Erasmus University Rotterdam

Erasmus School of Economics

Master Thesis Econometrics and Management Science: Business Analytics and Quantitative Marketing

---

Multimodal Prediction of Glaucoma Severity with Convolutional Neural Networks

---

Name student: Fenna ten Haaf

Student ID number: 450812fh

Supervisor: dr. Paul Bouman

Second assessor: dr. Nick Koning

Company supervisors (BIGR): dr. Luisa Sánchez Brea & dr. Danilo Andrade de Jesus

Date final version: November 5, 2021

Abstract

Glaucoma is a progressive eye disease where early detection is critical, as the progression is irreversible and can lead to blindness. Additionally, the staging of glaucoma severity can aid in treatment, but the test for severity is unreliable and time-consuming. This thesis aims to predict glaucoma severity using deep learning methods by addressing two tasks: the classification of glaucoma versus healthy subjects and the prediction of the visual field mean deviation (VF MD) score. We approach these tasks by using and combining different imaging modalities: fundus photographs, optical coherence tomography (OCT), and the relatively unexplored optical coherence tomography angiography (OCTA). To the best of our knowledge, we are the first study to use a combination of these modalities for investigating glaucoma severity. The dataset is obtained from 614 patients from the Leuven Eye Study (Abegão Pinto et al., 2016). Single-input convolutional neural networks (CNNs) are compared with a multi-input CNN and a stacked ensemble approach. Although we do not reach state-of-the-art results as seen in the literature so far due to the small dataset leading to lower generalizability, we find that the multi-input CNN and stacked ensemble methods are promising approaches and could be worth exploring further.

Contents

1 Introduction

Glaucoma is a progressive eye disease which is estimated to affect more than 80 million people globally, a number that is expected to increase due to ageing populations (Tham et al., 2014). Even so, the disease and its causes are still not fully understood. In most glaucoma cases, loss of sight happens slowly as a result of the optic nerve being damaged gradually over time and this damage is irreversible. If left untreated, this can result in blindness. Therefore, it is vital that tools exist to aid in early diagnosis and close monitoring of glaucoma patients.

In order to help ophthalmologists detect glaucoma and other diseases, Computer Aided Diagnostics (CAD) systems are being developed in the medical imaging and machine learning fields. Deep learning (DL), a subfield of artificial intelligence based on deep neural networks, has shown to be a highly interesting technology for medical imaging (Greenspan, Van Ginneken, & Summers, 2016). Applications range from segmentation of tissues (J. Lee et al., 2017) to disease detection (Ragab, Sharkas, Marshall, & Ren, 2019). In ophthalmology, studies have shown that applying deep learning for glaucoma detection with images of the eye has good performance (Ran et al., 2020). However, a large part of the literature focuses on distinguishing between healthy and glaucomatous eyes which, especially when trained on eyes with advanced glaucoma, has less added value because the most severe cases usually do not constitute a diagnostic dilemma for ophthalmologists (Medeiros, 2019). What composes a more challenging problem is the staging of glaucoma severity, as it is a progressive disease that can grow more severe over time, but the progression speed and rate varies widely among patients (Bengtsson & Heijl, 2008). Clinicians typically stage glaucoma based on visual field (VF) measurements, which indicate vision loss (European Glaucoma Society, 2020). However, these are time-consuming and not always reliable. Furthermore, glaucoma structural changes can occur before vision loss occurs, such that early detection of glaucoma relies on finding these changes in the eye structure (Sample, 2003). As such, modelling the relation between eye structure and severity of visual field loss can provide more insight on the disease and additionally aid in the treatment of glaucoma by giving an indication of the severity for a patient.

The majority of studies applying deep learning to retinal images use fundus photographs (photographs of the back of the eye) because they are easy to acquire and widely available (Ahn et al., 2018). However, different imaging modalities may have more potential to create insights into the progression of glaucoma. An important development in the field of ocular imaging is the optical coherence tomography (OCT). This is an imaging technique which uses low-coherence light to capture images up to 1 to 2 millimeters below the surface in biological tissue, providing a three-dimensional view (Huang et al., 1991). OCT has been vital for improving our ability to visualize both the front and back parts of the eye and it is now part of the standard-of-care in the management of glaucoma (Lai, 2020). With the recent introduction of optical coherence tomography angiography (OCTA), the OCT imaging devices can additionally be used to analyze the vascular health of the retina by showing the blood vessels and their flow (Fang et al., 2016). The majority of studies that deal with deep learning in glaucoma detection often focus on only one imaging modality. However, it has been shown that the combination of different images in one deep learning model can improve performance for estimating glaucoma progression (Yu et al., 2020).

Modalities of interest include fundus photographs, OCT scans, and, in particular, the newer OCTA modality. OCTA has not yet been widely explored, but has been shown to contain information that may be discriminative for glaucoma (De Jesus et al., 2020).

Therefore, the aim of this thesis is to develop and implement deep learning techniques to automatically predict the severity of glaucoma (as identified through visual field measurement) based on fundus, OCT, and OCTA data. We do this for a dataset of 614 subjects provided by the Leuven Eye Study (Abegão Pinto et al., 2016). Additionally, we aim to investigate the added value of the individual data modalities and to develop a model that combines the different data sources.

This thesis is structured as follows: in Section 2, some background knowledge is presented to understand glaucoma and how it is detected in clinical practice. Next, Section 3 explores related works that are relevant to the research problem at hand. In Section 4, the distribution of the dataset for each imaging modality is explained, as well as the preprocessing (quality criteria, data processing) that is applied before the data is used to train and evaluate our methods. In Section 5 the methods which will be used to classify and predict glaucoma are presented. In Section 6 the results are shown and discussed. Finally, a conclusion is given in Section 7.

2 Problem Background

In this section, we provide a background on glaucoma and the ways in which clinicians detect this condition. To do this, we first explain the anatomy of the (healthy) eye in Section 2.1. Next, the characteristics of glaucoma are presented in Section 2.2. Finally, we describe the techniques that can be used to visualize (glaucomatous) eyes in Section 2.3.

2.1 Anatomy of the Eye

There are three main layers to the eye: the fibrous layer, the vascular layer, and the retina. The fibrous layer is partly exposed on the outside of the eye. It consists of the transparent cornea, at the front of the eye, and the sclera, which is the white part of the eye. These structures mainly serve to provide shape to the eye. The sclera makes up the larger part of the fibrous layer and is connected to extraocular muscles, which act to control the movements of the eyeball. Some important nerves and vessels pass through the sclera into the eye, including the optic nerve. This can be seen in Figure 1, which presents a labelled diagram of the eye.

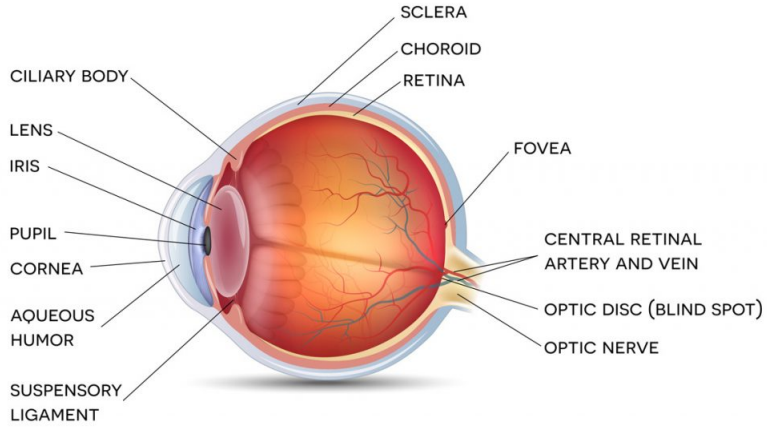


Figure 1. Labelled diagram depicting the anatomy of the human eye. Image obtained from (Kindsight, 2020).

Underneath the fibrous layer there is the vascular layer, which is the spongy middle layer between the sclera and the retina. The largest part of the vascular layer is the choroid, which is filled with blood vessels in order to nourish the outer layers of the retina. At the front of the eye, there are the ciliary body and the iris. The iris is the coloured, visible part of the eye in front of the lens, which can regulate the amount of light that enters the eye through the pupil. The ciliary body is a circular structure behind the iris that is attached to the lens. It has two functions: first of all, it can contract and relax, which can cause the lens to be less or more rounded. This way, the eye can focus at different distances. Secondly, it contributes to the formation of a fluid referred to as aqueous humour. This fluid fills the front of the eye and provides nutrition, as well as keeping the eye in a pressurized state.

Finally, the inner layer of the eye is the retina. This is a layer composed of light sensitive cells known as rods and cones, which help to see in dim and bright light, respectively. The highest concentration of cone cells is at the center of the retina, in the back of the eye. This region is also referred to as the macula, with a small indentation at the center that is referred to as the fovea. Due to the high concentration of cells at this center, the part of the image entering the eye that is focused on the fovea is seen most clearly.

Outside of the macula, there is a different area on the retina where the optic nerve enters. This is referred to as the optic disc and, since there are no light sensitive cells here, it is a blind spot. Within the optic disc, bundles of axons pass through a fibrous tissue called the lamina cribrosa. These axons are part of a type of neuron called retinal ganglion cells (RGCs), which allow the transmittance of electrical and chemical signals to other cells. In many eyes, the axons do not completely fill the disc. As a result, there is a 'neuroretinal rim' at the edges of the disc and a depression in its center called the optic cup.

2.2 Glaucoma Characteristics

Glaucoma is a group of eye conditions that damage the optic nerve, often (but not always) due to abnormally high pressure in the eye. The biological basis and causes of the disease are not completely understood yet, but all cases have in common that retinal ganglion cells and their axons degenerate over

time, resulting in changes to the optic disc and vision loss (Weinreb & Khaw, 2004). In most cases, this progression is very gradual, making changes in vision not very noticeable. Due to this, a large number of people with glaucoma remain undiagnosed (Quigley et al., 2001). However, without being treated, glaucoma can progress further, leading to visual disability and, eventually, even blindness. This is why early detection is very important.

Of the different types of the disease, primary open angle glaucoma (POAG) is the most common, particularly in populations of European or African ancestry (Weinreb & Khaw, 2004). In a healthy eye, aqueous humour flows from the ciliary body into the chamber in front of the iris, as was mentioned in Section 2.1. This fluid is drained out of the chamber through a spongy tissue, referred to as the trabecular meshwork, which is located at the sides where the cornea and the iris meet. In open-angle glaucoma, the angle between the cornea and the iris is still open, but the fluid cannot flow easily through through the trabecular meshwork. This causes a gradual increase in intraocular pressure (IOP), or pressure in the eye, resulting in damage to the optic nerve and vision loss.

On the other hand, angle-closure glaucoma (ACG), also called closed-angle glaucoma, happens when the iris comes forward and blocks or narrows the drainage angle between the cornea and iris. As a result, the fluid cannot circulate through the eye and pressure increases. It is possible for this condition to occur very suddenly (acute ACG), causing rapid and severe damage, or in short repeated intervals, causing damage more gradually (chronic ACG). Figure 2 illustrates the situation for the previously mentioned types of glaucoma in two diagrams.

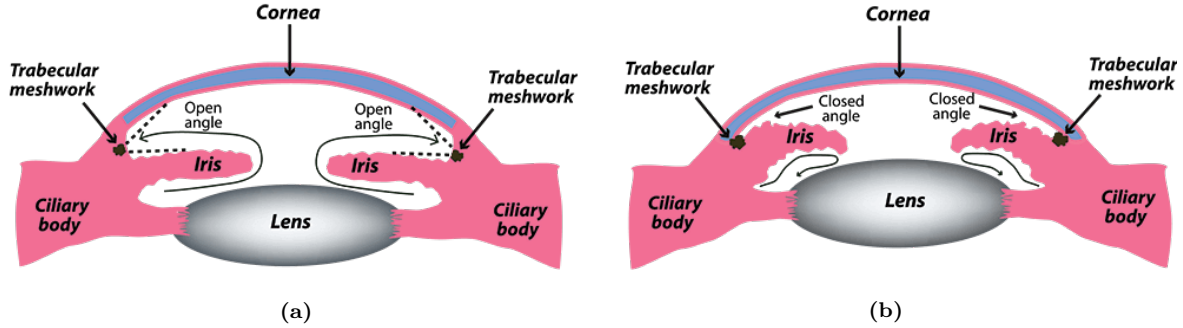


Figure 2. Diagram of (a) open-angle glaucoma and (b) closed-angle glaucoma. Illustration from (harvardeye, 2021).

In addition to open-angle glaucoma and closed-angle glaucoma, there is also a form of glaucoma where the optic nerve becomes damaged even though the intraocular pressure is within a normal range. This is referred to as normal-tension glaucoma (NTG). It is unknown what causes this variety of the disease, but there are some risk factors, such as family history, that are known to play a role (Trivli et al., 2019).

Although the exact pathological mechanisms of glaucoma onset and progression are not known completely (especially in the case of normal-tension glaucoma), all types of the disease share similar clinical features in changes to the eye structure and in vision loss (Ran et al., 2020). Treatment for glaucoma is usually aimed at reducing the intraocular pressure, for example with eye drops, laser treatment to open up blocked drainage tubes, or surgery.

Similarly, detection of glaucoma often starts by measuring the intraocular pressure during tonometry,

where a small amount of pressure is applied to the eye by a small device or by a warm puff of air, which allows the registration of pressure inside the eye. Another common diagnostic procedure is an ophthalmoscopy, which is an examination of the eye through a dilated pupil. After the pupil is dilated using eye drops, an instrument is used to light and magnify the optic disc such that its shape and colour can be observed.

If the intraocular pressure is above the normal range or the optic disc looks unusual, further diagnosis is often done through perimetry assessment of the visual field (VF). Standard automated perimetry (SAP), which uses a white target light on a white background at variable brightness, has been used for more than two decades in routine clinical practice to quantify the visual field (Quigley, 1993). Since the center of the macula is relatively resistant to glaucoma damage, it means that central vision often remains unchanged until the disease has progressed. Meanwhile, the peripheral vision is the most susceptible, leading to a kind of 'tunnel vision' during later stages of glaucoma (Weinreb & Khaw, 2004). Figure 3 shows results for a VF assessment for two stages of glaucoma, where the right pair of images corresponds to the most severe stage.

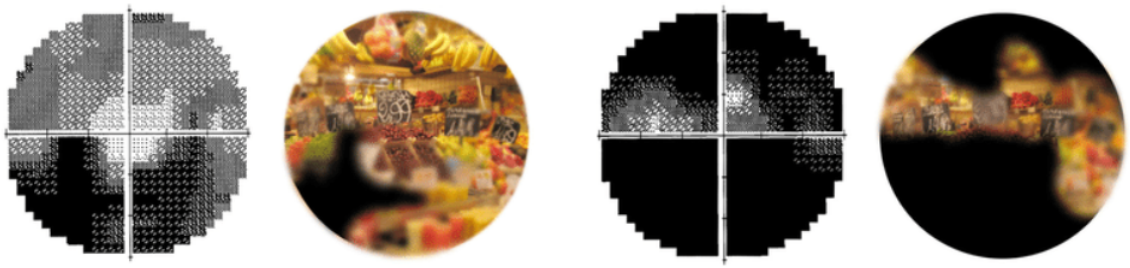


Figure 3. Visual field test result for two different stages of glaucoma. On the left for each pair a visual field greyscale, on the right a simulation of what patients see. Image pairs obtained from (Rulli et al., 2018).

2.3 Retinal Imaging Modalities

As mentioned in the previous section, glaucoma diagnosis is usually done using standard automated perimetry, giving an assessment of the visual function. However, detecting glaucoma at an early stage with this technique can be a challenge because there is a high degree of variability (Mikelberg et al., 1995). Moreover, the method is insensitive to loss of retinal ganglion cells, especially during early stages of the disease (Quigley, 1993). Often, even though a patient still has normal SAP results, changes are already present in the optic disc and nerve fiber layer (Sample, 2003). For this reason, the analysis of the optic disc and other relevant structures in the eye are of great importance, as it may be used to detect specific biomarkers that are already visible in early stages of glaucoma.

One imaging modality that is often used by ophthalmic specialists is fundus photography, which involves photographing the back part of the retina (the fundus), as depicted in Figure 4. Two key regions are usually visible on such an image, namely the macula, with the fovea at its center (seen as a darker spot), and the optic disc (seen as a brighter spot).

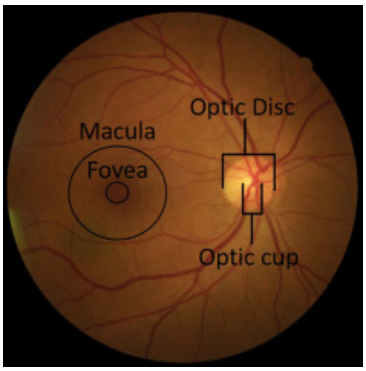


Figure 4. Fundus image, obtained from (Wigdahl et al., 2019).

The optic cup is of particular interest for glaucoma diagnosis. As nerve fibers die in patients with glaucoma, the neuroretinal rim of the optic disc tissue tends to become thinner and the cup becomes larger (Quigley, 1993). For this reason, the biomarker cup-to-disc ratio (CDR) has been used as a valuable measure for glaucoma screening and detection, as well as for patient monitoring after diagnosis (Almazroa, Burman, Raahemifar, & Lakshminarayanan, 2015). The CDR can be defined by the ratio between the area of the optic cup and the area of the optic disc:

$$CDR = \frac{A_{cup}}{A_{disc}}. \tag{2.1}$$

When the CDR is larger than 0.6, an eye is often determined as being a glaucoma suspect in clinical practice (Sevastopolsky, 2017). Figure 5 shows how the optic disc in a fundus image can differ between stages of glaucoma, with the optic cup being relatively larger and the neuroretinal rim relatively thinner in more severe stages.

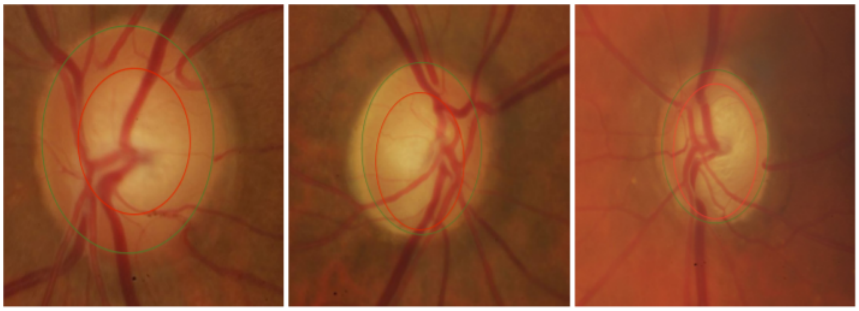


Figure 5. Healthy disc in comparison with two stages of glaucoma on fundus images. From left to right: Healthy disc, moderate glaucoma, and advanced glaucoma. Image obtained from (Norouzifard, 2020).

A different imaging modality that can be used to visualize the eye is optical coherence tomography (OCT). This technique is based on the principle of optical reflectometry, where light is back-scattered through transparent or semi-transparent media (for example, biological tissues). Different tissues will scatter the light in different ways and by measuring the intensity and the echo time delay of the light, an image can be created up to 1 to 2 millimeters below the surface of the eye (Duker, Waheed, & Goldman, 2013).

Compared to the two-dimensional fundus images, OCT not only provides the top view of the retina and optic nerve head, but also captures depth, providing a three-dimensional (3D) view (Ran et al., 2020).

The OCT 3D view is constructed as follows: a single OCT measurement gives a tissue depth-profile in a single point of the retina, called an A-scan. Multiple A-scans measured along a line over the retina surface can be put together to create a cross-sectional OCT image, referred to as a B-scan. Finally, a three-dimensional volume is acquired through an array of B-scans, side by side in a grid. For this three-dimensional volume, a single slice in depth is called a C-scan. By summing or averaging multiple C-scans, one can obtain a two-dimensional 'en face' image, also referred to as a projection (Braaf, 2015). Figure 6 illustrates the relation of these scans to each other.

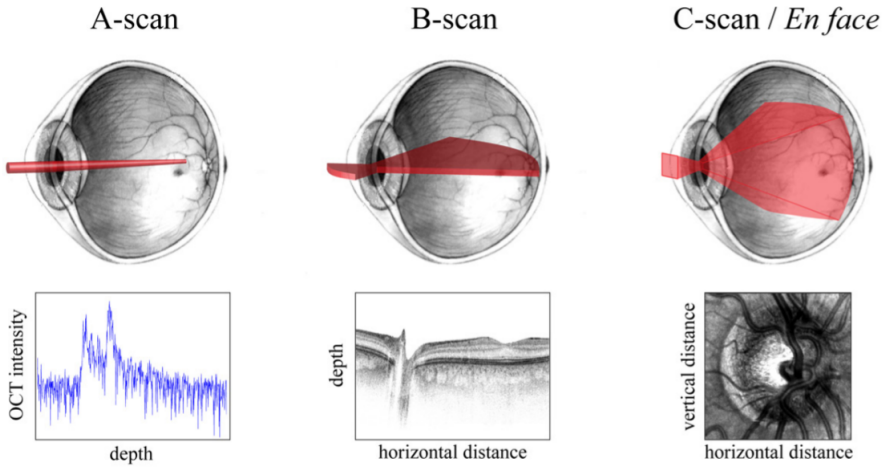


Figure 6. OCT A-, B-, and C-scans. Illustration from (Braaf, 2015).

One useful aspect of OCT scans is that they allow insight into different layers at the surface of the retina. Figure 7 shows a B-scan centered at the macula (the small indentation in the middle is the fovea), with clear layers being apparent.

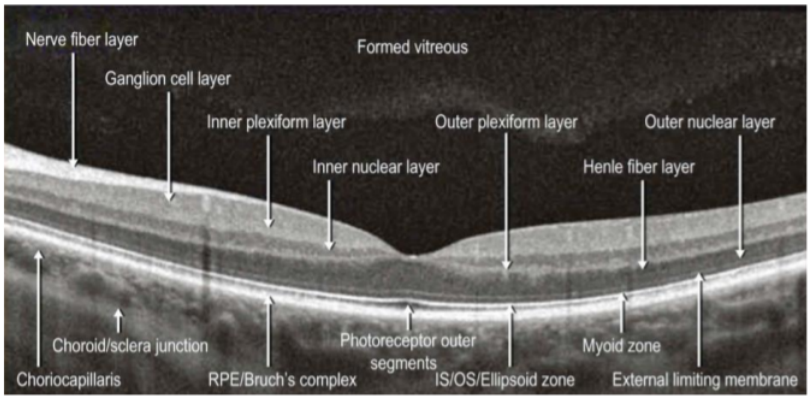


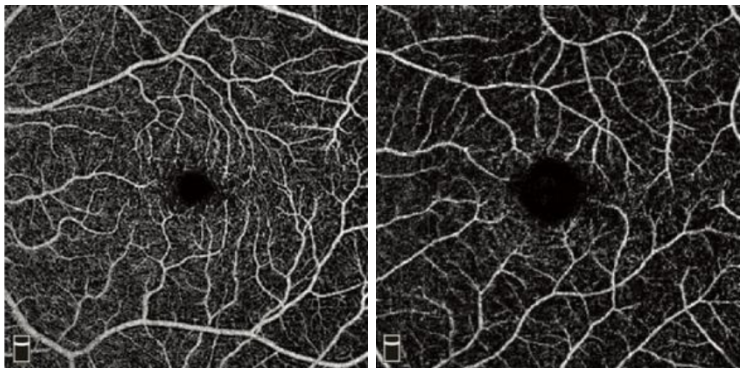
Figure 7. Macula-centered OCT B-scan with different layers labelled, from (Duker et al., 2013).

One of the most important layers in this image is the retinal nerve fiber layer (RNFL). As glaucoma damages this layer, measurable differences appear in RNFL thickness between normal and glaucomatous eyes (Bowd, Weinreb, Williams, & Zangwill, 2000). Similar to changes in the optic nerve head, changes in the retinal nerve fiber layer may precede the development of visual field loss, making these OCT scans useful for early detection of glaucoma (Bowd et al., 2001).

The final imaging modality discussed in this section is optical coherence tomography angiography

(OCTA). By making sequential OCT scans of the same area, it is possible to detect changes between the two different moments in time by looking at the differences in the intensity or phase of the back-scattered light. This way, OCTA can detect and illustrate movement in the structures in the back of the eye. In particular, most of the movement in these structures is caused by blood flow. Because the blood particles cause the most variation in back-scattered light, blood vessels become the brightest in OCTA images, while non-moving tissues are dark. Compared to other techniques that show the blood vessels, OCTA is non-invasive and acquires volumetric (3D) angiographic information without using dye (De Carlo, Romano, Waheed, & Duker, 2015).

Although the information in OCTA is 3D, it is common to compute a summation or 'projection' of the volumes at a certain depth in order to get a 2D representation, similar to the OCT C-scans. This is because 3D volumes are quite difficult to interpret for clinicians, due to the amount of noise in the volumes and to the inherent difficulty of inspecting 3D capillary information visually. Two projections of macula-centered OCTA scans are depicted in Figure 8.



(a) **(b)**

Figure 8. OCTA scans depicting superficial vessels of the (a) healthy eye and (b) glaucomatous eye. Images from (Fard & Ritch, 2020).

There is evidence that blood flow in the retina and optic nerve head is reduced in eyes affected by glaucoma (Flammer & Orgül, 1998; Grunwald, Riva, Stone, Keates, & Petrig, 1984), which can also be seen in Figure 8. For instance, averaging the decorrelation signal in OCT angiograms returns a 'flow index', which mainly measures the area and velocity of vessels. This index has been shown to have high sensitivity and specificity in differentiating glaucomatous eyes from normal eyes (Jia et al., 2014). Bekkers et al. (2020) also show that glaucoma vascular damage can be assessed using OCTA, where the added value for glaucoma diagnosis depends on the specific region being analyzed. Finally, Van Melkebeke, Barbosa-Breda, Huygens, and Stalmans (2018) provide an overview of studies using OCTA and show that it has good discriminatory power for differentiating normal eyes from glaucomatous eyes, also at early stages of glaucoma, and that it is more strongly correlated with visual function than conventional OCT.

3 Related Works

In this section, we aim to give a description of some of the literature that currently exists pertaining to glaucoma diagnosis. In Section 3.1, we first give a general overview of developments in machine learning (and, in particular, deep learning) techniques used to analyze medical images. Next, in Sections 3.2 and 3.3, we specifically look at how the OCT and OCTA data modalities, respectively, are used in the literature to diagnose glaucoma. Finally, in Section 3.4, we describe studies that combined two or more data modalities in their deep learning model.

3.1 Machine Learning in Medical Imaging

The use of machine learning has been increasing rapidly in the medical imaging field, including computer-aided diagnosis (CAD) and medical image analysis (Suzuki, 2017). Supervised learning, where the labels of the dataset are known, is frequently used to diagnose or predict disease outcomes. Unsupervised learning techniques, where training occurs without available information on the labels, can be applied to identify patterns of diseases (Caixinha & Nunes, 2017).

For image analysis in particular, the technique of convolutional neural networks (CNNs) has been shown to be especially effective. Deep convolutional neural networks usually consist of a series of layers of convolution filters, with a series of data reduction or pooling layers in between these convolutional layers. The input for a convolutional neural network normally is raw data, where the output is usually a classification (such as a stage of glaucoma). This end-to-end aspect is one of the advantages of CNNs, since it means that features do not need to be hand-crafted and reduced as they do in other machine learning approaches (Suzuki, 2017). Overall, it has been shown that CNNs are highly effective in object recognition and localization in natural images, which is why the technique is used more and more in medical image analysis (Greenspan et al., 2016).

When it comes to diagnosis and management of glaucoma specifically, deep learning approaches have been applied successfully (Mayro, Wang, Elze, & Pasquale, 2020; Thompson, Jammal, & Medeiros, 2020). CNNs have already been applied to glaucoma detection, mostly in the case of fundus images (Ahn et al., 2018). Jammal et al. (2020) have shown that, in the case of fundus imaging, a deep learning algorithm has the potential to outperform humans at detecting eyes with glaucomatous loss of visual field, meaning that deep learning could be used as a supplement or even as an alternative to human graders in glaucoma screening. Another frequent application with fundus images is for segmentation of the optic disc (Almazroa et al., 2015), allowing the calculation of useful metrics for glaucoma diagnosis.

Since the introduction of standard CNNs, alternative or extended architectures have been devised such as Deep Residual Learning for Image Recognition (ResNet). This architecture has been shown to be more powerful than simple CNNs, as it allows relatively fast training of very deep convolutional neural networks (He, Zhang, Ren, & Sun, 2016). Shibata et al. (2018) use ResNet for glaucoma screening with a dataset of around 3,000 fundus photographs, achieving a high diagnostic performance with an area under the curve (AUC) of 0.96. Similarly, a few studies have applied the architecture to assess glaucoma damage from OCT B-scans with success, achieving an AUC ranging from 0.92 to 0.96 for distinguishing

glaucoma from healthy patients (Thompson, Jammal, Berchuck, Mariottoni, & Medeiros, 2020).

One common problem in medical imaging data processing is the lack of annotated data. To tackle this issue, many models applied in medical imaging make use of transfer learning. This is an important technique where features learned to perform one task are applied to other tasks (Pan & Yang, 2009). This way, model knowledge can be transferred across tasks (related or unrelated) to solve a new task with minimal retraining. Especially when the target task has few high quality data points, transfer learning can improve performance while reducing the amount of data and training time needed. This is the case for most medical image classification tasks, where the scale of available data is much smaller in comparison with other tasks, such as general real world image classification, where there are very large datasets available (e.g. ImageNet, which consists of millions of labelled images (Deng et al., 2009)). For example, Christopher et al. (2018) use deep learning to detect glaucomatous eyes from fundus photographs. The authors show that, in all cases, transfer learning improved performance and reduced training time. Another study that applies transfer learning and uses several architectures for deep learning with glaucoma images is by Diaz-Pinto et al. (2019). They evaluate the ResNet50, InceptionV3 (Szegedy, Vanhoucke, Ioffe, Shlens, & Wojna, 2016), Xception (Chollet, 2017), and the VGG16 and VGG19 (Simonyan & Zisserman, 2014) architectures, all of them pre-trained on ImageNet. They fine-tune these architectures for the glaucoma assessment task by changing the last fully connected layer of each CNN for a global average pooling layer followed by a fully connected layer of two nodes (representing the glaucoma and healthy classes) and a softmax classifier. They find that the Xception architecture shows the best performance when it comes to trade-off between the area under the curve (AUC) score and the number of parameters of the CNN. Gómez-Valverde et al. (2019) also compare five different architectures for classifying glaucoma from fundus images and they find that VGG19 with transfer learning and fine-tuning has the best performance, with an AUC of 0.94.

Although many studies do make use of transfer learning, not all studies find that transfer learning works better. For example, the study by Mehta, Lee, Lee, Balazinska, and Rokem (2018) uses a model trained on the ImageNet dataset to build a multi-label classifier for OCT data and shows that transfer learning does not perform better in their case. One explanation they offer is that many of the low-level filters are tuned to colours, since ImageNet consists of colour images, while the OCT images are in greyscale. Therefore, although transfer learning can often increase performance, it should not be used blindly.

3.2 Automated Glaucoma Diagnosis with OCT

As explained in Section 2.3, optical coherence tomography scans provide a 3D view of the eye, allowing for calculation of different metrics, such as RNFL thickness, which have been shown to be structural indicators for glaucoma. The application of deep learning on OCT for glaucoma assessment has been shown to be efficient, accurate, and promising, with various studies using CNNs (Ran et al., 2020). However, it is less common in the literature to focus on detecting glaucoma severity with OCT scans. Nevertheless, it is a topic of rising interest and some studies acknowledge the added value of investigating (especially) earlier stages of glaucoma and how it progresses over severity levels (Medeiros, 2019).

For severity classification, one option is to perform multi-class classification, making each severity level a class in the model. However, most studies take the approach of directly estimating the visual field measurements from OCT data, as these (continuous) VF scores are the main factor used to stage glaucoma in clinical practice (European Glaucoma Society, 2020). Therefore the severity analysis is formulated as a regression problem, where the output of the model will be a continuous range of values. This way, these models aim to describe the 'structure-function' relationship in glaucoma, under the hypothesis that changes in structure could already be present before any changes in the visual function have occurred.

Hemelings et al. (2021) use an Xception21 architecture pre-trained on ImageNet to estimate a global VF index of mean deviation (MD), as well as VF values at individual locations. They show that using OCT data to predict pointwise and mean deviation VF measurements has a lot of potential and could be a solution for patients that are unable to reliably take an SAP test. For predicting VF MD, they reach an R^2 of 0.71 and mean squared error (MSE) of 11.10 on a dataset of 1600 4.7mm cicumpapillary OCT scans. The study of Christopher et al. (2019) reinforces this notion. They first use a binary classification model with spectral domain OCT (SDOCT) projections, in order to distinguish healthy and glaucomatous eyes, and then have a second model to predict quantitative VF measurements. They make use of a ResNet50 architecture with transfer learning from ImageNet and show that deep learning models outperform RNFL thickness measurements, currently used in clinical practice, in predicting VF MD.

While both of the studies mentioned use two dimensional data extracted from the OCT volumes, it is also possible to use the full volumes with 3D CNNs. Maetschke et al. (2019) propose a model which classifies eyes as healthy or glaucomatous directly from raw OCT volumes using a 3D convolutional neural network and show that the deep learning approach achieved better results compared to traditional machine learning techniques. Ran et al. (2019) employed a 3D deep learning model on a large dataset of OCT scans, showing that a 3D model with volumetric data significantly outperformed a 2D model with en face images as input in all the datasets.

3.3 Automated Glaucoma Diagnosis with OCTA

Although the application of deep learning methods to diagnose glaucoma from OCTA scans is very sparse in the literature, some studies do exist which apply CNNs to OCTA for feature segmentation (Hormel et al., 2021) and for diagnosis of other ophthalmic diseases, specifically diabetic retinopathy (Le et al., 2020). The latter study makes use of a VGG16 architecture in combination with transfer learning to deal with the fact that the sample size for OCTA data is generally more limited. Studies also exist which apply models for glaucoma prediction to features extracted from OCTA images, such as the prelaminar flow index, peripapillary vessel density, and RNFL thickness (Sarhan, Rokne, & Alhajj, 2019). These show that OCTA data indeed contains features which are informative towards glaucoma diagnosis.

De Jesus et al. (2020) show that OCTA has a lot of potential in the application of glaucoma classification, although the study does not focus on deep learning methods. The authors apply random forest, support vector machine, and extreme gradient boosting (XGBoost) techniques for three classification tasks, including the identification of glaucoma severity levels. To the best of our knowledge, Bowd et al. (2021) is the only study which applies deep learning to en face vessel density images for classifying

healthy and glaucomatous eyes. They compare results to gradient boosting machine learning analysis of measurements from OCTA data and find that the CNN has higher performance, achieving an AUC of 0.91.

Overall, we see that OCTA data is a relatively unexplored medium for detecting glaucoma progression and is, to our knowledge, not used with deep learning models to estimate glaucoma severity in any literature so far. However, studies show that it has added value towards glaucoma diagnosis and severity prediction and, as such, it may be a promising modality to incorporate in our models.

3.4 Multimodal Approaches

The majority of studies aiming to distinguish glaucoma patients use machine learning models that only deal with one type of images. This is different from how ophthalmologists will give a clinical diagnosis in practice, as they often use multiple information sources (i.e., multiple types of medical images) to make a decision. Only a few multimodal machine learning models have been reported for glaucoma diagnosis. There are, however, more examples of DL models applied to multiple imaging modalities in other ophthalmic diseases, such as the study by Yoo et al. (2019), who attempt multimodal categorization of age-related macular degeneration (AMD). The authors employ a VGG19 model pre-trained on ImageNet to extract visual features from fundus and OCT images. The model is trained for each imaging modality separately, and at the end the extracted features are concatenated and used as input for a random forest classifier. Yoo et al. (2019) show that the model using both OCT and fundus data outperforms the models using exclusively OCT or fundus images. Vaghefi, Hill, Kersten, and Squirrell (2020) also investigate a multimodal approach for the diagnosis of AMD. They use a CNN based on the Inception-ResNet-v2 design (Szegedy et al., 2016) and modify it to enable the network to be trained on multiple image modalities at the same time. In the end, they concatenate separate modalities using a global pooling layer. Combining OCT and OCTA data raised AMD diagnostic accuracy to 96%, compared to 94% and 91% accuracy on the individual models.

For glaucoma diagnosis specifically, An et al. (2019) built a machine learning classification model that combines the information of fundus and OCT data by adapting the VGG19 architecture. For combining the results from each CNN model, the authors remove the second fully connected layer to obtain feature vector representations of each input, similar to Yoo et al. (2019). Consequently, they use this to train a random forest to classify the eye into healthy or glaucomatous. Here, they show that the random forest combining separate CNN models improved performance to an AUC of 0.963.

One drawback of multimodal methods is the available data. Since not all the imaging modalities are acquired routinely in clinics for all of the subjects, there will be fewer patients who have all imaging modalities available compared to the sample sizes for the individual modalities. An interesting approach is the one by Wang et al. (2019) for multimodal AMD categorization. They introduce 'loose pairing', where images are grouped together based on labels instead of eyes. This means that, for the training data, they pair fundus images with OCT images if their labels are identical. This strategy expands the size of the training set quadratically, as images can be paired with multiple other images. They then combine the imaging modalities using a 'two-stream' CNN.

Another model that uses a multimodal approach and deals with smaller datasets (as is common in medical imaging) is the one by An, Akiba, Omodaka, Nakazawa, and Yokota (2021). The goal was to classify patients into glaucoma subtypes and, similar to Kang et al. (2021), they tackle this in a two-step approach which first classifies patients into glaucoma and healthy and then into subtypes. For their methodology, they use a VGG16 CNN architecture with transfer learning and extend the framework to handle multiple input images, such that it becomes a 'multi-input CNN'. In addition to this, they try a 'stacking ensemble method'. In this method, the confidence vectors calculated from separately trained single-input image models are extracted and concatenated to train a final (meta) model via a linear support vector machine (SVM). An et al. (2021) compare the multi-input CNN and the stacking ensemble method with separately trained single-input CNNs. They find that a multi-input CNN was not significantly better in performance than the single-input CNNs, but classification performance did significantly improve with the stacking ensemble method, resulting in a weighted accuracy of 83.9% using a dataset.

Finally, a study that is potentially the most relevant to this thesis is the one by Yu et al. (2020). They aim to estimate global VF indices in glaucoma using 3D OCT data from the macula and from the optic disc. For this purpose, they use two regional networks that analyze the macula and the optic nerve head separately. The outputs of these two networks are then concatenated after being pooled by global average pooling (GAP). They find that the combination of regions improves performance in advanced glaucoma in particular, suggesting that structural changes of the two regions have different courses over the spectrum of glaucoma severity. Their combined model reaches an R^2 of 0.76 for predicting VF MD.

Overall, the literature indicates that making use of multiple regions of interest and data modalities is certainly promising. Furthermore, from the studies described so far, we see that a common method to implement multimodal CNNs is by having separate models and pooling or concatenating them at the end. For the studies which specifically aim to detect glaucoma severity, we see that a two-step model is common, where first glaucomatous or healthy eyes are distinguished and then the severity is predicted. In this thesis, we combine these approaches and are, to the best of our knowledge, the first study to use a combination of fundus photographs together with OCT and OCTA data of both the macula and the optic disc for investigating glaucoma severity. The contributions of this thesis are as follows: first of all, we analyze the value of different regions of interest (optic disc and macula) and different imaging modalities (fundus, OCT, OCTA) for glaucoma diagnosis and prediction of visual field. Second of all, we propose multi-modal and multi-region approaches that use all the available data, combined in different manners, and study how the performance varies for glaucoma classification and visual field prediction, in comparison with individual imaging modalities.

4 Data

In this section, we discuss the dataset used to develop our models as well as the characteristics of the different data modalities. Section 4.1 presents an overview of the dataset and Section 4.2 provides details on the imaging modalities. Subsequently, Section 4.3 describes the data quality issues present in the dataset and, finally, Section 4.4 describes the data processing and augmentation techniques applied.

4.1 Dataset

The data has been provided by the Leuven Eye Study (Abegão Pinto et al., 2016). In this study, a total of 614 subjects (291 males and 395 females) were recruited between March and December 2013. Of these subjects, there are 214 subjects who have been diagnosed with primary open-angle glaucoma (POAG), 192 who have normal-tension glaucoma (NTG), and 41 who are glaucoma suspects. 27 subjects have ocular hypertension (OHT) and 140 subjects are healthy controls. As mentioned in Section 2, POAG and NTG share similar clinical features and it is theorized that they may even be part of the same continuum of open-angle glaucoma (Shields, 2008). Considering this factor and to increase the sample size of glaucoma patients, no distinction is made in this thesis between POAG and NTG glaucoma, such that they form a single group of 406 glaucoma patients. Furthermore, the glaucoma suspects and the ocular hypertension subjects are not used in this project to avoid any confounding factors and uncertainty.

In addition to the type of glaucoma, the subjects also have scores available for their visual field test. In particular, we look at the mean deviation, which compares the visual field sensitivity to the sensitivity of an 'average' patient of the same age. The lower the visual field mean deviation (VF MD), the worse the vision of the patient is. Based on the guidelines by the European Glaucoma Society, the glaucoma severity can be staged using these scores (European Glaucoma Society, 2020). Glaucoma subjects with a score above -6 are determined to have mild glaucoma. With a score between -6 and -12, the glaucoma is moderate. For scores of -12 and below, the disease is staged as severe glaucoma. Since one of the goals of this thesis is to predict these scores, subjects that have a glaucoma diagnosis but no associated visual field measurement are discarded. Figure 9 shows the distribution of the VF MD scores.

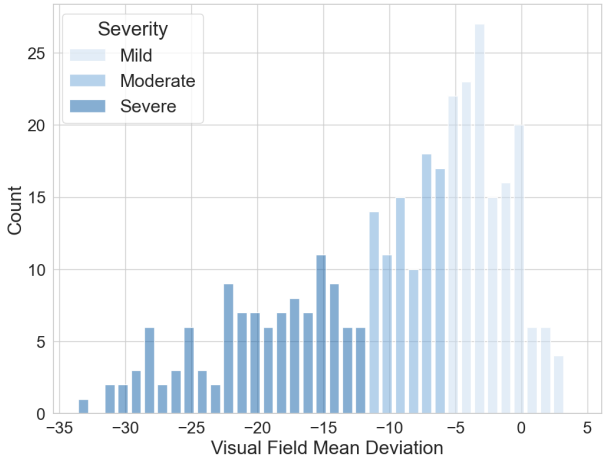


Figure 9. Distribution of visual field mean deviation scores for glaucoma patients in the dataset.

It can be seen that some patients have a mean deviation above zero (indicating better than average sight), even though they have a glaucoma diagnosis. For these patients, sight is only affected in local parts of the visual field because glaucoma is still in an early stage, such that they can still have an above average score for the visual field as a whole.

Table 1 illustrates the characteristics of the dataset after filtering out patients and dividing them into categories of healthy subjects and patients with mild, moderate, or severe glaucoma. In addition to age and gender distribution, it presents the percentage of POAG (as opposed to NTG), mean scores for visual field mean deviation, the maximum intraocular pressure measured (MaxIOP), and the RNFL thickness.

Table 1. Overview of dataset characteristics. Standard deviation in parentheses. POAG = primary open-angle glaucoma, VF MD = visual field mean deviation, IOP = intraocular pressure, RNFL = retinal nerve fiber layer.

	Healthy	Mild glaucoma	Moderate glaucoma	Severe Glaucoma	Overall
Number of subjects	95	146	79	103	423
Percentage male	53%	48%	43%	45%	47%
Percentage POAG	-	45%	53%	61%	53%
Mean age	66.18 (10.84)	68.26 (11.88)	69.54 (10.96)	72.92 (11.06)	69.18 (11.5)
Mean VF MD score	-	-2.23 (2.38)	-8.55 (1.78)	-20.25 (5.32)	-9.31 (8.4)
Mean MaxIOP	-	21.98 (6.73)	22.31 (5.93)	24.88 (6.28)	23.06 (6.59)
Mean RNFL thickness	0.21 (0.1)	0.19 (0.09)	0.14 (0.15)	0.1 (0.1)	0.16 (0.11)

4.2 Data Modalities

There are several imaging modalities available in the data from the Leuven Eye Study. For each subject, we have available fundus photographs centered at the optic disc, OCT volumes (one centred in the macula and other in the optic disc), and OCTA volumes (two volumes of two different sizes per region of interest, macula and optic disc).

The fundus photographs have dimensions of 1244 $\times$ 1420 pixels. The optic disc OCT volumes have dimensions of 200 $\times$ 200 $\times$ 1024 pixels. The macula OCT volumes have dimensions of 512 $\times$ 128 $\times$ 1024 pixels. There are OCTA scans available in two fields-of-view: 6$\times$6mm (dimensions 350$\times$350$\times$1024 pixels) and 3$\times$3 mm (dimensions 245$\times$245$\times$1024 pixels). The scans with a larger field of view show a larger portion of the retina, but have a lower density and resolution compared to 3$\times$3mm scans, because the number of cross-sectional OCT scans is limited by the scanning speed of the instrument. More importantly, they are more difficult to acquire due to the scans taking longer to complete, which means that patients have to concentrate and keep their eye fixed in a specific position for a longer period of time. For this reason, we opt to use the 3$\times$3mm scans.

To illustrate the different imaging modalities, Figure 10 shows the three imaging types, all centered at the optic disc.

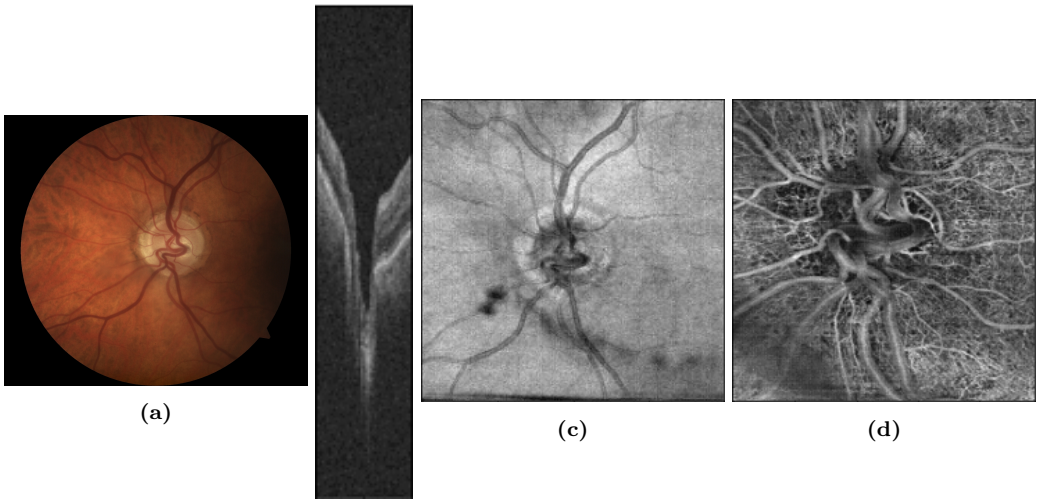


Figure 10. Data for right eye of a subject from the Leuven Eye Study (Abegão Pinto et al., 2016). All images are centered at the optic disc. Given: (a) fundus image, (b) OCT B-scan, (c) projection of the OCT scan, (d) projection of OCT angiography.

For each subject, only one image from one eye is used for a given data modality and region, although there can be multiple scans available. When a subject has data from multiple check-ups at different moments in time, a scan belonging to the earlier check-up is selected. This choice is made because glaucoma will always be in an earlier stage at earlier check-ups and, since diagnosing glaucoma at later stages is easier for clinicians, these images from earlier stages are expected to be more valuable to training the models. In many cases, the scans were not made on the exact same date as the check-up dates for the patient (when visual field scores are measured). An overview of how often a difference occurs and how many days is presented in Appendix A.1. In this project, scans are only used if they have been made within a window of 150 days from the check-up date. For a larger time difference, we cannot assume that the disease has not progressed further, changing the structure of the eye and, therefore, rendering the VF measurements outdated.

After applying the previously mentioned exclusion criteria, there are 222 subjects who have optic disc centered OCT scans (OCT D) and 231 who have macula centered OCT scans (OCT M). 165 subjects have 3×3mm optic disc centered OCTA scans (OCTA 3×3 D) and 153 have 3×3mm macula centered OCTA scans (OCTA 3×3 M). Finally, 416 subjects have fundus images available. Figure 11 shows the sample sizes for each imaging modality and region, with the glaucoma subjects divided into mild, moderate, and severe stages based on their visual field score.

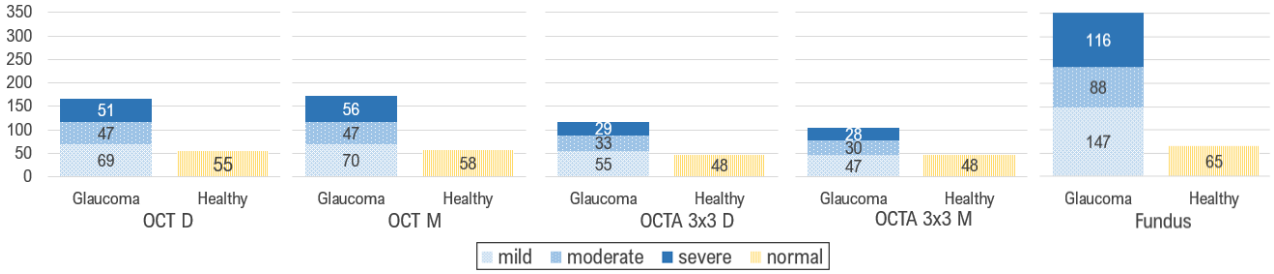


Figure 11. Overview of sample sizes for different data modalities and regions. OCT = Optical Coherence Tomography scans, OCTA 3x3 = OCT Angiography 3x3 mm scans, D = optic disc region, M = macula region.

When it comes to combinations of imaging modalities, 127 subjects (42 healthy and 85 glaucoma) have all of the above mentioned imaging modalities (and, for the case of OCT and OCTA, in both macula and optic disc) available from the same moment in time. Figure 12 shows these numbers in a graph.

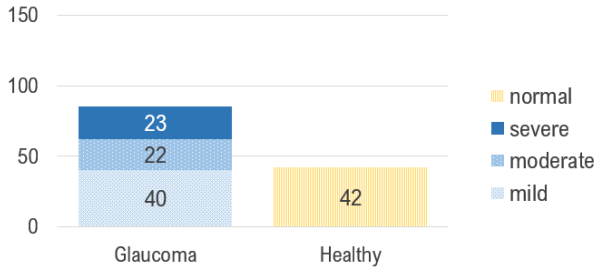


Figure 12. Overview of sample sizes for patients with all modality-region combinations.

4.3 Image Quality Issues and Selection

Now that a picture has been painted of the available data, we dive deeper into the limitations of the dataset and the techniques that are applied in order to deal with these limitations. Throughout the previous sections, we explained that OCT and OCTA are promising techniques that allow one to obtain deeper insights compared to normal fundus photographs. However, it is important to note that the scans are more difficult to acquire and, as a result, there are several types of artifacts that are common in these scans (Kraus et al., 2012). Patients must stay still and focus their sight on one point for the duration of the scan, which is especially difficult for older patients and patients with vision loss. If a patient moves their head, blinks, or even breathes, artifacts can occur. The axis in the 3D volumes that is most affected by this is the cross-section over sequential B-scans, also referred to as the 'slow axis'. Here, misalignments are created if the patient moves between the B-scans, causing them to be recorded at different locations. Figure 13 illustrates more clearly which axis is most prone to misalignments for a macula OCT volume example.

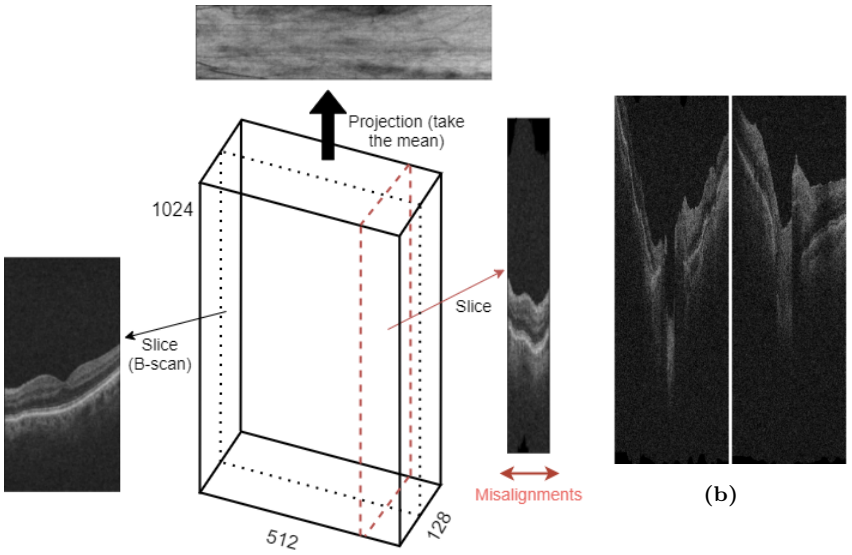


Figure 13. (a) Diagram to illustrate the axes of the 3D volume for a macula centered OCT scan, including the slow axis with misalignment. (b) Examples of the slow axis in OCT optic disc scans from two different eyes.

Correcting the misalignments is not an easy task. OCT is inherently noisy, which negatively affects the intensity-based motion correction approaches. Moreover, the OCT B-scans are not really images at the exact same position, and it is very difficult to correct the motion but keep the natural shape of the eye (the curvature of the retina). Note that, in the images included in our dataset, the top and bottom of the slow axis images have completely black regions that do not show the speckle noise characteristic of OCT. This is because the scanning device has already tried aligning the different B-scans, but not always successfully. In consideration of the misalignments, we do not use a 3D volume model for this thesis, but instead focus on the 2D B-scans and projections. This means that we miss out on volumetric information, but at the same time there is less risk of distorted images (the information is of higher quality) and it allows for simpler models.

For a full overview, Figure 14 shows more examples of what the 2D scans (B-scans and projections) look like for both the OCT and OCTA modalities on the macula and optic disc regions.

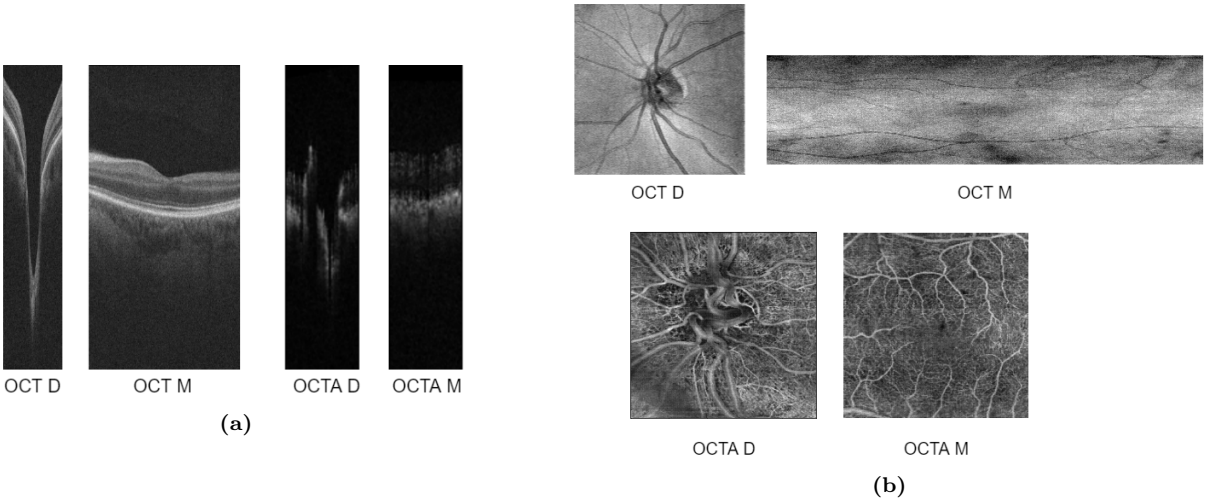


Figure 14. (a) B-scans and (b) projections of the macula and optic disc regions OCT and OCTA volumes

Although artifacts are more common on the slow axis, some motion artifacts can also show up in other views of the volume, such as in the projections. For example, eye movements (also referred to as 'saccades') will cause lines or shifts in the projections. In the OCTA scans, these lines are white because more movement will make the image lighter, as explained in Section 2.3. Meanwhile, blinks will show up as large black bars in both the OCT and the OCTA projections, because the signal is blocked and so there is no information (or movement) at all being registered by the scans. Figure 15 shows what these artifacts can look like.

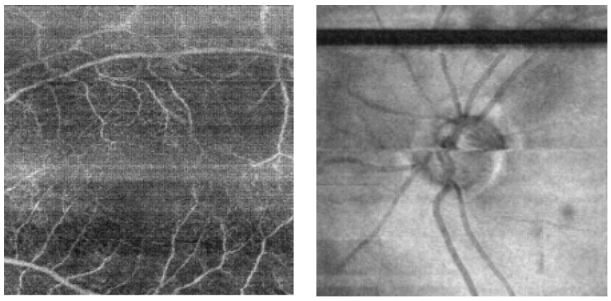


Figure 15. OCTA and OCT projections with artifacts

The OCT projection in Figure 15 shows a clear saccade, causing a misalignment right in the middle of the optic disc. It is, to some extent, possible to detect saccades in the OCT scans by looking at the correlation between B-scans (or rows in the coronal projection). Lezama et al. (2016) provide a calculation for an image I of H pixels high and W wide. They propose to filter each row of I with a one-dimensional Gaussian filter with $\sigma = 6$. The negative of the correlation between every two adjacent rows, $r_{i,i+1}$, can then be computed as

$$r_{i,i+1} = -\frac{\sum_{j=1}^{W} \left(\frac{I(j,i)-m_i}{s_i}\right) \cdot \left(\frac{I(j,i+1)-m_{i+1}}{s_{i+1}}\right)}{W}, \tag{4.1}$$

where m_i, m_{i+1} and s_i, s_{i+1} are the mean and standard deviation of rows i and $i+1$, respectively. The higher the value of $r_{i,i+1}$, the more dissimilar the two adjacent rows i and $i+1$ are. Next, the $H-1$ length vector r of the correlations between each adjacent row is subtracted from a locally averaged version of itself, so that small changes in correlation are taken out:

$$r = r - \left(\frac{r * \mathbb{1}_L}{L}\right), \tag{4.2}$$

where $*$ denotes convolution and $\mathbb{1}_L$ is a unit vector of length L (Lezama et al. (2016) use $L = 10$). Then, the rows of r which are above a certain threshold r_{min} are considered to be saccades. We try different values for r_{min}, as well as considering weighting the negative values at the center of the image more heavily than at the top (because saccades at the center are more disruptive). However, comparing the results of the saccade calculation with visual inspection, we see that in our application the method does not work as well as in the paper of Lezama et al. (2016), likely due to the noise in our images.

Computing blinks is more straightforward, as we can simply calculate the mean pixel value of each row and see whether it is too 'dark' (indicating a fully black row). In the OCT data, there are quite a

few projections with blinks (26 in the macula region and 5 in the optic disc region). In addition to that, there are many with disruptive misalignments due to saccades. Since the fundus images also provide en face information for the optic disc region and the macula projections have many artifacts, we decide for this thesis not to use the OCT projections and instead use only the B-scans. Meanwhile, for the OCTA volumes, we use only the projections and not the B-scans, because the OCTA B-scans are not very informative, as can be seen in Figure 14. That is, the difference between individual OCTA B-scans is not as interpretable as the projection or C-scan view.

The OCT B-scans are quite noisy, which can complicate the training of the model. One way in which this noise can be reduced is by taking the average over multiple B-scans (the central, selected B-scan, and a set of neighbouring B-scans around it) instead of taking only one cross-section. Figure 16 shows how the appearance of the resulting images vary after averaging different numbers of 'slices' (B-scans).

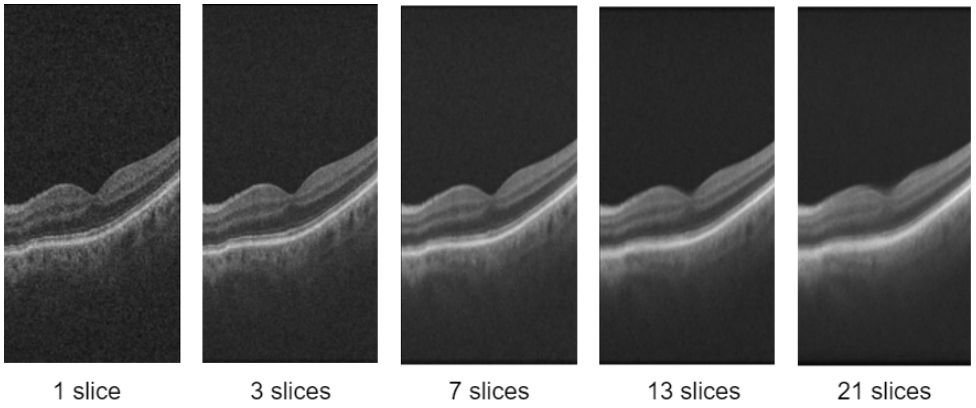


Figure 16. Result of averaging different numbers of B-scans. The reference/central B-scan is the center of a macula OCT volume.

It can be seen that averaging over more slices clearly reduces the speckle noise in the image but, at the same time, it causes the edges of the layers to be softened and less distinct. Furthermore, there is the issue of misalignments to be considered, which could potentially be located right between the images over which we average. For that reason, we opt to average over three slices (the B-scan at the center and one on each side of it) in order to reduce the noise while keeping sharp boundaries between layers and having lower risk of averaging over a misalignment.

The last challenge to mention under the topic of data quality is the task of selecting an image when there are multiple scans available (same eye, region, modality, and from the same date). In that case, we aim to select the image that has the least noise. We have explored several methods to estimate the noisiness of images, but in the end we use the method described by Mittal, Moorthy, and Bovik (2012), which is the Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE). We find that it works relatively well at selecting between images even with subtle differences in image quality, including noisiness, as can be seen in Figure 17.

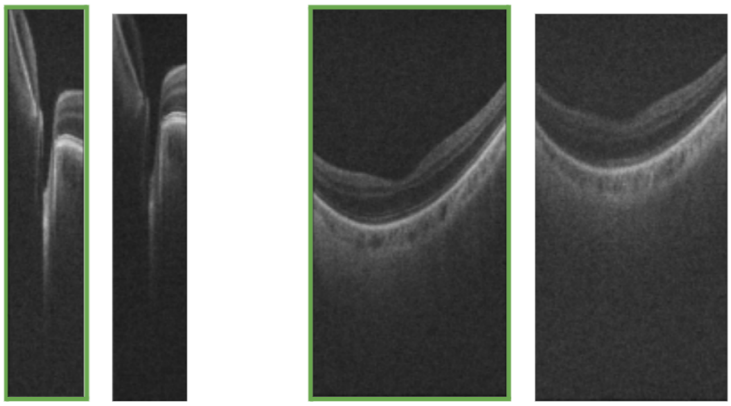


Figure 17. Pairs of B-scans from the optic disc and the macula, with slight differences in image quality, as measured with BRISQUE. The left (bordered) images for each pair are selected as highest image quality.

A higher BRISQUE score indicates worse image quality. Considering what was mentioned earlier in this section about how patients need to keep their fixation on one point for the scans, which is more difficult for patients with bad eyesight, it is interesting to see if there is a correlation between image quality and glaucoma severity. We use the Pearson correlation coefficient (Benesty, Chen, Huang, & Cohen, 2009) between the BRISQUE scores and the visual field scores for each modality and region. While the OCTA and fundus data do not show significant correlation (at a 5% significance level), we see that the OCT B-scans do have a significant negative correlation at a 5% significance level of -0.225 (p-value: 0.001) and -0.147 (p-value: 0.024) for the optic disc and macula regions, respectively. A negative correlation implies here that a lower VF score (and thus more severe glaucoma) correlates with a higher BRISQUE score (and thus worse image quality). This is something to take into account, as it may cause a model to learn that noise is associated with higher severity. In general, biases in the images such as this one can be addressed with data augmentation. In this case of poor image quality, random noise could be added to all images, in order to reduce the bias of particular images having more noise.

4.4 Data Processing and Augmentation

As explained in Section 4.3, we use projections for the OCTA data and B-scans for the OCT data. In this subsection, we go into more detail regarding processing and augmenting the data.

The first form of processing is horizontal flipping of the right eye. This is because the orientation of structures in the right eye is horizontally mirrored as compared to the left eye. Certain biomarkers for glaucoma will show more in specific sides of the eye (Jonas & Budde, 2002), which means that it is better if all images are in the same orientation. Next to horizontal flipping, the fundus images (such as the one shown in Figure 10a) are cropped by 250 pixels on each side in order to remove the black space on the outsides and leave more relevant information. Cropping has the additional advantage of speeding up computation, as the images are smaller from the beginning.

Additionally, we use measures to enhance the 'readability' of the images. This is mainly done by contrast enhancement, which has the goal to spread the pixel values of an image over a wider range by making light values lighter and dark values darker. For images in greyscale, this range is normally $0 - 255$. The specific method used is contrast limited adaptive histogram equalization (CLAHE) (Pizer

et al., 1987). Here, 'adaptive' refers to the fact that CLAHE changes the pixel distributions within local neighbourhoods rather than transforming the whole image in the same way, which helps when certain parts of the image are much darker or have much less contrast than others. The size of the regions is defined by a specified grid size. Furthermore, 'limited' refers to the fact that CLAHE uses a clip limit which limits the changes applied to the pixel values. For fundus images, which are colour images and, therefore, have three channels (red, green and blue), CLAHE is applied to each channel individually. We determine the clip limit and window size for each modality separately and empirically, selecting the parameters that visually improve the images the most (clearest structures without excessive contrast).

Furthermore, on the OCT B-scans and on the fundus images, we also apply some blurring in order to reduce the effect of the noise. This is done by median blurring, which makes each pixel value the median of the region of the pixels around it (Arce & McLoughlin, 1987). Increasing the size of this region will lead to a smoother and less noisy image, but, at the same time, it will make edges and details less distinct. We use 3×3 regions to introduce a small amount of blurring. We do not apply blurring to the OCTA projections to avoid losing the detail of the vessel edges. Figure 18 shows the exact augmentation parameters for each modality as well as examples.

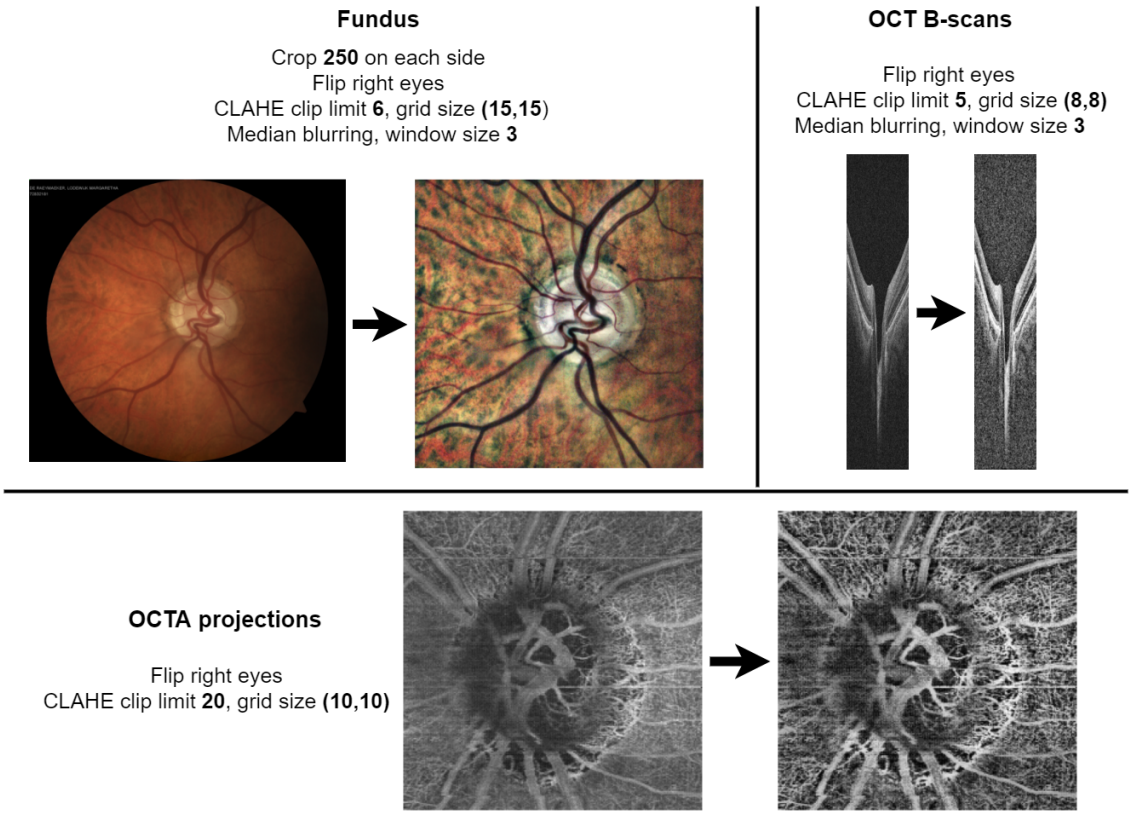


Figure 18. Data processing and augmentation settings for each data modality.

5 Methodology

We now present the methodology used to create and evaluate the models for glaucoma classification and visual field prediction. In Section 5.1, we give a short overview of the two types of models trained in this thesis. Section 5.2 dives deeper into the deep learning methodology for the single-input models, detailing the network architectures and other principles of convolutional models. Section 5.3 describes the methods used for training with multiple inputs. In Section 5.4 we explain the implementation of our models with data augmentation and hyperparameter tuning. Finally, we describe the techniques, metrics, and statistical tests applied for evaluation in Section 5.5.

5.1 Overview of Models

The aim of this thesis is to predict visual field mean deviation scores (as a metric for glaucoma severity) based on multiple input modalities and regions. However, for most patients who are not diagnosed with glaucoma, there is no visual field score available in our dataset. This is because it is not a measurement routinely performed in clinical practice, only when a patient has (or is suspected to have) glaucoma. For that reason, we need to use a two-step model, where we first aim to classify if a patient has glaucoma or not and, after this, we predict the visual field scores only for the glaucoma patients. These two models are trained separately, with the classification model having the full dataset as input and the VF model having only the glaucoma patients to train on. For both of the goals, we implement and compare various architectures. In Sections 5.1.1 and 5.1.2, we give a short overview of the different models that this thesis aims to evaluate.

5.1.1 Overview Single-input Models

One goal of this thesis is to compare how each modality and region performs individually for the goals of glaucoma classification and visual field prediction. Therefore, the data from each image input type is used to train a separate CNN. In order to improve our results, we try different sets of parameters and architectures. However, to reduce the number of models that we need to train and evaluate, we use the same parameter and architecture settings per modality. As such, for the B-scans, we only perform parameter tuning on the optic disc region dataset, and use the resulting optimal model settings to also train a CNN for the macula region. The same is applied to the OCTA data. In the end, for both the classification and the VF prediction, there will be results from each modality-region combination. A diagram to give a general overview of the models trained is presented in Figure 19.

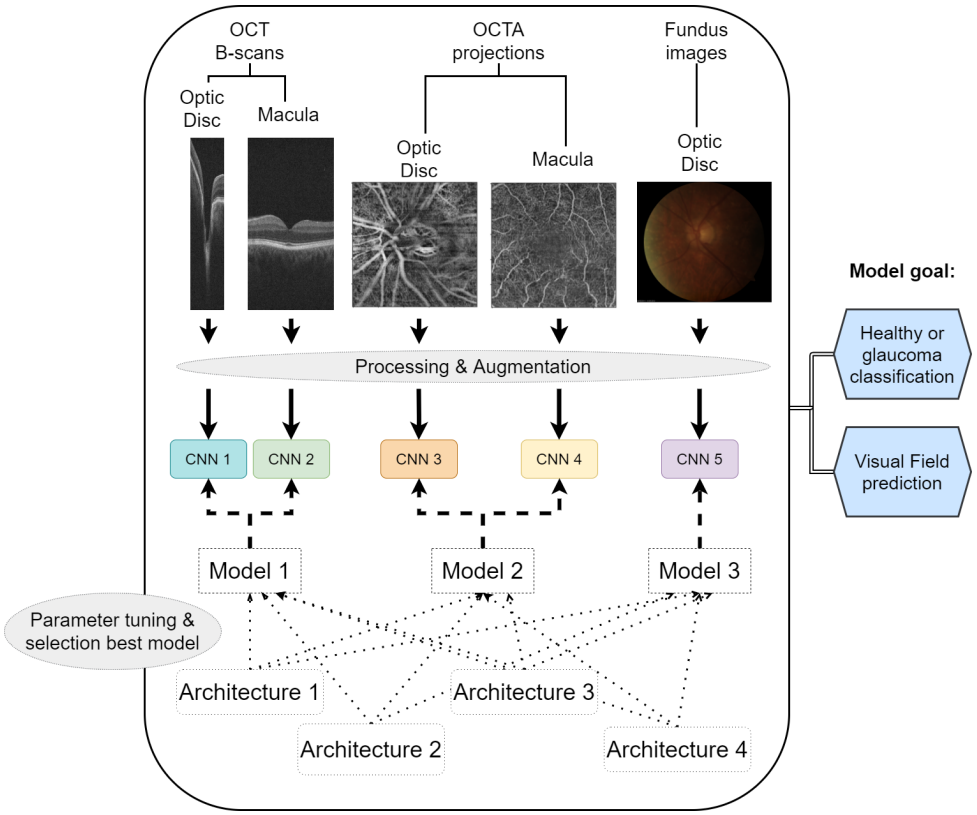


Figure 19. Overview of single-input models trained. $Architecture_i$ for i ∈ [1..4] represents each of the tested architectures; $Model_j$ for j ∈ [1..3] represents each of the model configurations that are obtained with parameter tuning for each modality (Optic disc OCT B-scans, optic disc OCTA projections, and fundus); and CNN_k for k ∈ [1..5] represents each final model (trained in each existing dataset, separating modality and, when applicable, region). The same steps are repeated for both goals (classification and visual field prediction) separately.

5.1.2 Overview Input Combination Models

In addition to training models on each modality and region individually, we also aim to combine the modalities and evaluate if this improves performance. For this purpose, we try two approaches. The first is to use a multi-input CNN. Here, all inputs are used in a single model, with different branches that are eventually combined to form a single output. Therefore, the weights for each input are updated at the same time and evaluated based on the same loss function. This means that the whole training process is truly multimodal, because the images are used together. Similarly to the single-modality models, different settings and architectures can be tried in order to improve the performance. However, this approach is not as flexible compared to training individual CNNs for each modality, as some of the settings will need to be the same for all the modalities. Furthermore, a disadvantage of the approach is that only patients who have data for each data type can be used, which means that the dataset is, by definition, smaller than for the individual models.

The second approach to combine multi-modal data is to use a stacked ensemble method. In this approach, separate CNNs are trained individually, one for each modality and region. After they have finished training, the final output layers can be removed so that it is possible to get vectors of the features that each CNN has extracted from the images. Next, these features can be combined with a different model in order to obtain a classification or a prediction. The advantage of this approach is that we can

use the optimal settings from the individual models again to train these individual CNNs. Moreover, to train the individual CNNs, we do not need all input types available at the same time for a given subject. The model used to combine the individual feature vectors can vary widely (e.g. another CNN, a machine learning classifier, a statistical formulation), and we also try different settings and methods. Figure 20 shows an overview of the methods used to combine multiple inputs into one model.

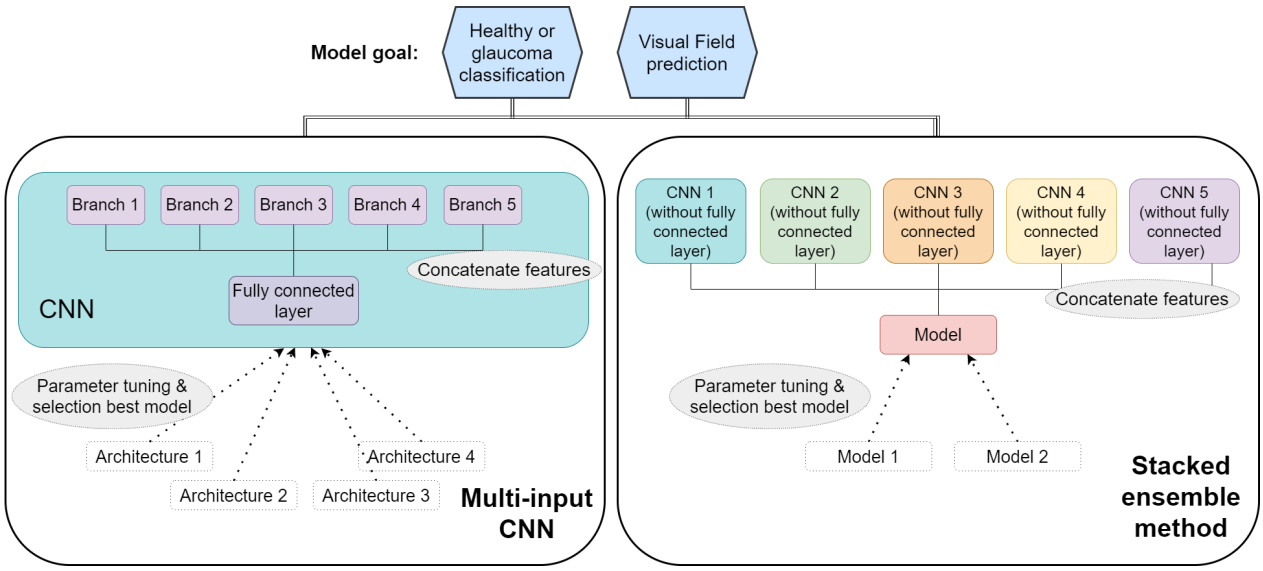


Figure 20. Overview of multi-input methodology. For the multi-input CNN, $Architecture_i$ for i $\in$ [1..4] represents each of the tested architectures; $Branch_j$ for j $\in$ [1..5] represents each branch of the CNN with its own modality-region input. For the stacked ensemble model, CNN_k for k $\in$ [1..5] represents each final single-input model trained on each modality-region dataset, without the last fully connected layer; $Model_l$ for l $\in$ [1..2] represents each of the tested models that take as input the concatenated features from CNN 1 to 5. The same steps are repeated for both goals (classification and visual field prediction) separately.

5.2 Deep Learning Methodology

In this section, we provide background information to the deep learning methods used in this thesis. In Section 5.2.1, we explain the basic convolutional neural network that we use in our classification and prediction models. Besides a more basic CNN architecture, we also apply three different state-of-the-art architectures that have been shown to perform well in the literature for similar medical imaging classification and regression tasks: VGG16 (Simonyan & Zisserman, 2014), Xception (Chollet, 2017), and MobileNetV2 (Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018). These architectures are explained in Sections 5.2.1.2, 5.2.1.3, and 5.2.1.4, respectively. In Section 5.2.2, we further explain how these three state-of-the-art architectures are implemented with transfer learning.

5.2.1 Convolutional Neural Networks

Convolutional neural networks are a type of deep learning model that can be used to process data in a grid or array pattern, such as images, which can be represented as numbers for each pixel value. The input layer for a two-dimensional CNN simply contains the pixel values of the image, given by a tensor I of size $H \times W \times c$, where H and W are the height and width of the image, respectively, and c is the number of channels. Fundus images are colour images and, hence, have three channels (red, green, and

blue), while the greyscale OCT and OCTA images have only one channel. Between the input and the output layer, CNNs have a number of intermediate layers, which typically belong to one of three different types: convolutional layers, pooling layers, and fully connected layers. A CNN architecture can be created by stacking these layers (O'Shea & Nash, 2015). The first two types, convolution and pooling, perform automatic feature extraction, while a fully-connected layer is commonly applied to map the extracted features into a final output, such as a classification (Yamashita, Nishio, Do, & Togashi, 2018).

We now explain the functionality of the CNN per layer. The first layer type, a convolutional layer, is illustrated in Figure 21. This layer has a filter which consists of weights and a bias term. A single filter has a 'kernel' (meaning a grid of weights) for each channel of the image. These kernels are matrices of numbers which are to be multiplied with a patch (the group of adjacent pixels which are covered by the filter at a given position) to get a new value. The filter starts at the top left of the image to select such a patch and compute a new value by matrix multiplication. Next, the filter slides to the right with a certain stride value s and continues to compute values until it completes the whole width. As it moves on, it goes down and back to the left of the image with the same stride value and repeats the process until the entire image has been parsed and an 'activation map' is produced.

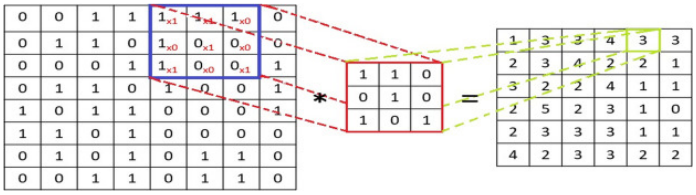


Figure 21. An example of a convolutional layer, image obtained from (Singh et al., 2020).

During training, the CNN learns kernels that 'fire' when they come across a specific feature in the image. These are commonly known as activations. Multiple filters which detect different features can be convolved on the input image to give a set of activation maps. For example, one filter may detect horizontal edges while another filter may detect vertical edges. In this way, each of the f filters can extract unique features from each patch that the filters slide across. In the cases when filters are larger than 1×1, padding can be used in order to deal with the border pixels of the image. Padding is when a number of pixels p are added which to each border of the image. These added pixel values will be convolved upon when the filter is at the borders of the image, for example in order to simply retain the original dimensions of the image (i.e., 'zero-padding'). An illustration of this is given in Figure 22.

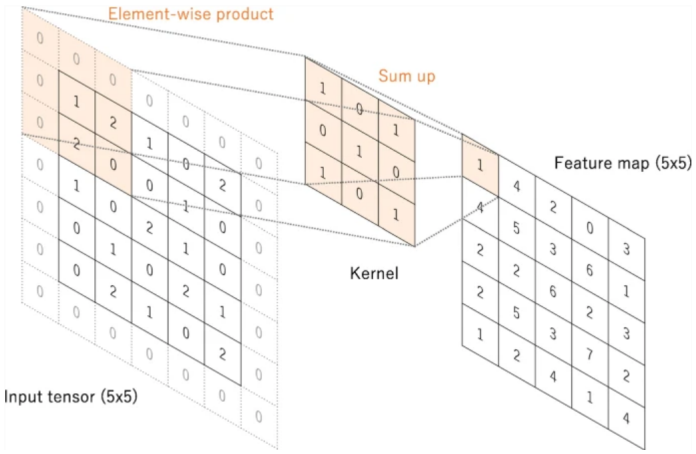


Figure 22. A convolution operation with kernel size 3×3, stride 1, and zero-padding in order to retain the original dimensions. Image from (O'Shea & Nash, 2015).

The stride value s, number of filters f, as well as the size k of the kernel(s) of the filter can all be adjusted. Increasing the stride value s means that the filter skips more pixels as it slides across the image, so that the number of operations decreases and the output activation map or 'feature map' is of a smaller size. Increasing the number of filters f will increase the number of features calculated, which allows the network to learn more relationships in the data, but at the same time increases computation time and complexity, as it may lead to the network having too many parameters.

Stacking multiple convolutional layers normally means that progressively more complex features can be selected from the input, as one layer sends its output into the next layer. For example, the first convolutional layer may detect edges, and the next convolutional layer could detect simple shapes using these edges. The network is said to be more 'wide' when more filters are added, whereas adding more layers makes it more 'deep'.

After applying a convolution layer, the outputs are usually passed through an activation function, which is there so that the network can learn nonlinear relations (otherwise, there is only multiplication and addition of the bias term). The most frequently used nonlinear activation function is the rectified linear unit (ReLU), which is given by

$$ReLU(x) = \max(0, x). \tag{5.1}$$

This means that any value below 0 is removed.

After the convolutional layers, a pooling layer can be used in order to reduce the number of parameters and computation in the network. This works in a similar way to convolution, but there are no trainable parameters. An example is max-pooling, which simply takes the maximum value from the patch that the filter is on. This operation is illustrated in Figure 23.

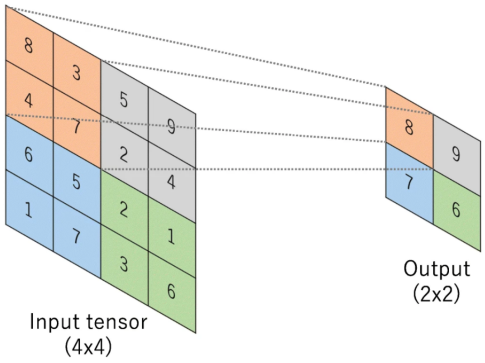


Figure 23. An illustration of a max pooling layer with a 2×2 filter and a stride of 2. Image from O'Shea and Nash (2015).

In the final layer type, the fully connected layer, an activation other than ReLU is used, which depends on the target problem. For classification, a function that is often used is the softmax function (also referred to as the multinomial logistic function). Its output is between 0 and 1, and can be interpreted as the relative probabilities of each of the output classes. The softmax function is computed by

$$\text{softmax}(u_{Lc}) = \frac{e^{u_{Lc}}}{\sum_{g=1}^{C} e^{u_{Lg}}}, \quad c = 1, \ldots, C, \tag{5.2}$$

where L refers to the final layer in the network and C is the number of target labels to classify. In our application, the number of classes in the classification model C has value 2 (healthy and glaucoma), such that the function becomes equivalent to the sigmoid activation function. For our visual field prediction models, the activation function used in the final layer is simply a linear activation, as the visual field values are continuous and not necessarily truncated to a specific range (in our dataset, which is representative of clinical practice, the values range between 2 and -30).

During training, the network trainable parameters are optimized in order to minimize the difference between outputs and the true label values. To guide this training, a certain loss function is used to describe the error of the output, which needs to be minimized. For classification, a common loss function is the cross-entropy loss function:

$$CE = -\sum_{i}^{C} t_i \log(s_i), \tag{5.3}$$

where t_i is the true value and s_i is the CNN output (e.g. the output from the sigmoid function) for class i out of C classes.

For linear activations, such as our visual field prediction model, the most common loss function is the mean square error (MSE), given by:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (t_i - s_i)^2, \tag{5.4}$$

where t_i is the true value and s_i is the CNN output. Minimizing the MSE means that the predicted values are closer to the true values.

The minimization of the loss function is done with a given optimizer. Although there are several

choices for the optimizer, most of them are based on stochastic gradient descent, such as Adam (Kingma & Ba, 2014), the optimizer used in this work. During training, the error gradient of the current state of the model is estimated using the training dataset and, after this, the weights of the model are updated through backpropagation. The 'learning rate' determines how much the weights are updated.

5.2.1.1 Custom CNN architecture

The specific architecture we use for an implementation of the standard CNN structure has 4 convolutional layers, which we refer to as '4layers'. The first layer has stride 2 and is not followed by max-pooling, while the other layers have stride 1 and max-pooling. The activation function used is ReLU and the activation function in the last fully connected layer is sigmoid for classification and linear for visual field prediction. This architecture is illustrated in Figure 24.

Figure 24. Illustration of the basic 4 layer CNN used in this thesis.

This simple architecture is designed for single-channel input images, so the fundus images are first converted to greyscale before training with this architecture.

Next to this architecture, we also try a version of the 4layers architecture where we insert a fifth convolutional layer with 64 filters and a stride of 1 after the first maxpooling layer. This is referred to as the '5layers' architecture. From preliminary testing, more layers than 5 and fewer layers than 4 does not lead to better performance. Therefore, we focus on these two versions of the base architecture.

5.2.1.2 VGG16

The VGG architecture is a popular architecture introduced by the Visual Geometry Group from the University of Oxford (Simonyan & Zisserman, 2014). It has a very uniform structure, having only 3×3 filters and stride 1 for the convolution layers, and using the same padding and max pooling with a 2×2 filter and stride 2 throughout all the architecture. There are a few variants of the architecture, but we use VGG16, which has 16 layers with trainable weights. Figure 25 shows the VGG16 architecture.

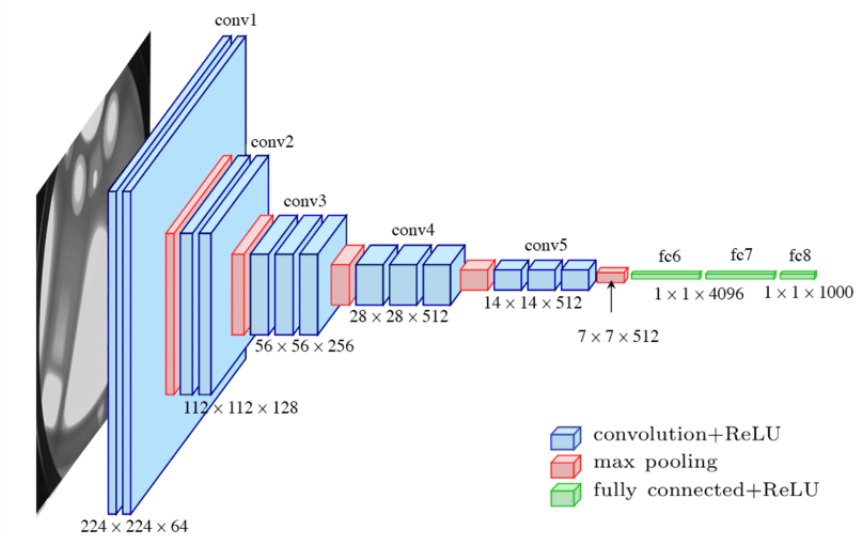


Figure 25. The VGG16 architecture, image from Ferguson et al. (2017).

The disadvantage of the VGG architecture is that it has a very large number of trainable parameters (more than 138 million when the fully connected layers are included), which means that it is computationally expensive and takes a large amount of memory. On the other hand, due to its ease of implementation, it is used by many authors, also in the glaucoma classification domain (An et al., 2019; Diaz-Pinto et al., 2019; Gómez-Valverde et al., 2019). This makes it good to use as a benchmarking model. Another advantage is that there are pre-trained VGG networks available, which means that we can use it with transfer learning.

5.2.1.3 Xception

The Xception architecture, introduced by Chollet (2017), has been shown to outperform previous high-achieving models on the ImageNet dataset, including VGG16. The key concept to understand this architecture is the depthwise separable convolution. Here, rather than having one convolution over all the channels of the input at once, the spatial convolutions are done independently per channel. This is followed by a pointwise convolution (1×1) over the channels to change the dimension. In Xception, a modified version of this occurs as the 1×1 convolution is performed before the channelwise convolution. The idea of decoupling the mapping of cross-channel correlations and spatial correlations is that it is easier to learn than a full 3D mapping, making more efficient use of the parameters (Chollet, 2017). A diagram of the depthwise separable convolution is shown in Figure 26.

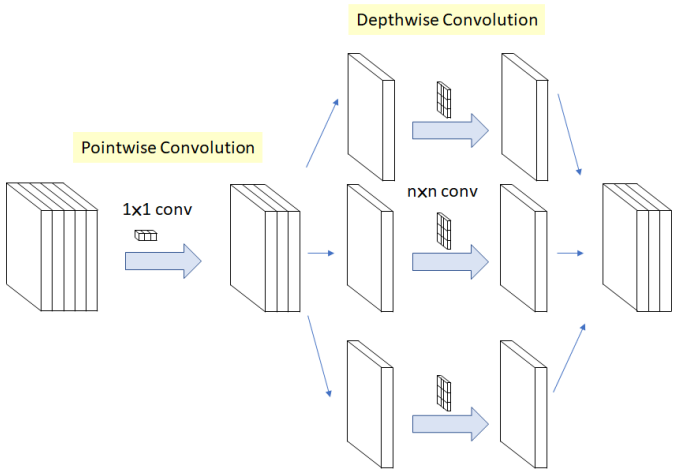


Figure 26. Modified depthwise separable convolution in Xception. Illustration by Tsang (2018).

An additional concept that Xception makes use of is residual connections. This was first introduced in the ResNet architecture (He et al., 2016) and allowed for much deeper networks to be trained. The main idea introduced by ResNet is to have, besides the normal blocks of sequential layers which send their output to the next layer, a connection to add the input of the block to the output afterwards. This is illustrated in Figure 27.

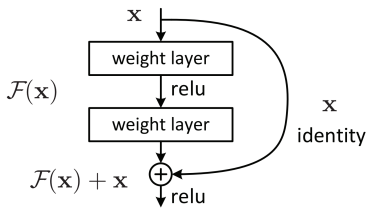


Figure 27. Residual connection. Image from He et al. (2016)

Before ResNet, very deep networks often saw a decrease in performance compared to networks with fewer layers. Due to the multiplicative nature of how the gradients are calculated in a sequential model, the gradients from the loss function can get increasingly smaller or increasingly larger when moving backwards through the layers during backpropagation. This is also known as the 'vanishing' or 'exploding' gradient problem, respectively. For a very deep architecture, the gradients could become too small for the network to learn. However, with residual connections, this problem is significantly reduced. This is because it becomes easier for the model to learn an 'identity' mapping, as it always receives the input of the block as a starting point for the output and, therefore, instead of decreasing performance with a larger number of layers, the model can just ignore the extra layers. The Xception architecture uses 36 convolutional layers, which are structured as 14 blocks. Other than the first and last block, these blocks all have linear residual connections.

5.2.1.4 MobileNet

MobileNetV2 is an architecture proposed by Sandler et al. (2018) which reaches competitive accuracy while having significantly fewer parameters. The structure uses depthwise separable convolution, which we explained in Section 5.2.1.3, to be less complex than normal convolution. Additionally, the MobileNetV2

architecture has an 'inverted residual structure'. Each block has three layers, where the first layer is a 1×1 convolutional layer. This layer has an activation function similar to the ReLU function given by Equation 5.1, except that this function is capped at 6 (also referred to as ReLU6). The second layer of the block is a depthwise separable convolution, and the final layer is a 1×1 convolution with a linear activation function. There are two versions of this block. One has convolutions with a stride of 1 and a residual connection around the block. The second has a stride of 2 for reducing the size of the inputs and no residual connection, since the elementwise addition operation in a residual connection requires that the input and the output are of the same size. Figure 28 shows the two types of blocks presented by Sandler et al. (2018).

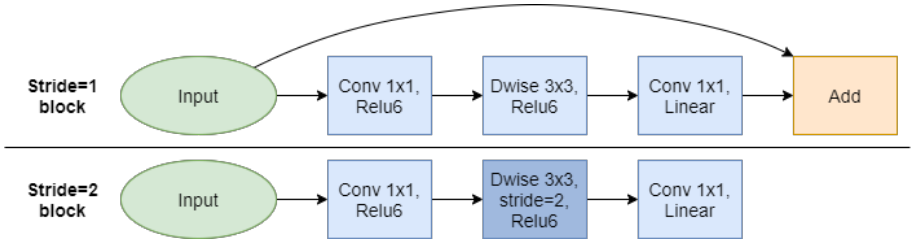


Figure 28. Two types of blocks used in the MobileNet architecture.

5.2.2 Transfer Learning

For the VGG16, Xception, and MobileNet architectures, pre-trained models are available with weights initialized based on the images from ImageNet (Deng et al., 2009). Because the number of samples in our dataset is small, this can be a useful way to introduce extra 'information' into the model from a very large dataset, as well as reducing the training time needed. It must be noted that the OCT and OCTA images are greyscale, as opposed to the colour images in ImageNet, and the datasets in our problem are very specific and of a different domain compared to the ImageNet images. Therefore, we trained the models both with and without transfer learning from ImageNet, but preliminary testing showed that transfer learning resulted in a better performance. In order to use the OCT and OCTA images with the VGG16, Xception, and MobileNet architectures, we modify the single channel greyscale images into three channels by copying the same information in the three channels. This is needed because the architectures are pre-trained on colour images, and expect to receive three channel inputs. Furthermore, we leave out the last fully connected layers from the pre-trained models (which map the features to the ImageNet classes) and instead use the same fully connected layers as presented in Figure 24, to achieve the desired outputs for our tasks.

5.3 Input Combination Methodology

As explained in Section 5.1.2, we take two different approaches to combining the different input types.

For the multi-input CNN, we use the same deep learning approaches described in Section 5.2.1, but leave out the last fully-connected layers. These architectures constitute the five branches of the combined model, one for each input type, after which the output of each branch is concatenated. Subsequently, the fully connected layers are applied to the concatenated output of the branches, and the result is evaluated

with the loss function (cross-entropy loss for classification and mean square error for the VF prediction).

For the stacked ensemble method, we use the best selected models for each individual input modality to train separate CNNs. Next, we can take the output of a specific layer from the architectures (such as the layer before the final fully-connected layer) to get a representation of the features of the image. These can then be concatenated and a separate model can be trained on the concatenated features. For this final model, we tried two options, random forest (RF) and extreme gradient boosting (XGBoost). Both of these methods are based on decision trees, which are simple algorithms consisting of nodes and leaves (Breiman, Friedman, Olshen, & Stone, 2017). At every node, the data is split into a subset based on the state of a specific feature in the data. Which features are used to split the dataset is decided based on a given measure that is minimized with respect to the training data. For a classification tree, a common criteria is the Gini index, whereas for a regression tree the MSE can be used (Breiman et al., 2017). The tree is built from the top downwards, until all features are used or a new split does not improve the evaluation measure. Figure 29 gives an example of a decision tree for a medical problem, namely the prediction of presence of a heart disease in a patient.

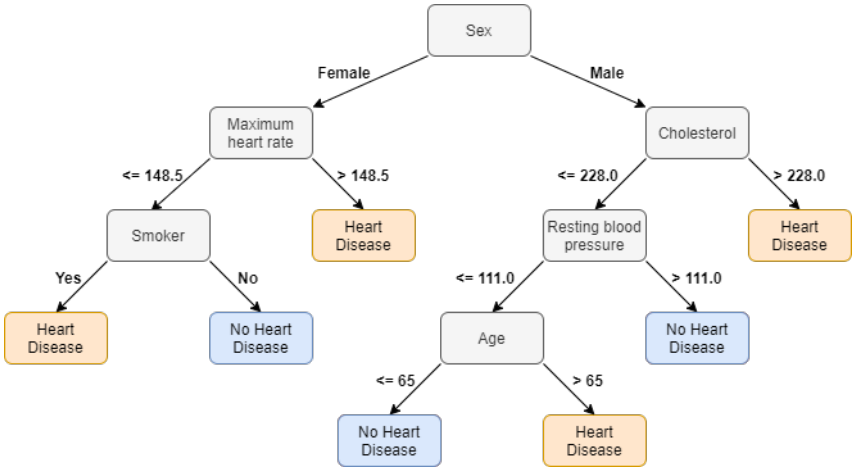


Figure 29. Example of a decision tree for predicting if a patient has a heart disease.

The disadvantage of decision trees is that they have high variance, as small changes in the data can change to a large degree what kind of tree is created. This also means that decision trees tend to be inaccurate when applied to testing data. This is why the performance of the decision trees improve greatly when using 'groups/sets' of trees, instead of individual models. The RF and XGBoost methods are ensemble techniques, meaning that they combine multiple trained models, trees in our case, to create predictions with a lower error overall. RF is referred to as a 'bagging' ensemble technique, whereas XGBoost is a 'boosting' ensemble technique.

In RF, multiple trees are created independently from each other (Breiman, 2001). Each tree is built using randomly selected subsets of the dataset and at each step of the tree only a given number of randomly selected features is considered, in order to make sure that the decision trees are different from each other. At the end, the prediction is based on an average of the individual predictions, which means that the prediction is much more stable even if individual trees can be quite deep and specific to the training data. The XGBoost method, introduced by Chen and Guestrin (2016), uses previous trees as a

starting point to build the subsequent trees, as opposed to building trees independently. Regression trees are used even in classification problems, as the target is to minimize the residual between the predicted and actual value. The trees are generally much less deep than in RF, because each tree is used to build the next tree and, therefore, having very deep trees would mean that the trees would not generalize well to testing data.

As an alternative to extracting the features from the individual CNNs and using those in the stacked ensemble method, we also try the RF and XGBoost methods with the outputs from each individual CNN. This means that we have a very simple model with only five features (one prediction from each CNN). An advantage of this method is that it is even simpler, and more interpretable, allowing the opportunity to see which individual input type contributes the most to each prediction.

5.4 Models Implementation

There are various challenges associated with the training of (deep learning) models, one of which is the problem of overfitting. This happens when the model learns weights that fit very well to the training set, but do not generalize well to the testing set. This is especially more likely to happen when the number of training samples is small, as it means that the model sees fewer and less varied examples. To deal with overfitting, we apply early stopping, using two criteria. The first rule is that the models stop training when the training loss drops below a certain boundary (loss of 0.1 for the classification model and 10 for the VF prediction model), unless the validation loss has reached a new minimum within the last 6 epochs (full passes through the dataset). The second rule is that we stop training when the validation loss does not reach a new minimum for more than 36 epochs. These criteria are based on preliminary testing and visual inspection of the loss graphs. Furthermore, we apply random data augmentation, which we explain in Section 5.4.1. Finally, we try many different hyperparameter settings for the models in order to find a balance where overfitting is not as much of an issue. These parameters and the procedure of tuning them are explained in Section 5.4.2.

5.4.1 Data Augmentation

Introducing random data augmentation into the training set is a very important tool when training convolutional neural networks. By creating (natural) variations in the dataset, the model is less likely to overfit to the training data and is more generalizable. We apply random 'online' augmentation, meaning that each time we pass through the data during training we cause a slight variation to the image. The augmentation techniques considered are rotation, shearing, shifting, elastic deformation (Simard, Steinkraus, & Platt, 2003), and scaling. Shearing slides one edge of an image along the x or y axis, causing a slanted shape or parallelogram. Elastic deformation causes local distortions according to a displacement field, which can create natural-looking variations in the curves of the vessel edges, for example. The hyperparameters for elastic deformation are α, which impacts the intensity of the deformation, and σ, which smoothes the deformations. Figure 30 shows examples of the base pre-processing and additional augmentation applied to an OCTA 3×3 D projection, with the optic disc each time being located and rotated in a slightly different place.

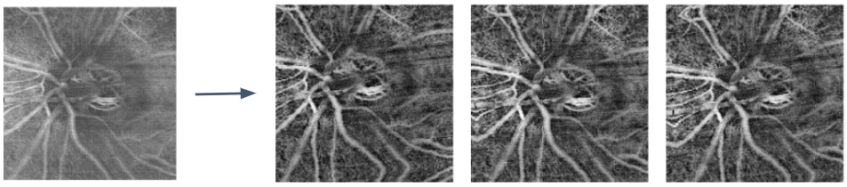


Figure 30. Raw OCTA 3×3 D projection (left) and three examples of the same image after base pre-processing and additional augmentation are applied.

The exact parameters for additional augmentation were decided by visual inspection. It is important to create variations in the image without creating very 'unnatural' images, as this would only distort the training process. For example, for the OCT B-scans, we do not apply rotation, because the direction of the light in these is always pointing downwards, and rotating the image could create unnatural intensity distributions. For the fundus images, we do not apply elastic deformation, as preliminary testing showed it did not necessarily improve performance while taking quite a bit of computation time. Table 2 shows the maximum and minimum hyperparameter values used when creating changes to the dataset. Actual values are randomly sampled from a uniform distribution over these ranges.

Table 2. Overview of ranges for augmentation parameters.

Data modality	Rotation & shearing	Horizontal & vertical shift	Elastic deformation	Scaling factor
OCT	N/A	[−15, 15] pixels	α = [-20,20], $\sigma = 10$	[1.0, 1.05]
OCTA	Rotation: [−12, 12] degrees Shearing: [−8, 8] degrees	[−20, 20] pixels	α = [-40,40], $\sigma = 15$	[1.0, 1.1]
Fundus	Rotation: [−12, 12] degrees Shearing: [−8, 8] degrees	[−20, 20] pixels	-	[1.0, 1.1]

5.4.2 Hyperparameter Tuning

We now describe which hyperparameter combinations and components have been tried for training the models.

The first tuning variable we consider is the *activation function*. In first place, we consider the ReLU, which is a commonly used activation function as described in Section 5.2. However, we also try an alternative activation function referred to as the scaled exponential linear unit (SELU). This function does not clip values at 0 and has a normalizing effect on the network (Klambauer, Unterthiner, Mayr, & Hochreiter, 2017).

The next parameter we tune is *batch size*, which determines how many training samples are fed into the network before calculating gradients and updating weights.

The *dropout* hyperparameter is a parameter introduced to deal with the issue of overfitting. With this technique, neurons are dropped randomly during training by setting their weights to 0 with a probability $1 - p$, such that it becomes harder for the model to fit very specifically to the training data. We add dropout between every layer in our model and try different values.

Next, the *filter factor* (FF) is a hyperparameter that we introduce to determine how many filters are used in each layer. The amounts are given by the base filter numbers shown in Figure 24, divided by the filter factor. For the architectures other than the 4layer and 5layer architectures, this filter factor only

has an impact on the filters in the fully connected layers.

The *input size* is another important hyperparameter, which determines to what size the images are downsized before being input to the model. The main advantage of downsizing is a reduction in the computational requirements of training the models. Moreover, it may contribute to decrease overfitting, since the model can train on fewer features. However, it also means that small structural differences will be lost and therefore unavailable for the model to learn from.

Another hyperparameter used for the architectures that are pre-trained with ImageNet is the number of *layers frozen*. For VGG16, all layers (except for the fully connected layers) are frozen and the model is simply used as a feature extractor, since retraining takes a long time due to the large number of parameters. However, for Xception and MobileNet, we also consider leaving the layers unfrozen and retraining them after initializing the weights.

Finally, the *learning rate* (LR) is a parameter that determines how much the weights in the network are adjusted during each iteration towards minimizing the loss function. A higher learning rate means larger step sizes, which means that training could go faster, but also that optimal values may be skipped over. On the other hand, a low learning rate could mean getting stuck in a local minimum.

Table 3 shows the options which are explored for each tuning parameter. For the selection of the best combination, we use random search.

Table 3. Overview of CNN hyperparameter options for tuning.

Hyperparameter	Options
Activation function	ReLU, SELU
Batch size	6, 10, 12, 18, 32
Dropout	0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7
Filter factor (FF)	1, 2, 3, 4
Input size	64×64, 96×96, 128×128
Layers frozen	All, None
Learning rate (LR)	0.0005, 0.001, 0.005

Within the input combination methodology (Section 5.3), some hyperparameters were also analyzed.

For XGBoosting, the *number of estimators* (trees) that are created can be tuned. The parameter η is similar to a learning rate, as it determines how large the weights are of new features. A higher η shrinks the feature weights so that the boosting process becomes more conservative. Another parameter that determines how conservative the boosting process is (and therefore can influence how much overfitting there is) is γ, which is the minimum loss reduction required to make a further partition on a leaf node of the tree. The *maximum tree depth* hyperparameter determines how deep (and therefore complex) the individual constructed trees can be. Finally, the *subsample* hyperparameter describes what proportion of the training data is sampled before constructing each tree. Table 4 presents the range of values explored for the XGBoosting hyperparameters. Again, the selection is conducted through random search.

Table 4. Overview of XGBoost hyperparameter options for tuning.

Hyperparameter	Options
Number of estimators	800, 1000, 1200
η	0.1, 0.2, 0.3, 0.4
γ	0, 0.5, 2, 4
Max tree depth	10, 30, 50, 60
Subsample	0.6, 0.7, 0.8, 1

For Random Forest, the hyperparameters being tuned include the *number of estimators* and *max tree depth*, which are defined in the same manner as in XGBoosting. Furthermore, the *bootstrap* variable describes whether bootstrap samples are used: if it is true, then sampling is done with replacement, if false then sampling is without replacement and so the whole dataset is used to make each tree. The *max features* hyperparameter determines the number of features to consider when searching for the best split, and has two options. It can be either 'all' features, or 'sqrt', the square root of the number of features. The *split min* is the minimum number of samples required to split an internal node in the decision tree, and the *leaf min* is the minimum number of samples that should be at a leaf node. Table 5 presents the range of values considered for each hyperparameter.

Table 5. Overview of Random Forest hyperparameter options for tuning.

Hyperparameter	Options
Number of estimators	800, 1000, 1200
Max tree depth	3, 6, 9
Bootstrap	True, False
Max features	all, sqrt
Split min	2, 5, 10
Leaf min	1, 2, 4

5.5 Evaluation

In order to have a consistent comparison, we evaluate each model using the same testing datasets. This means that the test set is only selected from the patients who have all data modality and region input types, as the combination models are limited to this dataset. The training set for each single-modality model consists of all the data from that modality that does not belong to the test set. The full sample sizes of the testing set and the remaining datasets used for training are given in Appendix A.2.

Parameter tuning is done using 5-fold cross-validation on the training datasets. During cross-validation, the dataset is split into folds containing 1/5, or 20%, of the dataset. One of the five folds is then left out to be used for validation, while the remaining four are used for training. This is done five times, so that each fold is used for validation exactly once, and the evaluation metrics achieved for each of the five splits are averaged in order to get a more stable assessment of the model performance. The model settings resulting in the best performance are then used to train the models on the complete training set, which are finally evaluated on the test set.

We split the data into folds with stratification, which means that the class label proportions are similar in each fold. For the visual field prediction model, we stratify based on the severity class of the VF MD score rather than the exact score, as these scores are continuous and it would be difficult to balance their

proportions. Figure 31 illustrates the cross-validation process for parameter tuning.

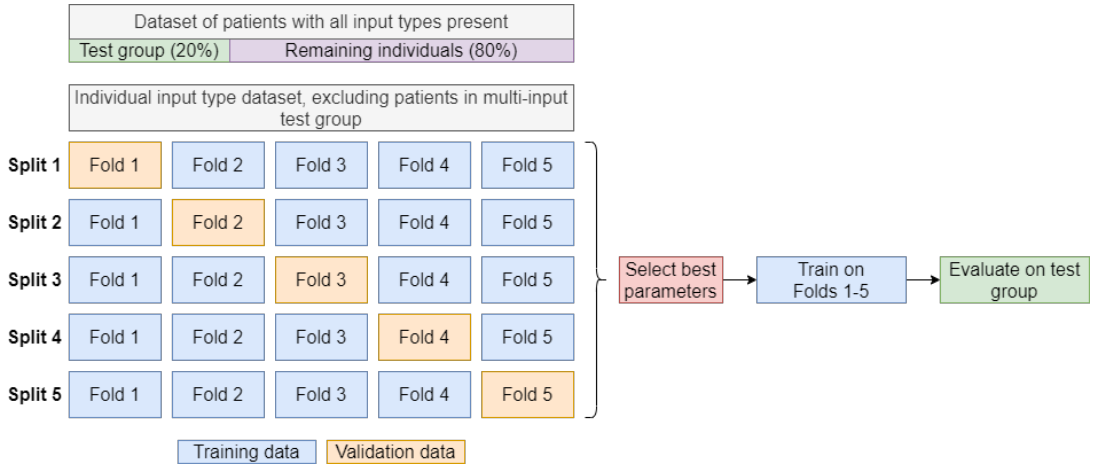


Figure 31. Cross validation methodology.

To evaluate our models, we use several different metrics. The performance of the classification models is evaluated using accuracy, precision, recall, and the area under the receiver operating curve (AUC). Here, AUC is the main metric used to decide between models. The AUC is given by the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance, and is independent of the class distribution. In our case, where the data is not balanced (around 75% has glaucoma), this is a better metric to use than the accuracy which would be already 75% for a model that predicts glaucoma for any input image. The formulas for accuracy, precision and recall are given in Appendix B.1. For the visual field prediction model, we calculate the R^2 and the mean square error (MSE), where MSE is the main metric used to decide between models. This is because it is a commonly used metric in the literature.

For the interpretation of the stacked ensemble methods, we can additionally calculate SHAP values. SHAP stands for Shapley additive explanations, and was proposed by Lundberg and Lee (2017) to increase the interpretability of machine learning techniques. SHAP values attribute to each feature in a model (e.g., a random forest regression) the change in the expected prediction when conditioning on that feature. This way, a ranking can be obtained of the importance of each feature towards the final predictions, as well as whether the features have a negative or positive impact on the model. By calculating SHAP values for the stacked ensemble methods, we can evaluate which modalities have the most important features according to the model towards classifying or predicting glaucoma severity.

To test whether there is a difference in the performance of one model compared to another model, the Welch t-test is used. This is a t-test which allows for variances to be different across different samples. The formula for this is given in Equation 5.5, where $\overline{X_i}$, s_i and N_i represent the sample mean, variance, and size, respectively, of sample i.

$$t = \frac{\overline{X_1} - \overline{X_2}}{\sqrt{\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}}} \tag{5.5}$$

6 Results

All models described in this thesis are trained on an NVIDIA RTX 2080 Ti GPU. The techniques are implemented using the Python programming language and the TensorFlow library (Abadi et al., 2016). In this section, we first present the results obtained for the hyperparameter tuning in Section 6.1 and next we present the testing results in Section 6.2.

6.1 Hyperparameter Tuning Results

As described in Section 5.4.2, the best hyperparameters are selected for each modality and architecture by random search of a range of hyperparameter values.

First of all, it is interesting to compare the training times of the architectures, before presenting which models had the best performance. The training times were similar between the classification and visual field prediction models, but quite different between the architectures (due to a difference in the number of parameters) and input types (due to a difference in the number of samples). Table 6 and Table 7 present the average training times duration over all 5-fold cross-validations per architecture, for the individual modalities and the multi-input CNN, respectively. They also show the number of parameters per architecture to give an indication of the size differences between the architectures.

Table 6. Training times (HH:mm:ss) per model and modality for the single-input CNNs. Duration is calculated from the cumulative run times over 5 folds, averaged over all parameter tuning runs. The number of epochs given is the average per individual fold. The parameters are for the base single-input models (filter factor = 1), rounded to the nearest half million.

	4layers		MobileNet		VGG16		Xception	
	Duration	Epochs	Duration	Epochs	Duration	Epochs	Duration	Epochs
OCT D	01:14:14	72.26	02:17:47	53.16	03:01:53	44.26	07:05:50	58.68
OCTA 3×3 D	00:42:05	77.67	00:56:41	26.53	01:47:29	28.96	02:57:33	66.61
Fundus	04:23:16	79.25	03:40:26	28.32	07:16:17	30.30	11:12:44	71.40
Parameters	3,000,000		5,500,000		16,000,000		25,000,000	

Table 7. Training times (HH:mm:ss) per model for the multi-input CNNs. Duration is calculated from the cumulative run times over 5 folds, averaged over all parameter tuning runs. The number of epochs given is the average per individual fold. The parameters are for the base multi-input CNNs (filter factor = 1), rounded to the nearest half million.

	4layers		MobileNet		VGG16		Xception	
	Duration	Epochs	Duration	Epochs	Duration	Epochs	Duration	Epochs
Multi	04:08:04	67.78	15:17:57	29.4	18:57:46	30.8	23:20:21	64.3
Parameters	9,000,000		19,000,000		79,000,000		126,000,000	

While VGG16 is mentioned in Section 5.2.1.2 to be known as a heavy architecture, after removing the fully connected layers and replacing them with our own, it can be seen that it is the Xception architecture which has the largest number of parameters due to the higher number of convolutional layers. This is reflected in the longer running times as compared to the 4layers and MobileNet architectures, which have fewer parameters to learn. At the same time, the 4layers model takes on average more epochs before it

stops training compared to the remaining three models. This is likely the case because it does not utilize transfer learning and instead learns the weights from scratch. This may explain why the 4layers model has a longer average duration on the fundus data compared to the larger MobileNet model, as the fundus data has relatively many samples and, therefore, the larger number of epochs, or full passes through the training data, increases the duration to a great extent.

Finally, the multi-input CNN has a longer duration than the single-input CNNs, even though the number of samples is smaller, as can be seen in Section 4.2. This can be explained by the fact that all five input types each have their own individual branch in the network, greatly increasing the number of parameters.

When it comes to the stacked ensemble method, the average duration of the cross-validation is only around 10 minutes. However, in order to run this model, first the individual input models need to be trained in order to be able to extract features for the stacked ensemble model. If this can be done in parallel, then the running time of the stacked ensemble method is roughly equal to the longest running single-input CNN.

We now describe the outcomes of our hyperparameter random search for the glaucoma versus healthy classification and for the visual field prediction problems in Sections 6.1.1 and 6.1.2, respectively.

6.1.1 Classification Hyperparameter Outcomes

We present the best model selected based on our hyperparameter tuning process for each of the datasets. We only show the model that achieved the highest AUC in each dataset. The top 5 best performing models for each input type are presented in Appendix A.3. Table 8, first of all, presents the performances and hyperparameters of the single-input classification CNNs.

Table 8. Selected hyperparameters for the single-input CNNs for the classification task. Arch. = architecture, FF = filter factor, LR = learning rate. The evaluation metric shows mean AUC (standard deviation) for the validation over the five cross-validation folds.

	Metric	Hyperparameters							
	AUC	Arch.	Activation	Batch size	Dropout	FF	Input size	Frozen	LR
OCTA 3×3 D	0.909 (0.037)	4layers	ReLU	18	0.4	1	[128, 128]	None	0.005
OCT D	0.926 (0.036)	VGG16	ReLU	18	0.5	1	[64, 64]	All	0.001
Fundus	0.922 (0.038)	MobileNet	ReLU	32	0.7	1	[128, 128]	All	0.001

In general, the classification performance is very good on the validation datasets. Looking at the results for these CNNs, we can see that a relatively high dropout is selected for most models, likely due to the overfitting problem. In addition to this, all models select a ReLU activation function as the best function. This may be because high dropout does not work as well in combination with SELU, as dropout sets neurons to zero while SELU does not clip values to zero (Klambauer et al., 2017). Furthermore, different input types have different architectures performing best, where interestingly the smaller OCTA dataset does not appear to benefit from transfer learning to the same extent as the larger OCT and fundus datasets. As expected, fundus images benefit the most from transfer learning, since their characteristics

are the most similar to ImageNet's (both datasets consist of color images). It is not straightforward to compare the performance of the best selected models with each other because the datasets used to train and evaluate them were all somewhat different, since the modalities were available for different subsets of patients. However, the low standard deviations suggest that in general the results are relatively stable over the folds in all cases.

Table 9 presents the highest performing settings on the validation data for the multi-input CNN, while Tables 10 and 11 show this information for the XGBoosting and random forest stacked ensemble methods, respectively. As mentioned in Section 5.3, we train the stacked ensemble models both on the predicted outputs from each CNN and on the extracted features from the first fully-connected layers.

Table 9. Selected hyperparameters for the multi-input CNN for the classification task. Arch. = architecture, FF = filter factor, LR = learning rate. The evaluation metric shows mean AUC (standard deviation) for the validation over the five cross-validation folds.

	Metric	Hyperparameters							
	AUC	Arch.	Activation	Batch size	Dropout	FF	Input size	Frozen	LR
Multi	0.940 (0.024)	MobileNet	ReLU	18	0.4	3	[128, 128]	None	0.0005

Table 10. Selected hyperparameters for the XGBoosting stacked ensemble model for the classification task. The evaluation metric shows mean AUC (standard deviation) for the validation over the five cross-validation folds.

	Metric	Hyperparameters				
	AUC	γ	Estimators	Max tree depth	η	Subsample
XGB - Outputs	0.877 (0.087)	4	1200	9	0.2	0.7
XGB - Features	0.817 (0.128)	0	1200	3	0.1	0.6

Table 11. Selected hyperparameters for the random forest stacked ensemble model for the classification task. The evaluation metric shows mean AUC (standard deviation) for the validation over the five cross-validation folds.

	Metric	Hyperparameters					
	AUC	Bootstrap	Estimators	Max tree depth	Max feat.	Split min	Leaf min
RF - Outputs	0.877 (0.087)	False	500	30	sqrt	2	1
RF - Features	0.849 (0.104)	True	50	30	sqrt	2	4

Once again, the performance is relatively good on the validation sets, especially in the multi-input method, which outperforms the single-modality models in terms of average AUC (Table 8). The multi-input CNN and the stacked ensemble methods use the same datasets for validation. Thus, we can test for a significant difference using the one-sided Welch t-test. However, the differences are not large enough to conclude that the multi-input CNN performs better than any of the stacked ensemble methods at 5% significance (lowest p-value: 0.092).

For both of the ensemble methods, the models did not find an improvement when predicting with the features extracted from the images and, instead, achieve a higher AUC when using only the output

predictions of the single-input CNNs. Apparently, the last fully-connected layer of the individual CNNs themselves works better at deriving a pattern from the features than the stacked ensemble methods do. Meanwhile, the multi-input CNN is able to better incorporate the different input types for the purpose of classification, possibly because the different branches are all trained at the same time and, as such, features can be created while taking into account the structures in the other images.

6.1.2 Visual Field Prediction Hyperparameter Outcomes

Table 12 presents the selected hyperparameters of the visual field prediction models for the single-input CNNs. The best performing hyperparameters, in terms of MSE, are shown. For additional insight, we also show the R^2 values. The top 5 best performing models can be found in Appendix A.4.

Table 12. Selected hyperparameters for the single-input CNNs for the visual field prediction task. Arch. = architecture, FF = filter factor, LR = learning rate. The metrics show the mean MSE (standard deviation) and R^2 for the validation over the five cross-validation folds.

Input	Metrics		Hyperparameters							
	MSE	R^2	Arch.	Activation	Batch size	Dropout	FF	Input size	Frozen	LR
OCTA 3×3 D	22.700 (4.236)	0.495	4layers	SELU	18	0	1	[64, 64]	None	0.001
OCT D	40.421 (12.775)	0.217	4layers	SELU	6	0.4	3	[64, 64]	None	0.001
Fundus	51.310 (10.517)	0.309	MobileNet	ReLU	12	0.3	2	[128, 128]	All	0.001

We see that for the single-input CNNs, lower dropout values are selected as compared to the classification models. Of note is also that the standard deviations are relatively large, indicating that the performance across different folds is varying. This may be because the folds are not stratified based on the exact visual field values, only based on the glaucoma severity. In addition, the sample sizes are lower for the visual field prediction compared to the classification model since only glaucoma patients are used, such that the performance on the training data does not generalize as well as if there were more samples available. This is also coherent with the fact that the best results seem to be obtained with the simplest models (4layers and MobileNet), which have fewer parameters and are less prone to overfitting.

Tables 13, 14 and 15 show the highest performing visual field model settings for the multi-input CNN, XGBoosting, and random forest methods, respectively. For the XGBoosting and RF methods, results are shown for the model using output predictions from the individual CNNs as inputs, as well as the model using the features extracted with the individual CNNs.

Table 13. Selected hyperparameters for the multi-input CNN for the visual field prediction task. Arch. = architecture, FF = filter factor, LR = learning rate. The metrics show the mean MSE (standard deviation) and R^2 for the validation over the five cross-validation folds.

	Metrics		Hyperparameters							
	MSE	R^2	Arch.	Activation	Batch size	Dropout	FF	Input size	Frozen	LR
Multi	29.424 (7.945)	0.322	VGG16	SELU	8	0	1	[64, 64]	All	0.005

Table 14. Selected hyperparameters for the XGBoosting stacked ensemble model for the visual field prediction task. The metrics show the mean MSE (standard deviation) and R^2 for the validation over the five cross-validation folds.

	Metrics		Hyperparameters				
	MSE	R^2	γ	Estimators	Max tree depth	η	Subsample
XGB - Outputs	19.198 8.874	0.620	4	800	3	0.2	0.7
XGB - Features	21.944 11.996	0.525	4	1200	6	0.3	0.8

Table 15. Selected hyperparameters for the random forest stacked ensemble model for the visual field prediction task. The metrics show the mean MSE (standard deviation) and R^2 for the validation over the five cross-validation folds.

	Metrics		Hyperparameters					
	MSE	R^2	Bootstrap	Estimators	Max tree depth	Max feat.	Split min	Leaf min
RF - Outputs	19.064 11.099	0.627	True	50	50	sqrt	5	1
RF- Features	23.870 5.748	0.493	False	50	50	auto	10	1

Again, we can test for a difference in performance between these methods using the one-sided Welch t-test. For the null hypothesis that the XGB - Outputs method does not perform better than the multi-input CNN, the p-value is 0.046, and therefore we can say that the XGB method does perform significantly better. The other differences are not significant.

Compared to the classification models, it appears that the multi-input CNN does not perform as well here, especially in comparison to the stacked ensemble method. This may be because the inputs into the stacked ensemble method are from the individual CNNs, and thus trained using more data compared to the multi-input CNN. When it comes to the visual field prediction task, which already has fewer samples available, this disadvantage of the multi-input CNN may play a bigger role, causing the multi-input CNN performance to stand out less.

For the ensemble methods it is notable that the model achieves the best performance when the output visual field predictions are from the single-input CNNs are used as inputs, just as with the classification method. The individually trained CNNs may be able to use the features more effectively, potentially because they are using more samples when training the last fully-connected layer (as the stacked ensemble method can only use the subset of patients who have all modalities available).

Furthermore, what is interesting about the visual field prediction results is that, in all cases, the standard deviations across the folds are relatively high. To show better what this variation in performance between the folds looks like, we dive deeper into the cross-validation results for the OCTA D data. Figure 32 shows the results for one of the cross-validation folds. On the left, the true VF values from the best performing epoch are plotted against the predicted VF values. On the right, the training and validation loss values at each epoch of the training process are presented.

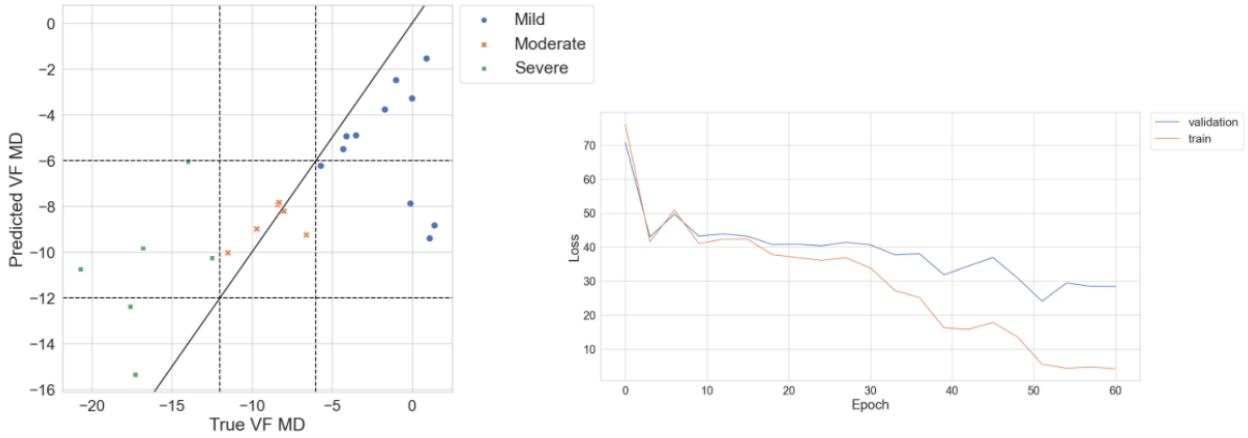


Figure 32. Loss plot and best validation data predictions for fold 1 on the OCTA 3×3 D data. The diagonal line is an identity line to indicate where a perfect prediction would be located.

In this figure, we can see that the performance of the model on the mild and moderate data is relatively good. There is a clear correlation, indicating that OCTA data is potentially informative about the stage of glaucoma at least in the mild and moderate cases. Moreover, the loss plot shows that the validation loss decreases along with the training loss, although this lessens towards the end. However, this performance does not generalize well to all folds. To illustrate, Figure 33 shows the worst performing fold from the same cross-validation process.

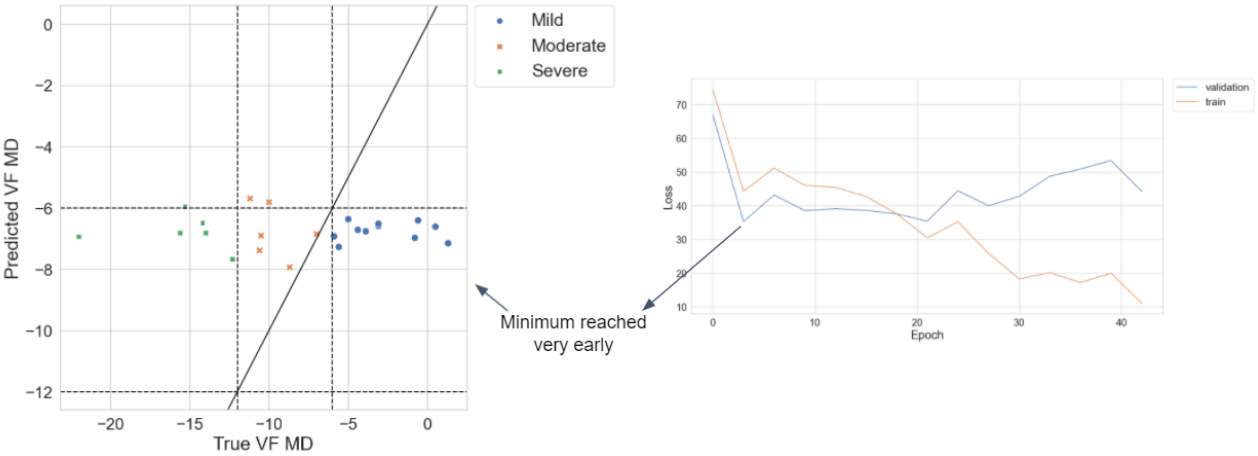


Figure 33. Loss plot and best validation data predictions for fold 2 on the OCTA 3×3 optic disc data.

This shows an extreme example of overfitting, where the validation loss does not decrease with the training loss. As a result, the minimum validation loss is reached early in the training process, when the model still only predicts the average mean deviation value of the training dataset. As a result, this fold does not have a great performance on the validation set, lowering the average performance over the folds and increasing the standard deviation.

Another notable point is that even the well-performing fold shown in Figure 32 does not perform well on the patients with a severe VF MD score. Viewing the performance in other folds and models shows that in general, the VF values on the severe class are the hardest to predict accurately for all models

and input types. Furthermore, the performance on the patients with a VF MD above zero is also quite variable. This may be because these patients only have local visual field deviations, such that changes to the structures visible in these images can still be quite different per person. We tested removing the patients with positive VF MD from the training sets (ranging from 5 to 9 samples per fold for the OCTA data), considering that these cases may introduce more variance into the model, but this did not improve performance.

6.2 Testing Results

We now present the results obtained when training on the full validation and training sets together, and evaluating on the test sets. The model settings used are the best performing hyperparameter settings, as presented in Section 6.1. Although these results were obtained on the optic disc regions, the same hyperparameter settings per modality are also used on the data for the macula regions. For the stacked ensemble method, we show the results for the XGBoosting method, as the performance of XGBoosting and RF methods were shown to be similar during hyperparameter tuning. In Section 6.2.1 and 6.2.2, we present and discuss the results for the glaucoma versus healthy classification task and for the visual field mean deviation prediction task, respectively.

6.2.1 Classification Results

Table 16 presents the classification model AUC, accuracy, precision and recall for each of the single modality-region input CNNs, as well as for the multi-input cnn and the stacked ensemble method. All the metrics are computed in the test portion of the data after training the models in the complementary part of the dataset, as described in Appendix A.2.

Table 16. Testing results for classification models. Standard deviations are given in parentheses.

Model	AUC	Accuracy	Precision	Recall
OCTA 3x3 D	0.800	0.810	0.841	0.906
	(0.155)	(0.111)	(0.100)	(0.107)
OCTA 3x3 M	0.475	0.670	0.670	1.000
	(0.149)	(0.014)	(0.014)	(0.000)
OCT D	0.864	0.834	0.836	0.965
	(0.163)	(0.096)	(0.112)	(0.053)
OCT M	0.678	0.724	0.740	0.929
	(0.155)	(0.045)	(0.062)	(0.097)
Fundus	0.797	0.796	0.829	0.888
	(0.147)	(0.162)	(0.112)	(0.115)
Multi	0.918	0.850	0.871	0.941
	(0.097)	(0.104)	(0.122)	(0.083)
XGB - Outputs	0.830	0.858	0.879	0.918
	(0.067)	(0.067)	(0.048)	(0.098)
XGB - Features	0.811	0.842	0.867	0.906
	(0.126)	(0.115)	(0.083)	(0.115)

It is immediately noticeable that the AUCs obtained on the test sets are lower than those obtained on the validation data during hyperparameter tuning. This is in part because the process of hyperparameter tuning allows us to train many models and select the best performance for the validation sets, so it is

common for the testing performance to be lower as the testing set is independent and, contrarily to the validation set, was not used to guide the training process. Another explanation may be that the test set is smaller than the validation sets were for the individual models, as we have only 26 samples in the test set for each fold. This means that the test samples are likely to be more different across each fold and simply due to chance could be more difficult to predict for. Even if they are not more different across different folds, they are likely to be less representative, as there are simply fewer cases that are tested. This is also reflected in the higher standard deviations compared to standard deviations seen in Table 8. Therefore, it is more difficult to draw conclusions from the test results. For example, we can see that the multi-input CNN again reaches the highest performance out of all models, but due to high standard deviations we cannot reject the null hypothesis that the multi-input CNN and stacked ensemble methods have equal performance with the one-sided Welch t-test (p-value: 0.069).

We do have significant evidence that the optic disc centered images have better performance than the macula centered images, both for the OCTA modality (p-value: 0.004) and for the OCT modality (p-value: 0.048). When interpreting the model trained with the stacked ensemble method, this finding is reinforced. By reviewing the (absolute) SHAP values for the XGB - outputs model, which has five features (one prediction from each single-input CNN), we can see which individual prediction was on average more 'valuable' towards the final predictions. These values are presented in Table 17, ordered from highest to lowest absolute SHAP value.

Table 17. Mean absolute SHAP values for the XGBoosting stacked ensemble classification model per input feature (averaged over 5 folds).

Input prediction source	Mean absolute SHAP value
OCT D	0.192 (0.161)
OCTA 3×3 D	0.176 (0.105)
Fundus	0.167 (0.128)
OCT M	0.030 (0.050)
OCTA 3×3 M	0.008 (0.012)

We will not go into the exact meaning of the magnitude of these SHAP values as this is difficult to interpret, but we can see that on average, the feature extracted from the OCT optic disc scans had the most contribution towards the final prediction. The standard deviations do show that, again, there is quite a bit of variability between the folds.

6.2.2 Visual Field Prediction Results

Table 18 presents the results for the visual field prediction models trained with the hyperparameter settings described in Section 6.1.2. We present the mean squared error and the R^2. In addition to that, we also present the classification accuracy of the predictions, that is, the multiclass accuracy if the ground truth and predicted values were converted to the mild, moderate and severe classes. Finally, we show a separate MSE for the data points within each severity class, since we saw in Section 6.1.2 that the predictive performance of our models can differ quite a bit per class.

Table 18. Testing results for visual field prediction models. Standard deviations are given in parentheses.

Model	MSE	R^2	Accuracy	MSE mild	MSE moderate	MSE severe
OCTA 3x3 D	33.959	0.295	0.424	27.196	10.028	70.772
	(11.904)	(0.193)	(0.121)	(6.097)	(6.811)	(55.102)
OCTA 3x3 M	35.693	0.170	0.388	27.599	6.664	78.484
	(11.534)	(0.090)	(0.089)	(10.068)	(5.051)	(35.365)
OCT D	38.299	0.131	0.435	27.629	6.092	90.023
	(13.964)	(0.148)	(0.107)	(10.022)	(4.250)	(55.627)
OCT M	40.190	0.069	0.294	21.577	6.762	107.055
	(11.848)	(0.049)	(0.072)	(7.194)	(2.244)	(52.764)
Fundus	28.749	0.353	0.529	20.661	11.841	61.512
	(13.931)	(0.162)	(0.199)	(5.571)	(6.071)	(57.453)
Multi	35.946	0.200	0.376	30.248	8.519	72.089
	(19.264)	(0.133)	(0.086)	(32.289)	(14.512)	(101.478)
XGB - Outputs	30.192	0.348	0.494	19.416	8.503	71.033
	(14.453)	(0.191)	(0.115)	(3.272)	(6.860)	(60.631)
XGB - Features	31.226	0.319	0.518	23.165	5.638	71.460
	(12.698)	(0.208)	(0.097)	(6.755)	(2.957)	(50.141)

Just as with the classification models, the performance of the VF prediction models is generally a bit worse on the testing sets than on the validation sets (except for the single-input CNN using fundus data, which performs much better on the test data). This could again be explained in part by the variety in the data, as reflected in the high standard deviations. It is interesting to see if these models perform better than if the average VF MD value of the training data were predicted for every training sample, as a model might do when it has not learnt anything yet from the training data. The MSE that would be obtained when predicting the average VF MD for every training sample is referred to as the 'baseline MSE'. In Table 19, we present the average baseline MSEs over the five folds of the testing data for each training dataset. Together with this, we present the p-values for a one-sided one-sample t-test comparing the obtained MSEs in Table 18 to their respective baseline MSEs (where the null hypothesis is that the obtained MSEs are not better than their baselines).

Table 19. Baseline MSEs (if the average VF MD value of the training set were predicted for every test sample) and p-values for the test comparing obtained test MSEs with baseline MSEs.

	OCTA 3×3 D	OCTA 3×3 M	OCT D	OCT M	Fundus	Multi	XGB - Out.	XGB - Feat.
Baseline MSE	38.156	37.636	36.882	36.963	38.197	42.079	42.079	42.079
p-value t-test	0.237	0.362	0.584	0.712	0.102	0.258	0.070	0.064

Out of all models, only the stacked ensemble model performs significantly better than predicting the average value of the training samples when evaluating with a one-sample t-test at a 10% significance level, and this is the case for none of the models at a 5% significance level. Nevertheless, the models do follow the same trend as was found during the hyperparameter tuning, which was that the stacked ensemble model has the best performance, followed by the OCTA data. The OCT data has the worst performance when compared to the baseline MSE. However, considering the high standard deviations as a result of the small sample size of the test set (17 samples), it is still possible that this difference is caused by chance.

Just as with the classification model, we can calculate the SHAP values for the stacked ensemble

method using the five output predictions as input, in order to give an indication of the contribution of each input. We average the absolute SHAP value for each individual feature over the 5 folds. Table 20 presents these values in order of magnitude.

Table 20. Mean absolute SHAP values for the XGBoosting stacked ensemble VF MD prediction model (averaged over 5 folds). Standard deviations in parentheses.

Input prediction source	Mean absolute SHAP value
OCTA 3×3 D	0.099 (0.033)
OCT D	0.052 (0.021)
Fundus	0.046 (0.028)
OCT M	0.041 (0.016)
OCTA 3×3 M	0.040 (0.037)

The OCTA predictions are determined to have the highest contribution to the model, which is in line with the best results obtained during hyperparameter tuning, and a further indication that the test results, where the fundus model achieves the best MSE, might suffer from not being representative.

7 Discussion and Conclusion

In this thesis we investigated approaches for predicting glaucoma severity using multiple imaging modalities and regions. We had two different model tasks: classifying healthy versus glaucomatous eyes and predicting the visual field mean deviation. Three imaging modalities were used: fundus photography, optical coherence tomography B-scans, and optical coherence tomography angiography projections, with the OCT and OCTA scans being available for both the optic disc and macula regions. For the use of these modality-region input types to train models individually, we investigated the implementation of convolutional neural networks with different architectures and hyperparameter settings, selecting the best performing approach for each modality. For the combination of all modalities and regions, we investigated a multi-input CNN approach and a stacked ensemble method approach, where XGBoost and random forest were used as combination methods.

In general, the variability of the performance is high for each of the models, especially when evaluating on the test sets. For the classification task, the only significant results are in the comparison of the OCTA macula and OCT macula single-input model with other models, which shows that the optic disc regions result in a higher AUC when classifying healthy and glaucoma subjects. This can be justified when looking at the established knowledge in the literature, as glaucoma is known as a disease that mostly affects the optic disc (European Glaucoma Society, 2020).

Furthermore, even though the standard deviation is high, the AUC of 0.918 achieved by the multi-input CNN is a promising result. It is lower than the AUC achieved by some single-input models in the literature, such as the AUC of 0.960 achieved by Shibata et al. (2018), but the multi-input model has achieved this using a much smaller sample of patients (127 as opposed to 3,000 patients). This shows that the multi-input CNN has potential to be a good option when a researcher has access to a limited number of patients for their study, but does have the opportunity to collect multiple imaging modalities. Additionally, if a higher sample size is available, the performance of the multi-input CNN could increase,

which warrants future research.

Finally, it is interesting to note for the classification model that during parameter tuning, the top-5 for all modalities included at least one 4layers model with an AUC of more than 0.9, which signals that this problem can be solved in all modalities with quite a simple, cost-effective model.

For the visual field prediction models, we find no significant differences between the single-input CNNs and the multi-input approaches, once more due to the high standard deviations. Disregarding this, the stacked ensemble model is the approach that appears to be the most promising here, unlike the classification model where the multi-input CNN seems to be more suitable. Additionally, the OCTA modality contributes the most to the predictions in the stacked ensemble model (reaching the highest average SHAP values), more than the OCT modality, which is interesting because RNFL measurements in circumpapillary OCT scans are one of the standard measurements used by clinicians to assess the retinal damage during glaucoma progression (Banegas et al., 2016). Although the results are not significant, they do give an indication that OCTA data may provide additional insights towards staging glaucoma which are not provided by the OCT modality.

Furthermore, we find that the visual field prediction models have a much worse performance and higher variability in predicting the severe cases compared to the moderate and mild cases. This may be explained by the fact that, at high severity levels, a 'flooring' effect can occur. For instance, the RNFL thickness becomes critically thinned and does not thin further after a given progression (Bowd, Zangwill, Weinreb, Medeiros, & Belghith, 2017). It is thought that this floor effect might be less of an issue in the OCTA modality (Moghimi et al., 2019), making it more promising for predicting later stages, but in our test set this modality also does not perform well on the severe cases. At the same time, the severe glaucoma group has the largest range of VF MD in its samples (ranging from -12 to -34) and it may be the case that there are not enough severe samples for the model to train properly on these cases.

Even with the stacked ensemble model reaching the best mean squared error (MSE) of 30.192 and R^2 of 0.348, our visual field prediction models do not achieve the performance that is found elsewhere in the literature. Yu et al. (2020) reach $R^2 = 0.76$ for predicting VF MD with a combination of macula and optic disc OCT data. Hemelings et al. (2021) reach $R^2 = 0.71$ and MSE $= 11.10$. There are different potential reasons for why we did not achieve the same performance. First of all, our dataset is much smaller, using only around 160 OCT scans while Yu et al. (2020) use more than 10,000 pairs of OCT volumes and Hemelings et al. (2021) use around 1,600 circumpapillary OCT scans. Smaller dataset means that models do not generalize well to new data, since they have fewer examples to learn from, and overfitting is more likely to happen. Additionally, the extents to which the information in the scans is used are different when comparing our approach with the literature. Yu et al. (2020) use the full 3D OCT volumes in 3D CNNs, providing more information compared to only using B-scans. Meanwhile, Hemelings et al. (2021) use the full resolution of the images rather than downsizing them, which may leave more of the smaller structures in the images intact. The disadvantage of these approaches is that training takes much longer, and that noise and disruptions will make them less effective. Lastly, the study of Hemelings et al. (2021) uses circumpapillary OCT. This is a circular scan around the optic disc, which has a different acquisition protocol that is very specific to glaucoma (as glaucoma is known to affect the region closer to the optic

disc first) (E. J. Lee, Kim, Kim, & Choi, 2014). Since this model requires a specific protocol, it would be more interesting to obtain a high accuracy with the regular OCT cubes which are used for a number of diseases and regular screening.

Another explanation for the lower than expected performances could be that our model approaches can be improved. While we have tried many different values for the hyperparameters and architectures, there are also many methods that we have not tried. For example, the basic 4layers model, which performed well for many cases, could be further refined by adding more layers, potentially with a residual connection, and experimenting further with regularization measures to reduce the amount of overfitting.

In other future research, the 'loose pairing' approach as described by Wang et al. (2019) could be investigated, in order to see if increasing the sample size by pairing patients with other patients that share the same label will increase performance. Alternatively, it would be interesting to create a method that can deal with multiple inputs but also with 'partial' data, such that the method can handle patients with all modalities but also make a prediction when a patient has only one of the modalities.

Finally, further research could be done to improve the interpretability of the techniques applied, in order to gain more insight into the value of each image type and even specific aspects of each image. For example, Liao et al. (2019) use a CNN for glaucoma diagnosis with a novel architecture that can produce clinically interpretable heatmaps from each image to show which features were given the most weight.

Overall, we have shown that it is possible to combine multiple input modalities and regions into single models for the purpose of glaucoma classification and severity prediction. For more conclusive evidence towards the value of individual modalities, more (high-quality) data is needed. Nevertheless, we have shown that the newer OCTA modality has potential to be valuable towards diagnosing glaucoma and its severity, and that the combination of modalities could result in better performing models, even for small sample sizes.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... Isard, M. e. a. (2016). Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)* (pp. 265–283).

Abegão Pinto, L., Willekens, K., Van Keer, K., Shibesh, A., Molenberghs, G., Vandewalle, E., & Stalmans, I. (2016). Ocular blood flow in glaucoma–the Leuven Eye Study. *Acta ophthalmologica*, *94*(6), 592–598.

Ahn, J. M., Kim, S., Ahn, K.-S., Cho, S.-H., Lee, K. B., & Kim, U. S. (2018). A deep learning model for the detection of both advanced and early glaucoma using fundus photography. *PloS one*, *13*(11), e0207982.

Almazroa, A., Burman, R., Raahemifar, K., & Lakshminarayanan, V. (2015). Optic disc and optic cup segmentation methodologies for glaucoma image detection: a survey. *Journal of ophthalmology*, *2015*.

An, G., Akiba, M., Omodaka, K., Nakazawa, T., & Yokota, H. (2021). Hierarchical deep learning models using transfer learning for disease detection and classification based on small number of medical images. *Scientific reports*, *11*(1), 1–9.

An, G., Omodaka, K., Hashimoto, K., Tsuda, S., Shiga, Y., Takada, N., ... Nakazawa, T. (2019). Glaucoma diagnosis with machine learning based on optical coherence tomography and color fundus images. *Journal of healthcare engineering*, *2019*.

Arce, G., & McLoughlin, M. (1987). Theoretical analysis of the max/median filter. *IEEE transactions on acoustics, speech, and signal processing*, *35*(1), 60–69.

Banegas, S. A., Antón, A., Morilla, A., Bogado, M., Ayala, E. M., Fernandez-Guardiola, A., & Moreno-Montañes, J. (2016). Evaluation of the retinal nerve fiber layer thickness, the mean deviation, and the visual field index in progressive glaucoma. *Journal of glaucoma*, *25*(3), e229–e235.

Bekkers, A., Borren, N., Ederveen, V., Fokkinga, E., Andrade De Jesus, D., Sánchez Brea, L., ... Stalmans, I. (2020). Microvascular damage assessed by optical coherence tomography angiography for glaucoma diagnosis: a systematic review of the most discriminative regions. *Acta ophthalmologica*, *98*(6), 537–558.

Benesty, J., Chen, J., Huang, Y., & Cohen, I. (2009). Pearson correlation coefficient. In *Noise reduction in speech processing* (pp. 1–4). Springer.

Bengtsson, B., & Heijl, A. (2008). A visual field index for calculation of glaucoma rate of progression. *American journal of ophthalmology*, *145*(2), 343–353.

Bowd, C., Belghith, A., Christopher, M., Goldbaum, M. H., Fan, R., Rezapour, J., ... Zangwill, L. M. e. a. (2021). Deep-learning enface image classifier analysis of optical coherence tomography angiography images improves classification of healthy and glaucoma eyes. *Investigative Ophthalmology & Visual Science*, *62*(8), 1024–1024.

Bowd, C., Weinreb, R. N., Williams, J. M., & Zangwill, L. M. (2000). The retinal nerve fiber layer thickness in ocular hypertensive, normal, and glaucomatous eyes with optical coherence tomography.

Archives of ophthalmology, *118*(1), 22–26.

Bowd, C., Zangwill, L. M., Berry, C. C., Blumenthal, E. Z., Vasile, C., Sanchez-Galeana, C., ... Weinreb, R. N. (2001). Detecting early glaucoma by assessment of retinal nerve fiber layer thickness and visual function. *Investigative ophthalmology & visual science*, *42*(9), 1993–2003.

Bowd, C., Zangwill, L. M., Weinreb, R. N., Medeiros, F. A., & Belghith, A. (2017). Estimating optical coherence tomography structural measurement floors to improve detection of progression in advanced glaucoma. *American journal of ophthalmology*, *175*, 37–44.

Braaf, B. (2015). *Angiography and Polarimetry of the posterior eye with functional Optical Coherence Tomography* (Unpublished doctoral dissertation). Vrije Universiteit Amsterdam.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (2017). *Classification and regression trees*. Routledge.

Caixinha, M., & Nunes, S. (2017). Machine learning techniques in clinical vision sciences. *Current eye research*, *42*(1), 1–15.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794).

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 1251–1258).

Christopher, M., Belghith, A., Bowd, C., Proudfoot, J. A., Goldbaum, M. H., Weinreb, R. N., ... Zangwill, L. M. (2018). Performance of deep learning architectures and transfer learning for detecting glaucomatous optic neuropathy in fundus photographs. *Scientific reports*, *8*(1), 1–13.

Christopher, M., Bowd, C., Belghith, A., Goldbaum, M., Weinreb, R., Fazio, M., ... Zangwill, L. (2019, 09). Deep Learning Approaches Predict Glaucomatous Visual Field Damage from Optical Coherence Tomography Optic Nerve Head Enface Images and Retinal Nerve Fiber Layer Thickness Maps. *Ophthalmology*, *127*. doi: 10.1016/j.ophtha.2019.09.036

De Carlo, T. E., Romano, A., Waheed, N. K., & Duker, J. S. (2015). A review of optical coherence tomography angiography (OCTA). *International journal of retina and vitreous*, *1*(1), 5.

De Jesus, D. A., Brea, L. S., Breda, J. B., Fokkinga, E., Ederveen, V., Borren, N., ... Klein, S. e. a. (2020). OCTA multilayer and multisector peripapillary microvascular modeling for diagnosing and staging of glaucoma. *Translational Vision Science & Technology*, *9*(2), 58–58.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 ieee conference on computer vision and pattern recognition* (pp. 248–255).

Diaz-Pinto, A., Morales, S., Naranjo, V., Köhler, T., Mossi, J. M., & Navea, A. (2019). CNNs for automatic glaucoma assessment using fundus images: an extensive validation. *Biomedical engineering online*, *18*(1), 1–19.

Duker, J. S., Waheed, N. K., & Goldman, D. (2013). *Handbook of Retinal OCT: Optical Coherence Tomography E-Book*. Elsevier Health Sciences.

European Glaucoma Society. (2020). *Terminology and Guidelines for Glaucoma* (5th ed.). Savona: Dogma.

Fang, P., Lindner, M., Steinberg, J., Müller, P., Gliem, M., Krohne, T., & Holz, F. e. a. (2016). Clinical applications of OCT angiography. *Der Ophthalmologe: Zeitschrift der Deutschen Ophthalmologischen Gesellschaft*, *113*(1), 14–22.

Fard, M. A., & Ritch, R. (2020). Optical coherence tomography angiography in glaucoma. *Annals of Translational Medicine*, *8*(18). Retrieved from `https://atm.amegroups.com/article/view/46397`

Ferguson, M., ak, R., Lee, Y.-T., & Law, K. (2017, 12). Automatic localization of casting defects with convolutional neural networks. In (p. 1726-1735). doi: 10.1109/BigData.2017.8258115

Flammer, J., & Orgül, S. (1998). Optic nerve blood-flow abnormalities in glaucoma. *Progress in retinal and eye research*, *17*(2), 267–289.

Gómez-Valverde, J. J., Antón, A., Fatti, G., Liefers, B., Herranz, A., Santos, A., ... Ledesma-Carbayo, M. J. (2019). Automatic glaucoma classification using color fundus images based on convolutional neural networks and transfer learning. *Biomedical optics express*, *10*(2), 892–913.

Greenspan, H., Van Ginneken, B., & Summers, R. M. (2016). Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. *IEEE Transactions on Medical Imaging*, *35*(5), 1153–1159.

Grunwald, J., Riva, C., Stone, R., Keates, E., & Petrig, B. (1984). Retinal autoregulation in open-angle glaucoma. *Ophthalmology*, *91*(12), 1690–1694.

harvardeye. (2021). https://harvardeye.com/glaucoma-orange-county/what-is-glaucoma/.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 770–778).

Hemelings, R., Elen, B., Barbosa-Breda, J., Bellon, E., Blaschko, M., De Boever, P., & Stalmans, I. (2021, 06). *Pointwise visual field estimation from optical coherence tomography in glaucoma: a structure-function analysis using deep learning.*

Hormel, T. T., Hwang, T. S., Bailey, S. T., Wilson, D. J., Huang, D., & Jia, Y. (2021). Artificial intelligence in OCT angiography. *Progress in Retinal and Eye Research*, 100965.

Huang, D., Swanson, E. A., Lin, C. P., Schuman, J. S., Stinson, W. G., Chang, W., ... Puliafito, C. A. e. a. (1991). Optical coherence tomography. *science*, *254*(5035), 1178–1181.

Jammal, A. A., Thompson, A. C., Mariottoni, E. B., Berchuck, S. I., Urata, C. N., Estrela, T., ... Medeiros, F. A. (2020). Human versus machine: comparing a deep learning algorithm to human gradings for detecting glaucoma on fundus photographs. *American journal of ophthalmology*, *211*, 123–131.

Jia, Y., Wei, E., Wang, X., Zhang, X., Morrison, J. C., Parikh, M., ... Edmunds, B. e. a. (2014). Optical coherence tomography angiography of optic disc perfusion in glaucoma. *Ophthalmology*, *121*(7), 1322–1332.

Jonas, J., & Budde, W. (2002). Is the nasal optic disc sector important for morphometric glaucoma diagnosis? *British journal of ophthalmology*, *86*(11), 1232–1235.

Kang, E. Y.-C., Yeung, L., Lee, Y.-L., Wu, C.-H., Peng, S.-Y., Chen, Y.-P., ... Lai, C.-C. (2021, May 31). A Multimodal Imaging–Based Deep Learning Model for Detecting Treatment-Requiring

Retinal Vascular Diseases: Model Development and Validation Study. *JMIR Med Inform*, *9*(5), e28868. Retrieved from "https://medinform.jmir.org/2021/5/e28868" doi: 10.2196/28868

Kindsight. (2020). https://kindsight.com.au/eye-conditions/eye-anatomy/.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klambauer, G., Unterthiner, T., Mayr, A., & Hochreiter, S. (2017). Self-normalizing neural networks. In *Proceedings of the 31st international conference on neural information processing systems* (pp. 972–981).

Kraus, M. F., Potsaid, B., Mayer, M. A., Bock, R., Baumann, B., Liu, J. J., ... Fujimoto, J. G. (2012). Motion correction in optical coherence tomography volumes on a per A-scan basis using orthogonal scan patterns. *Biomedical optics express*, *3*(6), 1182–1199.

Lai, T. Y. (2020). *Ocular imaging at the cutting-edge.* Nature Publishing Group.

Le, D., Alam, M., Yao, C. K., Lim, J. I., Hsieh, Y.-T., Chan, R. V., ... Yao, X. (2020). Transfer learning for automated OCTA detection of diabetic retinopathy. *Translational Vision Science & Technology*, *9*(2), 35–35.

Lee, E. J., Kim, T.-W., Kim, M., & Choi, Y. (2014, 02). Peripapillary Retinoschisis in Glaucomatous Eyes. *PloS one*, *9*, e90129. doi: 10.1371/journal.pone.0090129

Lee, J., Jun, S., Cho, Y.-W., Lee, H., Kim, G. B., Seo, J. B., & Kim, N. (2017). Deep learning in medical imaging: general overview. *Korean journal of radiology*, *18*(4), 570.

Lezama, J., Mukherjee, D., McNabb, R. P., Sapiro, G., Kuo, A. N., & Farsiu, S. (2016). Segmentation guided registration of wide field-of-view retinal optical coherence tomography volumes. *Biomedical optics express*, *7*(12), 4827–4846.

Liao, W., Zou, B., Zhao, R., Chen, Y., He, Z., & Zhou, M. (2019). Clinical interpretable deep learning model for glaucoma diagnosis. *IEEE journal of biomedical and health informatics*, *24*(5), 1405–1412.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems* (pp. 4768–4777).

Maetschke, S., Antony, B., Ishikawa, H., Wollstein, G., Schuman, J., & Garnavi, R. (2019). A feature agnostic approach for glaucoma detection in OCT volumes. *PloS one*, *14*(7), e0219126.

Mayro, E. L., Wang, M., Elze, T., & Pasquale, L. R. (2020). The impact of artificial intelligence in the diagnosis and management of glaucoma. *Eye*, *34*(1), 1–11.

Medeiros, F. A. (2019). Deep learning in glaucoma: progress, but still lots to do. *The Lancet Digital Health*, *1*(4), e151–e152.

Mehta, P., Lee, A., Lee, C., Balazinska, M., & Rokem, A. (2018). Multilabel multiclass classification of OCT images augmented with age, gender and visual acuity data. *bioRxiv*, 316349.

Mikelberg, F. S., Parfitt, C. M., Swindale, N. V., Graham, S. L., Drance, S. M., & Gosine, R. (1995). Ability of the heidelberg retina tomograph to detect early glaucomatous visual field loss. *Journal of glaucoma*, *4*(4), 242–247.

Mittal, A., Moorthy, A. K., & Bovik, A. C. (2012). No-reference image quality assessment in the spatial

domain. *IEEE Transactions on image processing*, *21*(12), 4695–4708.

Moghimi, S., Bowd, C., Zangwill, L. M., Penteado, R. C., Hasenstab, K., Hou, H., ... Weinreb, R. N. (2019). Measurement floors and dynamic ranges of OCT and OCT angiography in glaucoma. *Ophthalmology*, *126*(7), 980–988.

Norouzifard, M. (2020). *Computer Vision and Machine Learning for Glaucoma Detection* (Unpublished doctoral dissertation). Auckland University of Technology.

O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, *22*(10), 1345–1359.

Pizer, S. M., Amburn, E. P., Austin, J. D., Cromartie, R., Geselowitz, A., Greer, T., ... Zuiderveld, K. (1987). Adaptive histogram equalization and its variations. *Computer vision, graphics, and image processing*, *39*(3), 355–368.

Quigley, H. A. (1993). Open-angle glaucoma. *New England Journal of Medicine*, *328*(15), 1097–1106.

Quigley, H. A., West, S. K., Rodriguez, J., Munoz, B., Klein, R., & Snyder, R. (2001). The prevalence of glaucoma in a population-based study of Hispanic subjects: Proyecto VER. *Archives of ophthalmology*, *119*(12), 1819–1826.

Ragab, D. A., Sharkas, M., Marshall, S., & Ren, J. (2019). Breast cancer detection using deep convolutional neural networks and support vector machines. *PeerJ*, *7*, e6201.

Ran, A. R., Cheung, C. Y., Wang, X., Chen, H., Luo, L.-y., Chan, P. P., ... Young, A. L. e. a. (2019). Detection of glaucomatous optic neuropathy with spectral-domain optical coherence tomography: a retrospective training and validation deep-learning analysis. *The Lancet Digital Health*, *1*(4), e172–e182.

Ran, A. R., Tham, C. C., Chan, P. P., Cheng, C.-Y., Tham, Y.-C., Rim, T. H., & Cheung, C. Y. (2020). Deep learning in glaucoma with optical coherence tomography: a review. *Eye*, 1–14.

Rulli, E., Quaranta, L., Riva, I., Poli, D., Hollander, L., Galli, F., ... Weinreb, R. (2018, 01). Visual field loss and vision-related quality of life in the Italian Primary Open Angle Glaucoma Study. *Scientific Reports*, *8*. doi: 10.1038/s41598-017-19113-z

Sample, P. A. (2003). Glaucoma is present prior to its detection with standard automated perimetry: is it time to change our concepts? *Graefe's Archive for Clinical and Experimental Ophthalmology*, *241*(3), 168–169.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 4510–4520).

Sarhan, A., Rokne, J., & Alhajj, R. (2019). Glaucoma detection using image processing techniques: A literature review. *Computerized Medical Imaging and Graphics*, *78*, 101657.

Sevastopolsky, A. (2017). Optic disc and cup segmentation methods for glaucoma detection with modification of U-Net convolutional neural network. *Pattern Recognition and Image Analysis*, *27*(3), 618–624.

Shibata, N., Tanito, M., Mitsuhashi, K., Fujino, Y., Matsuura, M., Murata, H., & Asaoka, R. (2018). Development of a deep residual learning algorithm to screen for glaucoma from fundus photography. *Scientific reports*, *8*(1), 1–9.

Shields, M. B. (2008). Normal-tension glaucoma: is it different from primary open-angle glaucoma? *Current opinion in ophthalmology*, *19*(2), 85–88.

Simard, P. Y., Steinkraus, D., & Platt, J. C. e. a. (2003). Best practices for convolutional neural networks applied to visual document analysis. In *Icdar* (Vol. 3).

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Singh, S. A., Meitei, T. G., & Majumder, S. (2020). Short PCG classification based on deep learning. In *Deep learning techniques for biomedical and health informatics* (pp. 141–164). Elsevier.

Suzuki, K. (2017). Overview of deep learning in medical imaging. *Radiological physics and technology*, *10*(3), 257–273.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 2818–2826).

Tham, Y.-C., Li, X., Wong, T. Y., Quigley, H. A., Aung, T., & Cheng, C.-Y. (2014). Global prevalence of glaucoma and projections of glaucoma burden through 2040: a systematic review and meta-analysis. *Ophthalmology*, *121*(11), 2081–2090.

Thompson, A. C., Jammal, A. A., Berchuck, S. I., Mariottoni, E. B., & Medeiros, F. A. (2020). Assessment of a segmentation-free deep learning algorithm for diagnosing glaucoma from optical coherence tomography scans. *JAMA ophthalmology*, *138*(4), 333–339.

Thompson, A. C., Jammal, A. A., & Medeiros, F. A. (2020). A review of deep learning for screening, diagnosis, and detection of glaucoma progression. *Translational Vision Science & Technology*, *9*(2), 42–42.

Trivli, A., Koliarakis, I., Terzidou, C., Goulielmos, G. N., Siganos, C. S., Spandidos, D. A., ... Detorakis, E. T. (2019). Normal-tension glaucoma: Pathogenesis and genetics. *Experimental and therapeutic medicine*, *17*(1), 563–574.

Tsang, S.-H. (2018). https://towardsdatascience.com/review-xception-with-depthwise-separable-convolution-better-than-inception-v3-image-dc967dd42568.

Vaghefi, E., Hill, S., Kersten, H. M., & Squirrell, D. (2020). Multimodal retinal image analysis via deep learning for the diagnosis of intermediate dry age-related macular degeneration: a feasibility study. *Journal of ophthalmology*, *2020*.

Van Melkebeke, L., Barbosa-Breda, J., Huygens, M., & Stalmans, I. (2018). Optical coherence tomography angiography in glaucoma: a review. *Ophthalmic research*, *60*(3), 139–151.

Wang, W., Xu, Z., Yu, W., Zhao, J., Yang, J., He, F., ... Chen, Y. e. a. (2019). Two-stream CNN with loose pair training for multi-modal AMD categorization. In *International conference on medical image computing and computer-assisted intervention* (pp. 156–164).

Weinreb, R. N., & Khaw, P. T. (2004). Primary open-angle glaucoma. *The Lancet*, *363*(9422), 1711–

1720.

Wigdahl, J., Guimarães, P., & Ruggeri, A. (2019). Chapter 5 - Automatic landmark detection in fundus photography. In E. Trucco, T. MacGillivray, & Y. Xu (Eds.), *Computational retinal image analysis* (p. 79-93). Academic Press. Retrieved from `https://www.sciencedirect.com/science/article/pii/B9780081028162000058` doi: 10.1016/B978-0-08-102816-2.00005-8

Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, *9*(4), 611–629.

Yoo, T. K., Choi, J. Y., Seo, J. G., Ramasubramanian, B., Selvaperumal, S., & Kim, D. W. (2019). The possibility of the combination of OCT and fundus images for improving the diagnostic accuracy of deep learning for age-related macular degeneration: a preliminary experiment. *Medical & biological engineering & computing*, *57*(3), 677–687.

Yu, H.-H., Maetschke, S., Antony, B., Ishikawa, H., Wollstein, G., Schuman, J., & Garnavi, R. (2020, 07). Estimating global visual field indices in glaucoma by combining macula and optic disc OCT scans using 3D convolutional neural networks. *Ophthalmology Glaucoma*, *4*. doi: 10.1016/j.ogla.2020.07.002

Appendices

A Figures and Tables

A.1 Recorded Image and Check-up Dates Time Differences.

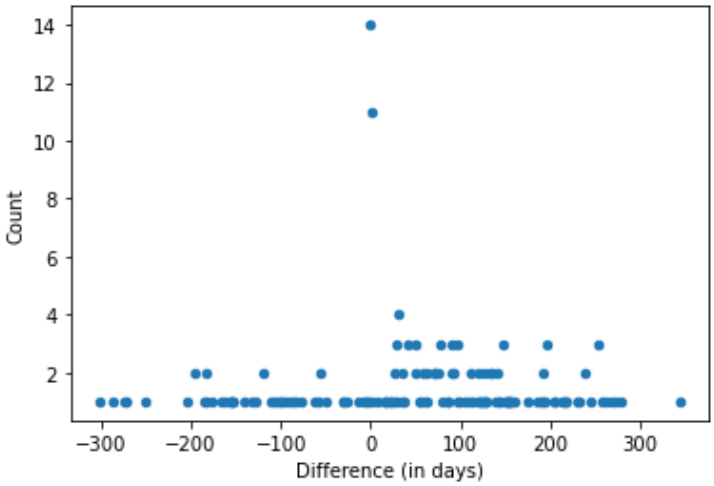


Figure 34. Time differences in recorded dates when matching images with the patient check-up dates, excluding a difference of 0 days.

A.2 Sample Sizes for Datasets Used To Train and Test Classification and VF Prediction Models.

Table 21. Number of samples in the datasets used for classification.

	Glaucoma		Healthy		**Total**
	Count	%	Count	%	
Test set	17	65.38%	9	34.62%	**26**
Combined train	68	67.33%	33	32.67%	**101**
OCT D train	150	76.53%	46	23.47%	**196**
OCT M train	156	76.10%	49	23.90%	**205**
OCTA D train	100	71.94%	39	28.06%	**139**
OCTA M train	88	69.29%	39	30.71%	**127**
Fundus train	334	83.62%	56	14.36%	**390**

Table 22. Number of samples in the datasets used for visual field prediction.

	Mild		Moderate		Severe		**Total**
	Count	%	Count	%	Count	%	
Test set	8	47.06%	5	29.41%	4	23.53%	**17**
Combined train	32	47.06%	17	25.00%	19	27.94%	**68**
OCT D train	61	40.67%	42	28.00%	47	31.33%	**150**
OCT M train	62	39.74%	42	26.92%	52	33.33%	**156**
OCTA D train	47	47.00%	28	28.00%	25	25.00%	**100**
OCTA M train	39	44.32%	25	28.41%	24	27.27%	**88**
Fundus train	139	41.62%	83	24.85%	112	33.53%	**334**

A.3 Parameter Tuning Top-5 Results for Classification Models.

Table 23. Classification OCTA 3 × 3 D. Arch. = architecture, FF = filter factor, LR = learning rate. The evaluation metric shows mean AUC (standard deviation) for the validation over the five cross-validation folds.

	Metric	Hyperparameters							
#	AUC	Arch.	Activation	Batch size	Dropout	FF	Input size	Layers frozen	LR
1	0.909 (0.037)	4layers	ReLU	18	0.4	1	[128, 128]	None	0.005
2	0.895 (0.112)	MobileNet	ReLU	18	0.6	1	[128, 128]	All	0.001
3	0.890 (0.088)	4layers	ReLU	6	0.4	1	[64, 64]	None	0.001
4	0.889 (0.041)	MobileNet	ReLU	10	0.2	1	[128, 128]	All	0.001
5	0.883 (0.137)	4layers	ReLU	32	0.4	1	[96, 96]	None	0.005

Table 24. Classification OCT D. Arch. = architecture, FF = filter factor, LR = learning rate. The evaluation metric shows mean AUC (standard deviation) for the validation over the five cross-validation folds.

	Metric	Hyperparameters							
#	AUC	Arch.	Activation	Batch size	Dropout	FF	Input size	Layers frozen	LR
1	0.926 (0.036)	VGG16	ReLU	18	0.5	1	[64,64]	All	0.001
2	0.923 (0.037)	MobileNet	ReLU	18	0.7	1	[96, 64]	All	0.001
3	0.923 (0.032)	5layers	ReLU	14	0.2	4	[96, 96]	None	0.0005
4	0.921 (0.030)	VGG16	ReLU	18	0.7	1	[64,64]	All	0.001
5	0.921 (0.029)	4layers	ReLU	14	0.4	2	[64, 64]	None	0.001

Table 25. Classification Fundus. Arch. = architecture, FF = filter factor, LR = learning rate. The evaluation metric shows mean AUC (standard deviation) for the validation over the five cross-validation folds.

	Metric	Hyperparameters							
#	AUC	Arch.	Activation	Batch size	Dropout	FF	Input size	Layers frozen	LR
1	0.922 (0.038)	MobileNet	ReLU	32	0.7	1	[128, 128]	All	0.001
2	0.916 (0.035)	4layers	ReLU	18	0.4	2	[64, 64]	None	0.001
3	0.911 (0.041)	VGG16	ReLU	18	0.6	1	[128, 128]	All	0.001
4	0.910 (0.052)	VGG16	SELU	12	0.6	2	[128, 128]	All	0.001
5	0.910 (0.019)	MobileNet	ReLU	6	0.4	1	[128, 128]	None	0.001

Table 26. Classification Multi-input CNN. Arch. = architecture, FF = filter factor, LR = learning rate. The evaluation metric shows mean AUC (standard deviation) for the validation over the five cross-validation folds.

	Metric	Hyperparameters							
#	AUC	Arch.	Activation	Batch size	Dropout	FF	Input size	Layers frozen	LR
1	0.940 (0.024)	MobileNet	ReLU	18	0.4	3	[128, 128]	None	0.0005
2	0.900 (0.067)	4layers	ReLU	12	0.2	1	[128, 128]	None	0.001
3	0.792 (0.133)	4layers	ReLU	4	0.6	1	[64, 64]	None	0.001
4	0.790 (0.101)	4layers	ReLU	4	0.2	1	[64, 64]	None	0.005
5	0.778 (0.173)	5layers	ReLU	18	0.5	1	[96, 96]	None	0.005

Table 27. Classification ensemble method: XGBoosting with outputs from single-input CNNs. The evaluation metric shows mean AUC (standard deviation) for the validation over the five cross-validation folds.

	Metric	Hyperparameters				
#	AUC	γ	Estimators	Max tree depth	η	Subsample
1	0.877 0.087	4	1200	9	0.2	0.7
2	0.870 0.083	0.5	1000	3	0.2	1
3	0.870 0.083	2	1200	6	0.3	0.6
4	0.870 0.083	4	1000	3	0.2	0.6
5	0.863 0.082	4	1000	6	0.2	1

Table 28. Classification ensemble method: XGBoosting with features from single-input CNNs. The evaluation metric shows mean AUC (standard deviation) for the validation over the five cross-validation folds.

	Metric	Hyperparameters				
#	AUC	γ	Estimators	Max tree depth	η	Subsample
1	0.817 0.128	0	1200	3	0.1	0.6
2	0.811 0.138	0.5	1000	9	0.3	1
3	0.810 0.125	4	1200	3	0.2	1
4	0.804 0.139	0.5	800	9	0.2	1
5	0.804 0.121	4	1200	3	0.2	0.6

Table 29. Classification ensemble method: random forest with outputs from single-input CNNs. The evaluation metric shows mean AUC (standard deviation) for the validation over the five cross-validation folds.

	Metric	Hyperparameters					
#	AUC	Bootstrap	Estimators	Max tree depth	Max feat.	Split min	Leaf min
1	0.877 (0.087)	FALSE	500	30	sqrt	2	1
2	0.870 (0.083)	TRUE	100	10	auto	10	2
3	0.870 (0.083)	TRUE	50	30	auto	2	1
4	0.870 (0.083)	TRUE	500	60	sqrt	5	1
5	0.870 (0.083)	TRUE	100	60	sqrt	5	2

Table 30. Classification ensemble method: random forest with features from single-input CNNs. The evaluation metric shows mean AUC (standard deviation) for the validation over the five cross-validation folds.

	Metric	Hyperparameters					
#	AUC	Bootstrap	Estimators	Max tree depth	Max feat.	Split min	Leaf min
1	0.849 (0.104)	TRUE	50	30	sqrt	2	4
2	0.832 (0.087)	TRUE	50	10	auto	5	1
3	0.825 (0.081)	FALSE	100	60	auto	10	4
4	0.823 (0.112)	TRUE	500	60	auto	5	1
5	0.823 (0.128)	TRUE	100	60	sqrt	2	4

A.4 Parameter Tuning Top-5 Results for Visual Field Prediction Models.

Table 31. VF OCTA 3×3 D. Arch. = architecture, FF = filter factor, LR = learning rate. The evaluation metric shows mean AUC (standard deviation) for the validation over the five cross-validation folds.

Rank	Metrics		Hyperparameters							
#	MSE	R^2	Arch.	Activation	Batch size	Dropout	FF	Input size	Layers frozen	LR
1	22.700 (4.236)	0.495	4layers	SELU	18	0	1	[64, 64]	None	0.001
2	23.873 (7.568)	0.383	4layers	ReLU	6	0	1	[96, 96]	None	0.005
3	23.953 (3.968)	0.406	4layers	ReLU	12	0	2	[128, 128]	None	0.001
4	24.026 (5.037)	0.428	4layers	ReLU	12	0	1	[96, 96]	None	0.001
5	24.052 (10.733)	0.442	4layers	ReLU	12	0.1	3	[96, 96]	None	0.001

Table 32. VF OCT D. Arch. = architecture, FF = filter factor, LR = learning rate. The evaluation metric shows mean AUC (standard deviation) for the validation over the five cross-validation folds.

	Metrics		Hyperparameters							
#	MSE	R^2	Arch.	Activation	Batch size	Dropout	FF	Input size	Layers frozen	LR
1	40.421 (12.775)	0.217	4layers	SELU	6	0.4	3	[64, 64]	None	0.001
2	41.356 (14.279)	0.194	4layers	SELU	4	0.3	3	[96, 96]	None	0.001
3	41.814 (15.874)	0.184	4layers	SELU	6	0.1	3	[64, 64]	None	0.001
4	41.872 (17.570)	0.216	Xception	SELU	32	0.2	4	[128, 128]	None	0.001
5	41.891 (16.452)	0.219	4layers	SELU	10	0	2	[64, 64]	None	0.001

Table 33. VF Fundus. Arch. = architecture, FF = filter factor, LR = learning rate. The evaluation metric shows mean AUC (standard deviation) for the validation over the five cross-validation folds.

	Metrics		Hyperparameters							
#	MSE	R^2	Arch.	Activation	Batch size	Dropout	FF	Input size	Layers frozen	LR
1	51.310 (10.517)	0.309	MobileNet	ReLU	12	0.3	2	[128, 128]	All	0.001
2	51.524 (10.040)	0.291	MobileNet	ReLU	32	0.2	2	[128, 128]	All	0.001
3	51.974 (10.141)	0.283	MobileNet	ReLU	18	0.2	1	[128, 128]	All	0.001
4	51.999 (7.720)	0.288	MobileNet	SELU	6	0.3	2	[128, 128]	All	0.001
5	52.038 (7.307)	0.277	MobileNet	SELU	32	0.4	1	[128, 128]	All	0.001

Table 34. VF Multi-input CNN. Arch. = architecture, FF = filter factor, LR = learning rate. The evaluation metric shows mean AUC (standard deviation) for the validation over the five cross-validation folds.

	Metrics		Hyperparameters							
#	MSE	R^2	Arch.	Activation	Batch size	Dropout	FF	Input size	Layers frozen	LR
1	29.424 (7.945)	0.322	VGG16	SELU	8	0	1	[64, 64]	All	0.005
2	29.896 (13.005)	0.343	MobileNet	ReLU	12	0	2	[128, 128]	All	0.0005
3	30.372 (9.912)	0.376	MobileNet	ReLU	8	0	1	[128, 128]	None	0.0005
4	31.404 (13.235)	0.285	5layers	SELU	12	0.4	1	[96, 96]	None	0.0005
5	33.178 (10.045)	0.259	5layers	SELU	18	0.1	4	[128, 128]	None	0.0005

Table 35. VF ensemble method: XGBoosting with outputs from single-input CNNs. The evaluation metric shows mean AUC (standard deviation) for the validation over the five cross-validation folds.

	Metrics		Hyperparameters				
#	MSE	R^2	γ	Estimators	Max tree depth	η	Subsample
1	19.198 (8.874)	0.620	4	800	3	0.2	0.7
2	19.281 (8.338)	0.620	0.5	1000	3	0.3	0.8
3	19.287 (7.193)	0.620	0	1200	3	0.4	0.8
4	19.716 (12.872)	0.614	4	1000	3	0.1	0.7
5	19.786 (9.629)	0.591	0.5	1000	9	0.4	0.8

Table 36. VF ensemble method: XGBoosting with features from single-input CNNs. The evaluation metric shows mean AUC (standard deviation) for the validation over the five cross-validation folds.

	Metrics		Hyperparameters				
#	MSE	R^2	γ	Estimators	Max tree depth	η	Subsample
1	21.944 (11.996)	0.525	4	1200	6	0.3	0.8
2	24.335 (10.346)	0.468	0	800	9	0.4	0.8
3	24.659 (13.533)	0.493	0.5	1200	3	0.2	0.7
4	25.020 (11.365)	0.458	0.5	1000	3	0.4	0.8
5	25.219 (11.295)	0.504	0.5	1000	6	0.2	0.7

Table 37. VF ensemble method: random forest with outputs from single-input CNNs. The evaluation metric shows mean AUC (standard deviation) for the validation over the five cross-validation folds.

	Metrics		Hyperparameters					
#	MSE	R^2	Bootstrap	Estimators	Max tree depth	Max feat.	Split min	Leaf min
1	19.064 (11.099)	0.627	TRUE	50	50	sqrt	5	1
2	20.046 (11.922)	0.597	TRUE	100	60	sqrt	2	1
3	20.065 (11.821)	0.605	TRUE	500	50	sqrt	5	1
4	20.296 (11.481)	0.593	TRUE	500	50	sqrt	2	1
5	20.638 (12.559)	0.593	TRUE	500	50	sqrt	5	2

Table 38. VF ensemble method: random forest with features from single-input CNNs. The evaluation metric shows mean AUC (standard deviation) for the validation over the five cross-validation folds.

	Metrics		Hyperparameters					
#	MSE	R^2	Bootstrap	Estimators	Max tree depth	Max feat.	Split min	Leaf min
1	23.870 (5.748)	0.493	FALSE	50	50	auto	10	1
2	24.111 (5.811)	0.489	FALSE	500	30	auto	10	1
3	24.180 (10.302)	0.488	FALSE	100	50	sqrt	10	2
4	24.254 (5.428)	0.486	FALSE	100	60	auto	10	2
5	24.254 (5.428)	0.486	FALSE	100	30	auto	10	2

B Equations

B.1 Classification Metrics Formulas

In the following equations, TP = true positives, TN = true negatives, FP = false positives, and FN = false negatives.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{B.1}$$

$$Precision = \frac{TP}{TP + FN} \tag{B.2}$$

$$Recall = \frac{TP}{TP + FP} \tag{B.3}$$

C Code Description

Scripts are listed in alphabetical order.

`config.py`: Contains the settings for which models to run and which parameter values to use.
`dataAugmentation.py`: Contains functions for image processing (e.g., contrast enhancement) and random data augmentation (e.g., shifting).
`dataSelection.py`: Contains methods to select which images to use for single-input and multi-input models.
`dataServerTransfer.py`: Code to obtain data from server and check the exact numbers of the dataset.
`dataProcessing.py`: Code to process raw information about data and connect the scan data with the patient metadata files, as well as check the exact numbers of the dataset.
`dataQualityEvaluation.py`: Contains functions to make an evaluation of image quality (noise, saccades, etc.) and functions to plot an overview of image quality.
`multiInput.py`: Contains multi-input processor and multi-input CNN classes for classifying glaucoma or predicting VF MD using all 2D data types.
`network2d.py`: Contains single-input processor and single-input CNN classes for classifying glaucoma or predicting VF MD using individual 2D data types.
`resultsEvaluation.py`: Contains functions to evaluate and visualize results.
`stackingEnsemble.py`: Implementation of stacking ensemble methods for classifying glaucoma or predicting VF MD.
`utils.py`: This file contains extra functions for saving files, reading scans, etc. that could be necessary in other classes.