is not easily classified by the human eye, OCT image samples with different scanning speed are shown in figure 3.1
2. The novel technique we utilize can form a bag of features that stores the strongest features in an OCT image and match similar features in any other OCT image, which will help us in recovering distorted OCT images as we will see later in section 4. More details about bag of features and matched features detection can be found in [18-20].

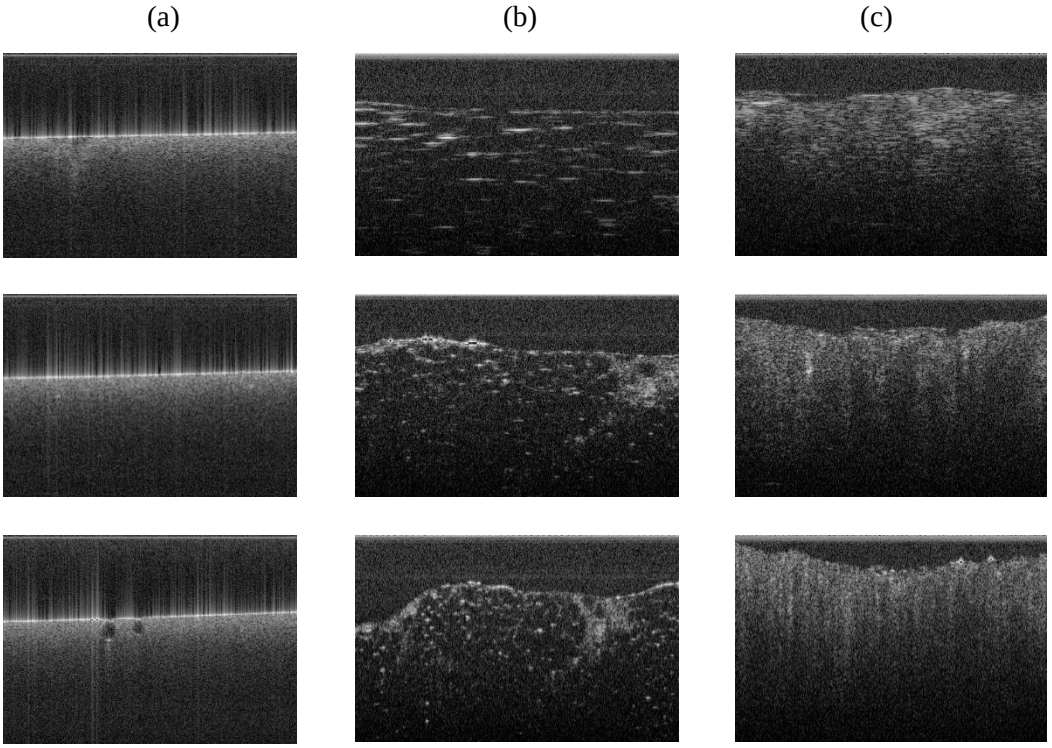


Figure 3.1 Phantom, adipose and dense tissue OCT images, in each column, there are 3 images with 3 different scanning speed from each tissue type. Column (a), 3 Phantom images acquired with 3 different lateral scanning speeds. Column (b), 3 Adipose tissue images with 3 different scanning speeds. Column (c), 3 Dense tissue images with 3 different scanning speeds. All images are of size 300x851 pixels.

3.2 Data Pre-Processing

Pre-processing started by writing a code on Matlab to extract cross-sectional B-scan images from A-scan images. In other words, 2-D OCT images is extracted from the acquired 3-D OCT images.

Data augmentation is one of the efficient methods to overcome the overfitting problem and to make the CNN model more generalized and robust to classify the test data. In this study, data augmentation is used in the form of image rotation and resizing. While performing experiments, different kinds of image processing is tested and embraced, such as adding noise, distortion and image transformation to test the CNN model and the algorithms' robustness and their ability to match features between the base or reference image and the distorted noisy image.

3.3 Feature Extraction and Classification

In previous OCT studies that used deep learning models for classification, most of these studies used pre-trained CNNs with proper fine-tuning to match their application, the most popular CNN model used in OCT application is the AlexNet model.

In this study, we introduce a CNN model built from scratch, the architecture and parameters of our CNN model used in our experiments are presented in table 3.1. Generally, every Convolutional Neural Network architecture which is applicable in image processing builds on four main operations: convolution, Rectified Linear Unit (ReLU), pooling or subsampling, and classification. In CNN, each convolutional filter creates one feature map when it moves through the whole image with a defined stride. Therefore, the size of the kernel determines the depth of the network. After every convolutional operation, a Rectified Linear Unit (ReLU) is applied. Since, convolution is a linear operator, it is required to introduce the non-linearity by storing non-negative values in the feature map and replacing the negative values by zero. The pooling or sub-sampling is used for dimensionality reduction by keeping the most important information [21,22].

Adam, an algorithm for efficient stochastic optimization that only requires first-order gradients with little memory requirement. The method computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients; the name Adam is derived from adaptive moment estimation. The algorithm used is designed to combine the advantages of two recently popular methods: AdaGrad, which works well with sparse gradients, and Root Mean Square Propagation, which works well in on-line and non-stationary settings. Some of Adam's advantages are that the magnitudes of parameter updates are invariant to rescaling of the gradient, its stepsizes are

approximately bounded by the stepsize hyperparameter, it does not require a stationary objective, it works with sparse gradients, and it naturally performs a form of step size annealing. See [26] for more details about Adam algorithm.

In this CNN model, Naïve Bayes classifier is used for classification, Naive Bayes is a classification algorithm that applies density estimation to the data. The algorithm leverages Bayes theorem, and (naively) assumes that the predictors are conditionally independent, given the class. Though the assumption is usually violated in practice, naive Bayes classifiers tend to yield posterior distributions that are robust to biased class density estimates, particularly where the posterior is 0.5 (the decision boundary). See [27,28] for more information about Naïve Bayes classifier.

Naive Bayes classifiers assign observations to the most probable class (in other words, the maximum a posteriori decision rule). Explicitly, the algorithm:

1. Estimates the densities of the predictors within each class.
2. Models posterior probabilities according to Bayes rule. That is, for all k = 1,...,*K*,

$$\hat{P}(Y = k|X_1, \dots, X_P) = \frac{\pi(Y = K)\prod_{j=1}^{P} P(X_j|Y = k)}{\sum_{k=1}^{K} \pi(Y = K)\ \prod_{j=1}^{P} P(X_j|Y = k)} \tag{3.1}$$

Where Y is the random variable corresponding to the class index of an observation, $X_1, \dots, X_P$ are the random predictors of an observation and $\pi(Y = K)$ is the prior probability that a class index is k.

3. Classifies an observation by estimating the posterior probability for each class, and then assigns the observation to the class yielding the maximum posterior probability.

Table 3.1 CNN Architecture and Parameters

Layer number	Layer type	Layer parameters
1	Image input	300x601x1 images with 'zerocenter' normalization
2	Convolution	9 3x3x1 convolutions with stride [1 1] and padding [0 0 0 0]
3	Batch normalization	Batch normalization with 9 channels
4	ReLU	ReLU
5	Max pooling	3x3 max pooling with stride [1 1] and padding [0 0 0 0]
6	Convolution	9 3x3x1 convolutions with stride [1 1] and padding [0 0 0 0]
7	Batch normalization	Batch normalization with 9 channels
8	ReLU	ReLU
9	Max pooling	3x3 max pooling with stride [1 1] and padding [0 0 0 0]
10	Dropout	20% dropout
11	Convolution	9 3x3x1 convolutions with stride [1 1] and padding [0 0 0 0]
12	Batch normalization	Batch normalization with 9 channels
13	ReLU	ReLU
14	Fully connected	10 fully connected layer
15	Softmax	Softmax
16	Classification output	Crossentropyex with 'v1_Adipose' and 9 other classes

3.4 Algorithms

In the following sub-sections, we introduce the algorithms used to recover any distorted OCT image. The two algorithms work very well for feature extraction, representation and matching, the three algorithms used in this study are as follow:

1. Bag of features algorithm.
2. Matched Features and Image Retrieval algorithm.

We will give brief details about each algorithm. See [29-31] for more information about the algorithms.

1. Bag of features algorithm

The past five years have seen the rise of the Bag of Features approach in computer vision. Bag of Features methods have been utilized in image classification, object detection, image retrieval, and even visual localization for robots. Bag of Features approaches are characterized by the use of an orderless collection of image features. Ignoring any structure or spatial information, it is perhaps surprising that this choice of image representation would be powerful enough to match or exceed state-of-the-art performance in many of the applications to which it has been applied. Due to its simplicity and performance, the Bag of Features approach has become well-established.

A Bag of Features method is one that represents images as orderless collections of local features. The name comes from the Bag of Words representation used in textual information retrieval.

The common perspective for explaining the Bag of Features image representation is by analogy to the Bag of Words representation. With Bag of Words, one represents a document as a normalized histogram of word counts. Commonly, one counts all the words from a dictionary that appear in the document. This dictionary may exclude certain noninformative words such as articles (like "the"), and it may have a single term to represent a set of synonyms. The term vector that represents the document is a sparse vector where each element is a term in the dictionary and the value of that element is the number of times the term appears in the document divided by the total number of dictionary words in the document (and thus, it is also a normalized histogram over the terms). The term vector is the Bag of Words document representation – called a "bag" because all ordering of the words in the document have been lost.

The Bag of Features image representation is analogous. A visual vocabulary is constructed to represent the dictionary by clustering features extracted from a set of training images. The image features represent local areas of the image, just as words are local features of a document. Clustering is required so that a discrete vocabulary can be generated from millions (or billions) of local features sampled from the training data. Each feature cluster is a visual word. Given a novel image, features are detected and assigned to their nearest matching terms (cluster centers) from the visual vocabulary. The term vector is then simply the normalized histogram of the quantized features detected in the image. See [29] for more information about Bag of Features.

2. Matched Features and Image Retrieval algorithm

A feature is a piece of information which is relevant for solving the computational task related to a certain application. Features may be specific structures in the image such as points, edges or objects. Features may also be the result of a general neighborhood operation or feature detection applied to the image [30].

Interest point or Feature Point is the point which is expressive in texture. Interest point is the point at which the direction of the boundary of the object changes abruptly or intersection point between two or more edge segments [30].

A feature descriptor is a method which takes an image and outputs feature descriptors/feature vectors. Feature descriptors encode interesting information into a series of numbers and act as a sort of numerical “fingerprint” that can be used to differentiate one feature from another. Ideally, this information would be invariant under image transformation, so we can find the feature again even if the image is transformed in some way. After detecting interest point, we go on to compute a descriptor for every one of them.

Features matching or generally image matching, a part of many computer vision applications such as image registration, camera calibration and object recognition, is the task of establishing correspondences between two images of the

same scene/object. A common approach to image matching consists of detecting a set of interest points each associated with image descriptors from image data. Once the features and their descriptors have been extracted from two or more images, the next step is to establish some preliminary feature matches between these images [31].

Generally, the performance of matching methods based on interest points depends on both the properties of the underlying interest points and the choice of associated image descriptors. Thus, detectors and descriptors appropriate for images contents shall be used in applications. For instance, if an image contains bacteria cells, the blob detector should be used rather than the corner detector. But, if the image is an aerial view of a city, the corner detector is suitable to find man-made structures. Furthermore, selecting a detector and a descriptor that addresses the image degradation is very important [31].

CHAPTER 4

EXPERIMENTAL RESULTS

4.1 Classification

Creating a deep learning model for classification is the main experiment in this study. We are training a deep learning network because we believe that the network can extract features human eye cannot and therefore, the deep learning network can outperform human in this particular task.

Human eye is incapable to distinguish between Adipose and Dense tissues with different scanning speeds for example (See figure 3.1), unlike the deep learning network which can classify the tissue type and its scanning speed with accuracy up to 97%.

After training and testing the CNN model on the phantom datasets (4500 images for training and 500 images for testing), we have 90% classification accuracy from the CNN model, which is considered acceptable result for data images that cannot be classified by the human eye (See figure 3.1, column (a)).

We used the phantom dataset initially to train the network to extract the features of the OCT images with a large dataset (5000 images). However, we care about classifying the real OCT data, not the phantom, which are the dense (500 images) and adipose (500 images) datasets.

The first test on the real data performed on the dense dataset, the model is trained on 350 dense images and we left out 150 images for testing. The image data store is a folder contains 5 sub-folders, each sub-folder contains 100 images and

each sub-folder carries the label that corresponds to the scanning speed of the images in this sub-folder. We have 5 speed labels ($v_1, v_2, v_3, v_4\ and\ v_5$) The test images are chosen randomly equally from each label folder and saved in a separate folder. The CNN model is trained in less than 15 minutes and results in 94% classification accuracy for the dense tissue dataset, figure 4.1 displays the CNN model prediction output of 6 samples chosen randomly from the test dataset.

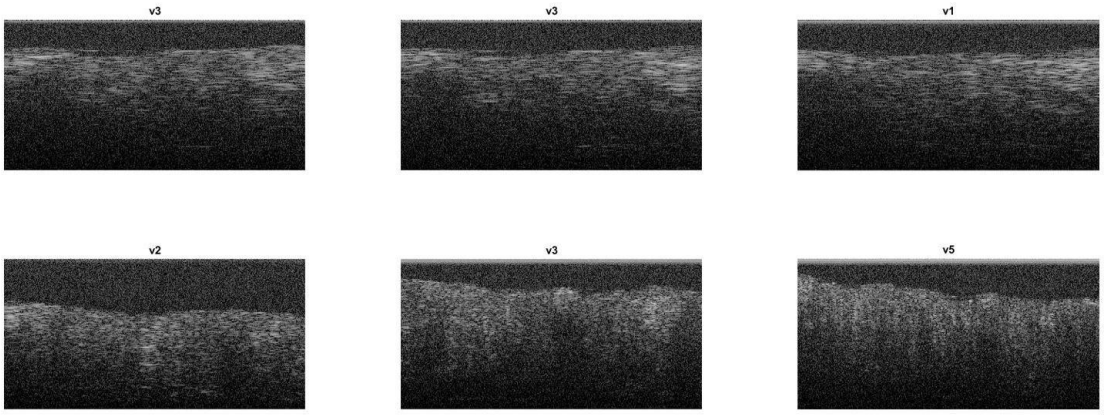


Figure 4.1 Prediction of the scanning velocity of 6 samples selected randomly from the test dataset of the dense tissue.

The same test is performed on the adipose tissue dataset, the dataset is divided into 350 images for training and 150 images for testing. The image data store is a folder contains 5 sub-folders, each sub-folder contains 100 images and each sub-folder carries the label that corresponds to the scanning speed of the images in this sub-folder. We have 5 speed labels ($v_1, v_2, v_3, v_4\ and\ v_5$) The test images are chosen randomly equally from each label folder and saved in a separate folder. The CNN model is trained in less than 15 minutes similar to the training time for the dense tissue training dataset and results in 100% classification accuracy for the adipose tissue dataset, figure 4.2 displays the CNN model prediction output of 6 samples chosen randomly from the test dataset.

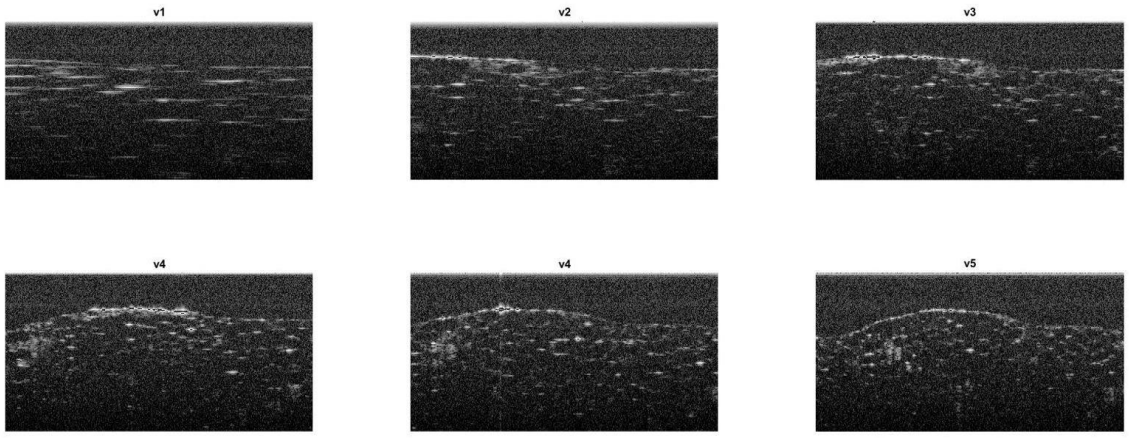


Figure 4.2 Prediction of the scanning velocity of 6 samples selected randomly from the test dataset of the adipose tissue.

The 100% classification accuracy for the adipose dataset sounds perfect, but this result brings up the overfitting problem. Overfitting is the production of an analysis that corresponds too closely or exactly to a particular set of data and may therefore fail to fit additional data or predict future observations reliably. An overfitted model is a statistical model that contains more parameters than can be justified by the data. The essence of

overfitting is to have unknowingly extracted some of the residual variation (i.e. the noise) as if that variation represented underlying model structure.

This result brings us to the next phase, we want to test the robustness of the model and also, we need to check if the model is overfitting or not. The first step to test the robustness of the model is to combine the dense tissue and the adipose tissue datasets into one dataset and test the combined dataset The new image data store is a folder contains 10 sub-folders, each sub-folder contains 100 images and each sub-folder carries the label that corresponds to the scanning speed and the tissue type of the images in this sub-folder. We have 10 speed and tissue type labels (v1_Adipose, v1_Dense, v2_Adipose, v2_Dense, v3_Adipose, v3_Dense, v4_Adipose, v4_Dense, v5_Adipose, v5_Dense). The test images are chosen randomly equally from each label folder and saved in a separate folder.

Similarly, like the previous 2 datasets, we tested the new dataset which we created as a combination between the dense and adipose tissue images on the CNN model. The result of the test is 97% classification accuracy, figure 4.3 displays the CNN model prediction output of 6 samples chosen randomly from the test dataset showing the new classification category, which is the tissue type and its associated scanning speed

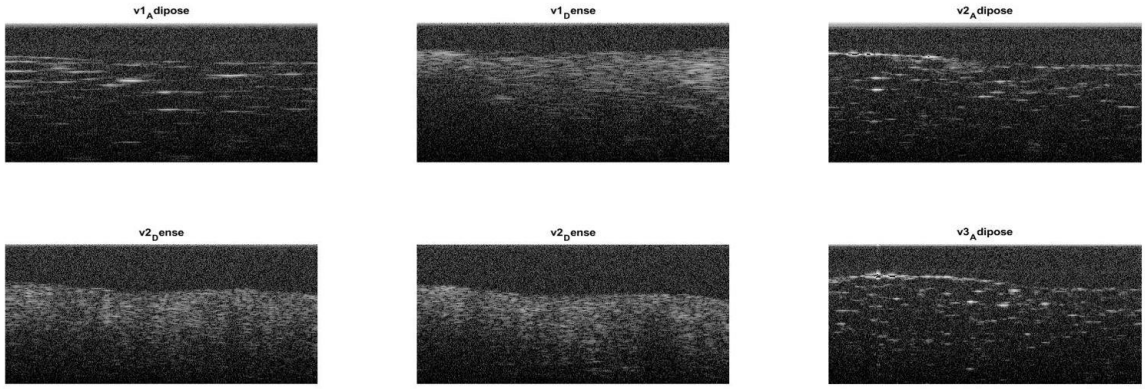


Figure 4.3 Prediction of the tissue type and its associated scanning velocity of 6 samples selected randomly from the test dataset of the adipose and dense tissue combined.

By knowing where exactly the misclassification occurs, we can easily determine the weak points in the dataset, the confusion matrix is generated for this purpose. A confusion matrix is a table that is often used to describe the performance of a classifier on a set of test data for which the true values are known, it allows the visualization of the performance of the algorithm.

The confusion matrix of the Naïve Bayes classifier of our CNN model is generated using Matlab and is displayed in table 4.1. In the table, the confusion matrix shows that v4_Dense label has the least classification accuracy 83%, the 17% error is divided into 10% misclassification as v5_Dense and the remaining 7% error misclassification as v3_Dense.

Table 4.1 Confusion Matrix of the Adipose and Dense Tissue Combined Test Dataset

	PREDICTED									
KNOWN	**v1_Adipose**	**v1_Dense**	**v2_Adipose**	**v2_Dense**	**v3_Adipose**	**v3_Dense**	**v4_Adipose**	**v4_Dense**	**v5_Adipose**	**v5_Dense**
v1_Adipose	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
v1_Dense	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
v2_Adipose	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
v2_Dense	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
v3_Adipose	0.00	0.00	0.00	0.00	0.93	0.03	0.03	0.00	0.00	0.00
v3_Dense	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
v4_Adipose	0.00	0.00	0.00	0.00	0.03	0.00	0.93	0.00	0.03	0.00
v4_Dense	0.00	0.00	0.00	0.00	0.00	0.07	0.00	0.83	0.00	0.10
v5_Adipose	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
v5_Dense	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

These results are good so far for the 3 different datasets; The adipose tissue dataset, the dense tissue dataset, and the mixed dataset with classification accuracy 100%, 94%, and 97% respectively.

The best way to make a deep learning model generalize better is to train it on more data. Of course, in practice, the amount of data we have is limited. One way to get around this problem is to create fake data and add it to the training set [4].

Dataset augmentation has been a particularly effective technique for classification problems. Images are high dimensional and include an enormous variety of factors of variation, many of which can be easily simulated. Operations like translating the training images a few pixels in each direction can often greatly improve generalization, even if the model has already been designed to be partially translation invariant by using the

convolution and pooling techniques. Many other data augmentation operations such as rotating the image or scaling the image have also proven quite effective [4].

After explaining what data augmentation is, we will apply 2 of the data augmentation operations to the mixed datasets and test on our CNN model, then we want to do some analysis to the results and compare it the results we had previously from the original mixed dataset.

Table 4.2 display the confusion matrix of the Naïve Bayes classifier for the augmented test dataset after training the model with the augmented dataset, the model is trained on 700 images as training dataset and tested on 300 images as test dataset. All images are resized to 300x601 pixels (300x851 pixels is the original image size) and rotation 30 degrees is applied to all images as the second data augmentation operation.

Table 4.2 Confusion Matrix of the Augmented Adipose and Dense Tissue Combined Test Dataset

	PREDICTED									
KNOWN	**v1_Adipose**	**v1_Dense**	**v2_Adipose**	**v2_Dense**	**v3_Adipose**	**v3_Dense**	**v4_Adipose**	**v4_Dense**	**v5_Adipose**	**v5_Dense**
v1_Adipose	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
v1_Dense	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
v2_Adipose	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
v2_Dense	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
v3_Adipose	0.00	0.00	0.00	0.00	0.93	0.00	0.07	0.00	0.00	0.00
v3_Dense	0.00	0.00	0.00	0.00	0.00	0.93	0.00	0.07	0.00	0.00
v4_Adipose	0.00	0.00	0.00	0.00	0.00	0.00	0.93	0.00	0.07	0.00
v4_Dense	0.00	0.00	0.00	0.00	0.00	0.17	0.00	0.83	0.00	0.00
v5_Adipose	0.00	0.00	0.00	0.00	0.00	0.00	0.13	0.00	0.87	0.00
v5_Dense	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

The result of the test is 95% classification accuracy. The interesting part of this result is when you look at the classification accuracy of v4_Dense still the same as the previous test on the original dataset, but the classification error for v4_Dense is now limited to one label, which is v3_Dense, and that is case for all the labels in the dataset that have classification error, unlike the previous test where each label that has classification error, this error was divided between two different labels (v4_Dense is misclassified as v5_Dense 10% of the total observations and misclassified as v3_Dense 7% of the total observations).

Although, the training and testing the original mixed data gives higher accuracy for classification than training and testing the augmented dataset, the augmented dataset is considered more reliable because it limits the error to only one different label, for example, v4_Dense is misclassified as v3_Dense 17% of the total observations of v4_Dense. It is

easier to improve the percentage of error between 2 labels than to improve it between 3 labels. This subject is beyond the scope of this study; therefore, we won't go any deeper on how to improve the percentage of error of a classifier.

To sum up, the confusion matrix is a very powerful matrix that evaluate the performance of a classifier. Also, it can give a hint on how to proceed to approach the percentage of error problem.

4.2 Features Representation

Imagine that we have a distorted image that is scanned using a manual OCT device (See figure 4.4) and we need to know which tissue type this image belongs to or even we would like to know the scanning speed of this particular image. In this sub-section and the following sub-sections, we introduce novel technique for image retrieval in the medical field, more specifically, in the field of the OCT imaging. we will build a systematic experimental procedure on how to recover a distorted OCT image using Bag of Features and Matched Features algorithms, then classify the recovered image using the deep learning network introduced in section 4.1 and get the best matching images in the OCT image dataset according to their probability scores.

As per our discussion in chapter 3, Bag of Features method extract and represent the features in an image. First step towards the image retrieval approach is to be able to extract and represent the features in our OCT datasets. Figures 4.4-4.13 display the feature representation of each OCT image associated with their scanning speed. Furthermore, we display an image of 100 strongest features in each image below its original image of feature representation. Table 4.3 provides the number of features extracted per each image.

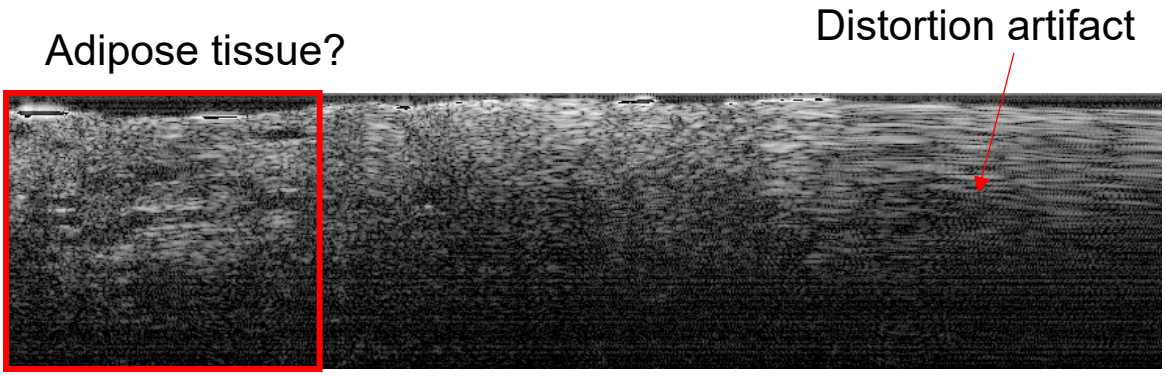


Figure 4.4 OCT image acquired using manual OCT device.

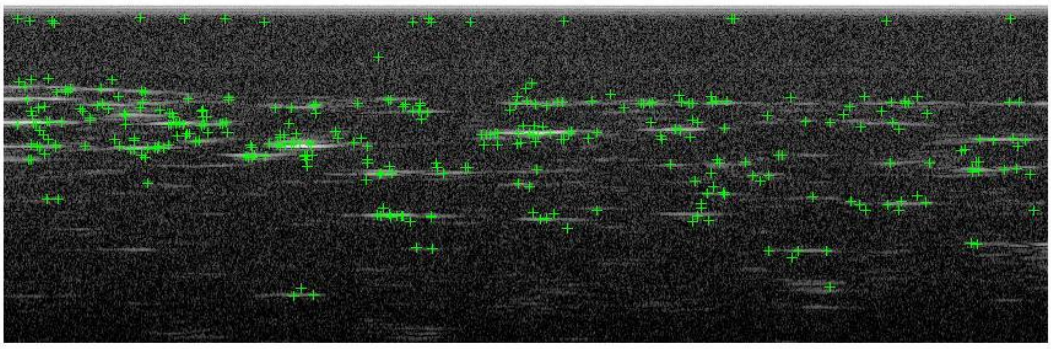


Figure 4.5 Representation of all features in v1_adipose OCT image.

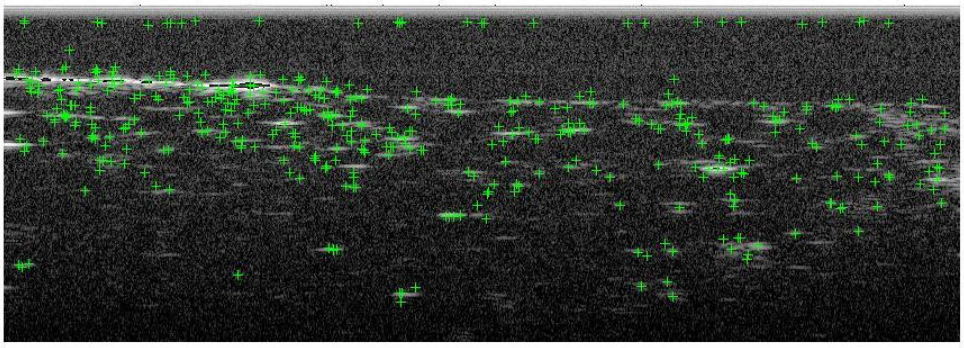


Figure 4.6 Representation of all features in v2_adipose OCT image.

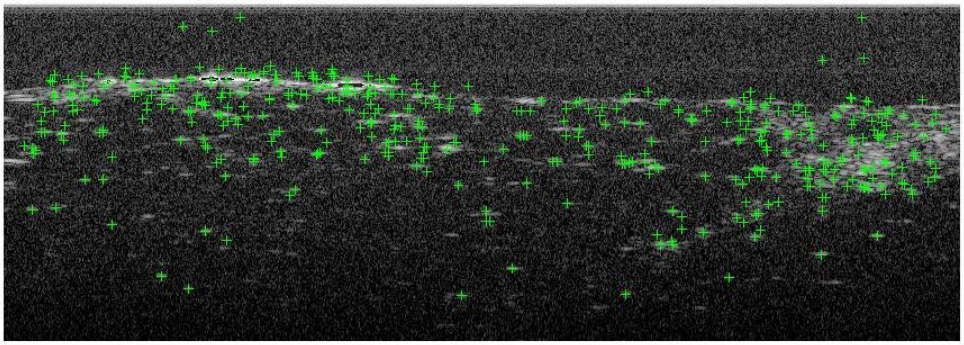


Figure 4.7 Representation of all features in v3_adipose OCT image.

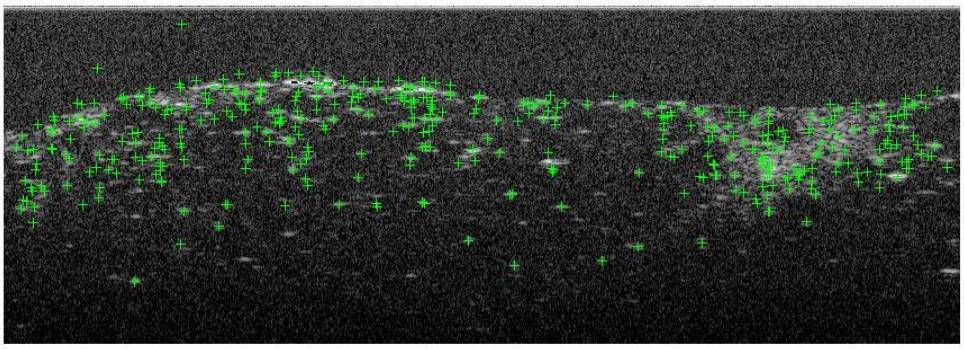


Figure 4.8 Representation of all features in v4_adipose OCT image.

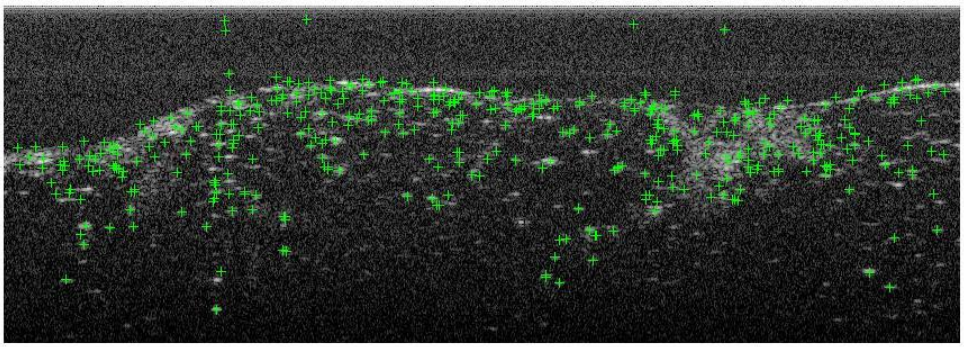


Figure 4.9 Representation of all features in v5_adipose OCT image.

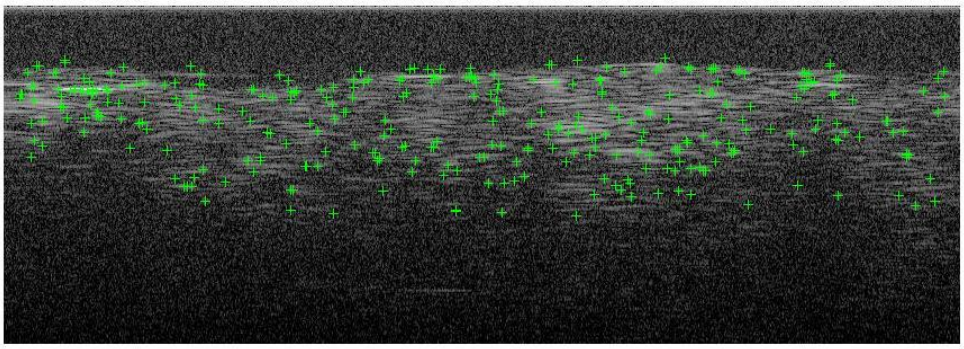


Figure 4.10 Representation of all features in v1_dense OCT image.

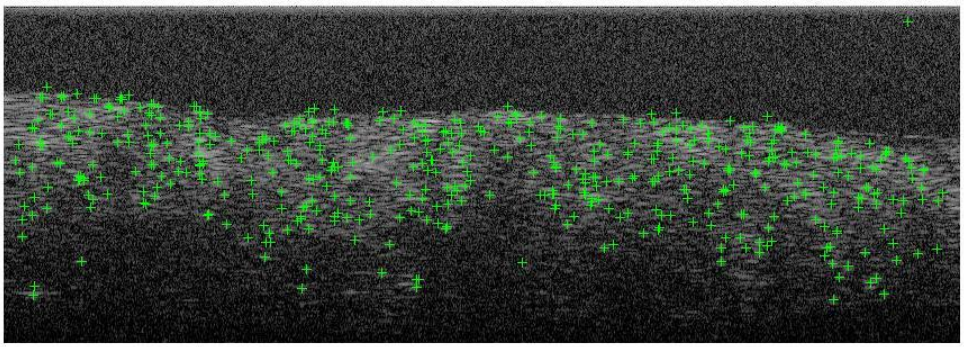


Figure 4.11 Representation of all features in v2_dense OCT image.

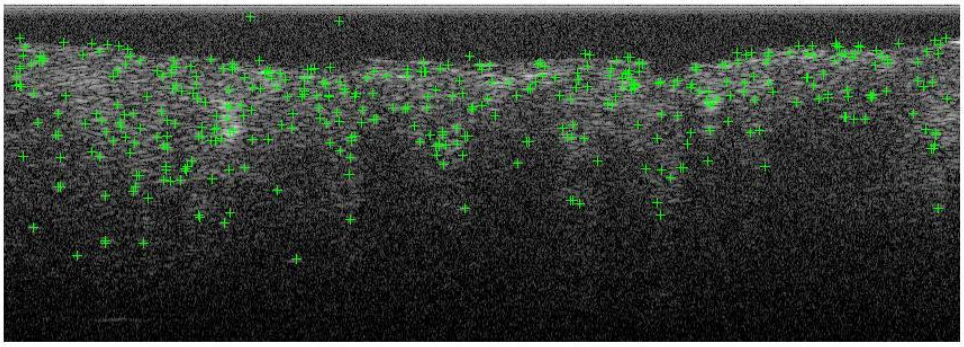


Figure 4.12 Representation of all features in v3_dense OCT image.

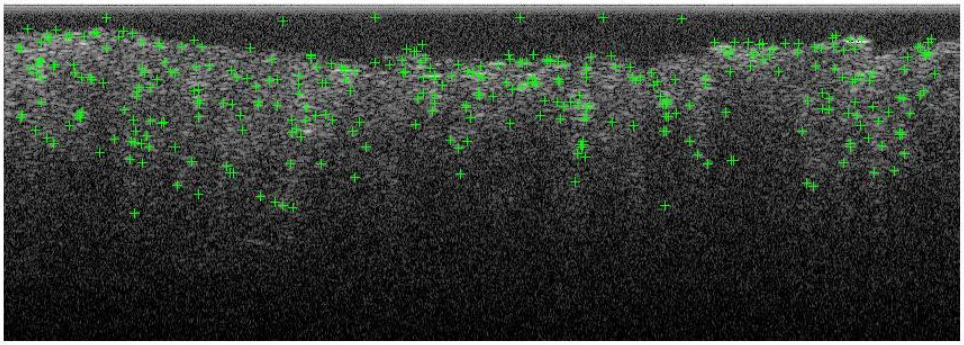


Figure 4.13 Representation of all features in v4_dense OCT image.

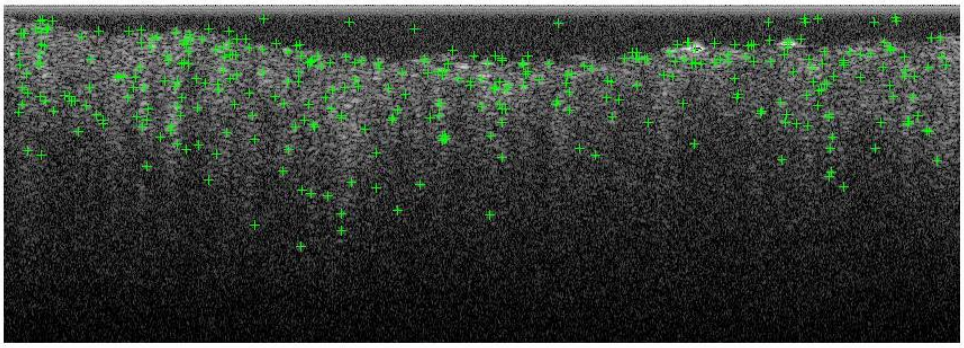


Figure 4.14 Representation of all features in v5_dense OCT image.

Table 4.3 Number of Features Extracted Per Each OCT Image

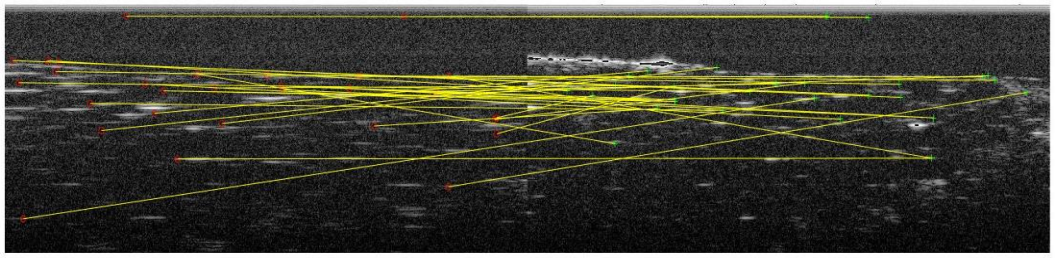
OCT image	Number of features extracted
v1_adipose	329
v2_adipose	424
v3_adipose	435
v4_adipose	419
v5_adipose	444
v1_dense	321
v2_dense	496
v3_dense	391
v4_dense	336
v5_dense	350

4.3 Matched Features Method

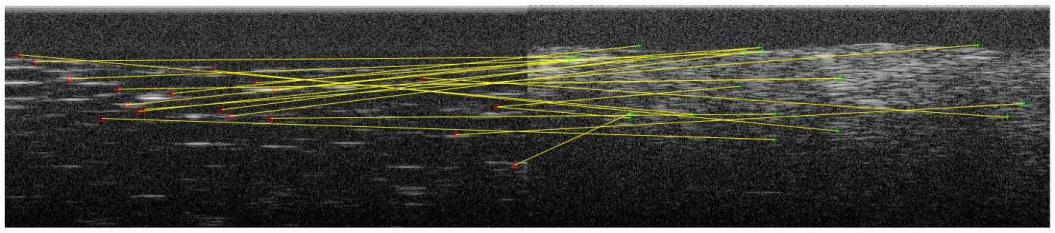
Matched features method consists of detecting a set of interest points each associated with image descriptors from image data. Once the features and their descriptors have been extracted from two or more images where we established in the previous step; Bag of Features. The next step is to establish some preliminary feature matches between these images.

In this test, we take v1_adipose as a reference image and test the matched features between v1_adipose and every other image in our dataset including v1_adipose with itself. Figure 4.14 displays the results of this test.

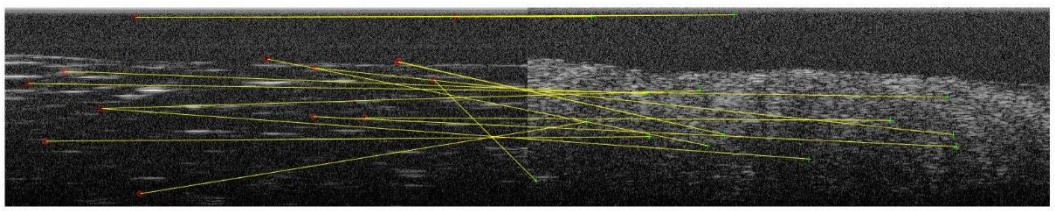
(a)



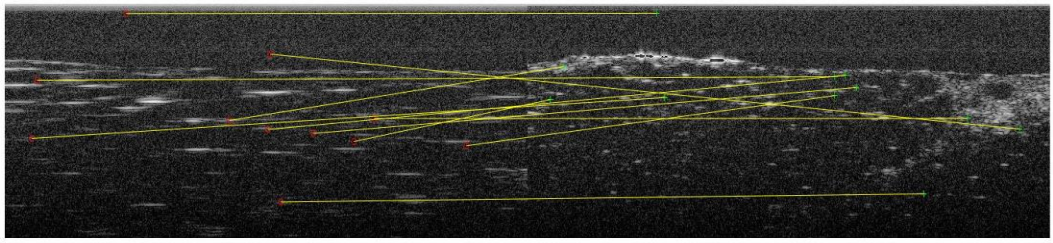
(b)



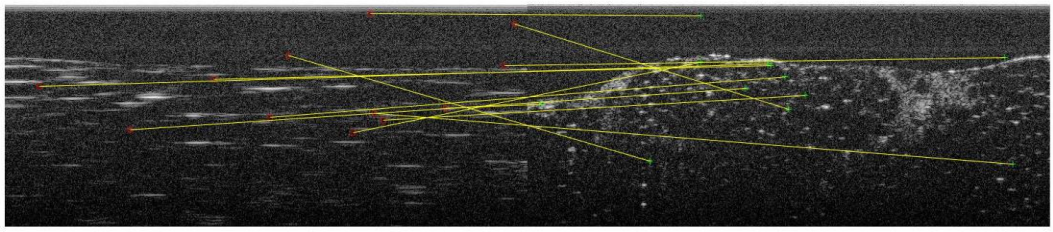
(c)



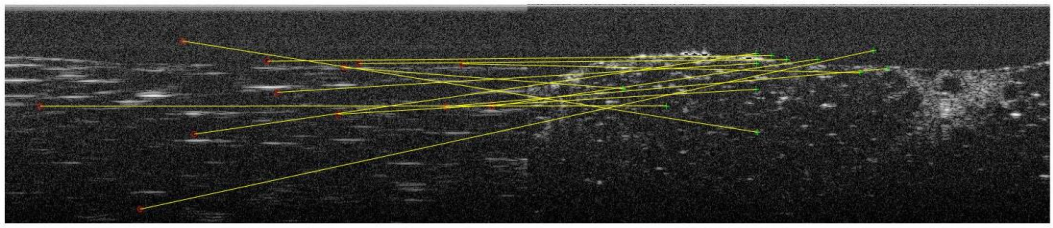
(d)



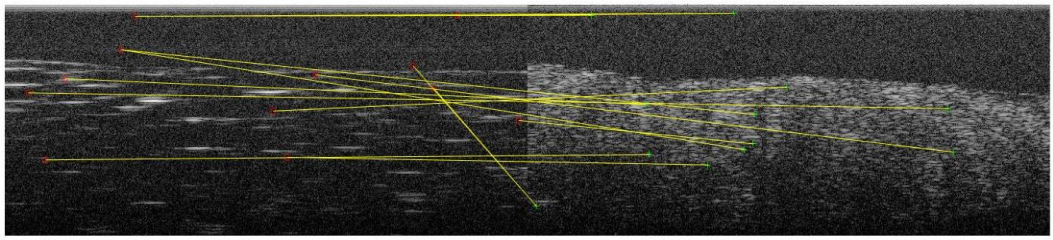
(e)



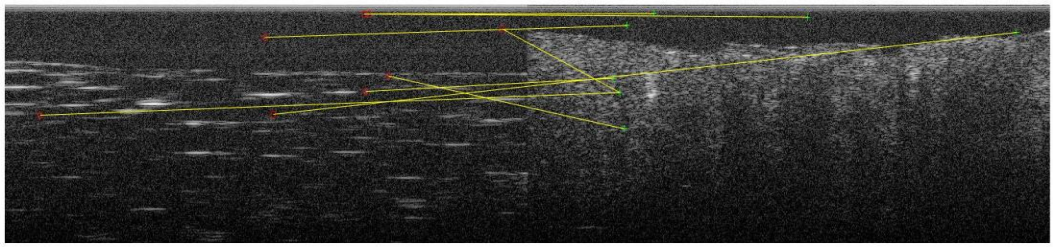
(f)



(g)



(h)



(i)

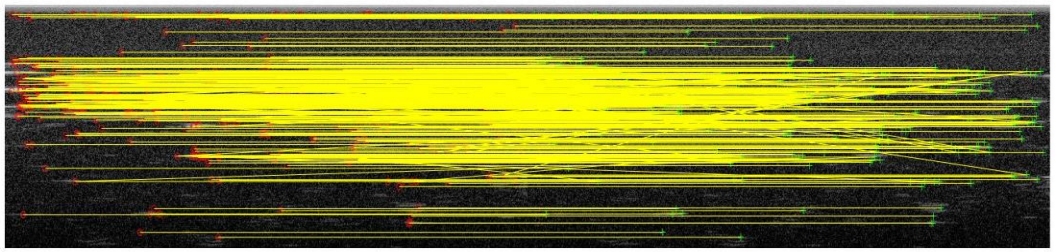


Figure 4.15 v1_adipose image as reference image on the left and the test image on the right. Matched features between v1_adipose and v2_adipose in (a). Matched features between v1_adipose and v1_dense in (b). Matched features between v1_adipose and v2_dense in (c). Matched features between v1_adipose and v3_adipose in (d). Matched features between v1_adipose and v4_adipose in (e). Matched features between v1_adipose and v5_adipose in (f). Matched features between v1_adipose and v3_dense in (g). Matched features between v1_adipose and v4_dense in (h). Matched features between v1_adipose and itself in (i).

4.4 Image Retrieval

In this sub-section we show the results of recovering an OCT distorted image using the Bag of Features and Matched Features algorithms discussed in sections 3.4, 4.2 and 4.3. The high-quality recovered image acquired from these algorithms make this technique outperform any techniques for image retrieval utilized in the past.

In this experiment, we chose v5_adipose (see figure 4.15) to be the reference image or the image that we want to recover from a distorted image with augmentation applied to it as well to test the robustness of the algorithm, the augmentation is in the form of resizing the photo to 70% of its original size and rotation is applied also. The distorted image is just a fake image created from v1_adipose, v3_adipose and v5_adipose stitched all together (see figures 4.16 and 4.17) to form the distorted image.

The role of the Image Retrieval algorithm is to match the common features between the reference image and the distorted image, then create a recovered version of the distorted image (see figure 4.18) according to the amount of features the algorithm is able to match between the reference image and the distorted image.

The recovered image is very similar to the reference image (see figure 4.19), this result emphasizes the robustness of the Image Retrieval algorithm in reconstructing images with complex features like the OCT images which cannot be noticed by human eyes.

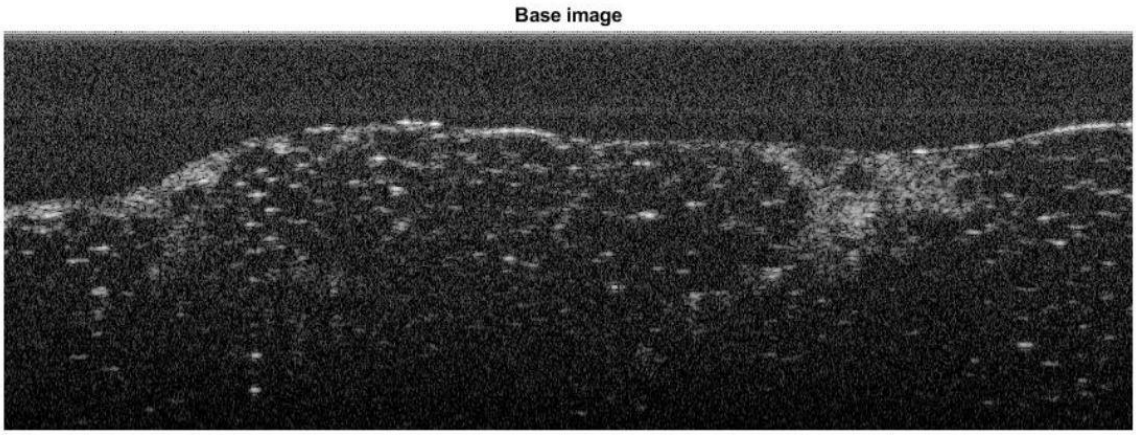


Figure 4.16 The reference image, v5_adipose.

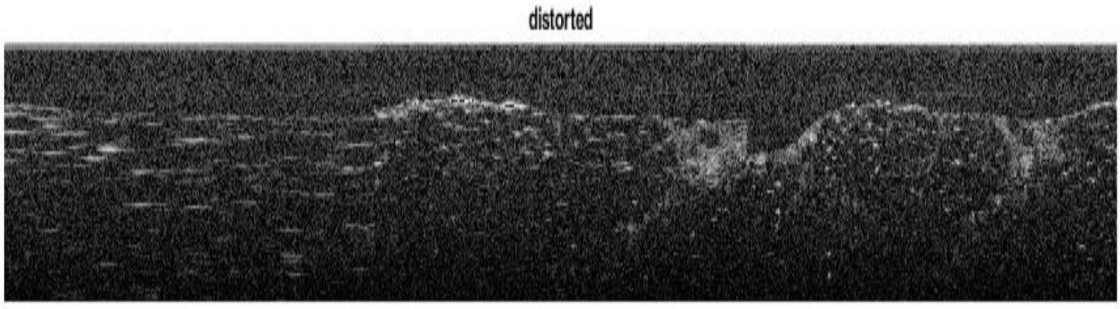


Figure 4.17 The distorted image created in Matlab using v1_adipose, v3_adipose and v5_adipose.

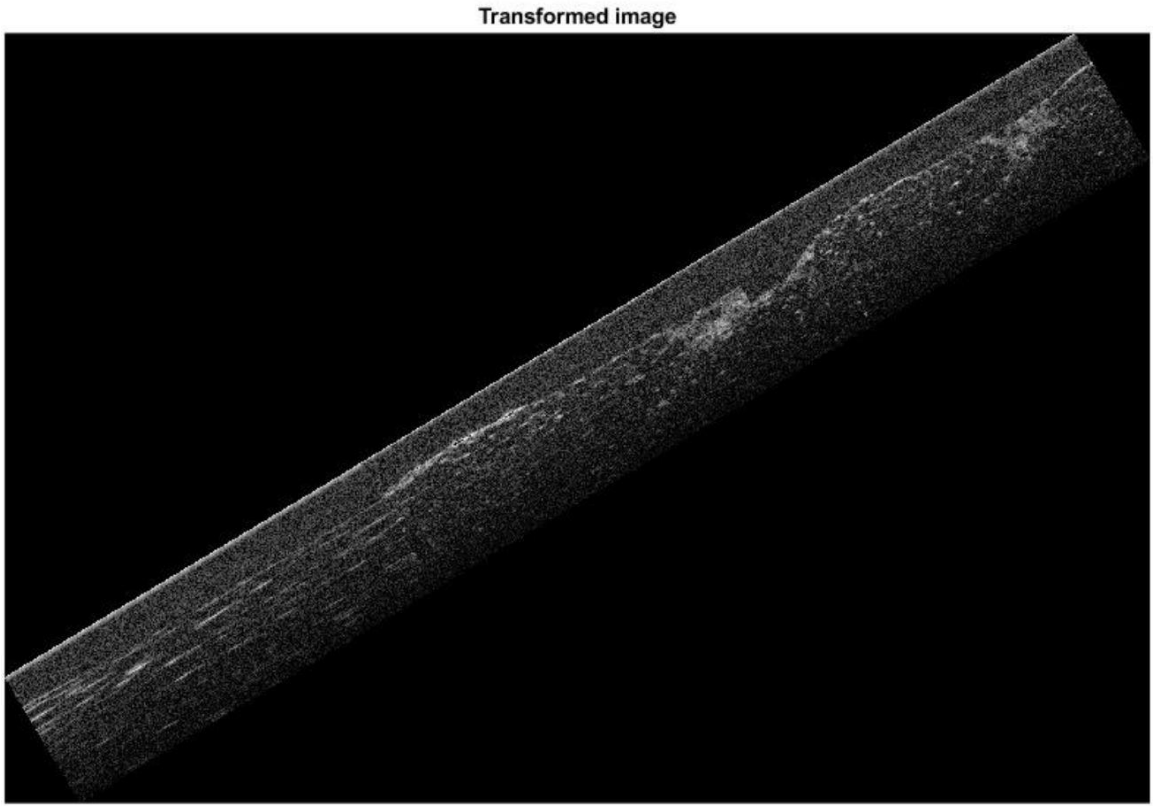


Figure 4.18 The distorted image with augmentation applied to it.

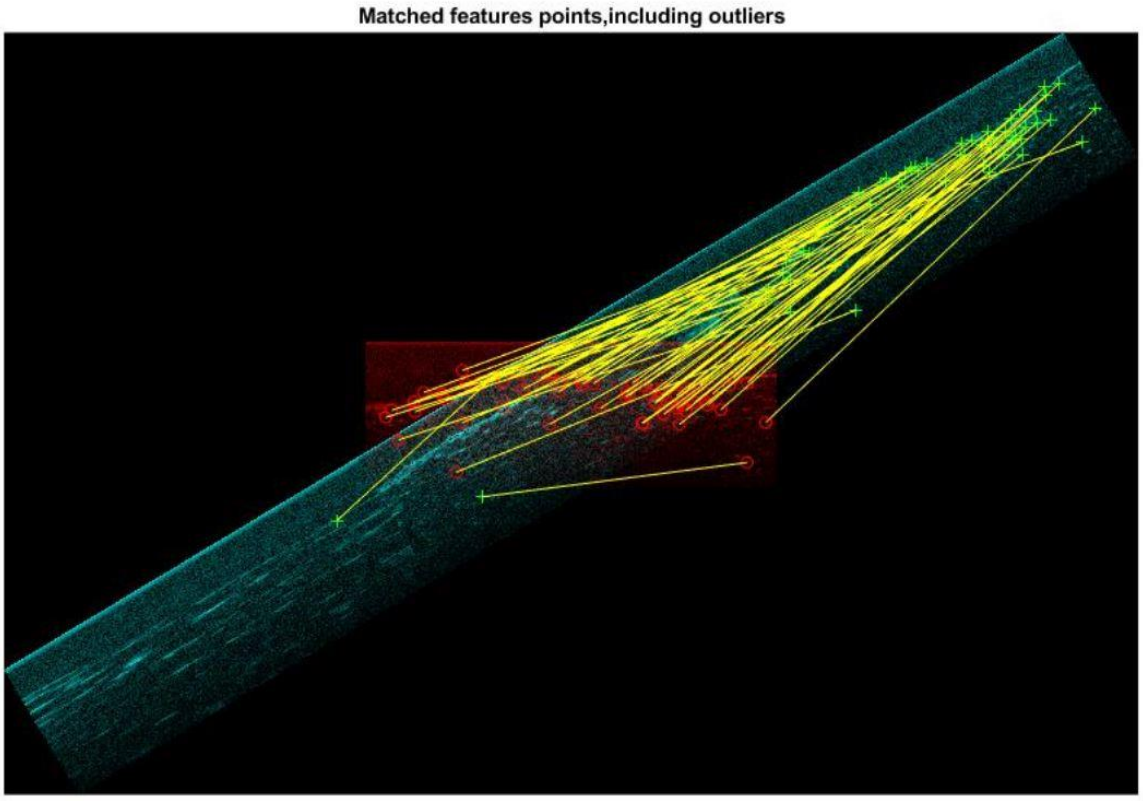


Figure 4.19 The red image is the reference image and the green image is the distorted image, the yellow lines are connecting the matched features between the 2 images.

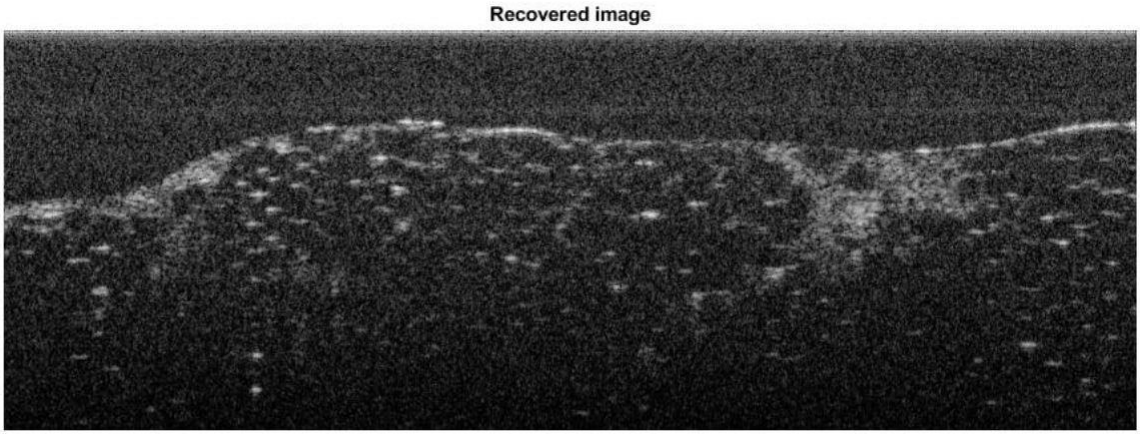


Figure 4.20 The recovered image by the Image Retrieval algorithm.

The final step, we need to test the recovered image on the CNN network and see if the CNN network is able to classify its tissue type and scanning speed or not. Also, we added additional algorithm to the CNN network to index the images in the dataset and display the best 6 matches of the recovered image and their probability scores from the original dataset that we used earlier in the tissue type and speed classification experiments which contains 1000 images (see figure 4.20).

The CNN network classifies the image recovered from the Image Retrieval algorithm correctly and displays the 6 best matches from the same tissue and speed category that the recovered image belongs to, which is v5_adipose.

This result proves and legitimize the robustness and efficiency of the algorithms used to do all the experiments in this study and open more applications and horizons for the deep learning and the matched features methods in the field of OCT.

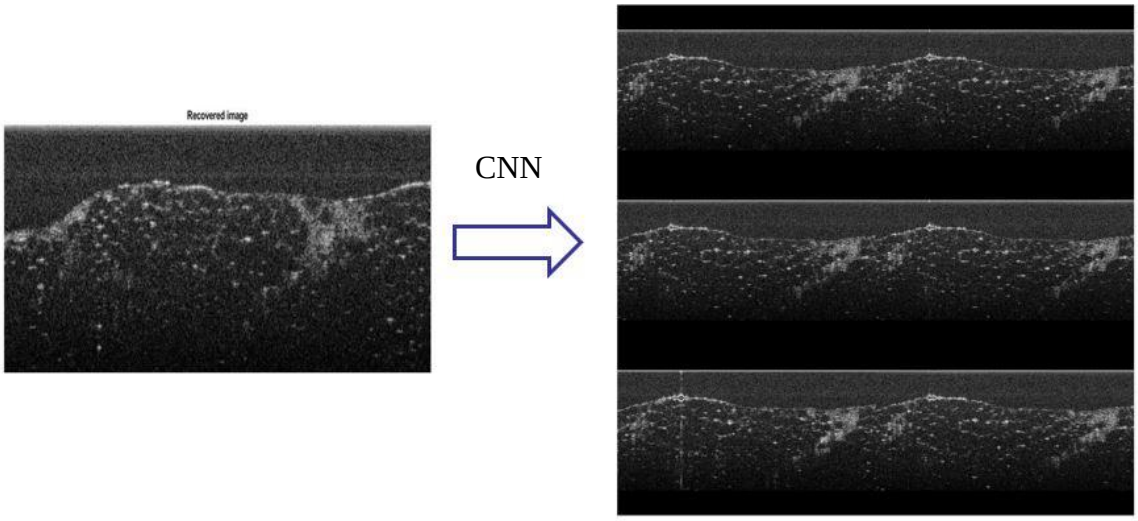


Figure 4.21 On the left the recovered image from the Image Retrieval algorithm. On the right, a tile of the 6 best matches from the dataset chosen by the CNN network, the top row has 2 images with probability scores 0.9947 and 0.9827, the middle row has 2 images with probability scores 0.9813 and 0.9800, the bottom row has 2 images with probability scores 0.9800 and 0.9762.

CONCLUSION

In this thesis, we presented a novel convolutional neural network that have very high accuracy in classifying the OCT test data. We presented a novel technique for OCT image retrieval, the experimental results show the accuracy and robustness of the algorithms in retrieving a distorted OCT image using the Matched Features algorithm as discussed explicitly in section 4 of this thesis (see figures 4.16 & 4.20).

The disadvantage of the proposed CNN is the dependence on big data. On the other hand, the CNN is very accurate and robust against noise compared to the traditional methods.

The results of this study are promising due to the adaptability and accuracy of the CNN and the matched features algorithms towards the OCT data. In future work, we will extend the proposed CNN and algorithms to handle other kinds of human tissues and help in breast cancer diagnosis using the OCT manual scanning device.

REFERENCES

[1] Xuan Liu, Yong Huang, and Jin U. Kang. *Distortion-free freehand-scanning OCT Implemented with real-time scanning speed variance correction.* OSA 20(15). 2012

[2] Stefan W. Hell, and Robert N. Weinreb. *High Resolution Imaging in Microscopy And ophthalmology.* Switzerland : Springer, 2019. Print.

[3] Xuan Liu, Yong Huang, Jessica C. Ramella-Roman, Scott A. Mathews, and Jin U. Kang. *Quantitative transverse flow measurement using optical Coherence tomography speckle decorrelation analysis.* OSA 38(5). 2013

[4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* Cambridge, MA : MIT Press, 2017. Print.

[5] Geron, Aurelien. *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. Sebastopol, CA : O'Reilly Media, Inc, 2017. Print

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

[7] Yibiao Rong, Dehui Xiang, Weifang Zhu, Kai Yu, Fei Shi, Zhun Fan, and Xinjian Chen. *Surrogate-assisted retinal oct image classification based on convolutional neural networks*. IEEE journal of biomedical and health informatics, 23(1):253–263, 2019.

[8] Usha Chakravarthy, Dafna Goldenberg, Graham Young, Moshe Havilio, Omer

Rafaeli, Gidi Benyamini, and Anat Loewenstein. *Automated identification of lesion activity in neovascular age-related macular degeneration*. Ophthalmology, 123(8):1731–1736, 2016.

[9] Philippe Burlina, David E Freund, Neil Joshi, Y Wolfson, and Neil M Bressler *Detection of age-related macular degeneration via deep learning*. In 2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI), pages 184–188. IEEE, 2016.

[10] Tae Keun Yoo, Joon Yul Choi, Jeong Gi Seo, Bhoopalan Ramasubramanian, Sundaramoorthy Selvaperumal, and Deok Won Kim. *The possibility of the combination of oct and fundus images for improving the diagnostic accuracy of deep learning for age-related macular degeneration*: *a preliminary experiment*. Medical & Biological Engineering & Computing, 57(3):677–687, 2019.

[11] J. W. Newburger, M. Takahashi, M. A. Gerber, M. H. Gewitz, L. Y. Tani, J. C. Burns, S. T. Shulman, A. F. Bolger, P. Ferrieri, R. S. Baltimore, W. R. Wilson, L. M. Baddour, M. E. Levison, T. J. Pallasch, D. A. Falace, K. A. Taubert. *Diagnosis, treatment, and long-term management of kawasaki disease a statement for health professionals from the committee on rheumatic fever, endocarditis and kawasaki disease*. Council on cardiovascular disease in the young, american heart association. Circulation 110, 2747–2771, 2004.

[12] J. M. Orenstein, S. T. Shulman, L. M. Fox, S. C. Baker, M. Takahashi, T. R. Bhatti,

P. A. Russo, G. W. Mierau, J. P. de Chadarévian, E. J. Perlman, C. Trevenen, A. T. Rotta, M. B. Kalelkar, A. H. Rowley. *Three linked vasculopathic processes characterize kawasaki disease: a light and transmission electron microscopic study*. PloS one 7, e38998, 2012.

[13] K. C. Harris, A. Manouzi, A. Y. Fung, A. De Souza, H. G. Bezerra, J. E. Potts, and M. C. Hosking. *Feasibility of optical coherence tomography in children with kawasaki disease and pediatric heart transplant recipients*. Circulation: Cardiovascular Imaging 7, 671–678, 2014.

[14] S. Celi and S. Berti. *In-vivo segmentation and quantification of coronary lesions by optical coherence tomography images for a lesion type definition and stenosis grading*. Med. Image Anal. 18, 1157–1168, 2014.

[15] H. R. Roth, A. Farag, L. Lu, E. B. Turkbey, and R. M. Summers. *Deep convolutional networks for pancreas segmentation in ct imaging*. in SPIE Medical Imaging. International Society for Optics and Photonics, pp. 94131G., 2015.

[16] F. Ciompi, B. de Hoop, S. J. van Riel, K. Chung, E. T. Scholten, M. Oudkerk, P. A. de Jong, M. Prokop, and B. van Ginneken. *Automatic classification of pulmonary peri-fissural nodules in computed tomography using an ensemble of 2d views and a convolutional neural network out-of-the-box*. Med. Image Anal. 26, 195–202, 2015.

[17] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle. *Brain tumor segmentation with deep neural networks*. Med. Image Anal. 2016.

[18] Lowe, David G. *Distinctive Image Features from Scale-Invariant Keypoints*. International Journal of Computer Vision. Volume 60, Number 2, pp. 91–110.

[19] Muja, M., and D. G. Lowe. *Fast Matching of Binary Features*. Conference on Computer and Robot Vision. CRV, 2012.

[20] Muja, M., and D. G. Lowe. *Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration.* International Conference on Computer Vision Theory and Applications.VISAPP, 2009.

[21] S. Hochreiter and J. Schmidhuber. *Long short-term memory*. Neural Computation 9, 1735–1780, 1997.

[22] D. H. Hubel and T. N. Wiesel, *Receptive fields of single neurones in the cat's striate cortex*. J. Physiol. 148, 574–591, 1959.

[23] Matas, J., Chum, O.: *Randomized RANSAC with* $T_{d,d}$ *test*. Image and Vision Computing 22(10), 837–842, 2004.

[24] Capel, D.: *An effective bail-out test for RANSAC consensus scoring*. In: Proc. BMVC, pp. 629–638, 2005.

[25] Torr, P., Zisserman, A.: MLESAC: *A new robust estimator with application to estimating image geometry*. CVIU, 138–156, 2000.

[26] Kingma, Diederik, and Jimmy Ba. *Adam: A method for stochastic optimization*. arXiv preprint arXiv:1412.6980 , 2014.

[27] Hastie, T., R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*, Second Edition. NY: Springer, 2008.

[28] Manning, C. D., P. Raghavan, and M. Schütze. *Introduction to Information*

Retrieval, NY: Cambridge University Press, 2008.

[29] Stephen O'Hara and Bruce A. Draper. *Introduction to the Bag of Features paradigm for image classification and retrieval*, arXiv:1101.3354v1, 2011.

[30] Sivic, J. and A. Zisserman. *Video Google: A text retrieval approach to object matching in videos*. ICCV, pg 1470-1477, 2003.

[31] Philbin, J., O. Chum, M. Isard, J. Sivic, and A. Zisserman. *Object retrieval with large vocabularies and fast spatial matching*. CVPR, 2007.