Semantic Segmentation of Finger Layer Boundaries in Optical Coherence Tomography Images via Deep Learning

Aaron Flanagan

College of Science & Engineering

National University of Ireland Galway

Supervisor

Dr Frank Glavin

In partial fulfillment of the requirements for the degree of

MSc in Computer Science (Artificial Intelligence - Online)

August 2021

DECLARATION I, Aaron Flanagan, do hereby declare that this thesis entitled Semantic Segmentation of Finger Layer Boundaries in Optical Coherence Tomography Images via Deep Learning is a bona fide record of research work done by me for the award of MSc in Computer Science (Artificial Intelligence - Online) from National University of Ireland Galway. It has not been previously submitted, in part or whole, to any university or institution for any degree, diploma, or other qualification.

Signature: _____

Dedication

This thesis is dedicated to my family and to the memory of a beloved friend and father Gerald Crosse. Gerry displayed an unparalleled level of strength and understanding as he cared for his four ill children for over 18 years. He provided us with the motivation to continue fighting during these difficult times battling the COVID-19 pandemic and dealing with his loss. We hope that he is resting peacefully and thank him for all that he has provided.

Acknowledgements

I would like to express my greatest appreciation to Dr. Frank Glavin for supervising this project, his guidance and knowledge provided a great contribution to the research. I would also like to offer my special thanks to Ryan McAuley and Ross Doherty for the very appreciated support they provided by contributing their time, wisdom, and encouragement.

Abstract

Medical imaging systems continue to advance and often require new and exciting methods to explore the rich and complex information they make available. Deep learning has become an important component in medical imaging analysis and plays a key role in the understanding of biological tissue structures, pathology, and ocular disease diagnosis. Optical coherence tomography (OCT) is light interference based imaging method for mapping the internal structures of tissue in a non-invasive and non-destructive way. Extensive research has been carried out on the semantic segmentation of boundary layers in OCT images, however, no previous study has investigated this task on finger tissue scans. Furthermore, OCT is only sensitive to changes in tissue at a micrometre level, a new method entitled nano-sensitive OCT (nsOCT) has been proposed that increases the sensitivity of the changes in the sample instead of the conventional increase in resolution. It has been demonstrated that this approach can successfully monitor changes at a nanoscale level. In this thesis, the aim is to assess the state-of-the-art and to achieve semantic segmentation of the finger layers in OCT and nsOCT images. The evidence suggests that it is achievable and to the best of my knowledge this is a novel task.

Keywords: optical coherence tomography, fully convolutional networks, semantic segmentation, nano-sensitive OCT, medical imaging techniques, convolutional neural networks, and image classification.

Contents

List of Figures

List of Tables

List of Acronyms

Adam Adaptive Movement Estimation. 33

AMD Age-Related Macular Degeneration. 1

ANN Artificial Neural Network. 10

CNN Convolutional Neural Network. 2

CNV Choroidal Neovascularization. 1

CT Computed Tomography. 1

FDOCT Fourier-domain OCT. 6

ICHEC Irish Centre for High-End Computing. 24

MRI Magnetic Resonance Imaging. 1

nsOCT nano-sensitive Optical Coherence Tomography. 2

NUIG National University of Ireland Galway. 2

OCT Optical Coherence Tomography. 1

ReLU Rectified Linear Unit. 12

RNN Recurrent Neural Network. 23

SDOCT Spectral-Domain OCT. 6

SESF Spectral Encoding of Spatial Frequency. 9

SGD Stochastic Gradient Descent. 33

SLURM Slurm Workload Manager. 24

SSOCT Swept-Source OCT. 6

TDOCT Time-domain OCT. 6

TOMI Tissue Optics and Microcirculation Imaging. 2

Chapter 1

Introduction

1.1 Overview

Medical imaging systems are a key instrument in biological tissue research, pathology, and other related disciplines. Magnetic Resonance Imaging (MRI), X-Ray imaging, Optical Coherence Tomography (OCT), Ultrasound imaging and Computed Tomography (CT) are popular modalities adopted in the medical industry for a variety of tasks and most widely known for their usage in the medical diagnosis and research domains. With the advancement of technology in the last few decades their potential has grown exponentially and provided researchers with a greater understanding of the structures and processes in biological tissue along with greater image processing techniques [8]. This is evident when considering the usage of such techniques in medical care to perform tasks such as x-ray scans for the diagnosis of bone breakages or fractures, OCT scans for diagnosing ocular diseases like Age-Related Macular Degeneration (AMD) and Choroidal Neovascularization (CNV) [9, 10], or Ultrasound imaging to capture sonograms during pregnancy. The central focus point of this thesis will be on the research conducted on the application of deep learning to OCT.

OCT is a light interference based approach for imaging the sub surface structures of biological tissues in-vivo and studying changes in the internal structure of the sample at the micrometre level. Methods have been demonstrated that extend these observations to the nanoscale level however these come with caveats

concerning the efficiency, sensitivity, and resolution of the results [11]. In an attempt to overcome these limitations a new technique referred to as nano-sensitive Optical Coherence Tomography (nsOCT) invented in the National University of Ireland Galway (NUIG) by the Tissue Optics and Microcirculation Imaging (TOMI) group has been proposed. This allows for the visualisation of structural changes in both time and space, at the nanoscale level, by increasing sensitivity to the changes happening in the sample instead of focusing on an increase in resolution [12].

Extensive research has been carried out on application of deep learning to OCT images for tasks such as disease classification, boundary segmentation or abnormality detection. There is distinct lack of work that attempts segmentation of finger layer boundaries in OCT B-scans, and due to the recent development of nsOCT, no such work exists. As mentioned the main objective of this research is to investigate the state-of-the-art for layer segmentation in OCT images. A popular approach for this type of task is a Convolutional Neural Network (CNN), a state-of-the-art deep learning technique. CNNs have inherited widespread popularity in the field of medical imaging due to their powerful properties and characteristics, such as learning complex features automatically [13, 14]. CNNs have the ability to deconstruct images, generate complex feature maps, capture their low-level properties, and reconstruct the original input. With this ability, CNNs can be trained to segment and classify individual pixels as belonging to a certain class or instance present in the image. These techniques are known as semantic and instance segmentation, respectively. This thesis will focus on achieving semantic segmentation of the layers, visualised in the OCT and corresponding nsOCT images. The OCT images are be based on grayscale intensity and the nsOCT images on their spatial frequency content.

The following sections in this chapter will provide the motivation and structure of this document. In the next chapters a background section explaining the technologies and a literature survey have been provided. The literature survey will focus on various permutations of deep learning techniques applied to OCT imaging. The knowledge gathered has been applied to the design and implementation of a new network with the goal of segmenting the spatial boundaries in the nsOCT images, that are easily visualised in the original OCT images.

1.2 Motivation

nsOCT research is still in its nascent period of development with new techniques and potential applications being discovered [3, 11, 12, 15–23]. This research is considered the first application and implementation of deep learning to nsOCT images, classifying it as a novel topic. Furthermore, the segmentation of the stratum corneum and papillary junctions, the surface and internal fingerprints, in OCT finger scans have been rarely explored in literature. Older conventional methods exist and proposals for fingerprint extraction through deep learning from volumetric data have been reviewed. However, the exploration and segmentation of these layers in a two-dimensional capacity has not been, to the best of my knowledge. The motivation behind this project is to determine if it possible to achieve semantic segmentation of the tissue layers spatial boundaries visualised in the OCT and matching nsOCT images. Positive results hold great potential for further research with many benefits including:

- Knowledge gathering - The results of this study may provide a greater understanding of the tissue structures in the samples while potentially highlighting patterns that are not easily perceptible to the human eye.
- Hypothesis testing - Cross analysing the segmented image with the original samples structure will enable a researcher to estimate and deduce results based on experimentation's they may be conducting.
- Automated disease classification - Successful results may lead to further development of different solutions capable of narrowing and classifying known diseases that exhibit changes at particular depths.
- Noise isolation - Tissue samples and imaging systems are susceptible to noise based on the environment, equipment, and general mechanisms of light. If segmentation is achieved on an input image the noise should be theoretically separable from the official structure and properties of the sample.
- Faster research time - Researchers will need to spend less time categorizing and locating the spatial co-ordinates of structures in the scans.

- Commercial applications - This work theoretically has the potential to become a product with market applicability's.

1.3 Research Questions

Q1. Can semantic segmentation of the finger layers in OCT images be achieved? The ability to identify and segment layer boundaries in medical imaging scans has undergone extensive research. No research has been conducted on OCT finger scans, so the main objective is to determine if the state-of-the-art methods can achieve segmentation of the two layers visually observable in the B-scans. Tissue structures vary from person-to-person, so it will require a flexible solution with a good generalisation of the local features that present in every sample.

Q2. Can semantic segmentation of the finger layers in nsOCT images be achieved? nsOCT images are produced from the raw interferograms recorded during the scanning process for OCT. Each pixel in an nsOCT image represents a dominant wavelength at which a scatterer or structure is resolved. As a result, the images can look noisy and the internal fingerprint layer is not visually distinguishable, therefore, the objective is to determine if the state-of-the-art can attempt to highlight the two layers.

1.4 Structure of Thesis

Chapter 2: Background
This thesis reviews expert knowledge in two academic domains: Medical Imaging and Deep Learning. The Background chapter will describe the technologies involved to provide context to this research.

Chapter 3: Literature Review
The Literature Review chapter will discuss previous work carried out on similar tasks with the aim of applying some of the methods and findings to the implementation of a new CNN.

Chapter 4: Project Configuration
This section will describe the environmental configurations, the data collection, data pre-processing and curation of the segmentation masks. They will act as the ground truth data set for training the solution.

Chapter 5: Experimentation
The Experimentation chapter will detail the model proposal and changes made when compared to the standard designs. It will cover the training and preliminary results and findings.

Chapter 6: Results
The Results chapter will detail the final findings, interpretations and propose extensions to consider for future work on this task.

Chapter 2

Background

2.1 Medical Imaging Systems

OCT, an optical analogue to ultrasound, is a light interference-based imaging technique for capturing high-resolution images of internal micro-structures in materials and biological tissues [9, 24]. The OCT process begins by splitting and directing light from a low coherence light source towards a sample arm and reference arm. The sample arm contains a scanning mechanism for focusing the beam to a specific point in the sample and it returns the reflected / backscattered light from scatterers in the sample. The reference arm delays the backscattered light reflected from a reference mirror. The reflected light from the sample and reference arms are then superimposed and manifested as interference on the surface of a detector and the resulting electrical signal is processed into an A-line. A-lines are axial scans in the depth direction of the image that are captured in succession by stepping or sweeping the focused beam from the scanning mechanism across the lateral direction of the sample. The resulting collection of A-lines, positioned around the focal point of the sample, are then concatenated to form a two-dimensional image referred to as a B-scan or B-frame. Refer to Figure 2.1 below for an example of a generic OCT schematic.

OCT systems are divided into two classes; Time-domain OCT (TDOCT) and Fourier-domain OCT (FDOCT), which includes Spectral-Domain OCT (SDOCT) and Swept-Source OCT (SSOCT) [2]. TDOCT gradually varies the reference

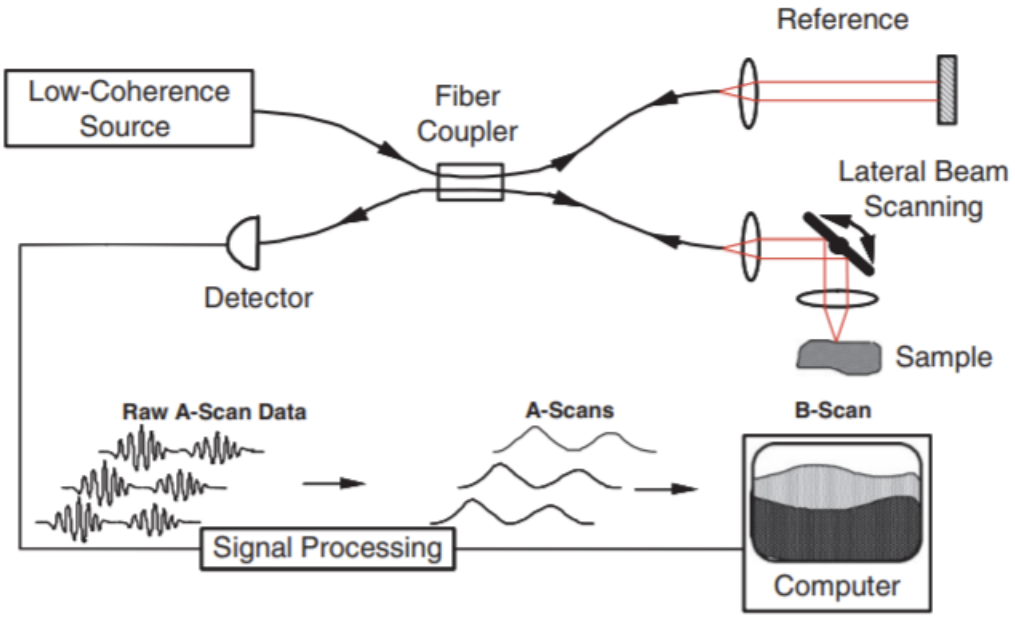

Figure 2.1: Schematic of a generic fiber optic OCT system. Bold lines represent fiber optic paths, red lines represent free-space optical paths, and thin lines represent electronic signal paths [1].

mirrors position at each lateral point of the sample. The interference pattern is generated from superimposing the varying positions of the reference arm and the sample arm which is then sent to a single detector and produces the A-line. FDOCT behaves similarly, however the reference mirrors position will remain static during the process. Capturing the A-line depends on the technique employed, SDOCT or SSOCT. In SDOCT when the remitted light from both arms is superimposed, the light is first dispersed into an array by a spectrometer based on their spectral frequencies and the interference patterns are detected by a charge-coupled device. After some pre-processing of the interferogram a Fourier transform is applied resulting in the A-line. In contrast to SDOCT, which uses a broadband and continuous-wave light source, SSOCT uses a single-frequency light source and a single detector. SSOCT captures the spectral frequencies over time, as opposed to a batch, by sweeping in varying wavelengths thus creating the interference patterns one at a time. The interference patterns are then superimposed to generate the interferogram and the same processing steps as in

SDOCT are applied to obtain the A-line [1].

When imaging sub surface structures via OCT many factors must be considered in order to obtain an accurate and representational output of the sample. These factors include the depth of the sample, the light source, wavelength, attenuation, and absorption rate, required resolution, and the heterogeneity of the tissue. For example, if the tissue at particular focal point is very scattering attenuation will become high and backscattering will be less, resulting in poor depth resolutions. Equally, if the tissue is very absorbing the intensity of the reflected light will be very low [25]. This can be easily visualised in the B-Scan images which are outputted in grayscale. Black, set at a value of zero, indicates no reflectivity and white, set at a value of one, indicates maximum reflectivity. An example of an OCT B-scan can be seen in Figure 2.2 showing the tactile portion of a human finger. The image includes two layers, the epidermis and dermis, imaged 1.5 mm below the skin layer. The ridges running from the top layer into the dermis via the epidermis are the sweat glands of the finger. An example of attenuation can be seen in the image from top to bottom where the intensity lowers, and this results in poorer depth resolution.

OCT is a very powerful method for imaging biological tissue, even when considering the important factors mentioned above. Even if the depth information

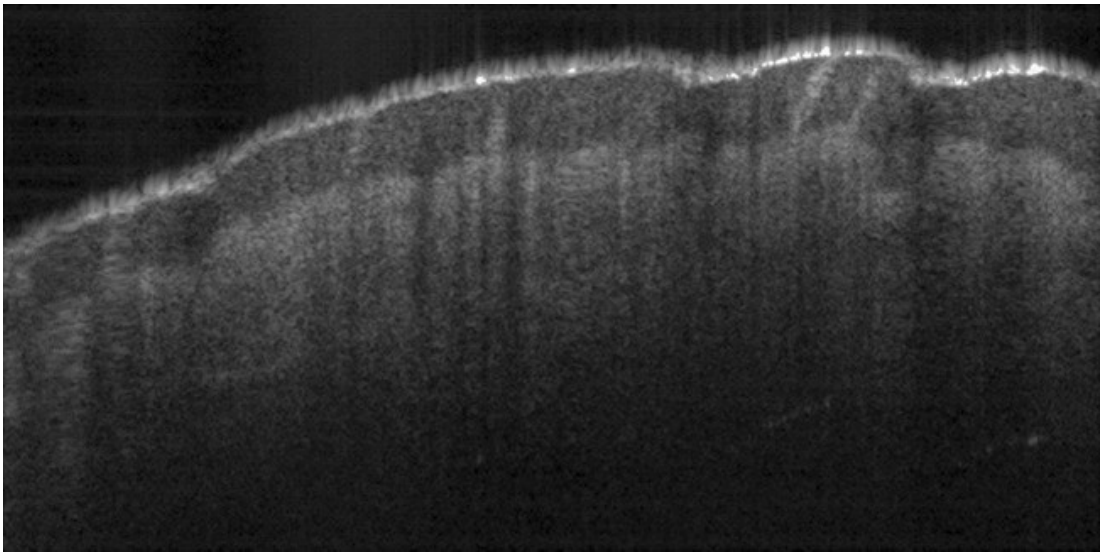

Figure 2.2: OCT image of a 3-mm-wide section of the tactile portion of a finger [2]

gathering is maximised during a scan, OCT still has limitations regarding the penetration depth and resolution. More specifically it is limited to the visualisation of structural changes in tissue samples at the microscale and as a consequence it is unable to resolve structural changes at the sub-micron scale, or nanoscale. Pathological changes in living tissue, like cancer for example, exhibit changes at the nanoscale level and require new techniques that provide deeper resolutions to visualise the pathogenesis [16].

A new technique known as nsOCT has been proposed that applies Spectral Encoding of Spatial Frequency (SESF) to OCT [11]. It aims to combat the limitations of conventional OCT and allow the probing of 3D structures at the nanoscale level in a single OCT image by transforming components of a 3D object, spectrally encoded in remitted light, from the Fourier domain into the voxels of an OCT image without comprising its sensitivity [16]. This is the achieved by dividing the sampled spatial frequencies into blocks and deriving the dominant power for each. The nsOCT images are then visualised with an arbitrary colour scale used for representing the portions of the sub surface structures that are uniform based on the dominant spatial period that can be resolved at certain wavelengths. The Telesto system used in this research specifies the range from 1176.72 - 1413.14 nanometers (nm) however during the conversion the range is split in half from approximately 588.36 to 706.57 nm, blue to red on the arbitrary colour scale the TOMI group imposed in the images. A single pixel or small cluster of blue pixels contained in the output image will be observed from a shorter wavelength like 588.36 nm for example. While this creates a more flexible representation of the sub surface structures it is very susceptible to noise making it very difficult to differentiate between them. The resulting image may contain sections with a high variation of colours indicating that there is no dominant structure or that there is a lot of variation in that region of the tissue sample. An example of an OCT and its resulting nsOCT image can be seen in Figure 2.3 below.

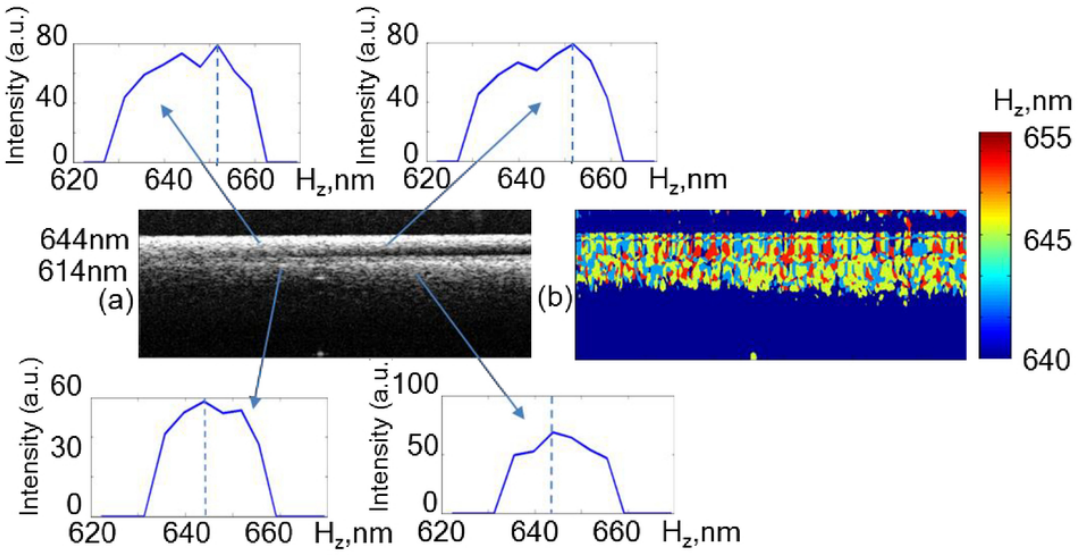

Figure 2.3: Images of nanospheres 614 nm and 644 nm diameters; a - conventional OCT image (B-scan) with axial spatial period profiles for selected locations. b - nsOCT image [3].

2.2 Convolutional Neural Networks

Convolutional Neural Networks are a state-of-the-art class of deep learning techniques inspired by the operations of the biological visual cortex [26]. Adapted from the classic Artificial Neural Network (ANN) design they provide less computationally expensive approaches to image classification, image segmentation, medical image analysis and computer vision approaches, along with other applications. More specifically, CNNs are a type of sparse multi-layer network that perform more accurately on tasks that require spatial or temporal awareness and where adjacent information may be highly correlated, examples include images, natural language, or time-frequency representations of speech [4]. By incorporating spatial information CNNs have the ability to detect low-level features in the early layers, and large complex patterns in the deeper layers. They are also able to identify similar features or objects in the input that may be varying in position or size, making them translation and shift invariant.

CNNs were pioneered by Yann LeCun et al. [27] with the LeNet architecture and later the LeNet-5. The objective for the LeNet model was to constrain the

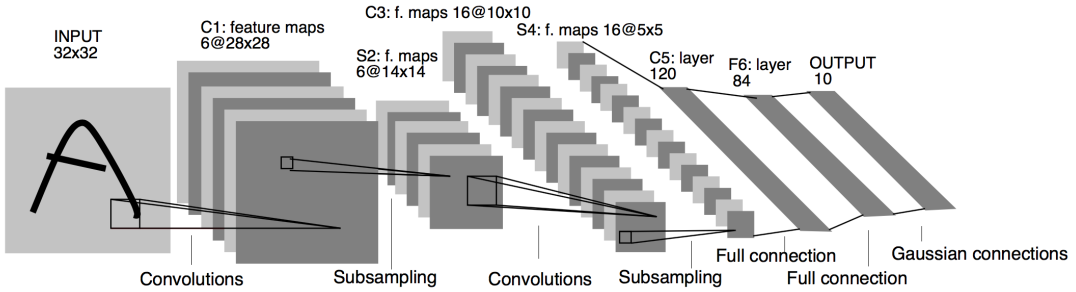

Figure 2.4: Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e., a set of units whose weights are constrained to be identical [4].

network by reducing the number of free parameters so that it may achieve a better generalisation of their task, classifying handwritten US zip code numbers. It combined a convolutional neural network, with a final classification layer, and the backpropagation training algorithm to train the weights. It was shown that sparse networks with constraints applied, reducing the number of free adjustable parameters, can generalise very efficiently. This work was later extended and introduced the LeNet-5 architecture, which was a huge contribution and motivation into the development of deep learning. Figure 2.4 below shows the original LeNet-5 seven layer architecture.

Referring to Figure 2.4, the most common layers in a CNN include the convolutional, pooling, fully connected, and classification layers. Layers are repeatable in sequence, or they can be stacked in different combinations depending on the required task. The fully connected and classification layers must be configured last, they are referred to as the head of the network. Examples of these layers can be observed in Figure 2.4 above, where the subsampling step refers to the pooling operation and the Gaussian connections are the final classification layer. Another important property of CNNs is that they adapt very efficiently to similar tasks without the requirement to train them from the beginning. This can be achieved in two ways; the CNN is designed to be fully convolutional or by using a modern deep learning framework such as Googles Tensorflow. The framework supplies the ability to export a classification CNN model without the head so that a practitioner can recycle a trained model for different tasks. The only requirement is

to create a new head for the network and train it for the new task, optionally freezing the imported layers and only training the updated head. As research continues the designs and testing of CNN models have tended towards deeper architectures and new techniques such as transfer learning with the attempt to improve performance. However, data processing and transformation techniques are equally as important when implementing a solution and often the feature engineering portion of a project consumes more time than the model design. Many research projects focus on how to experiment with and improve the performance of the current designs available [14, 28, 29].

As previously mentioned each layer performs some operation that transforms the input in some unique way. The main operation, the convolutional layer, operates by sliding a randomly generated $n \times n$ sized matrix of weights, denoted as a kernel, across the entirety of the input and performing an element-wise multiplication. It should be noted that the kernel is not reversed during the multiplication so the formal mathematical operation applied is cross-correlation, however I will continue to refer to it as a convolution step as it is described in the literature. The results of the convolution are then summed, and a bias is added before being passed through an activation function such as Rectified Linear Unit (ReLU), Sigmoid, or Tanh function. Then activation function determines if the value is above a certain threshold and should be activated. This operation is performed multiple times spanning across the input based on the number of slide steps and the kernel size. The number of sliding steps is referred to as the stride and the kernel size n is most commonly an odd number. The output is a $m \times m$ sized matrix called a feature map that represents how well a particular feature is resolved from the input image. A single value in a feature map is the output of the activation function at a particular point of the input image, termed the local receptive field. The term local receptive field corresponds to the spatial awareness concept described above where a single neuron / unit in the feature map constitutes spatial information from a $n \times n$ patch of the input. An example of a convolution with 1 kernel and stride 1 can be seen in 2.5.

A single convolution layer can have multiple kernels, each testing for different features, and generating a feature map per kernel. Referring to Figure 2.4 it can be observed that after the first layer, there is six feature maps sized 28×28,

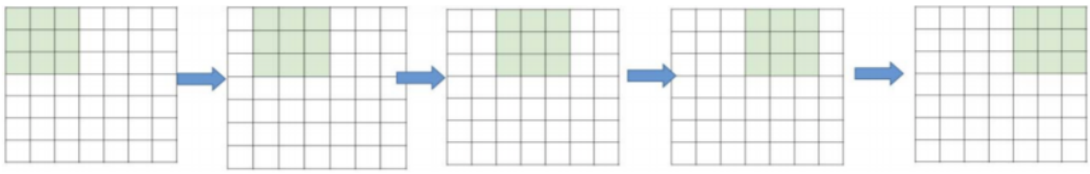

Figure 2.5: Convolution operation with stride=1, padding=0, kernel=3 × 3, and feature map output=5 × 5 [5].

resulting from applying this convolution operation to a single channel grayscale image of size 32 × 32 pixels. This means the first layer used six kernels, each sized 5 × 5 with a stride value of 1. The feature map size can be determined by applying the formula:

$$\frac{(n + 2p - f)}{s} + 1$$

where n represents the size of the input, f is the kernel size, p is the padding amount, and s is the stride value. If the feature map dimensions are required to equal in size with regards to the input image padding can be added before applying the convolution. The amount of padding required can be calculated by using the formula $p = (f - 1)/2$. After generating the features maps for each kernel, we now have a set of outputs, which feed as the inputs into the next layer. Layers that commonly follow a convolution step are another convolutional layer or the subsampling (pooling) step.

CNNs take advantage of spatial context during the convolution step however it is not necessary to store all the spatial information throughout the process. Once the convolution step is performed the spatial information is captured in the feature map which can lead to detecting features very early in the model. CNNs contain the property of translation and shift invariance, being able to detect similar features in arbitrary states and positions of the input. By incorporating all the spatial context captured early in the model the ability to detect more complex patterns deeper in the network becomes limited. The model becomes very sensitive to the position and orientation of features in the input. The pooling operation handles this limitation by reducing the size of the feature maps so that it only stores the prominent feature value information. The feature maps are reduced by taking the max or average value of the information in a similar

fashion to the convolution step. An $n \times n$ filter strides across each feature map and stores the max or average value calculated from the data in the window. If the filter size n and stride value s are equal, such that $n = s$, this will result in a non-overlapping reduction which greatly reduces the dimensions. If the filter size is set less such that $n < s$, this will result in overlapping information and less spatial context loss, which is hypothesized as a method to reduce over-fitting in the model [30].

The final layers in a CNN model include the fully connected and classification layers. Originating from the classic feed forward ANN they consist of a predetermined number of neurons that maintain connections with every neuron in the previous layer, and every neuron in the subsequent layer. The multiplication, summation, bias, and activation steps are identical when compared to the convolution steps described above, with the exception of the local receptive field concept. Two fully connected layers with a, metaphorically, infinite number of neurons can be used to approximate any mathematical function, this is termed function approximation. Each neuron contains a role in their respective layer and as training is performed the neurons learn to identify certain patterns, or "activate" in the presence of a feature. After training a number of fully connected layers the model begins to learn and recognise these patterns that enable it to perform classifications on the data. This is because ANNs take a simple generative modelling approach by creating a representation of the underlying distribution of the data and using this information they are able to recognise patterns and determine if new unseen inputs are similar. Increasing the number of neurons per layer, or an increase in the number of layers, provide the ability to model highly complex patterns and perform classifications on difficult tasks. Optionally the model can be created to be fully convolutional, opposed to the previous designs reviewed so far.

Fully convolutional models perform upsampling steps, also termed deconvolution or transpose convolution, instead of the feed-forward fully connected layers. The upsampling functions transform the feature maps into higher dimensions and construct an output with similar spatial dimensions. The upsampling functions were originally hand designed and hard-coded, until it was demonstrated by Long et al. [6] that nonlinear upsampling could be learned automatically in train-

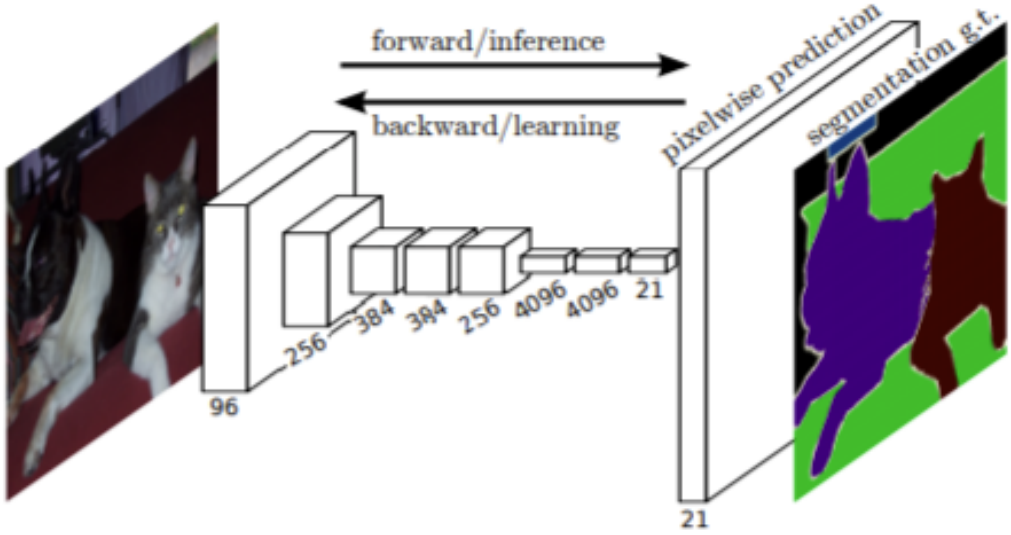

Figure 2.6: Fully Convolutional Network that performs dense pixel-wise predictions for semantic segmentation [6].

ing. They also described a new feature called skip connections that create links between the subsampling and upsampling layers. They are described as fusing the coarse semantic information from early layers with the fine-tuned inferences made at smaller dimensions deeper in the network. By utilizing the existing feature maps from the subsampling stages, the upsampling layers can reconstruct the image at scale with the learned spatial co-ordinates. This technique has been demonstrated to be more computationally efficient when training and it has the ability to make classification predictions at a pixel-to-pixel level.

Chapter 3

Literature Review

This chapter reviews previous studies conducted on the application of deep learning to OCT imaging for layer segmentation in tissue samples. The survey is divided into three sections, the state-of-the-art, the U-Net CNN, and the CIFAR-CNN. All articles were published between 1982 and 2021 and they were accessed through the NUIG library catalogue, the National Center for Biotechnology Information, IEEE Xplore Digital Library, Applied Physics Letters, Nanoscale, Science Direct, or the Nature Research Journal. This chapter does not contain any purchased or otherwise licensed literature, with the exception of the licensed material made available to NUIG. The search criteria included the terms "optical coherence tomography", "fully convolutional networks", "semantic segmentation", "nano-sensitive OCT", "medical imaging techniques", "convolutional neural networks" and "image classification".

3.1 Conventional Methods

Literature surveys have been conducted that investigate the growing interest and types of machine learning algorithms applied to medical imaging tasks. They propose that older conventional methods such as decision trees and support vector machines can be useful for topics like abnormality detection or image classification, but despite ongoing efforts they often produce a high number of false positives. In comparison, it is demonstrated that when deep learning is applied and

compared to the older conventional methods, higher accuracy and less sensitive results can be achieved, that also outperform human experts in their respective areas [13, 14, 31]. This is a result from the improvements in both machine learning models that can create complex representations of the data and a number of powerful feature engineering techniques that have been developed in recent years. These techniques include data processing, transfer learning, ensemble classifications, graph theory, and searching algorithms. Essentially, it is summarised that feature engineering in conjunction with a complex hypothesis language is what makes the deep learning methods successful and the current state-of-the-art.

The vast majority of the state-of-the-art segmentation methods for OCT study the retinal layers due to the ability to detect ocular diseases such as AMD and CNV. Methods have been proposed for achieving this through non deep learning methods. A paper by Sekulska-Nalewajko et al. [32] demonstrates the ability to detect the internal fingerprints by using the papillary junction. The papillary junction is a section of valleys and ridges that connects the stratum corneum, the topmost layer, to the epidermis sub layer, and these layers form the surface and internal fingerprints. They are identified by slicing a 3D volume and extracting a single B-scan, applying some speckle noise reduction and diffusion filtering, and then analysing the interferograms (A-line) for peaks in the signal based on the gradient, width, and intensity. Darlow et al. [33] adapt a similar approach but instead apply an unsupervised k-means clustering algorithm. They first identify the stratum corneum through peaks in the interferogram (A-line) extracted from a single B-scan in the volume. A high location precision of the stratum corneum leads to useful feature extractions that are then used as the basis for defining the clusters. Some post-processing and fine-tuning of the cluster data through image sharpening can identify the internal fingerprint of a single B-scan.

3.2 U-Net

The literature survey resulted in many studies referencing the U-Net architecture, similar to the papers reviewed above, but focusing on other tissue types like the retina. The motivation for the U-Net architecture was to train a CNN with minimum training data, which is a common requirement because data volumes

can be quite sparse for medical imaging tasks. This was achieved through data augmentation, applying deformations to the data, and the fully convolutional architecture style. The network operates by feeding the input images into two 3×3 convolutional layers, each producing a set of feature maps that are normalised and put through a ReLU activation function. The feature maps are then passed to a max-pooling layer for down-sampling where the number of feature channels are doubled, and the image dimensions are reduced by half. This is repeated four times until the fifth convolution block that does not apply pooling. This enables the network to learn very small and discriminative features at low levels of the input, a process known as the contracting path, or recently termed the encoder block. This design achieves low-level localization of each pixel in the input so it can later be classified as belonging to a surrounding object or class. The next phase is the expanding path, or decoder block, where upsampling is done via a 2×2 transpose convolution. The localized pixel information captured in the beginning is used during the expanding path to rebuild the feature maps at each scale. The feature maps from the contracting path, used to supply the contextual information learned, are cropped, and concatenated to the feature maps at the corresponding upsampling stage of the same channel dimension. This is followed by a further two convolutions with normalization and a ReLU activation function after every upsampling. The local and global information are continuously merged while the upsampling is performed, reducing the dimensions of the feature maps by half, and doubling the image dimensions. A 1×1 convolution layer with a softmax activation produces the final probability maps where the number of output channels is directly correlated to the number of classes. The softmax function is used to generate the probability of each pixel belonging to a certain class, where each channel dimension in the output contains a vector of prediction probabilities per pixel. That is, each pixel is represented as a vector size n, where n equals the number of classes, and the values correspond to the probabilities of the pixel belonging to one of the classes [7].

At the time of writing this review no literature was available that applied deep learning methods for the task of boundary segmentation in OCT, specifically for finger scans. However, two new papers have been published that focus on fingerprint extraction in 3D volumes, and to the best of my knowledge they are

the only existing work available. The first paper by Wang et al. [34] published in 2020 implements a 3D U-Net architecture for the identification of the stratum corneum and papillary junction contours in volumetric OCT finger scans, and the extraction of the 2D fingerprint. The architecture is a standard U-Net style design with 3D convolutions instead of 2D to take advantage of the inter-slice spatial dependencies in the volume. It starts with 32 feature maps and contracts to 256. The expansive path is a deconvolution upsampling with nearest neighbour interpolation and a final softmax layer generating three probability maps for the stratum corneum, papillary junction and the background. The model is trained with what they call regions of interest (ROI) which are slices extracted from the OCT volume on the x and y axis, a top-down view when looking at the tactile portion of the fingertip. After identifying and successfully segmenting the layers they produce a 2D representation of the fingertip that is compiled from both layers. This is because biometric scanning is susceptible to spoofing and not effective for handling damaged skin on the surface layer fingerprint.

The second paper by Ding et al. [35] was released in 2021 and they propose a new architecture entitled the BCL-U Net that extends the work of Wang et al. [34]. This architecture adapts the standard U-Net to include bi-directional convolutional long short-term memory blocks (BDC-LSTM), hybrid dilated convolutions (HDC) and residual learning, with three distinct phases. Each convolutional layer includes batch normalisation and a ReLU activation function. The contracting path contains three residual blocks, each with two 3×3 convolutions and a 1×1 convolution with batch normalisation is used for the skip connections. The second and third block contain an extra convolutional layer with a stride of 2 to perform down-sampling instead of the standard pooling layers. Lastly the HDC block has three dilated convolutions with a stride of $1, 2$, and 3, respectively. The HDC layers are introduced to enlarge the receptive fields without the loss of resolution and information of the small sweat glands. The BDC-LSTM part of the network is called the concatenate phase, consisting of three 3×3 convolutions with alternating LSTMs between them. This phase links multiple slices and leverages the inter-slice spatial dependence inherent in 3D volume scans. The upsampling stage then consists of two upsampling steps and three residual blocks with a final 1×1 convolutional layer with a softmax layer.

A study performed by De Fauw et al. [36] utilizes the U-Net design to build an independent two-tier deep network approach for the segmentation and classification of AMD and CNV in patient retinal scans. In this approach the input images are manually segmented and used to train the segmentation network in isolation and the classification network is trained with near fifteen thousand tissue maps with a confirmed diagnosis. New inputs are first sent through the segmentation network and the results are then streamed into the classification network for diagnosis. The benefit of this approach is to decouple the classification from the segmentation. When images are captured they are susceptible to noise, machine malfunction, image resolution and various other factors that can affect the sequential segmentation process. The trade-off with this approach is the development, training time and cross validation of results from multiple sources.

F. G. Venhuizen et al. [37] applied semantic segmentation via a U-Net based CNN to segment the full area of the retina. This takes advantage of the U-Net architectures ability to perform spatially dense classifications of pixels and produce a probability map that indicates the likeness that a pixel sits inside or outside the retina. The goal was to develop a robust algorithm that can handle invariances, deformations, and fluctuations usually visible in retina scans with disease present. Variations are caused by retinal fluid build-up in some of the layers, causing segmentation performed by conventional methods to fail. Adjustments are made to this generalized approach by widening the receptive field used during the contractive path phase. By making the network deeper using more down-sampling steps they widen the receptive field by one pixel in all directions per network layer. The typical receptive field is 140×140 pixels, covering 1.6×0.5 mm of the image. Abnormalities can spread to at least 5 mm, so by including six down-sampling layers they were able to create a receptive field of 572×572 pixels, capable of capturing and classifying these abnormalities as part of the retina.

T. Kepp et al. [24] performed a study based on the U-Net architecture to segment the layers of mouse skin tissue samples captured via OCT in-vivo. The report focuses on the comparison of their approach, using densely-connected convolutions, to the baseline U-Net described above and an older method which per-

formed a random forest classification with graph-based refinements. The densely-connected convolutions are realized by adding another 1×1 convolution step, which they denote as extra skip connections, before the second ReLU activation function in every convolution step in order to concatenate the input with the outputted feature maps and associated spatial and localization information. This will carry forward more information about each scale of the feature maps to the expanding path phase. This resulted in a solution with the capacity to accurately segment multiple layers of an OCT B-scan and outperform the baseline U-Net architecture. Its key advantages are noted as enhanced feature propagation, feature reuse and the increase in gradient flow during backpropagation as a direct result of the former points.

3.3 CIFAR-CNN

L. Fang et al. [38] proposed combining an adopted CIFAR-CNN architecture with graph search algorithms to outline the boundaries of the retinal layers. Images are pre-processed and intensity normalization is applied to remove outliers and improve the overall computation speed of the network along with the use of ReLU activation functions to further normalise the data. 33×33 pixel patches, deviating from the typical 32×32 pixels approach, are manually produced, and selected at random from single A-scans around the retinal layers and used to train a CNN. Each pixel in the image patches is represented as a node with associated weights to interconnected pixels in every direction. The graph search algorithm applies Dijkstra's shortest path algorithm to traverse the image and find the minimum weighted path from the probability map that the CNN produces before finally outputting an image highlighting the segmented boundaries. The results of this paper were compared with manually annotated and segmented images produced from existing software by experts for comparison. It was found that the proposal outperforms older support vector machines and random forest classifier approaches previously done with results closely accurate to the manually generated results.

Hamwood et al. [29] extended on the work conducted by L. Fang et al. [38], by proposing a number of modifications that can be made to increase the per-

formance of the CNN. Their approach operates in the same manner, but they experiment with different configurations that can be made to the network to find a more optimal solution. They adopt a 65×65 pixel image patch size to further segregate the layer boundary and avoid confusing them with neighbouring boundaries, which should improve the performance of the probability generation and subsequent classification. They also decided to experiment by removing some of the pooling layers and replace them with convolution layers, the double convolution and down-sampling technique used in U-Net. This works when the input is not zero padded, that is where extra zeros are added around the boundary of the input to avoid the convolution wrapping around the image. The fixed window size and carefully selected stride implemented in this approach avoids this issue and as a result the output will be smaller and only contain pure input information. The results of these modifications show a slightly higher error rate at some layers, but a significant difference at others. The consequences are reflected in the many versions of the solution that need to be created. When the patch size is altered the fully connected layers need to be reconfigured to reflect the fixed size input expected, reflected by the selected patch size.

Expanding on these ideas further K. Hu et al. [39] created a multi-scale CNN, termed as MCNN, based on several deep CNNs and motivated by the CIFAR-CNN adaptation by L. Fang et al. [38]. They also adopt and improve the implementation of Hamwood et al. [29] by applying an updated graph search algorithm that replaces the original eight neighbour connection per pixel approach with a unidirectional three neighbour approach. This significantly reduces the amount of redundant information and helps distinguish background data more clearly to avoid misclassification. The network operates by extracting multiscale features from the inputs with two or three different patch sizes, 33 x 33 pixels in the CIFAR adapted network, 65×65 pixels, 17×17 pixels or both in their custom adaptations. The extra feature extraction layers with varying window sizes obtain more spatial and localization information from the pixels and generates finely-tuned features maps at different scales. They then merge the feature maps from the two extra layers back into the CIFAR-CNN implementation before running the features maps through the convolution, normalization, upsampling, classification and softmax layers respectively. The probability map is then provided as input

to the updated graph search algorithm to segment and produce the final output image with the boundaries highlighted.

Kugelman et al. [40] reviewed the patch-based CIFAR-CNN design proposed by Fang et al. [38] and the Complex CNN, extending and reviewing the effects of the patch-sizes, by Hamwood et al. [29]. They apply the patched based approaches to both CNNs and a third model extending on their previous work which proposed the Recurrent Neural Network (RNN) as an efficient model for handling image tasks [41]. All three models are trained to perform classifications with varying patch sizes; 32×32, 64×32, 64×64 and 12×32. They also applied a number of variations to the original U-Net architecture and trained the models on the same data to perform semantic segmentation of the full B-Scans instead. They concluded that the patch-based methods performed comparably for two retina layers well represented in the data, and that the increased patch sizes aided in the classification of the third layer, the choroid-scleral interface (CSI). The semantic segmentation tests behaved similarly, with the exception that changes in architecture did not have a noticeable effect on the classification accuracy of the CSI layer. The semantic segmentation models did achieve lower mean absolute errors (MAE) when compared to the patch-based models, with an MAE of 0.33 pixels, compared to 1.82 pixels, respectively. This is attributed to the additional contextual information available when processing a full B-Scan, in comparison to the manually extracted patches of the image.

Chapter 4

Project Configuration

4.1 Environments

4.1.1 Hardware

The training of the model was compared on three platforms using a CPU, and two GPUs. The data pre-processing, model design and the first training cycle was performed on a personal laptop. The Irish Centre for High-End Computing (ICHEC) [42] supercomputer *Kay* was then tested; this is a free service that NUIG Postgraduate students can utilise. Jobs are submitted via a bash script to the job manager Slurm Workload Manager (SLURM) requesting the resources for a predefined time window and number of nodes to execute across. The ProdQ with four CPUs and a wall-time of three hours were requested. Lastly, the subscription based cloud service Google Colab Pro was used for training on the GPUs. Table 4.1 below contains the hardware details and average training times, for each. The benchmarks consider 300 labelled images shuffled and trained over 10 epochs for both data sets.

4.1.2 Software

The project was developed using Python version 3.8.10 via the Anaconda distribution. The Keras deep learning API version 2.4.3 was used with the Tensorflow 2.3 framework. The API library and framework were selected for two reasons:

Table 4.1: Environment Training Times

Environment	Hardware	Epoch (mins)	Total (hours)
Local Machine	Intel(R) I7-7700HQ	120	20
Kay (ICHEC)	4x Intel Xeon Gold 6148	50	8
Google Colab	Nvidia Tesla K80	7.2	0.7
Google Colab	Nvidia Tesla P100-PCIe	1.2	0.2

- Keras and TensorFlow are open-source
- TensorFlow can efficiently scale to multiple CPU/GPUs

The latter point is a requirement to take advantage of the parallel processing properties offered by *Kay* and Google Colab. Other libraries include matplotlib for loading and visualising the data and results, imageio for the conversion and storage of the OCT grayscale images, and numpy as the main in-memory data utility used for loading the data, performing the pre-processing steps, and feeding it through the model.

4.2 Data

This section will cover the collection, preparation and curation of the data used in the project. The data collection and ground truth creation posed some challenges due to the ongoing COVID-19 pandemic. With restricted access to the research labs in place, along with the unforeseen family responsibilities and the illness of my colleague, the data collection was delayed until May 2021. Many attempts were made to collect sample data, however the quality of the scans was quite low. Issues with saturation and noise in my personal scans required the TOMI group to evaluate the data and make some changes such as noise reduction to accommodate my requirements. The conversion from OCT to nsOCT is also a computationally expensive approach, taking $2-3$ days to convert the data, review the output, filter the extra noise from the images, and export the binary file. It should also be noted that the stratum corneum in the OCT B-scans, the topmost layer observably separable from the background, contains a very weak signal not

visually observable in the OCT images. As a result, the stratum corneum coordinates on the y axis do not match spatially when compared to the nsOCT converted images.

4.2.1 Data Collection

The data was gathered using a Thorlabs Telesto SDOCT system. The collection method involved capturing a 3D volumetric B-scan of the right ring finger from a member of the TOMI group. The scan measured $3.6 \times 5 \times 5$ mm in height, length, and width respectively. It was exported as a .oct file that was imported into MATLAB, sliced into an array of 1000 2D samples, and converted using their nsOCT algorithm. The algorithm divides each image into 153 sub-bands depth wise and calculates the dominant wavelength per pixel, per band. This results in a wavelength value ranging between $588.36 - 706.57$ nm, each stored as a floating point number. The main author of the algorithm applied some extra thresholding and filtering techniques to the images before conversions, deviating slightly from how the technique is described in the publications. This was to reduce the autocorrelation and amount of saturation in the images so that the topmost layer can be easily distinguished visually in the image and separated from surrounding noise. The array of images was exported as a binary file formatted as a vector of length 1,024,000,000, with a size of 8 GBs.

4.2.2 Data Pre-processing

The binary files were imported, and some one time pre-processing steps were performed to ready the data. Each image has a single channel with dimensions 1024×1000. This was reshaped to dimensions $1000 \times 1000 \times 1024$, corresponding to the columns, sample dimension, and rows of the images. The axes were rotated to shape $1000 \times 1024 \times 1000$, with the sample dimension first as required in Keras. The binary file contains one blank image at the end of the array, so this was removed. Lastly the data contains empty background information and some noise at the bottom of each image. As previously described the scanning procedure operates depth wise, so this corresponds to the lack of information resolved at a greater depth in the OCT images before conversion. Each image

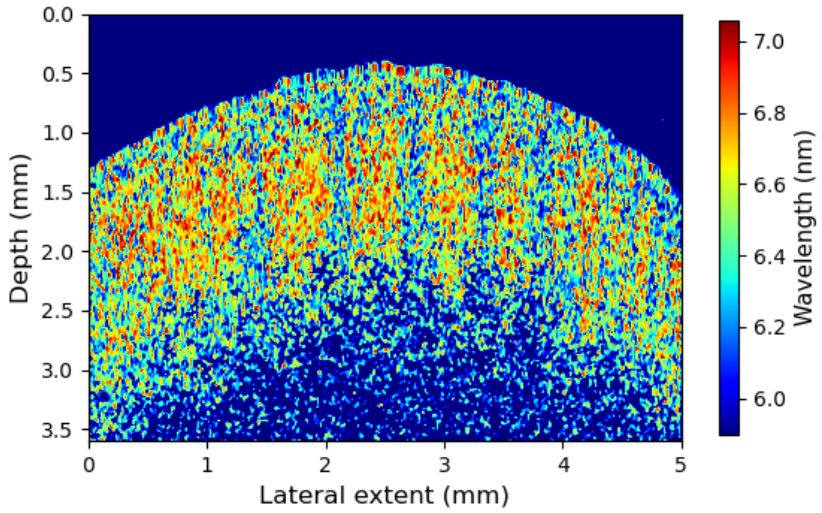

Figure 4.1: An nsOCT sample image with dimensions 3.6×5 mm displayed using a jet colour map with wavelength ranges between 588.36 - 706.57 nm.

was plotted and analysed to confirm that important information will be lost. After confirming, images were cut by 56% and exported as a final .npy binary file, reducing the files to roughly 4.5 GBs. A sample image can be seen in Figure 4.1 after the pre-processing was complete.

4.2.3 Segmentation Masks

The common method to developing deep learning applications for automated image analysis tasks is supervised learning. Supervised learning techniques typically require a sufficiently large amount of data and ground truth labels to influence the learning of the network and provide a wide array of samples to achieve a good generalisation. A particular challenge with medical imaging tasks is the lack of ground truth data sets available. Image processing techniques and existing tools were tested to try and automate the procedure or provide some basic heuristics that a human annotator could take advantage off. The approaches included applying Canny and Sobel filters and various image processing techniques like Gaussian blurring, mean, median and percentile filtering, and sharpening techniques to try highlight the boundaries slightly more for easier detection. It was decided that to achieve the most accurate set of border labels, they would be

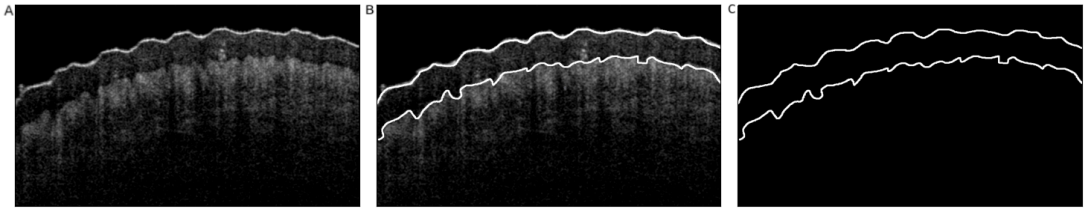

Figure 4.2: A - Original OCT B-Scan image, B - OCT B-Scan with projected layer boundaries, C - Final segmentation mask used for training.

applied manually to the OCT images, a common and time costly approach for imaging classification tasks.

The data set was imported and extracted from the "FImage" key item in the binary file applied by MATLAB. The last image was removed, the data was cut by 56% to reduce noise and the memory size, the images where then flipped horizontally by 180 degrees to fix how MATLAB exports them, and finally stored on disk as .png files. Each PNG file was stored as a single channel grayscale image with dimensions 580×1000. The files on disk were then ready for the segmentation mask curation. The masks were manually segmented in the Windows Paint 3D app. Each segmentation mask is manifested as a 5 pixel tick border super-imposed over the original OCT images. Each file was updated on disk and then some filtering was required to create the distinguishment between the background and the skin boundaries. The paint tool superimposed a vector of 1's representing the skin boundaries over the original image, the remainder of the pixels were set as 0 to represent the background. The masks were saved to disk as .png files and when they were loaded in for Keras an extra dimension was added to represent the single channel dimension, resulting in the shape $(N, 580, 1000, 1)$, where N represents the number of samples. Refer to Figure 4.2 to observe the segmentation process steps as described.

Chapter 5

Experimentation

5.1 Model Proposal

It has been demonstrated that the U-Net design, when applied to medical imaging tasks, performs outstandingly for image classification and segmentation. U-Net was designed as a method for handling low amounts of annotated images, which is a common occurrence in medical imaging. The proposal is to implement a standard U-Net architecture with some variations applied to achieve the desired output of the task at hand. A description of the U-Net process has been provided in Chapter 3, this section details the implementation and the deviations that will be applied. For context, please refer to Figure 5.1 below for the standard architecture.

The first half of the U-Net is the contractive path, consisting of four double convolution blocks followed by the max pooling layer, and one convolution block without pooling. The number of feature channels starts at 64 and is doubled in each block. The padding scheme is defaulted to valid, so padding is not applied. For this task I would like to output a predicted segmentation map with the same dimensions of the input. By segregating the background from the layers, the original image and the prediction mask can be compared either spatially or temporally. the padding scheme "same" was applied, which pads zeros around the borders of the input before the convolution, resulting in an output feature map of the same dimensions to the input. See Chapter 2 for the details on calculating

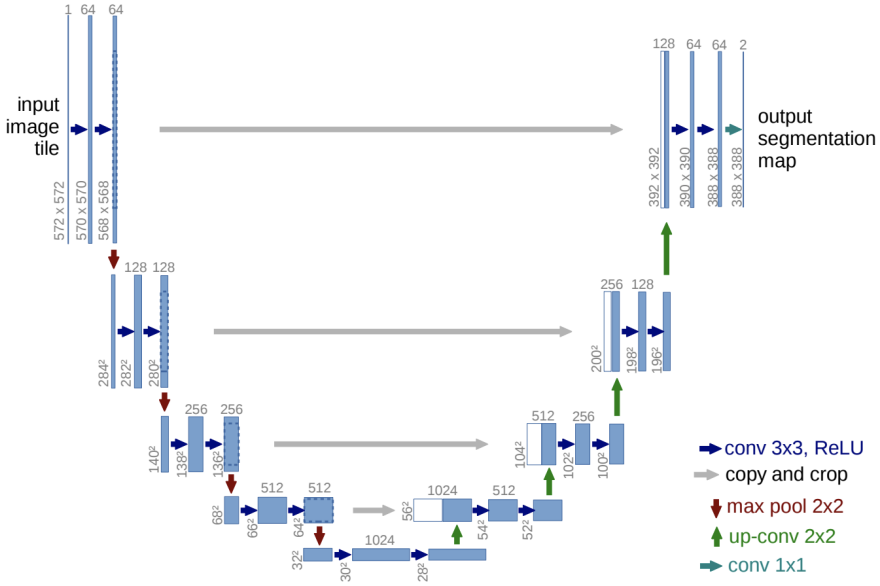

Figure 5.1: Original U-Net Design [7].

the padding.

The convolution operation behaves similarly to the standard feed-forward ANN design. Each pixel in the input is multiplied by the weight of the sliding kernel (neuron) of the same position. In the 3×3 kernel this will result in nine multiplications, per stride step. The final value is then the summation of the dot product and the bias that is put forward through a ReLU activation. This results in ten learnable parameters per kernel, including the bias, per convolution. In total the first block has 64 feature channels, which results in $64 \times 10 = 640$ parameters to train for the first layer of the first block.

The convolution formula is defined as:

$$a = f((W \circledast X) + b)$$

where W is the set of weights in a kernel, X is the input value the kernel strides across, and b is the applied bias. The output a represents the summed value from

the convolution and bias after applying the ReLU activation function:

$$\mathrm{a} = \max\{0, x\} = \begin{cases} x, x >= 0 \\ 0, x < 0 \end{cases}$$

The ReLU function returns 0 for all negative values and returns the input x if it is positive. In the original U-Net architecture they propose that the weight initialisation for this type of architecture should be sampled from a Gaussian distribution with a standard deviation of $\sqrt{\frac{2}{N}}$ where N is the number of weights of the kernel [7]. This is achieved using the He normal weight initialisation method presented by He et al. [43] and it is defined as the "he_normal" value for "kernel_initializer" in Keras models.

The max pooling layer is a 2×2 kernel with a stride of 2. Max pooling calculates the max value in the scope of the kernel at each stride step and stores that value in a new feature map. This will store the context of the features but discard the boundaries and location, to allow for a better generalisation. The problem that max pooling presents is that unless the image dimensions are constructed as power of 2, it can result in odd dimensions when down-sampling. This can be observed in Figure 5.1, during the expansive path phase the dimensions do not match the dimensions of the opposite contractive path step. In the original design this was handled by cropping the feature maps from the contractive path before the concatenation, resulting in a smaller output then the input.

The next phase of the design is the expansive path, consisting of repeating Conv2DTranspose layers followed by two convolutions. The Conv2DTranspose layer generates a random kernel, like the convolutional layers, that can be learned during training. At each upsampling step the number of feature channels are divided in half and the feature maps, before the pooling layers, from the contractive path are concatenated. The original network was designed to handle even dimension images, $N \times N$, and the convolutions were unpadded. As previously mentioned I opted for equal dimension outputs, which required padding and emitting the cropping of image dimensions for the skip connections. This created an issue when concatenating the layers because of the non-matching dimensions on the upsampling steps. The data for this task is rectangular, so to overcome this issue

ZeroPadding2D layers were added to the right and bottom of the image when the dimensions did not match during upsampling.

The final layer is a 1×1 convolution layer with a sigmoid activation function layer and one output channel. This differs from the softmax activation used in the standard network, however this task is binary so the sigmoid function will produce the same output. The sigmoid function is defined as:

$$\frac{1}{1 + e^{-x}}$$

where e is Eulers number and x represents the input value to the activation function. The sigmoid function will output values between 0 and 1 and they will be interpreted as the class label prediction per pixel. As x approaches very large numbers the value will tend towards one, while large negative numbers will always tend towards zero. This is required because the segmentation masks contain one channel specifying the background with a class label 0 and the foreground (layers) with a class label of 1. This design was considered to be more efficient when manually annotating the images as compared to the two channel approach were the background and foreground are separated into a single channel each. The loss function defined for the model is the binary cross entropy / log loss:

$$L(\theta) = -\frac{1}{N}\sum_{i=1}^{n}(y_i \cdot log f_\theta(x_i) + (1 - y_i) \cdot log(1 - f_\theta(x_i)),$$

where N represents the number of samples in the batch, y_i is the ground truth label and $f_\theta(x_i)$ is the predicted class label [44].

Additions to the design include batch normalisation and dropout regularisation. Batch normalisation is added after every convolutional layer, resulting in two normalisation layers per block in the down-sampling stage. Batch normalisation will adjust the underlying distribution of the data so that each convolutional layer will receive an input with the same mean and standard deviation as the previous layers [45]. To handle the overfitting of the model observed early in training a single dropout regularisation layer was added before each pooling layer in the contractive path. This was originally proposed by Hinton et al. [46] as a means of adding constraints to the neurons of hidden layers in fully connected

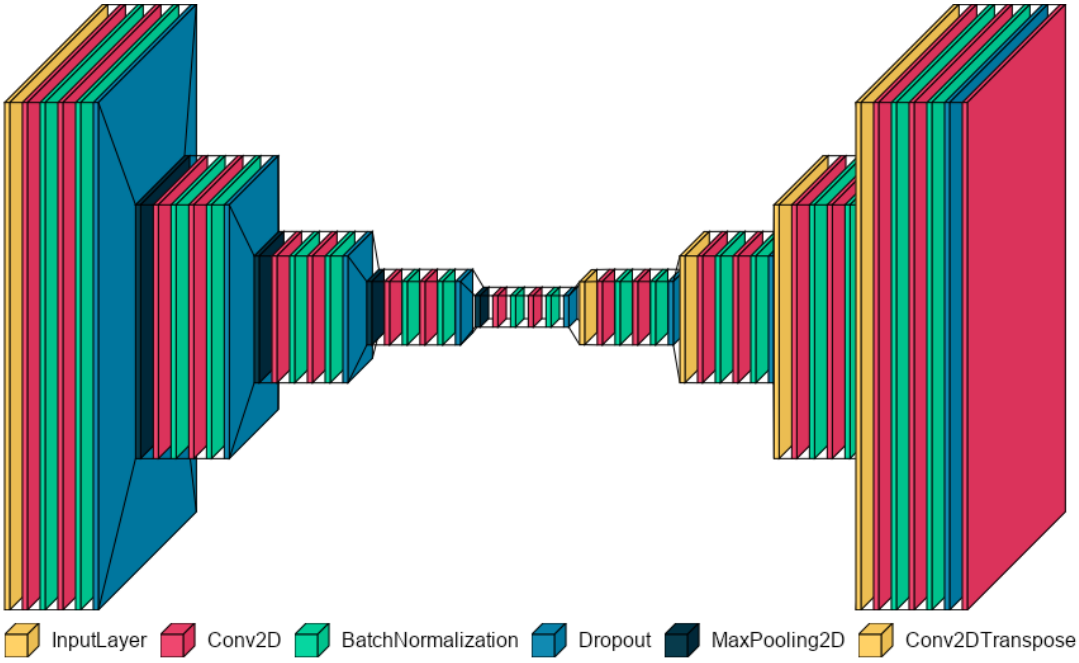

Figure 5.2: Model Proposal: U-Net style architecture including ZeroPadding2D, Batch Normalisation and Drop-out layers.

layers. Recent research suggests that dropout layers can provide similar benefits to convolutional layers to avoid overfitting [47]. During each training phase a percentage of hidden units are emitted, set to 0, based on the probability rate specified before training.

The final change is replacing the standard Stochastic Gradient Descent (SGD) method with the Adaptive Movement Estimation (Adam) optimizer [48]. The Adam optimizer improves on two previous optimization algorithms AdaGrad and RMSProp, by considering the first and second moment gradient estimates, mean and variance, with bias correction. It uses exponential moving averages to calculate parameter specific learning rates and it only requires the addition of two new hyperparameters, the exponential decay rates for the first and second moment estimates. This enables the parameters to change at an appropriate rate by considering the most recent changes and direction into the process while still being computationally efficient. Please refer to 5.2 for the adapted architecture proposal.

5.2 Experimentation

The experiments were conducted on both the OCT and nsOCT data sets. The models were trained using the standard backpropagation algorithm comparing the Adam optimizer with SGD. The standard model applies SGD and has approximately 31 million trainable parameters. The batch size, that determines how often the weights are adjusted during training, for both methods is set to 1, this is due to low RAM memory availability. The data contains 350 labelled examples in total, 301 samples were shuffled and split by 80% for training and 20% for validation with a final hold-out set of 49 samples used for model evaluation. Each epoch used 240 random samples for training and the remaining 60 for validation. A further 600 unlabelled examples were used to test the prediction masks visually. Both data sets were feature scaled / min-max normalised before input. The OCT data was scaled from $[0 - 255]$ to $[0 - 1]$ and the nsOCT images were scaled where 0 corresponds to the minimum wavelength 5.88e-7 and 1 corresponds to the maximum wavelength 7.06e-7.

A challenge that will be present in both scenarios is the class imbalance in the segmentation mask labels. Each mask contains exactly $580,000$ pixels and the superimposed boundaries with a class label of 1 constitute, on average, $12,000 - 12,500$ pixels in total. This means that the model, when predicting the class label 0, will be correct 97% of the time because the boundary layers only cover $2-3\%$ of the image. An accuracy around 98% must be achieved for a solution to distinguish between the two classes successfully and confidently. Forcing an accuracy rate this high can lead to overfitting on the training data, so as mentioned dropout regularisation was added. Dropout was tested at 10% and 40% in the initial tests, it was found that a rate of 40% was more efficient and had little negative impact on the final model performance. Lastly, it was observed that any solution that reached 98% accuracy was capable of segmenting the stratum corneum with high precision, although it could not fully identify the internal layer with the same confidence.

5.2.1 OCT

The first set of tests applied SGD varying the learning rate, three rates at $10e$-4, $10e$-3 and $5e$-2 were tested for 5 and 10 epochs each. It was observed that it takes a minimum of 7-8 epochs for a strong prediction mask to be created that can partially segment the boundary layers. The learning rate of $10e$-4 was too small, the model was not able to fully identify either boundary, the output prediction mask was a weak reconstruction of the full input. The learning rate of $5e$-2 resulted in lots of noise below the second layer, resolving to much information. After testing for 5 epochs with a learning rate of $10e$-3, the topmost layer was successfully segmented but the bottom layer was not identified. Further training for 10 and 15 epochs yielded better results, with the bottom layer now observable, but only partially and containing noise. It was found that SGD could partially achieve segmentation of the stratum corneum using these experimental settings, with a low confidence on the internal fingerprint up to 15 training iterations.

The Adam optimizer was successful in generating a strong prediction mask fairly quick. It was tested for 5 and 10 epochs respectively with learning rates $10e$-4, as suggested in the original paper, and $10e$-3 as observed when testing with SGD. The exponential decay rates for the first and second moment estimates were set as default, 0.9 and 0.999, respectively. The learning rate of $10e$-4 was too small and could not correctly structure the predication masks. After 5 epochs of training with a rate of $10e$-3 both layers were segmented partially and it contained some noise, after 10 epochs a strong prediction mask is generated. A further 15 epochs produced results with little difference from the previous test of 10 epochs. Testing for 20 epochs resulted in a more accurate solution, however, still containing some noise, and missing small portions of the layers. Additional training for 30, 40 and 50 iterations yielded higher accuracy results with very low loss rates but failed to fully segment the layers with higher precision then the lesser trained solutions. This observation was further quantified by visually observing 600 unseen samples, each containing noise and gaps in the boundaries. The Adam optimizer was more efficient in finding a good generalisation, reaching a 97% accuracy in $2 - 3$ epochs.

5.2.2 nsOCT

The key challenge specific to the nsOCT data is that the spatial co-ordinates of pixels do not align correctly with the corresponding pixels of the source OCT B-scan. This is because the OCT images have a weak signal above the stratum corneum and the nsOCT conversion is sensitive to this. This results in an image with the top layer starting higher on the y-axis when compared to the source B-scan. The challenge will be confidently predicting an accurate segmentation mask from the nsOCT images, that correspond to correct spatial co-ordinates of the layer visually observable in the OCT B-scans. The nsOCT images were tested with SGD in a similar fashion to the previous tests but compared to the OCT results it was unable to identify or segment the layers in any capacity. This included the same learning rates tested at 5 and 10 epochs each.

The successful experimental settings applied to the OCT images with the Adam optimizer were applied for 5, 10 and 15 epochs respectively to the nsOCT set. The Adam optimizer configured with a learning rate of $10e$-3, beta values of 0.9 and 0.999 and trained for 10 epochs performed the best. Both layers are identified but contain some gaps and noise, similar to the OCT data. As the training iterations increase the level of noise the model resolves does too. The next experiment consisted of twenty training phases in succession varying the hyperparameters for 10 epochs each. For each learning rate, where $lr = [0.01, 0.02, 0.05, 0.1, 0.2]$, the first moment beta value was increased, where $\beta 1 = [0.8, 0.85, 0.9, 0.99]$. The hyperparameters $lr = 0.01$ and $\beta 1 = 0.9$ and $\beta 2 = 0.999$ were observed to perform comparably to the OCT tests. Further training applying these parameters were executed for $20, 30, 40$ and 50 epochs.

Chapter 6

Results

The mean accuracy for the predictions on the training and validation for the OCT data set was positioned around 98%. During training the accuracy quickly converges in $2-3$ epochs, as can be seen in Figure 6.1. As training continues this increases linearly finishing at 98.25%. The validation data fluctuates quite frequently, this is contributed to the relatively small number of samples available. As each epoch begins the data is shuffled and the validation data from the previous

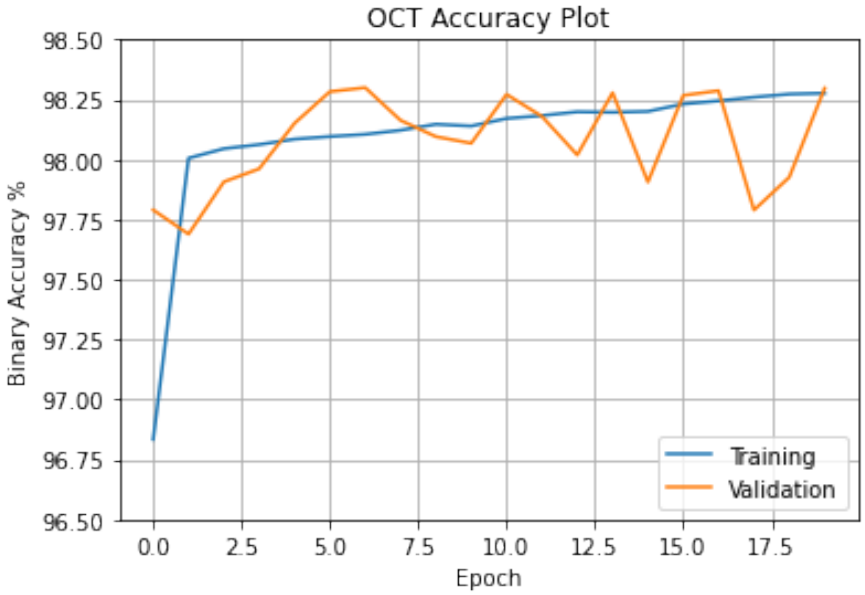

Figure 6.1: OCT Binary Accuracy per Epoch

epoch is possibly being sampled for training in the current iteration.

Figure 6.1 was generated from a model that was trained for 20 epochs where the learning rate $lr = 0.01$, $\beta 1 = 0.9$ and $\beta 2 = 0.999$. It achieved a final accuracy of 98.27% with a loss of 0.035. The Area under the ROC curve (AUC) was also measured during training, and for each validation set, both finalising at 0.98, which plateaued and remained unchanged for roughly 18 epochs. Similar to the accuracy rates the AUC values for the validation sets fluctuate around the training line, typically converging at the same value to the corresponding training set. In comparison, the validation sets did not fluctuate heavily with regards to the loss, instead the validation scores tended to spike in every test that lasted longer than 10 epochs. The log loss value per epoch can be observed in 6.2.

The training metrics plots are slightly misleading in regard to the intensity of the fluctuations during training. The model was evaluated using 49 unseen samples resulting in a binary accuracy of 0.98%, and a final loss of 0.0359. This indicates that the model has not overfit, and it is performing comparably to the training phase. However, an important factor is the small sample size of the hold-out set. The hold-out set also consists of B-scan slices that were split

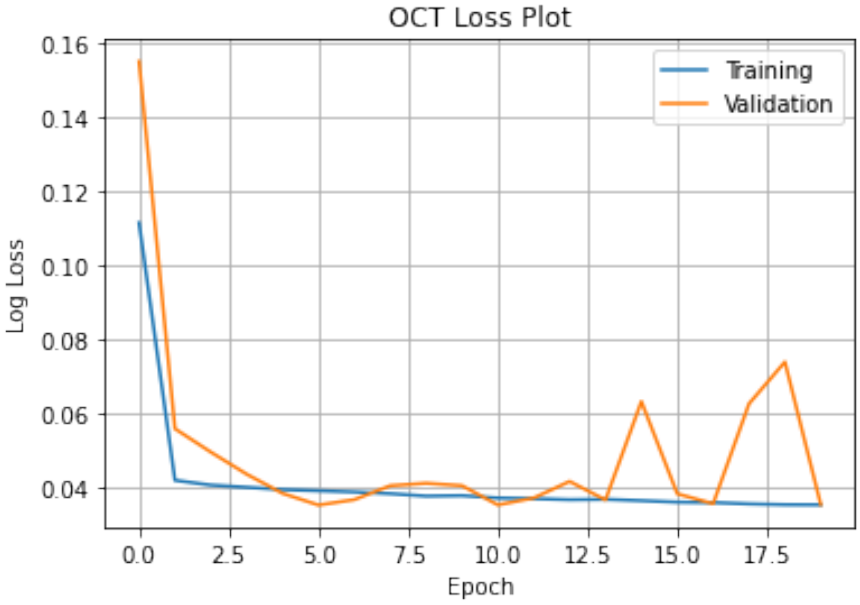

Figure 6.2: OCT Log Loss per Epoch

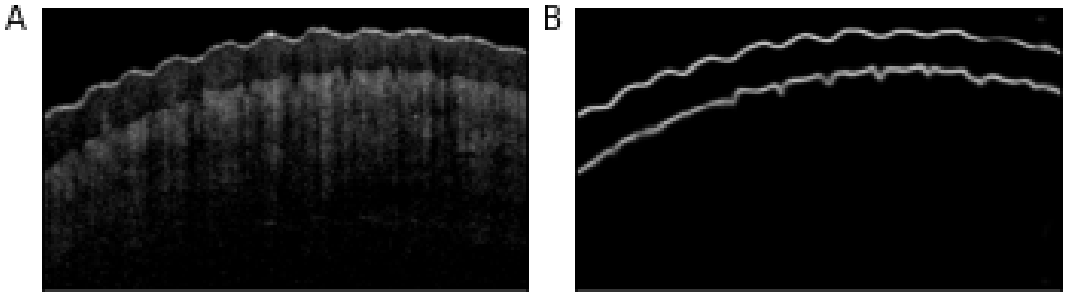

Figure 6.3: Sample OCT B-scan and prediction mask. A - Input image, B - Prediction mask

directly after the training set was constructed, which results in scans similar in structure to those used in training due to the close proximity of each slice in the volume. Please refer to figure 6.3 for a sample prediction mask extracted from the unseen sample set. Visual analysis suggests that this is a near perfect prediction, however, 600 samples from a second hold-out that contained no ground truth labels were visually observed. Many inconsistencies in the structure along with "false positive" patches at the bottom of the images were observed.

In comparison, the nsOCT data has been more difficult to measure and the final solution does not contain a very high level of precision, although it can generalise and learn the structure of the data. As previously mentioned the spatial co-ordinates of the pixels on the y-axis do not match in the nsOCT images when compared to the corresponding source B-scan. The outputted prediction masks must be visually compared to both the source OCT image, ground truth and the original input so that the dimensions can be aligned and analysed, this is demonstrated in Figure 6.4.

The experimental settings applied to the OCT data were also applied here. Training to upwards of 50 epochs and varying the hyperparameters revealed the most efficient model found for the nsOCT data was 10 epochs with $lr = 0.01$, $\beta 1 = 0.9$ and $\beta 2 = 0.999$. The model is capable of identifying the layers, and with some visual interpretations the spatial co-ordinates, the y-axis match with the source B-scans to some degree. The small volume of ground truth labels presents the challenge of quantifying the accuracy. The ground truth set is also

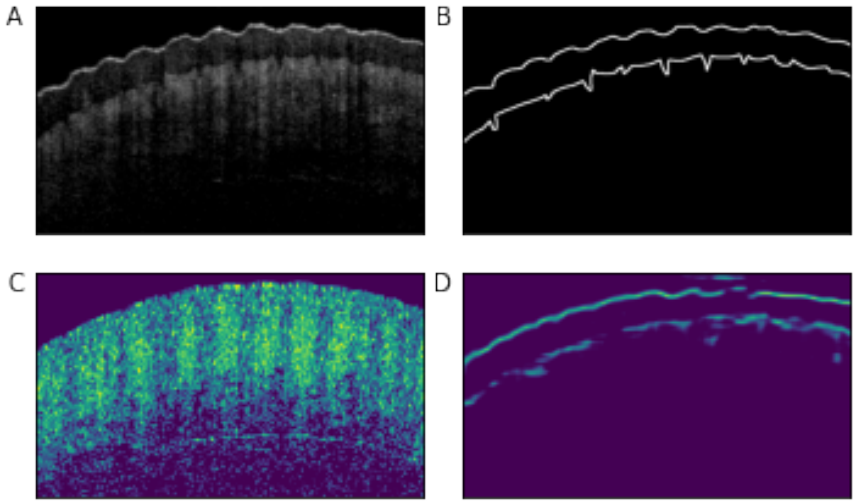

Figure 6.4: A - Source B-scan, B - Ground truth, C - Input nsOCT image, D - Prediction mask

human annotated so it is unknown if the accuracy threshold is limited or if it can be pushed higher with further tests. The output prediction masks do not fully capture the full structure of the boundaries.

It is noticeable that the validation metrics are not stable. The final solution achieved an accuracy rate of 98% and a final loss of 0.04. The validation metrics resulted in an accuracy of 97.5% and a loss of 0.07, this can be seen in Figure 6.5 and 6.6. This does not appear to be a significant difference, but if we recall the background class makes up 97% of the data. This means the validation data is performing as well as a just predicting each pixel is part of the background. Stability is observed with the hold-out set. The same hold-out data used in the previous OCT tests were applied to this model, resulting in an accuracy of 97% and a loss of 0.092, not much worse than the validation data. The same hold-out set used in the previous OCT tests were applied to this model, resulting in an accuracy of 97% and a loss of 0.092.

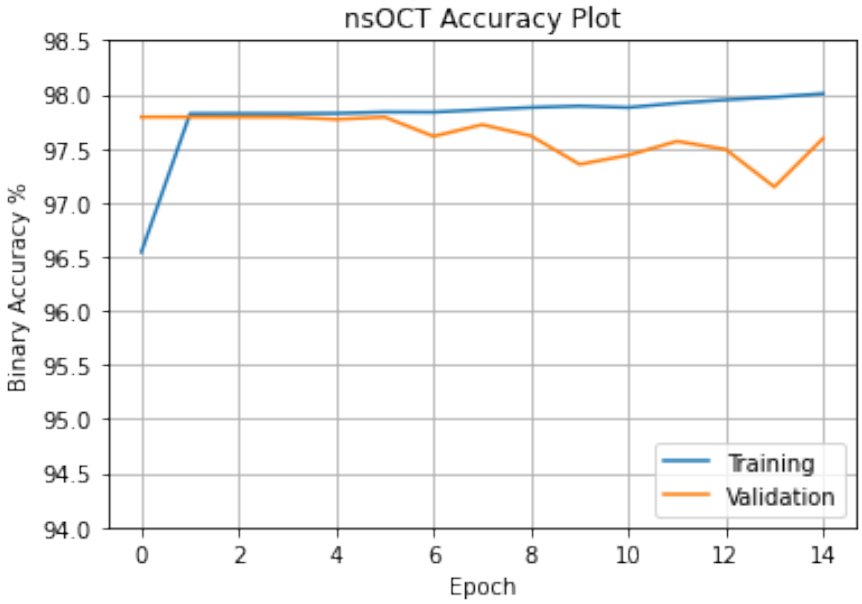

Figure 6.5: nsOCT Binary Accuracy per Epoch

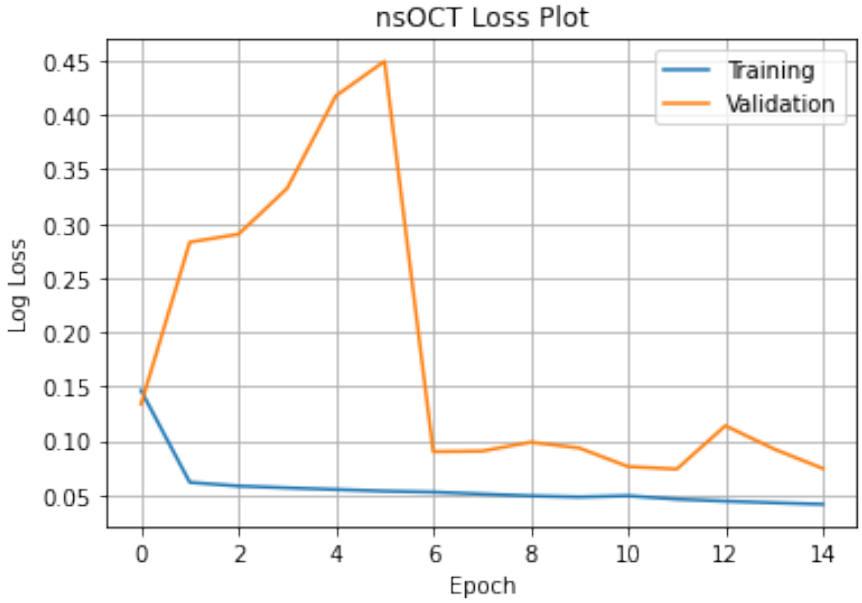

Figure 6.6: nsOCT Log Loss per Epoch

Chapter 7

Conclusion

It has demonstrated that the proposed design, derived from the current state-of-the-art, is capable of performing segmentation of the skin layers in two-dimensional scans of the human finger in the OCT and nsOCT images. It is difficult to quantify the performances with little reference work and a small manually annotated set of ground truth labels. Many weaknesses have been identified throughout the experimentation. The data is very sensitive to hyperparameter tuning and number of training iterations it is exposed too. More samples are required to produce a better generalisation of the tissue structures, this is because tissue will vary person to person. Furthermore, it is difficult to reproduce similar results due to the random nature of the kernel generation in the model. After the sample set is increased a number of experiments should be conducted and an average performance should be measured. Other approaches should be considered on how to handle the class imbalance also.

These limitations should be considered for future work and the main focus should be on the curation of a larger gold standard set with per pixel precision. Other methods explored in literature could be applied for comparison. The patch based extraction methods have shown high performance in retinal and mouse skin scans for segmentation. Extensive research into the dimensions of the patches and the effectiveness of each is also a popular region of study currently. The CIFAR-CNN with graph searching algorithms have also been demonstrated as efficient techniques. For this study I have also hypothesized two new methods not yet explored.

The first method consists of designing a two-tier segmentation network. The ground truth labels for the nsOCT images were gathered and curated from the source OCT B-scans. The curation of a high precision per pixel annotated ground truth data set could be utilized to produce a solution capable of segmenting the layers in the OCT images with higher accuracy and reduced noise. If this can be achieved the solution can utilized to generate a higher quantity of ground truth labels that can be applied to the nsOCT data, which requires a vast amount more of data.

The second method is an unsupervised learning approach, that would focus on clustering the spectral data of each pixel. It was observed that the stratum corneum was identified through peaks in the interferograms in some recent studies. When converting the OCT images to their nano-sensitive counterparts, the spectral data per pixel can be attached. This data ranges between 0 and the number of sub-bands created over the A-line, each value from range $0 - 153$, in this case, corresponding to a certain wavelength. A peak in this data is used to decide which wavelength the pixel value is mapped too. If this data is attached to each pixel, hypothetically, a clustering method should be able to identify similar arrays of this spectral data. Indexing the feature vectors based on pixel co-ordinates would be used for pixel identification in the source image, and a cluster that will contain pixels positioned in the layer could be identified.

References

[1] J. Fujimoto and W. Drexler, "Introduction to optical coherence tomography," in *Optical coherence tomography*. Springer, 2008, pp. 1–45. vii, 7, 8

[2] D. P. Popescu, C. Flueraru, Y. Mao, S. Chang, J. Disano, S. Sherif, M. G. Sowa *et al.*, "Optical coherence tomography: fundamental principles, instrumental designs and biomedical applications," *Biophysical reviews*, vol. 3, no. 3, p. 155, 2011. vii, 6, 8

[3] S. Alexandrov, H. Subhash, A. Zam, and M. Leahy, "Nano-sensitive optical coherence tomography (nsoct) for depth resolved characterization of 3d submicron structure," in *Optical Coherence Tomography and Coherence Domain Optical Methods in Biomedicine XVIII*, vol. 8934. International Society for Optics and Photonics, 2014, p. 89340Z. vii, 3, 10

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. vii, 10, 11

[5] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6. vii, 13

[6] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440. vii, 14, 15

[7] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241. vii, 18, 30, 31

[8] K. Doi, "Diagnostic imaging over the last 50 years: research and development in medical imaging science and technology," *Physics in Medicine & Biology*, vol. 51, no. 13, p. R5, 2006. 1

[9] M. R. Hee, C. R. Baumal, C. A. Puliafito, J. S. Duker, E. Reichel, J. R. Wilkins, J. G. Coker, J. S. Schuman, E. A. Swanson, and J. G. Fujimoto, "Optical coherence tomography of age-related macular degeneration and choroidal neovascularization," *Ophthalmology*, vol. 103, no. 8, pp. 1260–1270, 1996. 1, 6

[10] S. Farsiu, S. J. Chiu, R. V. O'Connell, F. A. Folgar, E. Yuan, J. A. Izatt, C. A. Toth, A.-R. E. D. S. . A. S. D. O. C. T. S. Group *et al.*, "Quantitative classification of eyes with and without intermediate age-related macular degeneration using optical coherence tomography," *Ophthalmology*, vol. 121, no. 1, pp. 162–172, 2014. 1

[11] S. A. Alexandrov, S. Uttam, R. K. Bista, K. Staton, and Y. Liu, "Spectral encoding of spatial frequency approach for characterization of nanoscale structures," *Applied physics letters*, vol. 101, no. 3, p. 033702, 2012. 2, 3, 9

[12] S. Alexandrov, P. M. McNamara, N. Das, G. L. Yi Zhou, J. Hogan, and M. Leahy, "Spatial frequency domain correlation mapping optical coherence tomography for nanoscale structural characterization," *Applied Physics Letters*, vol. 115, no. 12, p. 121105, 2019. 2, 3

[13] H. Greenspan, B. van Ginneken, and R. M. Summers, "Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1153–1159, 2016. 2, 17

[14] R. Zemouri, N. Zerhouni, and D. Racoceanu, "Deep learning in the biomedical applications: Recent and future status," *Applied Sciences*, vol. 9, no. 8, p. 1526, 2019. 2, 12, 17

[15] S. Alexandrov, H. Subhash, and M. Leahy, "Adaptation of the spectral encoding of spatial frequency approach to optical coherence tomography (oct)," in *Biomedical Optics*. Optical Society of America, 2014, pp. BS2B–4. 3

[16] S. A. Alexandrov, H. M. Subhash, A. Zam, and M. Leahy, "Nano-sensitive optical coherence tomography," *Nanoscale*, vol. 6, no. 7, pp. 3545–3549, 2014. 3, 9

[17] R. Dsouza, J. Won, G. L. Monroy, M. C. Hill, R. G. Porter, M. A. Novak, and S. A. Boppart, "In vivo detection of nanometer-scale structural changes of the human tympanic membrane in otitis media," *Scientific reports*, vol. 8, no. 1, pp. 1–13, 2018. 3

[18] S. Alexandrov, N. Das, J. McGrath, P. Owens, C. J. Sheppard, F. Boccafoschi, C. Giannini, T. Sibillano, H. Subhash, and M. Leahy, "Label free ultrasensitive imaging with sub-diffraction spatial resolution," in *2019 21st International Conference on Transparent Optical Networks (ICTON)*. IEEE, 2019, pp. 1–4. 3

[19] C. Lal, S. Alexandrov, S. Rani, T. Ritter, and M. Leahy, "Probing temporal structural changes within cornea using 200 khz swept source nano sensitive optical coherence tomography (nsoct)," in *Dynamics and Fluctuations in Biomedical Photonics XVII*, vol. 11239. International Society for Optics and Photonics, 2020, p. 1123909. 3

[20] N. Das, A. Sergey, Y. Zhou, K. E. Gilligan, R. M. Dwyer, and M. Leahy, "Nanoscale structure detection and monitoring of tumour growth with optical coherence tomography," *Nanoscale Advances*, vol. 2, no. 7, pp. 2853–2858, 2020. 3

[21] Y. Zhou, S. Alexandrov, A. Nolan, N. Das, R. Dey, and M. Leahy, "Noninvasive detection of nanoscale structural changes in cornea associated with cross-

linking treatment," *Journal of biophotonics*, vol. 13, no. 6, p. e201960234, 2020. 3

[22] C. Lal, S. Alexandrov, S. Rani, Y. Zhou, T. Ritter, and M. Leahy, "Nanosensitive optical coherence tomography to assess wound healing within the cornea," *Biomedical Optics Express*, vol. 11, no. 7, pp. 3407–3422, 2020. 3

[23] N. Das, S. Alexandrov, K. E. Gilligan, R. M. Dwyer, R. B. Saager, N. Ghosh, and M. Leahy, "Characterization of nanosensitive multifractality in submicron scale tissue morphology and its alteration in tumor progression," *Journal of Biomedical Optics*, vol. 26, no. 1, p. 016003, 2021. 3

[24] T. Kepp, C. Droigk, M. Casper, M. Evers, G. Hüttmann, N. Salma, D. Manstein, M. P. Heinrich, and H. Handels, "Segmentation of mouse skin layers in optical coherence tomography image data using deep convolutional neural networks," *Biomedical Optics Express*, vol. 10, no. 7, pp. 3484–3496, 2019. 6, 20

[25] J. M. Schmitt, A. Knuttel, M. Yadlowsky, and M. Eckhaus, "Optical-coherence tomography of a dense tissue: statistics of attenuation and backscattering," *Physics in Medicine & Biology*, vol. 39, no. 10, p. 1705, 1994. 8

[26] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and cooperation in neural nets*. Springer, 1982, pp. 267–285. 10

[27] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989. 10

[28] A. G. Howard, "Some improvements on deep convolutional neural network based image classification," *arXiv preprint arXiv:1312.5402*, 2013. 12

[29] J. Hamwood, D. Alonso-Caneiro, S. A. Read, S. J. Vincent, and M. J. Collins, "Effect of patch size and network architecture on a convolutional neural net-

work approach for automatic segmentation of oct retinal layers," *Biomedical optics express*, vol. 9, no. 7, pp. 3049–3066, 2018. 12, 21, 22, 23

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012. 14

[31] S. M. Anwar, M. Majid, A. Qayyum, M. Awais, M. Alnowami, and M. K. Khan, "Medical image analysis using convolutional neural networks: a review," *Journal of medical systems*, vol. 42, no. 11, p. 226, 2018. 17

[32] J. Sekulska-Nalewajko, J. Goclawski, and D. Sankowski, "The detection of internal fingerprint image using optical coherence tomography," *Image Processing Communications*, vol. 22, pp. 59–72, 12 2017. 17

[33] L. N. Darlow, J. Connan, and S. S. Akhoury, "Internal fingerprint zone detection in optical coherence tomography fingertip scans," *Journal of Electronic Imaging*, vol. 24, no. 2, p. 023027, 2015. 17

[34] H. Wang, X. Yang, P. Chen, B. Ding, R. Liang, and Y. Liu, "Acquisition and extraction of surface and internal fingerprints from optical coherence tomography through 3d fully convolutional network," *Optik*, vol. 205, p. 164176, 2020. 19

[35] B. Ding, H. Wang, P. Chen, Y. Zhang, Z. Guo, J. Feng, and R. Liang, "Surface and internal fingerprint reconstruction from optical coherence tomography through convolutional neural network," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 685–700, 2020. 19

[36] J. De Fauw, J. R. Ledsam, B. Romera-Paredes, S. Nikolov, N. Tomasev, S. Blackwell, H. Askham, X. Glorot, B. O'Donoghue, D. Visentin *et al.*, "Clinically applicable deep learning for diagnosis and referral in retinal disease," *Nature medicine*, vol. 24, no. 9, pp. 1342–1350, 2018. 20

[37] F. G. Venhuizen, B. van Ginneken, B. Liefers, M. J. van Grinsven, S. Fauser, C. Hoyng, T. Theelen, and C. I. Sánchez, "Robust total retina thickness segmentation in optical coherence tomography images using convolutional

neural networks," *Biomedical optics express*, vol. 8, no. 7, pp. 3292–3316, 2017. 20

[38] L. Fang, D. Cunefare, C. Wang, R. H. Guymer, S. Li, and S. Farsiu, "Automatic segmentation of nine retinal layer boundaries in oct images of non-exudative amd patients using deep learning and graph search," *Biomedical optics express*, vol. 8, no. 5, pp. 2732–2744, 2017. 21, 22, 23

[39] K. Hu, B. Shen, Y. Zhang, C. Cao, F. Xiao, and X. Gao, "Automatic segmentation of retinal layer boundaries in oct images using multiscale convolutional neural network and graph search," *Neurocomputing*, vol. 365, pp. 302–313, 2019. 22

[40] J. Kugelman, D. Alonso-Caneiro, S. A. Read, J. Hamwood, S. J. Vincent, F. K. Chen, and M. J. Collins, "Automatic choroidal segmentation in oct images using supervised deep learning methods," *Scientific reports*, vol. 9, no. 1, pp. 1–13, 2019. 23

[41] J. Kugelman, D. Alonso-Caneiro, S. A. Read, S. J. Vincent, and M. J. Collins, "Automatic segmentation of oct retinal boundaries using recurrent neural networks and graph search," *Biomedical optics express*, vol. 9, no. 11, pp. 5759–5777, 2018. 23

[42] I. C. of High-End Computing. Python / conda. [Online]. Available: https://www.ichec.ie/academic/national-hpc-service/software/python-conda 24

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1026–1034. 31

[44] D. Chen, Y. Ao, and S. Liu, "Semi-supervised learning method of u-net deep learning network for blood vessel segmentation in retinal images," *Symmetry*, vol. 12, no. 7, p. 1067, 2020. 32

[45] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456. 32

[46] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012. 32

[47] S. Park and N. Kwak, "Analysis on the dropout effect in convolutional neural networks," in *Asian conference on computer vision*. Springer, 2016, pp. 189–204. 33

[48] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 33