Automatic Segmentation of Retinal Layers in Optical Coherence Tomography using Deep Learning Techniques

Master's Thesis Computing Science - Data Science

Chris Kamphuis

August 23, 2018

Supervisor RU:
Prof. dr. Elena Marchiori

Supervisor RUMC:
Dr. Clarisa Sánchez

Supervisor RUMC:
Bart Liefers MSc

Second reader RUMC:
Prof. dr. Bram van Ginniken

Radboud University

Contents

Chapter 1

Introduction

1.1 Introduction

1.1.1 The retina

The retina is a multilayered structure that covers a large surface inside the eye. The function of the retina is to convert light to a neural response for further use by the brain. The retina contains two different types of photoreceptors: rods and cones. A healthy eye consists of around 60 million rods and around 3 million cones [1]. Rods are located at the peripheral part of the retina, they are responsible for peripheral vision, motion detection and the perception of light/dark contrast. Cones are mostly located in the macula lutea region of the retina, with most of the cones living at the central part of the macula lutea, the fovea centralis. Cones allow for color and central vision. The retina however consists of many layers, and photo receptors only constitute a small part of these. Figure 1.1 shows an illustration of the layers of the retina and their respective cellular composition.

1.1.2 Retinal imaging

In order to investigate the retina, one could make use of several retinal imaging techniques. Examples of such techniques are Color Fundus Photography (CF), Fundus Autofluorescence (FAF), Near-Infrared Reflectance (NIR) and Optical Coherence Tomography (OCT). CF, FAF and NIR are en-face imaging techniques while OCT is a cross-sectional imaging technique. Figure 1.2 shows an example of images created using these techniques. The dashed line indicates the location of where the OCT scan is taken. Depending on what one wants to investigate, one of these techniques might be more useful than the others. OCT allows for accurately measuring the thickness of the retina, something that is not possible using the other imaging techniques mentioned. In this thesis I will focus on the OCT imaging technique.

OCT is an imaging technique to create cross-sectional views of the retina non-invasively. OCT scans are often acquired as multiple linear slices. Stacking these slices, it is possible to create a 3D view of the retina and its different layers. In OCT volumes 18 different retinal layers can be identified, Figure 1.3 shows these layers as proposed by Staurenghi et al.[4]. An OCT scan is acquired by directing a beam of near-infrared light to both a mirror and the retina. Different retinal layers have different optical properties, some reflecting more light back than others. The

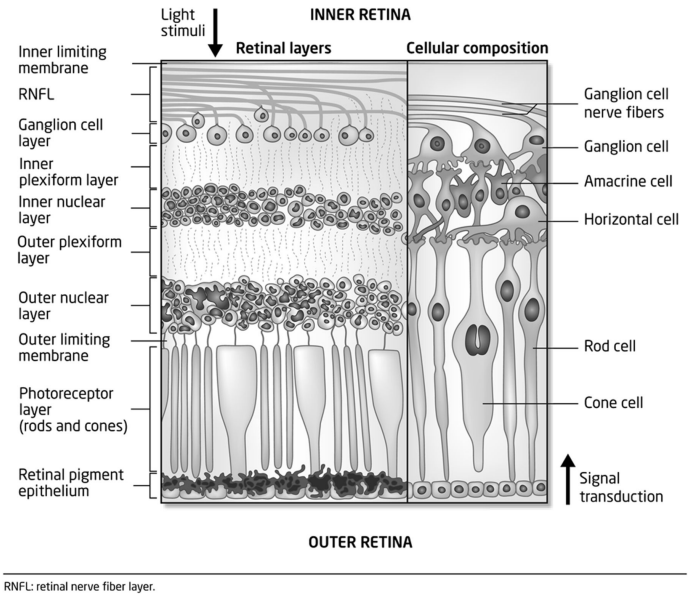


Figure 1.1: Illustration of layers of the retina by Ratchford et al. [2]

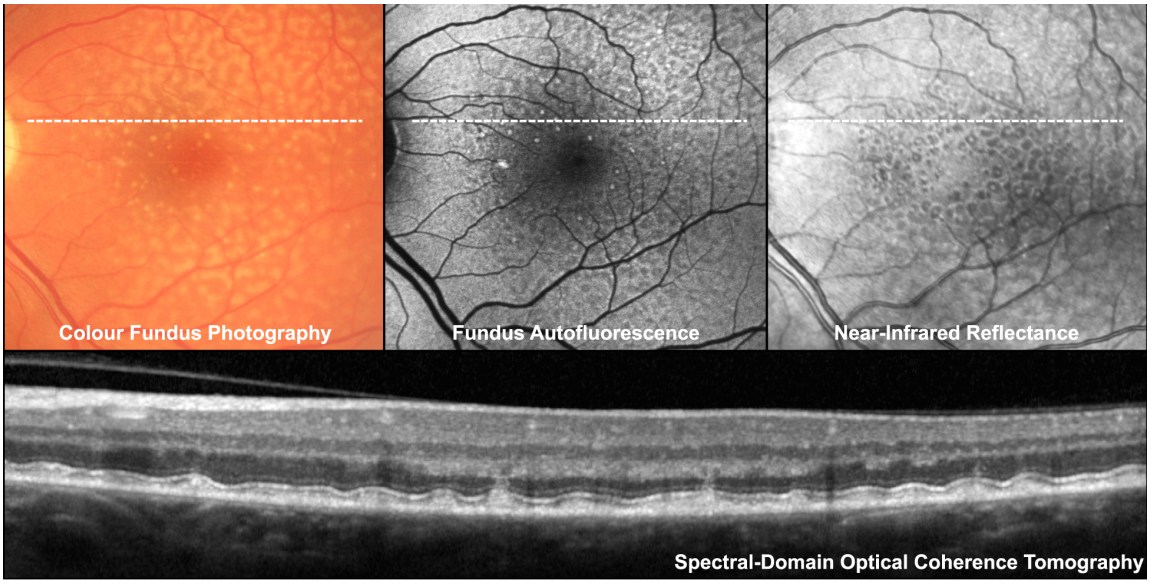


Figure 1.2: Examples of retinal imaging techniques by Wu et al. [3]

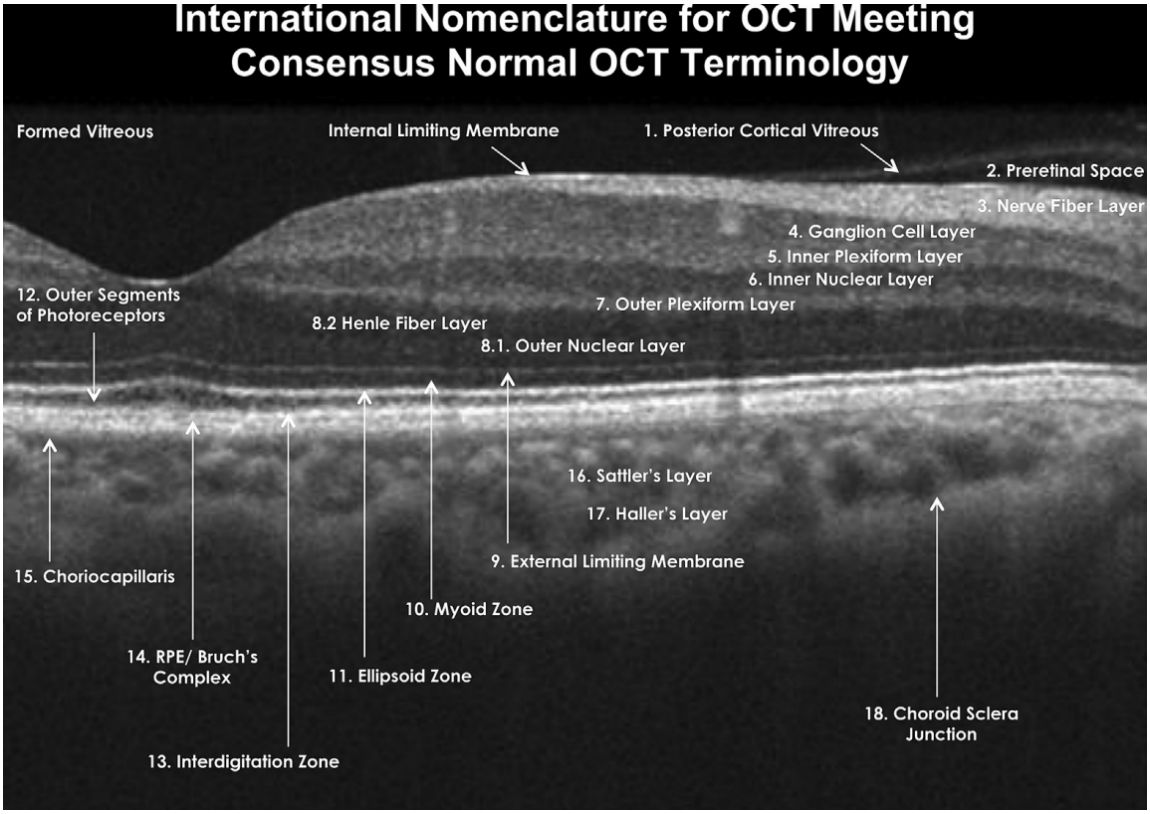


Figure 1.3: Visible retinal layers in OCT by Staurenghi et al. [4]

interference of light reflected back by the layers of the retina, and the light reflected back as reference by the mirror can be measured [5]. This wave interference can be analyzed using a spectrometer. By applying a Fourier transform on this signal the amount of light reflected back by different layers of the retina can be quantified. The quantification of one measurement is called an a-scan. An image can be created by measuring many a-scans next to each other. Such an image is called a b-scan. Figure 1.3 shows an example of a b-scan. By repeating this procedure one can stack such b-scans and create a 3D view of the retina and its layers.

1.1.3 Retinal diseases

There are different retinal diseases that affect the thickness of layers of the retina. Accurately measuring these changes in the retina can help with diagnosis, prognosis and severity determination of multiple retinal complications. Examples of such complications are glaucoma [6][7], diabetic retinopathy [8] and age-related macular degeneration (AMD) [9].

Glaucoma is a disease that affects more than six million people world wide. Symptoms of glaucoma are vision loss and eye pain. Glaucoma results in damage in the optic nerve. As it is possible the measure the thickness of the nerve fiber layer in OCT imaging, it is possible to detect

the presence and severity of glaucoma in OCT scans [7].

Diabetic retinopathy is a condition that damages the retina. This condition affects many people that suffer from diabetes. Properly monitoring and treating the eyes can reduce symptoms at 90 percent of new cases [10]. Diabetic retinopathy may cause macula edema, resulting in vessels leaking blood in the retina. OCT can show which areas are thickened when fluid accumulates. Other effects of diabetic retinopathy that can be seen in OCT images are retinal swellings and damaged nerve tissue.

Finally, AMD is a medical condition that typically happens to older people. A symptom of this disease in an advanced stage is blurred or no vision in the center of visual field. This disease specifically damages the macula lutea. When suffering from AMD there is an accumulation of a yellow fluid, these accumulations are called drusen. AMD can also cause other fluids to appear in the retina. Moreover, AMD alterates the retinal pigment epithelium (RPE) layer of the retina. The presence of drusen, other fluids and the alteration of the RPE can been seen in OCT images.

1.1.4 Layer segmentation

Segmentation of the individual layers of the retina provides a quantitative tool for ophthalmologists to analyze the retina more thoroughly. Segmentation of retinal layers helps with measuring the thickness of individual retinal layers, making it possible to detect the presence of the diseases described. Recently a new technique, OCT-angiography, has been introduced [11]. This technique can be used to create en-face images of vasculature in different vascular plexi. OCT-angiography requires the retinal boundaries to be delineated precisely, for which an accurate segmentation is necessary [12].

Segmenting the scans manually is however a tedious and subjective task, and as a result automatic methods for segmentation of these images have been a topic of interest in research. Automatic segmentation allows for more quantitative measurements, and therefore more objective health care and better reporting/documentation. As automatic segmentation allows for the investigation of a lot of data, it is interesting to link segmentation results to prognosis and treatment response.

1.2 Related work

1.2.1 Rule-based segmentation

Many layer segmentation techniques have been proposed in the literature. Often such techniques apply a set of given rules to extract the boundaries of the layers that need to be segmented: Ishikawa et al. developed an approach by analyzing the intensity variation and detecting the boundaries using an adaptive thresholding techniques in order to identify retinal structures [13]. The authors identified four different layer structures. Using this method the authors were able to discriminate between glaucomatous and normal eyes. Koozekanani et al. introduced an intensity-based Markov boundary model that was able to detect the retinal boundaries in OCT-images [14]. This method was able to produce a thickness measurement that differs less than 10 μm compared to the manually segmented images in 74% of the images. Kajic et al. created a statistical model for layer segmentation based on texture and shape analysis [15]. Using the model they were able to segment eight different retinal layers on OCT images of healthy retinas. This model is quite robust in the sense that it performed well on images with high speckle noise. Chiu et al. used the graph-cut technique to segment eight retinal boundaries [16]. Using the fact that there is a

high intensity gradient between layers it is possible to find the graphs weights. This method was able segment the retinal boundaries more closely to an expert grader compared to a second expert grader on healthy eyes. Garvin et al. approached this problem by using a multi-surface graph cut approach [17]. As stacks of multiple b-scan can be visualized as a volume, it is possible to used adjacent b-scans for extra information. Using this method the researchers were able to segment six different layers on healthy eyes with an error that was comparable to inter-observer variability. Mishra et al. proposed a two-step kernel based optimization scheme that first tries to approximate the locations of the layers, which is then refined to obtain an accurate segmentation of the layers [18]. This method achieved accurate segmentation results on images of healthy eyes. It was able to perform well on images having a low image contrast with the presence of irregular shapes in the images.

The problem with techniques based on rules/assumptions however is that they often do not generalize well, especially in the cases of severely affected retinas. Changes in images might also mean that the set of rules used might need to change in order to get a good performance.

1.2.2 Learning-based segmentation

In order to tackle the problems ruled based methods face, researchers have started using machine learning models. Instead of creating the decision rules by hand, machine learning models learn their parameters given data and develop their own decision rules. Machine learning models learn by being presented examples. Different kinds of machine learning models have been used in the literature to approach the OCT layer segmentation problem. Vermeer et al. used a support vector machine to segment six different layers [19]. They extracted multiple statistical features from individual a-scans to train the support vector machine. Lang et al. trained a random forest model and used it with either an edge detector or graph-search to identify nine different retinal layers [20]. When machine learning approaches are being used for images segmentation, researchers often use deep neural networks as segmentation models. Venhuizen et al. trained a convolutional neural network for the total retina segmentation problem [21]. They showed that their proposed algorithm was capable of modeling the variability in retinal appearances. The algorithm was able to reliably segment the retina even in severe pathological cases. Roy et al. proposed a CNN that used for retinal layer and fluid segmentation [22]. The output of this network was used as the input for a rule based segmentation method. Recently, researches from Google's Deep Mind published their research on OCT images, using neural networks they were able to detect many retinal diseases in OCT images [23].

1.3 Background

1.3.1 Neural networks

Neural networks are learning systems that have won many competitions in pattern recognition and machine learning [12]. Figure 1.4 shows a schematic diagram of a simple feed forward neural network. Neural networks are decision machines consisting of multiple layers of artificial neurons. Figure 1.4 shows a neural network with two hidden layers, but often neural networks have many. In this thesis I will specifically explain the workings of feed-forward neural networks, other kinds (like recurrent neural networks) exist and work slightly different from how it is explained in the next paragraph.

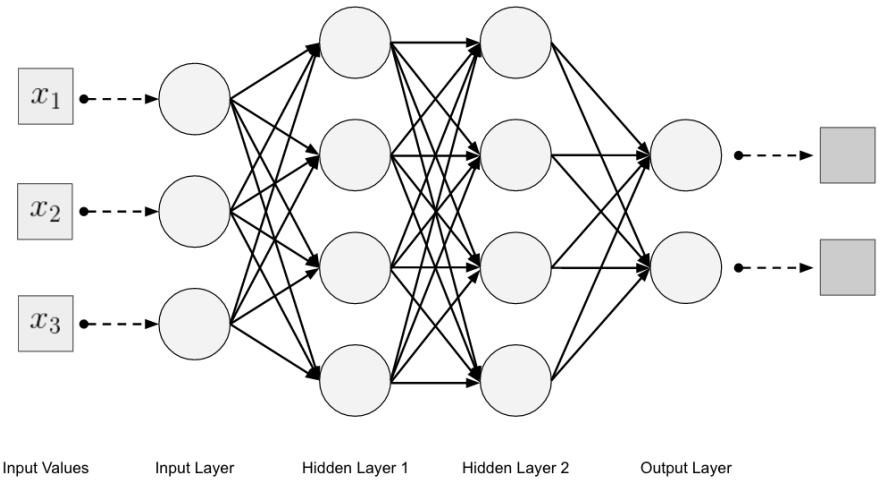


Figure 1.4: Example of feed forward neural network illustrated by Patterson et al. [24]

The input layer of a neural network takes sensor data as input. Other layers in the network receive their input through weighted connections, that are connected to preceding layers of the network. These weighted connections are the parameters of the model that need to be optimized. There are different types of layers that each have their own characteristics on which I will elaborate later. An activation function will often be applied on the output of the neuron. This activation function is often a non-linear function. The main purpose of activation functions is to introduce non-linearities in the network, this allows for solutions that are not a linear combination of the input. The output produced by the last layer of neurons is the prediction of the network. One may represent a neural network as a function $f(X, \mathbf{w}) = Y$ where the parameters $\mathbf{w}$ should be optimized in such a way that it finds an $\hat{Y}$ given input X that is as close as possible to the true value Y. Layers that are not the input or output layer of the network are called hidden layers, when a neural networks has many hidden layers it is called a deep neural network.

Activation functions

Activation functions are often applied to the output of a neuron. Depending on the location in the network researchers often choose different kinds of activation functions. A small overview of the most common activation functions is given:

Sigmoid is an activation function that scales its input to a value between 0 and 1:

$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1} \tag{1.1}$$

Classically, the sigmoid function or functions with similar properties were used as activation functions throughout the whole neural network. The derivative of a sigmoid function goes to zero for large values of x. This can lead to small gradient values throughout the network. As neural networks now typically have multiple layers, applying the sigmoid function often leads to a vanishing

gradient. This makes it hard for a neural network to learn. Nowadays, this activation function is often only used at the end of a neural network. By scaling the value at the end of the network, the outcome can easily be used as a decision score.

Rectifier Linear Unit [25] (ReLU) is the activation function that is used most often. The ReLU function is more biologically plausible than the sigmoid function [26]. This type of activation takes the maximum value of 0 and its input:

$$f(x) = \max(0, x) \tag{1.2}$$

The main advantage of this activation function is that allows for better gradient propagation compared to activation functions with a vanishing gradient problem. Big changes in the input of the function do not necessarily map to a small change in the output. A disadvantage of this type of activation function is that it allows for parts of the network to be deactivated (the output becomes 0 independent of all possible inputs). However, as neural networks often contain many layers, this activation function is very useful in order to properly train the early layers of the network as it allows for proper gradient propagation.

Softmax is a group version of the sigmoid function:

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \tag{1.3}$$

Instead of taking one value, the softmax function takes a vector of values. All these values are scaled to a value between 0 and 1, together the values sum to 1. This activation function is generally only used at the final layer of a neural network. The softmax functions allows for a decision value for every class in the decision problem. Often these scores are interpreted as the probability of the output belonging to the class the value represents. Typically, the class with the highest value is taken as decision.

1.3.2 Training neural networks

The weights $\mathbf{w}$ of the connections in the network need to be optimized in order for the network to achieve good performance. The process of optimizing these weights $\mathbf{w}$ is called network training. This is done by letting the network predict an output $\hat{Y}$ given an input X, this predicted output $\hat{Y}$ and the real output Y are then used as input for a loss function. This loss function will determine the loss value. The loss function returns a value that gets smaller when the difference between the real and predicted output gets smaller. This loss function should be a differentiable function in order to calculate the partial derivative of f with respect to each weight in $\mathbf{w}$. Given these derivatives, the weights $\mathbf{w}$ of the network are updated using gradient descent [27] or a similar optimizing technique. The technique used to determine the partial derivatives with respective to the weights $\mathbf{w}$ throughout the network is called back-propagation [28]. Using this technique, the weights $\mathbf{w}$ throughout the whole network can be updated. Two commonly used loss functions are Mean Squared Error (MSE) and Cross Entropy (CE). MSE is calculated using the predicted values and the true values, both represented as a vector of length n. CE is calculated using the values of the pixels o over M classes. MSE is often used for regression problems and CE is often used for (multi-)class decisions problems.

$$MSE = \frac{1}{n}\sum_{i=0}^{n}(Y_i - \hat{Y}_i)^2 \tag{1.4}$$

$$CE = -\sum_{c=0}^{M} \hat{Y}_{o,c} \log(Y_{o,c}) \tag{1.5}$$

1.3.3 Convolutional neural networks

Specifically Convolutional Neural Networks (CNNs) [29] have successfully been used in (biomedical) image segmentation. CNNs are a type of neural network where the operations carried out by many of the layers are discrete 2D-convolutions. Figure 1.5 illustrates the convolution operation. The values used in the kernels of the convolution operations are the weights. The patterns in the kernels can be used to recognize similar patterns in the input data, this creates new features where the value of the pixels in these features is dependent on the value of the pixel on the same spot as the input and its surrounding pixels. By chaining multiple convolutions, more complex features can be recognized, and more context can be taken into account. The kernel is applied on the complete input, this allows for fewer weights as the parameters are shared. Not having a lot of parameters make convolutions well suited for extracting information at a low computational cost, making the operation useful when deep learning architectures are used. Another effect of the kernel being applied on the complete input is that the convolution operation is translation invariant; the convolution operation is not affected by the location of a local pattern.

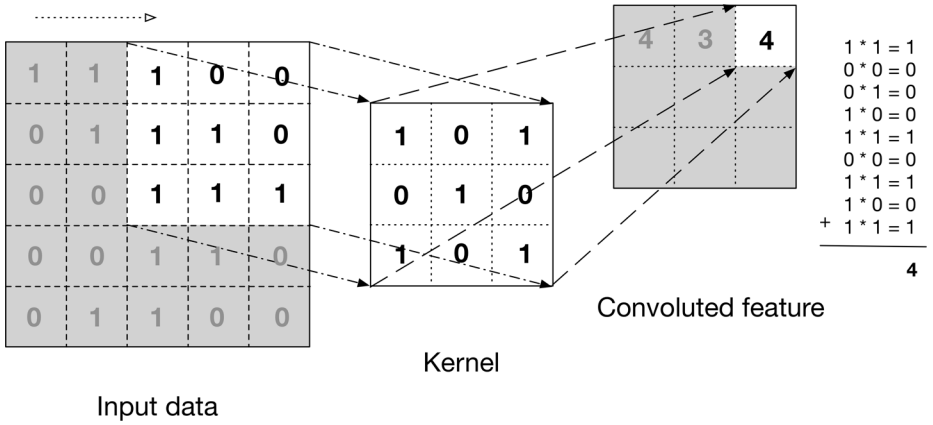


Figure 1.5: The convolution operation illustrated by Patterson et al. [24]

Next to convolutions, a CNN typically contains multiple types of layers. A small overview of other important layers other than the convolutional layers and their characteristics is given:

Max-pooling layers move a window over the input, the maximum value in that window will be the output. Typically, the windows move differently from the convolution operation in the sense that the windows do not overlap. The two main purposes of max-pooling is to reduce the pixel resolution of the features in the network and generalize the results of the previous layers (typically convolutional layers). This effect is also called down-sampling.

Fully connected layers contain the most simplest form of neurons. A neuron in a fully connected layer takes as input a weighted value of all neurons in the previous layer, these values are then

all summed. Often simple neural networks only contain these kinds of neurons. In convolutional neural networks they are often used as last layers and their output is then the predicted value.

Dropout layers [30] behave quite differently compared to other layers. Dropout layers are only being used during the training process of the network. During training they take the input and set a certain amount of these values to 0 before passing them to the next layer. This might sound counterintuitive as you make it more difficult for the network to learn. The advantage is that the network can not rely on specific neurons for certain behaviour, this lowers the chance of overfitting.

Up-convolution layers are the counter part of max-pooling layers; these layers apply a convolution on an up sampled input. This kind of operation up-samples the data. These layers are often used to create a specific desired dimensionality of the output data.

1.3.4 Network architecture

To solve a particular problem with a neural network it is often required to choose the right network architecture. The amount of layers, the order in which they appear, the activation functions and the settings per layer make up the architecture of a network. Different network architectures are able to perform differently on particular tasks. In order to test which architecture is best suited for a specific problem, multiple architectures should be compared to each other. However, depending on how an architecture is trained, the performance might also differ within a network. One could for example train a network using varying learning rates. Literature of course gives a good indication which kinds of network architectures might be suited for specific problems.

One specific architecture often used for biomedical image segmentation is U-net [31]. Figure 1.6 shows an illustration of the implementation of U-net as implemented by Ronneberger et al.[31]. Variations of this architecture have also been successfully deployed for OCT segmentation [21]. The architecture of U-net consists of two paths, a contracting path to capture the context and a symmetric expanding path that allows for precise localization. In this context, paths refer to a way how information flows from the input of the network to the output. For the image segmentation problem, network architectures are often designed like the U-net architecture. They consist of a down sampling (contracting) path followed by an up sampling (expanding) path, these paths are often connected at multiple places through so called skip connections. These skip connections concatenate features of earlier stages of the network to features calculated later in the network.

Another architecture that uses these properties is called ReLayNet [22]. This network has been successfully deployed for the OCT layer segmentation problem. Another architecture that has been successfully deployed for segmentation problems is called fully convolutional densenet [32]. It also uses a down and up sampling path and skip connections. The regular layers are however replaced by densely connected blocks of layers. These blocks allow for easier training of the network and have shown excellent results in classification tasks [33]. Figure 1.7 shows an illustration of a densely connected block. Generalizing this to a general segmentation architecture was fruitful as this kind of network is able to achieve state-of-the-art performance on regular segmentation tasks [32].

1.4 Problem statement

The goal of this thesis is to investigate which deep neural networks can be employed for the OCT layer segmentation problem. In order to find a fitting network for this problem, different network

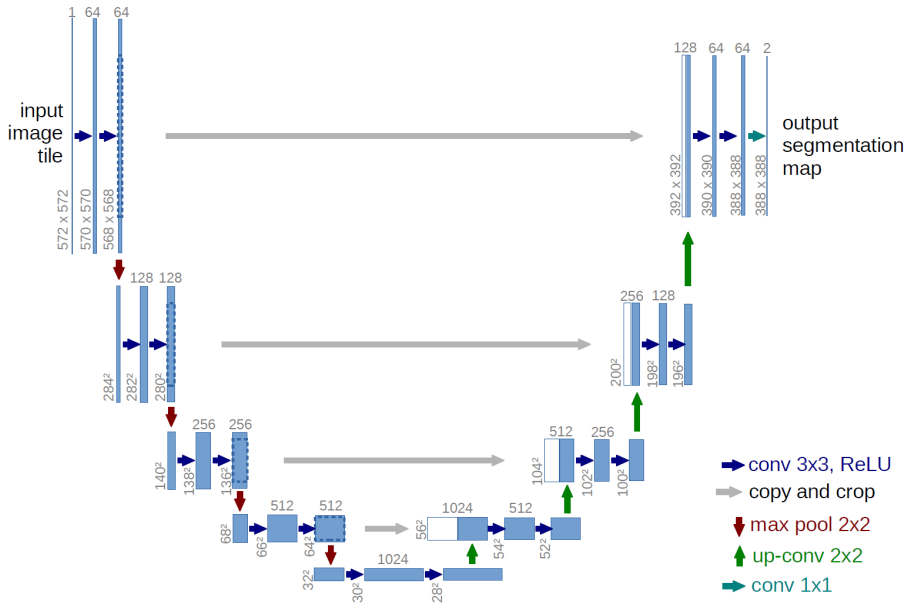


Figure 1.6: U-net architecture by Ronnerberger et al. [31]. The left side of the network is the contracting path, the right side is the expanding path.

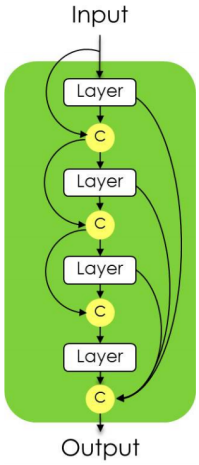


Figure 1.7: Illustration of a denseblock by Jegou et al. [32], the circle with a C represents a concatenation.

architectures are compared. The goal is to give an overview of different approaches that can be used and investigate the performance on different datasets.

Chapter 2

Data

2.1 Farsiu dataset

Dataset *Farsiu* [34] is a dataset containing 38400 b-scans from 269 subjects with AMD and 115 healthy subjects. The scans were required with a Bioptigen system. The scans have an average axial resolution of 3.19μm and an average lateral resolution of 6.55μm. The b-scans have a 512x1000 pixel resolution. The b-scans contain three different segmentation boundaries on a 5mm diameter centred at the fovea. The three segmentation boundaries are the inner aspect of the inner limiting membrane (ILM), the inner aspect of the retinal pigment drusen complex (RPEDC) and the outer aspect of Bruch's membrane (BM). Figure 2.1 shows an example of an image in this dataset including the provided annotations. This particular b-scan is a scan at the centre of the fovea. Often the length of the annotated lines is shorter. This happens when the scans lay further away from the fovea. A test and validation set are randomly sampled on a subject level. The test set contains b-scans from 50 subjects and the validation set contains b-scans from 20 subjects. Both sets contain an equal amount of healthy subjects and subject with AMD. The size of test set was chosen such that it will represents most of the variances of the data. Although more data is available of healthy subjects, an equal amount of healthy subjects and subjects with AMD is included. The reason for this is that the model should not be able to get a good performance by performing well on one subject class specifically well. The size of the validation set was chosen to make sure that it could be used properly for model selection. Both types of subjects are represented equally in the validation set as well in order to make the variance in the data resemble as closely to that of the testing set as possible. All other subjects, 314 in total, are used for training. The volume of one subject contains 100 b-scans, however not all b-scans are annotated as some scans are scanned at a distance greater than 5mm from the fovea. This results in 23992 b-scans included in the training set, 1480 b-scans in the validation set and 3814 b-scans in the test set.

2.2 Chiu dataset

Dataset *Chiu* [35] is a dataset made publicly available by the same research group as the *Farsiu* dataset. This dataset contains 110 b-scans from ten different subjects. The scans are measured on a Heidelberg system. These b-scans all contained 768 a-scans. All a-scans contained 496 val-

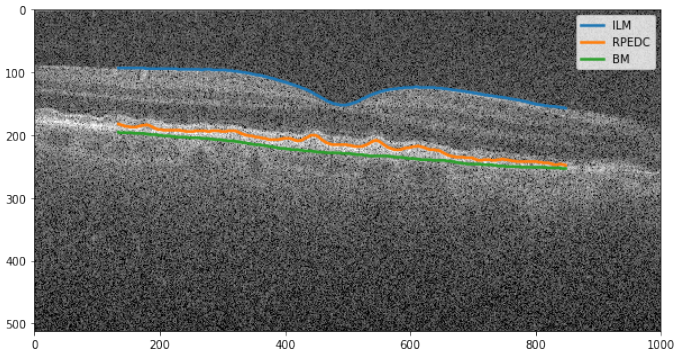


Figure 2.1: Annotated b-scan from Farsiu dataset

ues. The a-scans had an axial resolution of 3.87 μm/pixel and a 11.07 – 11.59 μm/pixel lateral resolution. The b-scans were annotated with nine different retinal boundaries and fluid. For this thesis only the retinal boundaries were taken into account. The boundaries were determined using the following structures of the retina: inner limiting membrane (ILM), Nerve fiber layer (NFL), Inner nuclear layer (INL), Outer plexiform layer (OPL), Outer nuclear layer (ONL), Inner segment myeloid (ISM), Inner segment ellipsoid (ISE), Outer segment (OS), Retinal pigment epithelium (RPE), Bruch's membrane (BM). The nine retinal boundaries are: ILM, NFL/GCL, IPL/INL, INL/OPL, OPL/ONL, ISM/ISE, OS/RPE, BM. Figure 3 2.2 shows an example of a b-scan from this dataset with annotations. The validation set and test set both have the b-scans of one (differ-

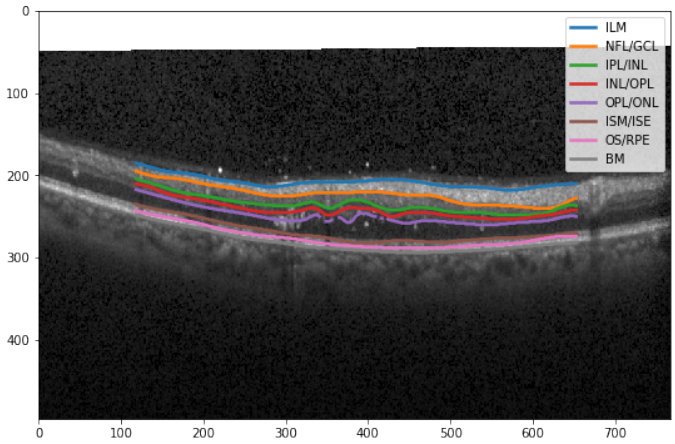


Figure 2.2: Example b-scan from Chiu dataset with annotations

ent) subject. The scans of all other subjects were assigned to the training set. Ideally more subjects would be included in the testing and validation set, but in order to be able to properly train the network all other subjects were included in the training set.

2.3 Eugenda dataset

This data is a subset of the Eugenda database made available by Radboud UMC. The set used contains 57 different b-scans, all from different subjects. The scans are acquired on a Heidelberg system. The dataset contains b-scans from patients with different severity levels of AMD. The OCT scans in this dataset were not all measured using the same scanner settings. This resulted in scans of different pixel resolutions and different spatial resolutions. The b-scans had either a 466x1536 pixel resolution, 469x1024 pixel resolution or a 496x512 pixel resolution. All scans have a axial resolution of 3.872μm. The 466x1536 pixel resolution and 469x1024 pixel resolution scans had a lateral resolution of 5.5μm while the 496x512 pixel resolution scans had a lateral resolution of 11.5μm. The b-scans are annotated with twelve retinal boundaries. The boundaries are constructed using the following retinal structures: inner limiting membrane (ILM), retinal nerve fiber layer (RNFL), ganglion cell layer (GCL), inner plexiform layer (IPL), inner nuclear layer (INL), outer plexiform layer (OPL), outer nuclear layer (ONL), external limiting membrame (ELM), elipsoid zone (EZ), retinal pigment epithelium (RPE), Bruch's membrane (BM) and choroid-sclera junction (CSJ). The boundaries annotated are: ILM, RNFL/GCL, GCL/IPL, IPL/INL, INL/OPL, OPL/ONL, ELM, EZ-top, RPE-top, RPE-bottom, BM, CSJ. Both the validation and test set are assigned five b-scans, the other b-scans are used for training. Figure 2.3 shows an example of a b-scan from this dataset with annotated boundaries. Ideally all sets would contain more scans as the amount of scans in all sets is quite small. The amount was chosen as now the testing and validation set are around 1/10 of the size of the training set.

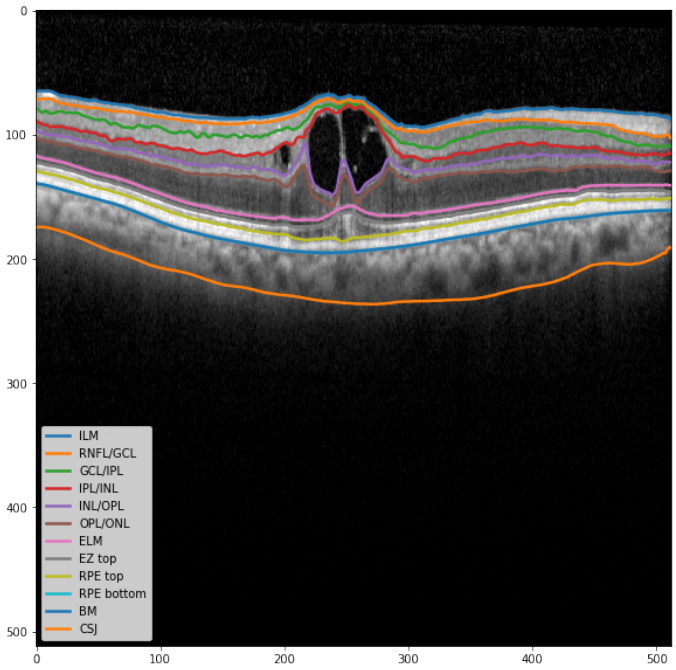


Figure 2.3: Annotated b-scan from Eugenda dataset

Chapter 3

Experiments

The *Farsiu* dataset is used for exploratory research. The models that work well on this dataset will be tested on the other datasets that contain less b-scans and more severly affected retinas.

3.1 Experiments on Farsiu dataset

3.1.1 U-net architecture

As the U-net architecture has been shown to work well for segmentation problems, the first method that is tried is applying a U-net model. Venhuizen et al. [21] introduced a generalization of U-net. This generalization describes a way to create deeper versions of U-net. Figure 3.1 shows a schematic diagram of the architecture they proposed. For this thesis an implementation of a generalized version of U-net is used, this way the depth of U-net can be considered as a parameter. The shape of the output is the same way as the input pixel resolution with a channel for every class it tries to predict, the $n'th$ channel represents the probability of every pixel belonging to class n. One big difference between the original implementation of U-net and the implementation used in these experiments is that the original U-net used convolutions without padding (valid convolutions) the input. The U-net used in these experiments do pad their input (same convolutions). By using same convolutions, the output probability map will have the same dimensionality as the input (except for the number of channels).

Pre-processing

In order for the model to be able to handle the data it should be converted in a way that U-net can train on it. First the input is padded with zeros to a width of 1024. This way the max-pooling operation can be applied up to nine times without padding. This is necessary in order for the expanding path to be able to expand the output to the same size as the input. The pixel intensities are normalized such that all values fall between zero and one.

The lines annotated should be converted to a label that can be trained on for this network. A label is constructed by making four classes where the boundaries separate the classes. If the a-scan does not contain an annotation will not be used for training. Figure 3.2 shows an example of a label generated from the annotations provided.

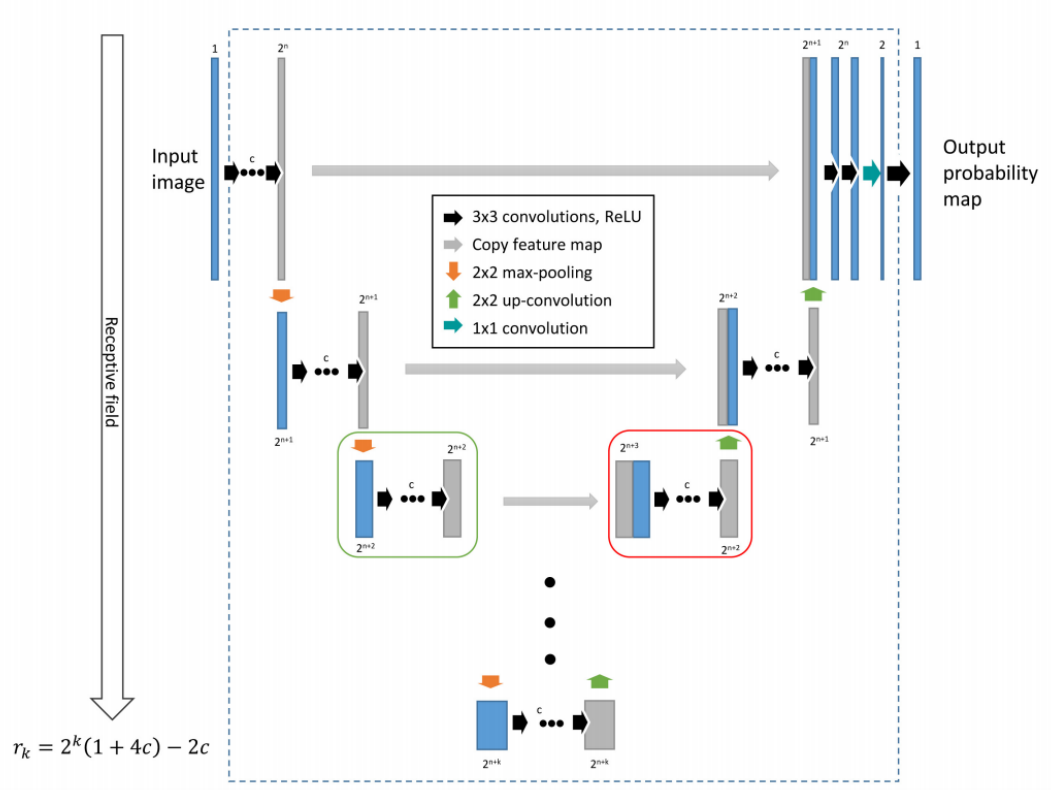


Figure 3.1: Generalization U-net described by Venhuizen et al. [21]

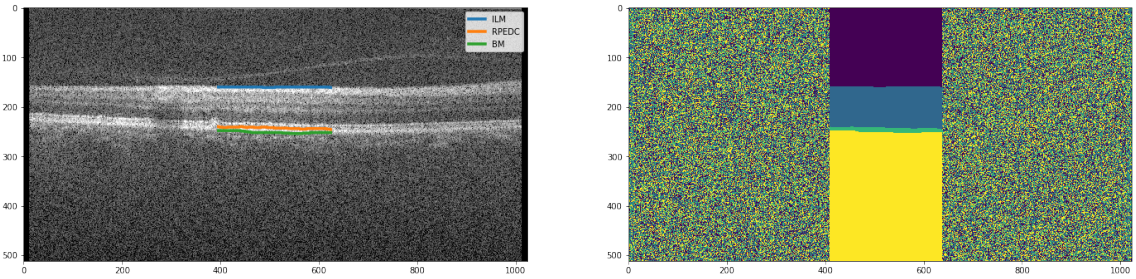


Figure 3.2: Example of mask generated from provided annotations. A random value was assigned to a-scans without annotation, they will not be used during training.

Network training

The network is trained using weighted cross entropy as loss function as shown in equation 3.1. The weighted cross entropy is calculated over M classes of images with N pixels. If a label is present for the a-scan where pixel o resides in $w_o = 1$, if a label is not present $w_o = 0$.

$$Weighted\ CE = \frac{-\sum_{o=0}^{N} w_o(\sum_{c=0}^{M} \hat{Y}_{o,c} \log(Y_{o,c}))}{\sum_{o=0}^{N} w_o} \tag{3.1}$$

The weighted version of the CE function is used in order to deal with the absence of annotations in images. This way the loss value for a-scans is set to zero whenever no annotation is available.

The network is trained on batches with a batch size of two, as bigger batches do not fit on the GPU as a result of memory constraints. One epoch during training contains all train b-scans exactly once. Validation happens after every epoch. When validating, the weighted cross entropy loss and the MSE is calculated. The MSE is calculated by converting the predicted labels back to boundaries that separate the classes. The highest index (calculated from the top) per a-scan of every class predicted is chosen as a value for the boundary. The MSE is only calculated for a-scans that contain annotations. If the mean MSE value over all b-scans and all layers found is lower than the MSE values of the epochs before that, the model is saved.

Evaluation

The MSE can be calculated for every boundary separately. The training loss, validation loss and validation MSEs are also saved so they are investigated after training. After training the validation MSE of the best model is investigated to see whether the boundaries generated by the model are shifted in a direction. If this is the case, the average shift is calculated and used as a correction for the boundaries generated for the test set.

Experiments

Optimizers Using the generalized version of U-net, it is possible to create a U-net of greater depth. The first U-net trained is of depth six [21]. This way the network can take more context into account. This network was trained using stochastic gradient descent (SGD) [27] and Adam [36] as optimizers using various learning rates. Table 3.1 shows the lowest MSE in pixels on the validation set for the variations of optimizers and learning rate tested.

	ILM	RPEDC	BM
SGD 1e-2	5.93	8.05	8.62
SGD 1e-3	2.25×10^4	2.75×10^2	3.77×10^2
SGD 1e-4	1.74×10^4	4.52×10^4	4.92×10^4
Adam 1e-3	3.35	6.56	5.43
Adam 1e-4	**3.33**	**6.42**	**4.02**

Table 3.1: Lowest MSE in pixels on validation set for optimizers tested (U-net depth 6)

Further experiments are carried out with an Adam optimizer and a learning rate of 10^{-4} as those settings yield the best results in the optimizer / learning rate experiment.

Depth In order to confirm whether the deeper version of U-net actually helps for the segmentation problem, the results are compared to that of a network with depth four and five. Depth four corresponds to the original U-net. A depth of more than six was not possible due to limitations of the GPU memory. Table 3.2 shows the lowest MSE in pixels found for the depths tested.

	ILM	RPEDC	BM
U-net 4	3.80	**6.20**	4.17
U-net 5	3.39	6.44	4.31
U-net 6	**3.33**	6.42	**4.02**

Table 3.2: Lowest MSE in pixels on validation set for depths tested

As depth six provided the best results on average, this depth is used for further comparisons.

Class weights and initialization Given the best performing architecture and optimizer, it is interesting to see whether it is possible to optimize this. Two variations of the network will be tested. Firstly, instead of using the default weight initialization (glorot-uniform) of the network, the weights of the network as initialized with a he-normal distribution. He-normal tends to work better when Relu is used as an activation function [37]. Secondly, class-weighting is applied. The amount of pixels belonging to the classes between the boundaries is much smaller than the amount of pixel outside of the boundaries. The class between ILM and RPEDC is given a class weight of two, the class between RPEDC and BM is given a class weight of four. These values were chosen arbitrary, but higher class weights are assigned to smaller areas. Table 3.3 shows the lowest MSE in pixels found for these variations.

	ILM	RPEDC	BM
Glorot	3.33	6.42	4.02
He-normal	**3.17**	**6.32**	3.73
Weighted	3.49	6.71	**3.18**

Table 3.3: Lowest MSE in pixels on validation set on variations tested (U-net 6)

Regressing the boundaries

Instead of calculating the boundaries directly it might also be possible to extend the architecture to predict the index of the segmenting boundaries directly. The network predicts a class probability for every pixel for each of the four classes. The n'th channel of the output can be interpreted as a matrix that contains the probabilities of every pixel belonging to the n'th class. Consider the situation that the network produces the perfect output, in that case the value of a pixel belonging to class n would be 1 in the n'th channel. For the first channel this means that all pixels above the first boundary are a one, and the rest of the pixel are a zero. This means that the sum of all these pixels in an a-scans produces the index of the boundary at that a-scan. For the second boundary in an a-scan the index can be produced by summing all values of the first two channels. This carries on, so given an perfect output, the index of the n'th boundary can be calculated by summing the values of the first n channels (at that a-scan). So by taking a locally weighted sum on the produces output, it might be possible to find the index values of the separation boundaries immediately.

As the convolution operation actually is taking a locally weighted sum, producing the decision boundaries should be possible by adding one convolution layer. As the goal is to take the weighted sum of every a-scan, the convolution kernel should have the same shape as one a-scan (512x1). As an experiment a trained U-net was taken, its layers were frozen and then one convolution layer with a kernel of 512x1 was added. This layer was then trained with the goal of predicting the decision boundaries directly. SGD with a learning rate of 0.01 was used as optimizer. The network was trained on a custom version of MSE where only the a-scans with annotations were taken into account as shown in equation 3.2. If an annotation is available for i'th a-scan then weight $w_i = 1$, else weight $w_i = 0$.

$$\textit{Weighted MSE} = \frac{\sum_{i=0}^{n}((Y_i - \hat{Y}_i)^2 \cdot w_i)}{\sum_{i=0}^{n} w_i} \tag{3.2}$$

The network was however not able to reconstruct the boundaries using this method. The model was only able to produce near horizontal lines as boundaries at a height around the same place where the boundaries normally appear in the b-scans.

3.1.2 Densenet architecture

Densenet is an architecture that also successfully has been deployed for segmentation problems. Jegou et al. [32] found state-of-the-art results using a densenet for image segmentation. Figure 3.3 shows a schematic diagram of their implementation of this network.

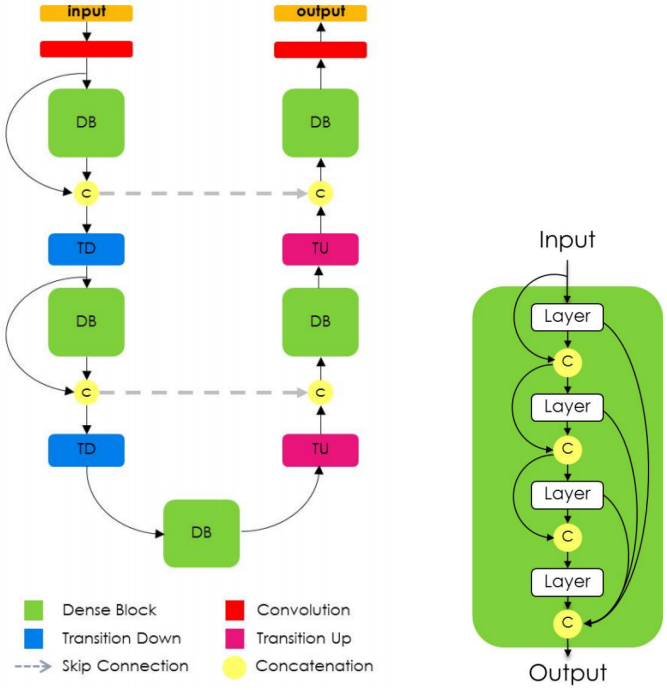
As the output of this network is the same as the output of U-net, the data is pre-processed the same. The training of this network can also be done using the same loss function and validation strategy as with U-net. In order to train the network, it however needs to be cut in size severely (compared to the implementation by Jegou et al.[32]). This is needed, because otherwise the feature maps would not fit on the memory of the GPU. The result of this is that only three dense blocks containing 2 layers per block could be used. A few variations of this network are tested, this architecture was tested with a SGD optimizer with a learning rate of 10^{-4} and 10^{-3} and an Adam optimizer with a learning rate of 10^{-4}. Table 3.4 shows the lowest MSE in pixels found for these variations.

	ILM	RPEDC	BM
SGD 1e-4	553.57	20.11	13.45
SGD 1e-5	846.01	57.53	17.01
Adam 1e-4	**444.08**	**13.41**	**7.27**

Table 3.4: Lowest validations MSE in pixels on densenet experiments

3.1.3 Pyramid architecture

There is a big difference between the OCT segmentation problem and regular segmentations problems. Instead of identifying objects in an image, the goal is now to identify the boundary that separates the classes. As the layers of the retina always appear in a specific order, it might be possible to make use of this property. Instead of identifying the classes that are separated by the boundaries, it might be possible to identify the boundaries immediately. In order to identify the



(a) Densenet (b) Denseblock

Figure 3.3: Densenet for segmentation by Jegou et al. [32]

location of the boundary it might be possible to predict its location through regression. For this a custom network architecture, called the pyramid architecture is designed. The input of this network is the 512x1000x1 b-scan and the output is a 1x1000x3 tensor that represent the boundary location. For every a-scan three values will be predicted: the index of the pixel where the separation boundaries reside.

The network consists of nine blocks of two convolutions and one max-pooling layer. All convolutions are 3x3 convolutions followed by a Relu activation. The first two blocks have regular 2x2 max-pooling layers. All other block have 2x1 max-pooling layers. The first two max-pooling layers allows for more context to be taken into account, however as the width of the output should be the same as with the input only two 2x2 max-pooling operations are used. After block five and six the network is up sampled with a 2x1 pool size, this expands the features to the input width. The last layer of the network is a 1x1 convolution layer with three channels with a sigmoid activation. This produces a 1x1000x3 tensor that contains the predicted values of the boundaries. Figure 3.4 shows a schematic diagram of the Pyramid architecture.

Pre-processing

If the a-scan does not contain any annotations, the value of the label is set to zero. However, just as with the U-net and Densenet training, the absence of an annotation should be taken into account during training.

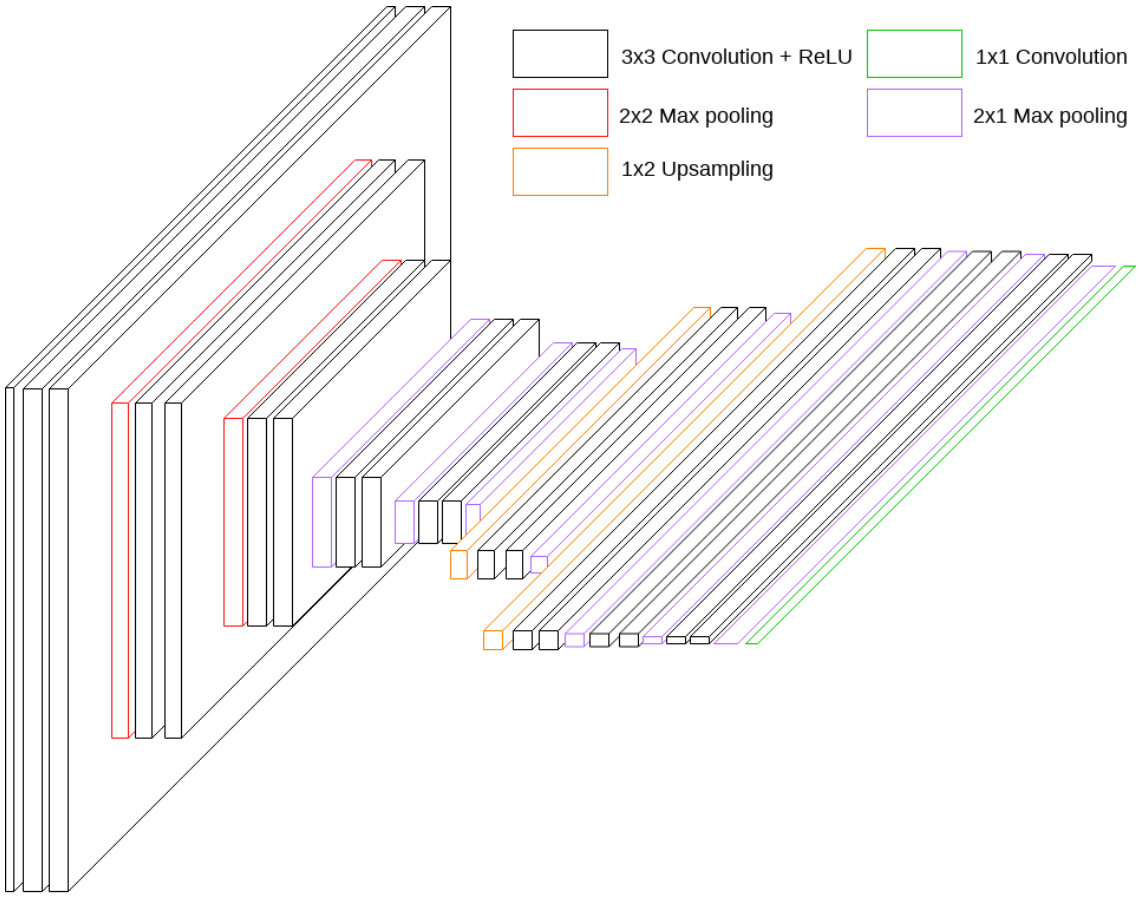


Figure 3.4: Schematic diagram of the pyramid architecture

The network is also trained using the weighted MSE loss function as shown in Equation 3.2. This network is also trained on batches with a batch size of two.

Experiments

This network was trained using SGD as an optimizer. The labels the network is trained on contain the indices representing the boundaries in the a-scans. The indices were trained on the real values of the indices and the indices normalized. When training on the indices normalized they were divided by 512, scaling all values between 0 and 1. When training on the not normalized values however, the loss exploded resulting in the network not training. When the output was normalized, the loss was extremely low. For example, if the model predicted 10 pixel over or under, the squared error would be $(10/512)^2$. This resulted in the network not able to train really well either. The lowest found value for the MSE in pixels is shown in 3.5.

	ILM	RPEDC	BM
Pyramid	9580.02	25413.00	27748.57

Table 3.5: Lowest validations MSE in pixels pyramidnet experiments

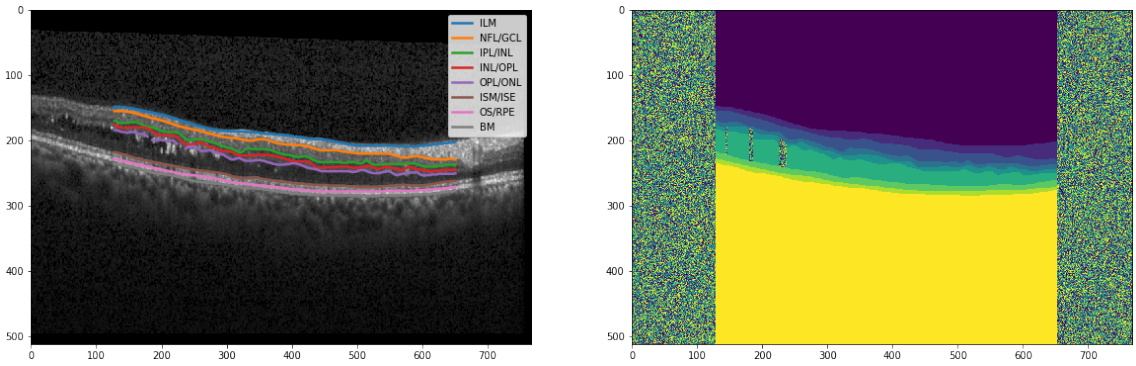


Figure 3.5: Example of mask generated from provided annotations

3.2 Chiu dataset

3.2.1 U-net architecture

The U-net architecture provided the best results for the first dataset. The same architecture as used for the *Farsiu* dataset is used for this dataset. The network is trained the same as with the *Farsiu* dataset, the only difference is that it now produces an output of eight classes instead of four.

Pre-processing

In order for the U-net to be able to apply six max-pooling operations, the image is padded with zeros in height until six max-pooling operations without striding are possible. This means that the image is padded until the amount of pixels is divisible by 2^6, resulting in a height of 512.

The mask is generated similarly as with the *Farsiu* dataset. The boundaries annotated separate the classes the U-net will try to predict. The a-scans that are not annotated are assigned random labels. At some places the annotators were not able to give an exact judgment. In these cases the boundary contains a hole. If this is the case it is not possible to identify where layers start or stop. When this is the case, the layers that are separated by that boundary also get random values assigned at that particular a-scan. Figure 3.5 shows an example of a mask generated from the provide annotations.

Experiments

Optimizers For the first experiments on this dataset also a U-net of depth six was used. The network was again trained with different optimizers and different learning rates. The network was trained with SGD with a learning rate of 10^{-2} and Adam with a learning rate of 10^{-4} and 10^{-5}. Table 3.6 shows the MSE in pixels found on the validation set.

	ILM	NFL/GCL	IPL/NFL	INL/OPL	OPL/ONL	ISM/ISE	OS/RPE	BM
SGD 1e-2	1.84×10^4	90.96	177.12	174.80	563.55	10.2×10^3	507.57	14.6×10^3
Adam 1e-4	**1.76**	**4.24**	**6.16**	**8.68**	**9.14**	**1.43**	**1.32**	**1.39**
Adam 1e-5	2.10	37.25	13.99	18.94	23.91	166.45	2.00	3.17

Table 3.6: Lowest MSE in pixels found by model best performing on validation data

Depth In order to confirm that a U-net of depth six works the best, it is again compared to U-net architectures of depth four and five. Table 3.7 shows the lowest MSE in pixels found on the validation set.

	ILM	NFL/GCL	IPL/NFL	INL/OPL	OPL/ONL	ISM/ISE	OS/RPE	BM
Depth 4	3.86	5.16	5.96	10.21	14.74	**1.15**	1.50	1.45
Depth 5	2.26	7.73	**5.76**	**7.90**	11.07	1.72	1.42	1.65
Depth 6	**1.76**	**4.24**	6.16	8.68	**9.14**	1.43	**1.32**	**1.39**

Table 3.7: Lowest MSE in pixels found by model best performing on validation data

Augmentation As the number of b-scans trained on is much lower than with the *Farsiu* dataset it might help to augment the data in this dataset. Three augmentations were implemented to enrich the training data. The first augmentation applied is a flip over the vertical axis. The second augmentation is applying a Gaussian blur using a distribution with a standard deviations of one, two or three pixels. The last augmentation is applying Gaussian salt and pepper noise, with a standard deviation of five. All three augmentations have a probabilty of 0.5 of being applied. Table 3.8 shows the lowest MSE in pixels found on the validation set for the model trained on data with and without augmentations.

	ILM	NFL/GCL	IPL/NFL	INL/OPL	OPL/ONL	ISM/ISE	OS/RPE	BM
Normal	**1.75**	4.24	6.16	8.68	**9.14**	1.43	**1.32**	**1.39**
With augmentation	1.97	**3.34**	**5.78**	**5.37**	12.42	**1.20**	1.60	1.62

Table 3.8: Lowest MSE in pixels found by model best performing on validation data

3.3 Eugenda dataset

3.3.1 U-net Architecture

The same U-net architecture is also used for experiments on this dataset. As the b-scans are annotated with twelve retinal boundaries, the network now has to classify thirteen classes. The network is trained and validated in the same way as the previous experiments.

Pre-processing

The scans with a lateral resolution of 5.5μm are down-sampled to have the same spatial resolution as the other images. A slice with a width of 512 of the images with a width of 768 is taken so

the pixel resolution is the same for all images. The images are then padded with zeros until six max-pooling operations are possible. The label is generated the same as labels generated from the *Chiu* dataset. Figure 3.6 shows an example of a label generated that is used for training.

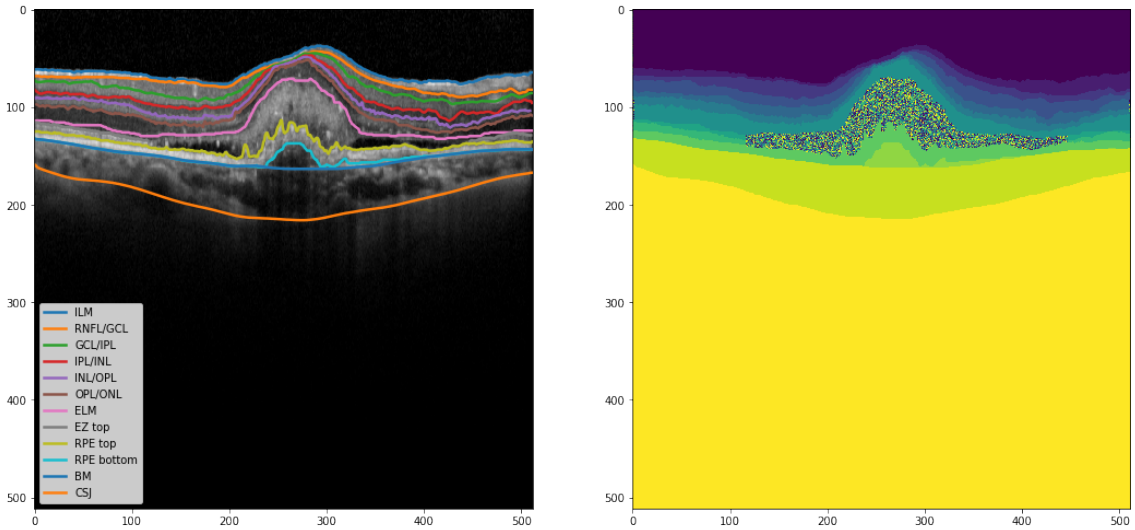


Figure 3.6: Example b-scan from Eugenda dataset with annotations

Experiment

For this dataset only the U-net with depth six architecture was applied. The network was trained using an Adam optimizer with a learning rate of 10^{-4}. Table 3.9 shows the lowest MSE in pixels found on the validation set.

ILM	RNFL/GCL	GCL/IPL	IPL/INL	INL/OPL	OPL/ONL
8.73	6.07	10.26	9.05	7.03	7.06
ELM	EZ-top	RPE-top	RPE-bottom	BM	CSJ
21.20	15.02	9.53	10.47	12.39	9.48

Table 3.9: Lowest MSE in pixels found by model best performing on validation data

Chapter 4

Results

4.1 Farsiu dataset

Table 4.1 shows the results of the trained models on the testing data. The mean absolute error in μm is reported together with the standard deviation of the absolute error distribution.

	AMD			Control		
	ILM	RPEDC	BM	ILM	RPEDC	BM
U-net 6 SGD 1e-2	3.89 (7.21)	5.77 (7.24)	6.64 (37.39)	3.57 (4.21)	4.82 (4.43)	4.05 (3.67)
U-net 6 SGD 1e-3	4.65 (10.31)	9.52 (48.26)	10.30 (40.47)	4.20 (5.54)	5.16 (4.37)	4.94 (4.34)
U-net 6 SGD 1e-4	431.76 (261.01)	56.16 (74.07)	64.04 (109.34)	417.12 (241.08)	41.01 (27.76)	53.69 (89.22)
U-net 6 Adam 1e-3	132.83 (93.51)	133.67 (94.72)	132.89 (95.377)	102.85 (72.19)	107.75 (73.57)	107.82 (74.37)
U-net 6 Adam 1e-4	**3.62 (3.93)**	5.82 (7.20)	4.46 (4.87)	3.64 (4.11)	4.75 (4.47)	3.69 (3.63)
U-net 4	35.34 (158.64)	6.62 (20.57)	12.78 (46.06)	**3.56 (7.72)**	5.32 (4.56)	4.14 (3.43)
U-net 5	3.77 (9.14)	5.85 (11.29)	4.95 (23.19)	3.64 (4.15)	4.93 (4.65)	3.56 (3.59)
U-net 6 He-normal	3.64 (4.21)	5.83 (6.40)	**3.91 (4.19)**	3.63 (4.32)	4.52 (4.14)	**3.35 (3.31)**
U-net 6 Weighted	3.74 (4.57)	**5.73 (6.82)**	4.80 (5.51)	3.61 (4.00)	**4.51 (4.29)**	3.86 (3.24)
Densenet SGD	48.43 (166.67)	11.06 (48.126)	10.89 (47.56)	13.95 (62.02)	4.84 (7.07)	4.14 (6.00)
Densenet Adam 1e-4	32.35 (121.90)	9.76 (30.03)	8.49 (34.51)	8.02 (31.84)	4.91 (7.35)	4.16 (3.97)
Densenet Adam 1e-5	54.33 (172.89)	14.98 (57.35)	10.13 (42.51)	24.57 (93.20)	5.53 (11.87)	4.45 (5.29)
Pyramid	123.23 (50.85)	207.55 (51.25)	217.61 (51.17)	110.36 (37.46)	192.78 (38.59)	202.10 (38.63)

Table 4.1: Mean absolute error and standard deviation of absolute error in μm on test set

Examples of some predictions on the *Farsiu* testing set by the U-net network with depth 6 trained with an Adam optimizer using a learning rate of 10^{-4} are shown in Figure 4.1.

4.2 Chiu dataset

The results found by the models trained on the *Chiu* dataset is shown in Table 4.2.

Examples of some predictions on random samples from the *Chiu* testing set by the U-net network with depth 6 trained with an Adam optimizer using a learning rate of 10^{-4} are shown in Figure 4.2.

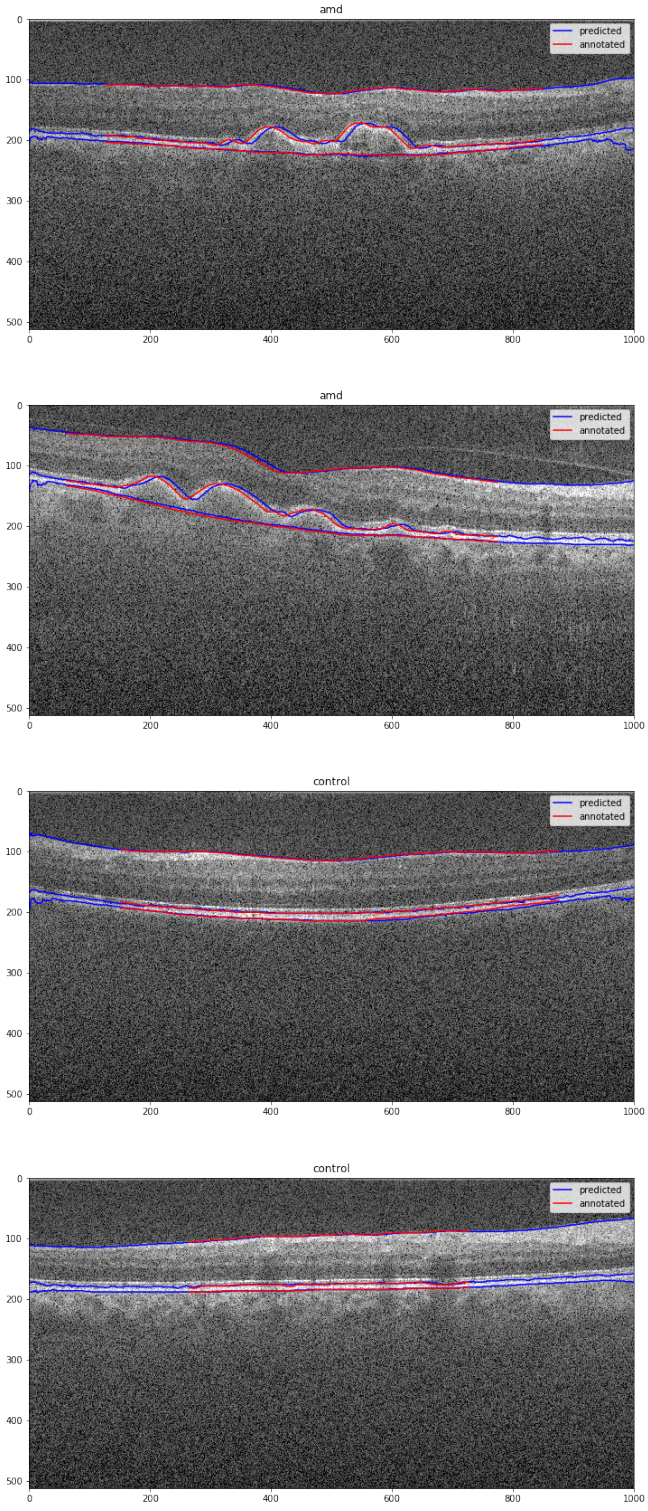


Figure 4.1: Examples of boundaries predicted on b-scans from *Farsiu* testing set

	ILM	NFL/GCL	IPL/NFL	INL/OPL	OPL/ONL	ISM/ISE	OS/RPE	BM
U-net 6 Adam 1e-4	**3.61 (2.47)**	**5.55 (6.42)**	10.26 (12.02)	9.29 (9.01)	10.08 (8.82)	3.70 (2.69)	3.47 (2.35)	3.05 (2.22)
+ augmentation	3.68 (2.77)	6.84 (7.38)	**7.24 (8.23)**	**8.77 (9.22)**	**9.36 (8.33)**	3.88 (2.73)	3.94 (2.85)	**2.34 (2.77)**
U-net 6 Adam 1e-5	4.73 (4.13)	20.72 (22.94)	23.49 (19.79)	20.53 (18.71)	15.38 (17.12)	5.66 (6.39)	4.47 (4.07)	5.79 (11.68)
U-net 6 SGD 1e-2	280.45 (276.08)	53.05 (84.35)	42.47 (71.19)	45.10 (67.89)	44.65 (66.45)	85.55 (67.65)	70.32 (66.10)	69.19 (65.84)
U-net 5 Adam 1e-4	3.65 (3.90)	7.39 (6.53)	13.82 (15.59)	9.40 (11.22)	10.87 (10.67)	3.47 (2.85)	3.31 (2.66)	3.42 (2.47)
U-net 4 Adam 1e-4	3.92 (3.65)	23.10 (22.48)	19.56 (17.75)	14.03 (19.68)	13.38 (17.56)	**3.27 (2.41)**	**3.29 (2.36)**	3.17 (2.83)

Table 4.2: Mean absolute error and standard deviation of absolute error in μm on test set

4.3 Eugenda dataset

The results found by the model trained on the *Eugenda* dataset are shown in Table 4.3.

ILM	RNFL/GCL	GCL/IPL	IPL/INL	INL/OPL	OPL/ONL
6.02 (6.43)	8.00 (8.28)	10.86 (9.55)	10.34 (11.08)	11.40 (10.55)	14.00 (14.19)
ELM	EZ-top	RPE-top	RPE-bottom	BM	CSJ
13.00 (15.38)	12.35 (13.30)	15.10 (14.08)	14.96 (13.58)	14.30 (14.02)	26.94 (23.63)

Table 4.3: Mean absolute error and standard deviation of absolute error in μm on test set

Examples of predictions on b-scans out of the testing set by the trained model are shown in Figure 4.3.

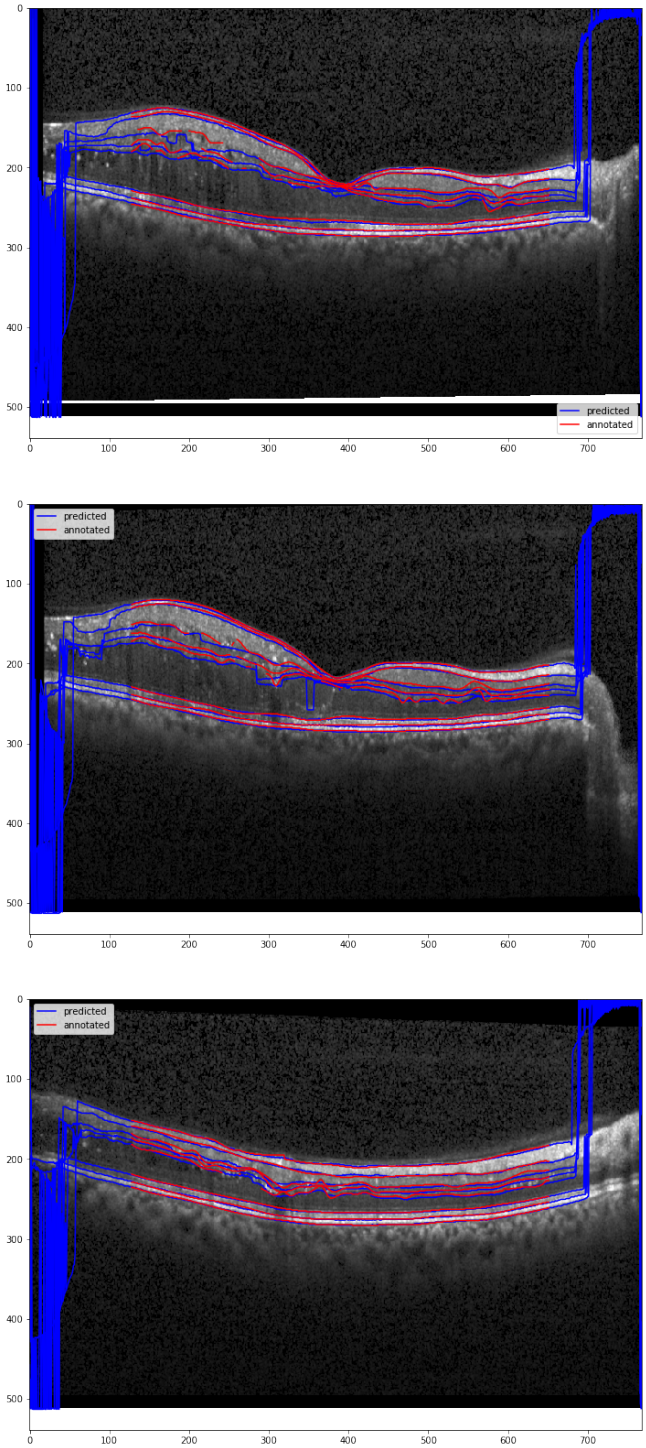


Figure 4.2: Examples of boundaries predicted on b-scans from *Chiu* testing set

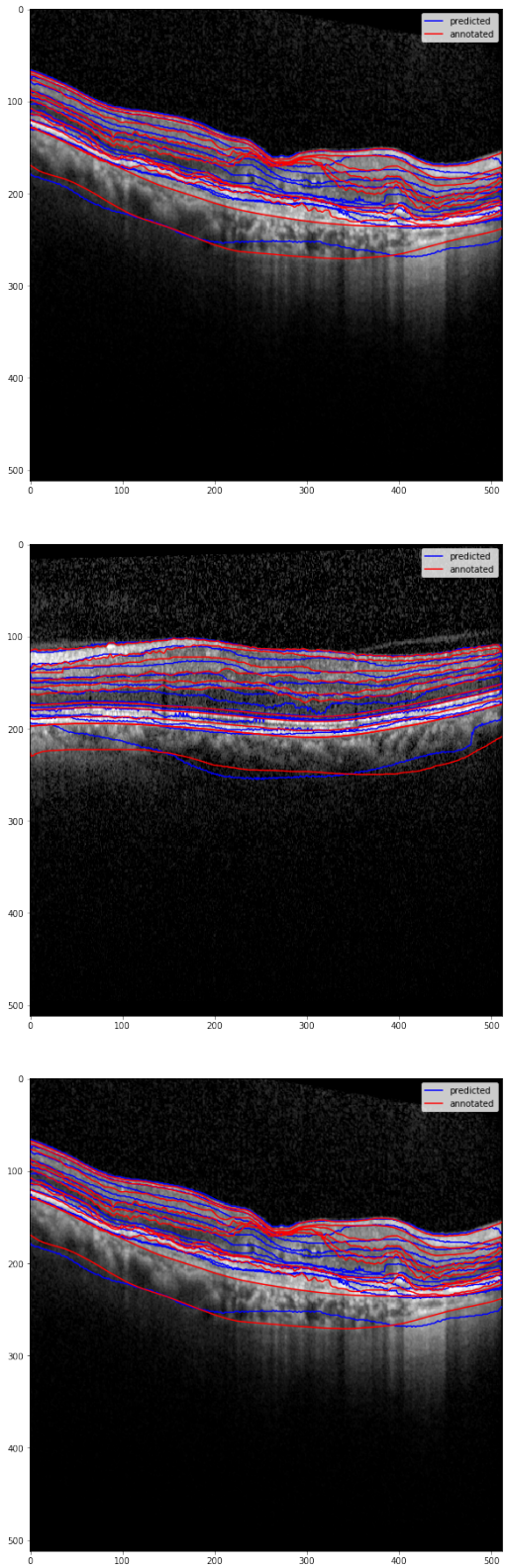


Figure 4.3: Examples of boundaries predicted on b-scans from *Eugenda* testing set

Chapter 5

Discussion

5.1 Datasets

5.1.1 Farsiu dataset

A great advantage of using the *Farsiu* dataset is that it is the largest publicly available annotated OCT dataset. Having many b-scans enables different network architectures to be compared to each other properly. If a network does not perform well on this dataset, it will not perform well on datasets with less b-scans while more layers might need to be predicted. A disadvantage of this dataset is that it is only annotated with three segmentations boundaries. Another disadvantage of this dataset is that the data is quite homogeneous in its nature, only healthy scans and scans of (non-severe) AMD are included. In order to create a clinically relevant system, predicting eight or nine boundaries is desirable. This dataset is especially interesting for exploitative experiments. Farsiu et al. [34] show that especially the location of the inner aspect of the RPE is indicative of AMD. If it is possible to find this boundary precisely using this dataset, the models can still be clinically relevant.

5.1.2 Chiu dataset

In the *Chiu* dataset more retinal boundaries are annotated compared to the *Farsiu* dataset. Although the *Farsiu* dataset is useful for testing methods, the results you find using that dataset might not generalize well to more layers. The results from the *Chiu* dataset give better insight in how well the methods work when multiple layers need to be segmented. As segmenting more layers is more clinically relevant, it provides more information on which models are more interesting. This dataset also has been used in the literature for layer segmentation, making it easier to benchmark models. The amount of b-scans available is however much lower compared to the *Farsiu* dataset. Having more b-scans available might improve the models more.

In one of the experiments augmentations were used to increase the variability in the training data. The model was able to perform better on the test data when the training data was augmented during training. It would be interesting to see if more augmentations would yield even better results.

5.1.3 Eugenda dataset

In the Eugenda dataset even more retinal boundaries are annotated. The layers that are separated by these boundaries were chosen based on their clinical relevance and their visibility. This dataset is especially clinically relevant, as patients in these images sometimes had severe AMD. The results found by the models on the other datasets would probably not generalize well to these images, as large drusen in the images. These do not appear in the other datasets. The amount of annotated b-scans available from this dataset was however minimal. Models trained on this dataset would probably become better if more data is available.

5.2 Networks and results

5.2.1 Training

When training the networks, the architecture was tested by varying certain hyper parameters. However in order to find the best parameters, a more exhaustive search would have to be done. But, the experiments done give an insight in which models work well for this problems and what the effect is on the training process and the eventual results. When really trying to optimize one specific network architecture a variable learning rate would be preferred. Instead of using a fixed learning rate, the learning rate should be gradually decreased during the training phase.

5.2.2 U-net

As shown in the results, the U-net architecture performed the best on the OCT segmentation task. When increasing the depth of the network the performance tended to become better. Figure 5.1 shows the validation loss of the U-net of different depths trained with the Adam optimizer and a learning rate of 10^{-4}. It shows that although a U-net of depth 6 works slightly better than the models with lower depth, the improvement over the other models is small. Increasing the depth of the U-net allows for context to be taken into account, and by having more convolutional layers, more complex patterns might be recognized by the network. This gives the models with greater depth some advantages over the models with a lower depth. It was not possible to increase the architecture to a depth of seven due to GPU memory constraints. If might be interesting to test what the effect would be when increasing the depth even more.

Using he-normal weight initialization and class weights improved the performance slightly on some of the boundaries, but lowered the performance on others. On average he-normal initialization improved the performance of the network. This might also indicate that the network did not train long enough and the same error might be feasible when training longer.

Class weighting improved the performance on some of the boundaries, fine tuning the class weighting might also lead to a small performance gain. The performance gain will be small as the train loss was already low. Instead of trying to fine tune the hyper parameters of the model it might be more interesting to over sample b-scans on which the performance was relatively poor. This technique is also called hard example mining [38]. Figure 5.2 shows an example from the *Farsiu* dataset where this might help.

The best performing models on the *Farsiu* and *Chiu* dataset find mean absolute errors on the boundaries comparable to values found in literature. Table 5.1 shows values found in the literature

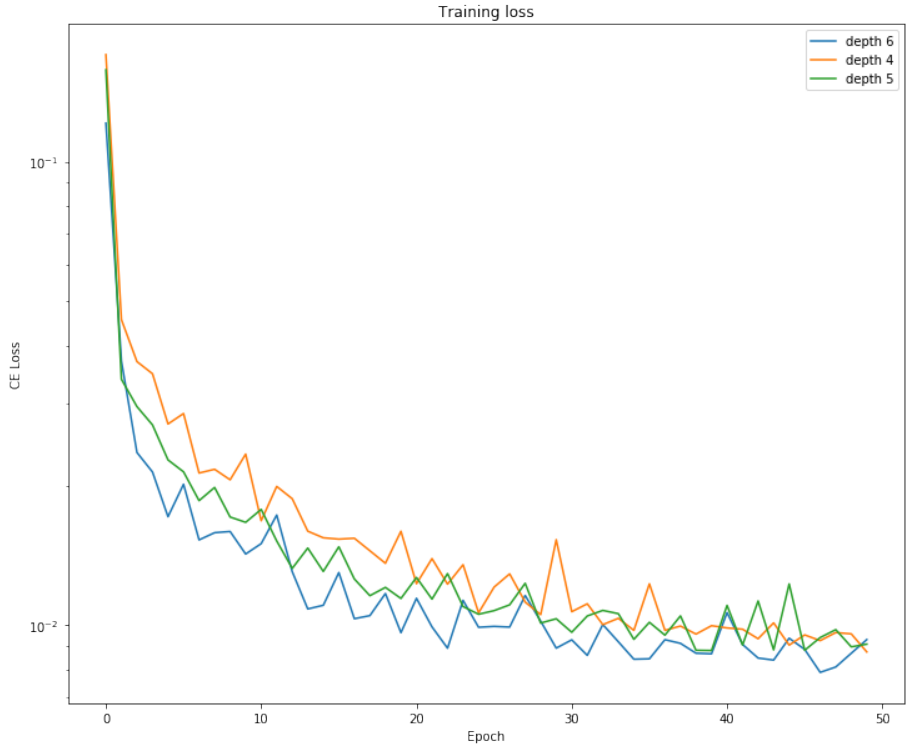


Figure 5.1: CE loss on validation set per epoch for different U-net depths

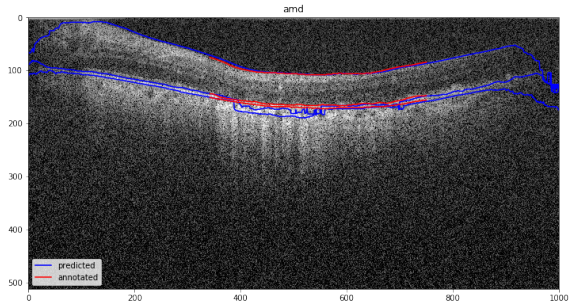


Figure 5.2: Example from *Farsiu* dataset where model performed poorly

compared to the best values found on the *Chiu* dataset. The datasets used in the literature are different from the *Chiu* dataset.

The models are able to segment the b-scans from the *Chiu* dataset relatively well, while less data is available compared to the *Farsiu* dataset. I suspect that this effect is found because the b-scans from the *Chiu* dataset contain much less jitter noise compared to scans from the *Farsiu* dataset.

	ILM	NFL/GCL	IPL/INL	INL/OPL	OPL/ONL	ISM/ISE	OS/RPE	BM
Best this thesis	3.61 (2.47)	5.55 (6.42)	7.24 (8.23)	8.77 (9.22)	9.36 (8.33)	**3.27 (2.41)**	**3.29 (2.36)**	**2.34 (2.77**)
Lang et al. [20]	2.60 (3.33)	**4.00 (6.11)**	**3.87 (4.54)**	3.57 (3.75)	**3.27 (4.06)**	na	4.32 (4.23)	3.50 (3.56)
Yang et al. [39]	**2.20 (0.66)**	4.85 (1.01)	4.31 (1.04)	**3.26 (0.86)**	na	na	na	2.35 (1.21)

Table 5.1: Mean absolute error and standard deviation of absolute error in μm

Directly trying to predict the boundaries from the U-net output failed. Although it theoretically should be possible to do it with one convolution layer, the network does not seem to be able to learn the correct parameters. It would be interesting to experiment more with this and maybe its an idea to initialize the filters with the weights suited for a network that gives a perfect output and try to learn from there.

The higher absolute errors on the *Eugenda* dataset can be explained by the fact that the amount of b-scans available is much lower compared to the other datasets while also having more variability in the data. The *Eugenda* dataset contained b-scans of severely affected retinas where the model was not able to segment the classes well. This results in the model not being able to construct the boundaries.

Google's Deepmind recently publicized a paper regarding OCT analysis [23]. Instead of segmenting retinal layers, they segmented areas that could be affected by a retinal disease. For this they used 3D U-nets. They trained their segmentation model on the full volume instead of the b-scans. It would be interesting to test how well a 3D U-net would segment the retinal layers as it can take more context into account.

5.2.3 Densenet

When training a densenet architecture, the architecture was severely cut in size. Only three dense blocks with two layers could be used. When using bigger/more blocks, the GPU was not able to fit the feature maps in memory. When using U-net, several convolutions happen before the feature maps are down-sampled. When using densenet however, the input of every convolution gets concatenated to the output of that convolution. This results in more features map that need to be saved for the network to train. The size of the b-scans are quite large compared to the images used by Jegout et al. [32]. The result of this is that the original implementation of densenet for image segmentation does not work for bigger images given the GPU limitations. When a small version of densenet is used, it just does not perform as well compared to an architecture as U-net. When GPU's would have more memory it would definitely be worth trying this kind of architecture again.

5.2.4 Pyramid architecture

The pyramid architecture predicts the boundaries directly. This model down sampled the a-scans until almost all information of an a-scan was compressed to a few features. From these features the location of the separation boundaries should be constructed. The network was able to get a general idea where the boundaries reside. However, exactly delineating the boundaries for every a-scan did not work at all. If you compare this network to traditional segmentation models, this model only has the contracting path that captures information but it lack an expanding path for precise localization. In a way this model acts like an encoder. However to exactly decode the information more information is needed for localization. In a model like U-net this information is provided

through skip connections. These skip connections can however not be incorporated in this network as its output shape is quite different from its input shape.

Predicting the boundaries directly might however not be the best idea. The retinal layers are structured in a certain order. However, when patients suffer from retinal diseases it might happen that for example a fluid starts to settle between retinal layers. It is not really possible to construct a boundary between the two layers if there is fluid between them. Instead of trying to predict the boundaries between the layers, it might be more useful to segment the layers themselves. Thickness of layers can still easily be calculated when the complete layer is segmented and models that learn to do this might be more useful as they are less constrained due to assumptions made.

5.3 Conclusion

In this thesis deep learning models were applied on the OCT layer segmentation problem. Different kind of architectures were compared on one dataset. Specifically a deeper version of the U-net architecture worked really well for the layer segmentation problem. A new kind of architecture, the pyramid architecture, was introduced to regress the boundaries immediately. This model was however not able to predict the values of the boundaries close the the annotated boundaries. Instead of calculating the boundaries manually an extension on the U-net architecture was proposed. This extension was however not able to reconstruct the boundaries from the U-net output. The best working U-net architecture was applied on two different OCT datasets showing it generalizes to images measured by different devices. In order to get a good performance for these problems it is however important to have enough data available. Experiments on the *Farsiu* dataset show that the proposed models are able to segment retinal layers well, however for the models to be clinically relevant more layers need to be segmented. When trying to segment more layers on the *Chiu* or *Eugenda* dataset the models were not able to perform a segmentation as accurately compared to the *Farsiu* dataset. More training data would probably help the models segment the retinal layers in those datasets, and therefore make the models more clinically relevant.

Bibliography

[1] J. B. Jonas, U. Schneider, and G. O. Naumann, "Count and density of human retinal photoreceptors," *Graefe's Archive for Clinical and Experimental Ophthalmology*, vol. 230, no. 6, pp. 505–510, 1992.

[2] J. N. Ratchford, S. Saidha, E. S. Sotirchos, J. A. Oh, M. A. Seigo, C. Eckstein, M. K. Durbin, J. D. Oakley, S. A. Meyer, A. Conger, *et al.*, "Active ms is associated with accelerated retinal ganglion cell/inner plexiform layer thinning," *Neurology*, vol. 80, no. 1, pp. 47–54, 2013.

[3] Z. Wu, L. N. Ayton, C. D. Luu, P. N. Baird, and R. H. Guymer, "Reticular pseudodrusen in intermediate age-related macular degeneration: prevalence, detection, clinical, environmental, and genetic associations," *Investigative ophthalmology & visual science*, vol. 57, no. 3, pp. 1310–1316, 2016.

[4] G. Staurenghi, S. Sadda, U. Chakravarthy, and R. F. Spaide, "Proposed lexicon for anatomic landmarks in normal posterior segment spectral-domain optical coherence tomography: The in•oct consensus," *Ophthalmology*, vol. 121, no. 8, pp. 1572 – 1578, 2014.

[5] D. Huang, E. A. Swanson, C. P. Lin, J. S. Schuman, W. G. Stinson, W. Chang, M. R. Hee, T. Flotte, K. Gregory, C. A. Puliafito, *et al.*, "Optical coherence tomography," *science*, vol. 254, no. 5035, pp. 1178–1181, 1991.

[6] F. A. Medeiros, L. M. Zangwill, C. Bowd, and R. N. Weinreb, "Comparison of the gdx vcc scanning laser polarimeter, hrt ii confocalscanning laser ophthalmoscope, and stratus oct optical coherence tomographfor the detection of glaucoma," *Archives of Ophthalmology*, vol. 122, no. 6, pp. 827–837, 2004.

[7] J. E. DeLeón-Ortega, S. N. Arthur, G. McGwin, A. Xie, B. E. Monheit, and C. A. Girkin, "Discrimination between glaucomatous and nonglaucomatous eyes using quantitative imaging devices and subjective optic nerve head assessment," *Investigative ophthalmology & visual science*, vol. 47, no. 8, pp. 3374–3380, 2006.

[8] W. Goebel and T. Kretzchmar-Gross, "Retinal thickness in diabetic retinopathy: a study using optical coherence tomography (oct)," *Retina*, vol. 22, no. 6, pp. 759–767, 2002.

[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.

[10] R. J. Tapp, J. E. Shaw, C. A. Harper, M. P. De Courten, B. Balkau, D. J. McCarty, H. R. Taylor, T. A. Welborn, and P. Z. Zimmet, "The prevalence of and factors associated with diabetic retinopathy in the australian population," *Diabetes care*, vol. 26, no. 6, pp. 1731–1737, 2003.

[11] Y. Jia, J. C. Morrison, J. Tokayer, O. Tan, L. Lombardi, B. Baumann, C. D. Lu, W. Choi, J. G. Fujimoto, and D. Huang, "Quantitative oct angiography of optic nerve head blood flow," *Biomedical optics express*, vol. 3, no. 12, pp. 3127–3137, 2012.

[12] V. Schreur, A. Domanian, B. Liefers, F. G. Venhuizen, B. J. Klevering, C. B. Hoyng, E. K. de Jong, and T. Theelen, "Morphological and topographical appearance of microaneurysms on optical coherence tomography angiography," *British Journal of Ophthalmology*, pp. bjophthalmol–2018, 2018.

[13] H. Ishikawa, D. M. Stein, G. Wollstein, S. Beaton, J. G. Fujimoto, and J. S. Schuman, "Macular segmentation with optical coherence tomography," *Investigative ophthalmology & visual science*, vol. 46, no. 6, pp. 2012–2017, 2005.

[14] D. Koozekanani, K. Boyer, and C. Roberts, "Retinal thickness measurements from optical coherence tomography using a markov boundary model," *IEEE transactions on medical imaging*, vol. 20, no. 9, pp. 900–916, 2001.

[15] V. Kajić, B. Považay, B. Hermann, B. Hofer, D. Marshall, P. L. Rosin, and W. Drexler, "Robust segmentation of intraretinal layers in the normal human fovea using a novel statistical model based on texture and shape analysis," *Optics express*, vol. 18, no. 14, pp. 14730–14744, 2010.

[16] S. J. Chiu, X. T. Li, P. Nicholas, C. A. Toth, J. A. Izatt, and S. Farsiu, "Automatic segmentation of seven retinal layers in sdoct images congruent with expert manual segmentation," *Optics express*, vol. 18, no. 18, pp. 19413–19428, 2010.

[17] M. K. Garvin, M. D. Abramoff, X. Wu, S. R. Russell, T. L. Burns, and M. Sonka, "Automated 3-d intraretinal layer segmentation of macular spectral-domain optical coherence tomography images," *IEEE transactions on medical imaging*, vol. 28, no. 9, pp. 1436–1447, 2009.

[18] A. Mishra, A. Wong, K. Bizheva, and D. A. Clausi, "Intra-retinal layer segmentation in optical coherence tomography images," *Optics express*, vol. 17, no. 26, pp. 23719–23728, 2009.

[19] K. Vermeer, J. Van der Schoot, H. Lemij, and J. De Boer, "Automated segmentation by pixel classification of retinal layers in ophthalmic oct images," *Biomedical optics express*, vol. 2, no. 6, pp. 1743–1756, 2011.

[20] A. Lang, A. Carass, M. Hauser, E. S. Sotirchos, P. A. Calabresi, H. S. Ying, and J. L. Prince, "Retinal layer segmentation of macular oct images using boundary classification," *Biomedical optics express*, vol. 4, no. 7, pp. 1133–1152, 2013.

[21] 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.

[22] A. G. Roy, S. Conjeti, S. P. K. Karri, D. Sheet, A. Katouzian, C. Wachinger, and N. Navab, "Relaynet: retinal layer and fluid segmentation of macular optical coherence tomography using fully convolutional networks," *Biomedical optics express*, vol. 8, no. 8, pp. 3627–3642, 2017.

[23] J. De Fauw, J. R. Ledsam, B. Romera-Paredes, S. Nikolov, N. Tomasev, S. Blackwell, H. Askham, X. Glorot, B. O'Donoghue, D. Visentin, G. van den Driessche, B. Lakshminarayanan, C. Meyer, F. Mackinder, S. Bouton, K. Ayoub, R. Chopra, D. King, A. Karthikesalingam, C. O. Hughes, R. Raine, J. Hughes, D. A. Sim, C. Egan, A. Tufail, H. Montgomery, D. Hassabis, G. Rees, T. Back, P. T. Khaw, M. Suleyman, J. Cornebise, P. A. Keane, and O. Ronneberger, "Clinically applicable deep learning for diagnosis and referral in retinal disease," *Nature Medicine*, 2018.

[24] J. Patterson and A. Gibson, *Deep Learning: A Practitioner's Approach.* " O'Reilly Media, Inc.", 2017.

[25] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, vol. 30, p. 3, 2013.

[26] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315–323, 2011.

[27] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, pp. 177–186, Springer, 2010.

[28] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in neural information processing systems*, pp. 396–404, 1990.

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

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[31] 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*, pp. 234–241, Springer, 2015.

[32] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE*, pp. 1175–1183, IEEE, 2017.

[33] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks.," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE*, vol. 1, p. 3, IEEE, 2017.

[34] 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.

[35] S. J. Chiu, M. J. Allingham, P. S. Mettu, S. W. Cousins, J. A. Izatt, and S. Farsiu, "Kernel regression based segmentation of optical coherence tomography images with diabetic macular edema," *Biomedical optics express*, vol. 6, no. 4, pp. 1172–1194, 2015.

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

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

[38] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 761–769, 2016.

[39] Q. Yang, C. A. Reisman, Z. Wang, Y. Fukuma, M. Hangai, N. Yoshimura, A. Tomidokoro, M. Araie, A. S. Raza, D. C. Hood, *et al.*, "Automated layer segmentation of macular oct images using dual-scale gradient information," *Optics express*, vol. 18, no. 20, pp. 21293–21307, 2010.