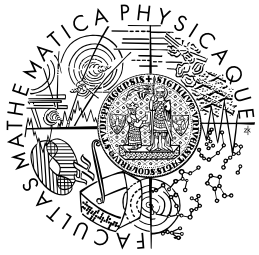

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Matúš Goliaš

Gradient boosted segmentation of retinal fundus images

Department of Software and Computer Science Education

Supervisor of the master thesis: Doc. RNDr. Elena Šikudová PhD.
Study programme: Computer Science (N1801)
Study branch: IPGVPH (1801T053)

Prague 2021

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ....

Author's signature

I want to thank my supervisor Doc. RNDr. Elena Šikudová PhD. for her assistance, advice and insurmountable patience. Also, I would like to thank my family for their support in the uncertain times that accompanied my writing of this work.

Názov práce: Gradientová segmentácia snímok očného pozadia

Autor: Matúš Goliaš

Katedra: Katedra softwaru a výuky informatiky

Vedúci práce: Doc. RNDr. Elena Šikudová PhD., Katedra softwaru a výuky informatiky

Abstrakt:
V posledných rokoch sa zvýšilo využívanie automatických metód v lekárskej diagnostike. Značný počet publikácií bol zameraných na analýzu očných poškodení a chorôb. Jedným z najzávažnejších ochorení oka je zelený zákal (glaukóm). Spôsobuje poškodenie očných nervov a postupnú stratu zraku. Podstatný krok k rýchlejšej diagnóze tejto choroby je presná segmentácia terča zrakového nervu a jeho exkavácie. Táto úloha je náročná z dôvodu mnohých druhov poškodenia očnej sietnice, rôznych prístupov k získavaniu obrázkov očného pozadia a chýb spôsobených zachytávaním obrazu v kamere. Táto práca popisuje prahovací algoritmus založený na postupnom zlepšovaní zvoleného prahu pre segmentáciu terča zrakového nervu. Definujeme funkciu podobnosti objektu k terču ako riadiaci prvok vylepšovania prahu. Následne poskytneme algoritmus pre nájdenie exkavácie zrakového nervu založený na klasifikácií superpixelov. Predkladáme prístup používajúci gradientom zosilnené rozhodovacie stromy, ktoré ukazujú lepšie výsledky oproti náhodnému lesu a mechanizmus podporných vektorov. Ďalej vyhodnotíme predstavené algoritmy na verejne dostupnej dátovej množine snímok očného pozadia. Nakoniec prediskutujeme získané výsledky a rozdiely medzi jednotlivými metódami. Poslednou súčasťou tejto práce je implementácia našich algoritmov v programovacom jazyku Python.

Kľúčové slová: segmentácia, prahovanie, superpixely, terč zrakového nervu, exkavácia zrakového nervu

Title: Gradient boosted segmentation of retinal fundus images

Author: Matúš Goliaš

Department: Department of Software and Computer Science Education

Supervisor: Doc. RNDr. Elena Šikudová PhD., Department of Software and Computer Science Education

Abstract: Over the recent years, there has been an increase in the use of automatic methods in medical diagnosis. A significant number of publications have analysed eye disorders and diseases. One of the most severe eye conditions is glaucoma. It damages optic nerves and causes gradual loss of vision. An essential step towards a faster diagnosis of this disease is accurate segmentation of the optic disc and cup. This task is difficult due to many retinal defects, different image acquisition techniques, and artefacts caused by imaging devices. This thesis describes an iterative threshold-based algorithm for extraction of the optic disc. An objective function quantifying object similarity to the optic disc is defined to direct the iteration. Following that, we introduce a superpixel-based classification algorithm for extraction of the optic cup. We propose the use of gradient boosted decision trees which outperform random forest and support vector machine. In addition, we evaluate the proposed algorithms and their alternatives on a publicly available retinal fundus image dataset. Finally, we discuss the reason for performance differences and implement our algorithms in the programming language Python.

Keywords: segmentation, thresholding, superpixels, optic disc, optic cup

Contents

Introduction

Glaucoma is a disease of the major nerve of vision, called the optic nerve. The optic nerve receives light-generated impulses from the retina and passes them to the brain. We recognize those electrical signals as vision. Glaucoma is characterized by progressive damage to the optic nerve, which starts with a subtle loss of peripheral vision. If it is not treated, it can progress to central vision and cause blindness. Glaucoma is generally associated with higher pressure in the eye (intraocular pressure). In some cases, it can appear in the eyes without elevated pressure. In these cases, it is often believed to be caused by poor blood flow to the optic nerve. The damage to the eye as a result of this disease is irreversible. It cannot be cured with medication, and it cannot be operated on either. However, if caught early, the treatment of glaucoma can halt the progressive damage and loss of vision [1].

Furthermore, glaucoma is the leading cause of blindness, which is the result of untreated disease. Even so, approximately 10% of people who receive proper treatment experience loss of vision. It is a chronic condition, and it has to be monitored for life. Diagnosis is the first step to the preservation of vision. Everyone is at risk for glaucoma. For instance, babies can be born with it. Approximately 1 in 10000 babies born in the United States have glaucoma. Certain demographics are more susceptible than others, and older people are at a higher risk of developing the disease. Next, there almost no symptoms of glaucoma. Gradual loss of peripheral vision is difficult to notice, and there is no pain associated with the disease. Let us include several statistics to put a perspective on the severity of glaucoma. It is estimated that over 3 million Americans have the disease, but only half of them know about it. Also, in the U.S., more than 120000 are blind from glaucoma, which accounts for 9% to 12% of all cases of blindness. Next, according to the World Health Organization, glaucoma is the second leading cause of blindness in the world. Estimates put the total number of suspected cases of glaucoma at over 60 million worldwide. These figures are taken from the webpage of Glaucoma research foundation [2]. We would like to specifically mention the source of worldwide statistics in different demographics projected into the year 2010 and 2020 [3].

Diagnosis of glaucoma using retinal imaging is carried out by calculating Cup to Disc Ratio (CDR), Inferior Superior Nasal Temporal (ISNT) rule, Disc Damage Likelihood Scale (DDLS) and Glaucoma Risk Index (GRI) which can be achieved by extracting the optic disc and the optic cup to calculate their ratio. The optic disc is the place where nerve fibres are bundled together and leave the eye towards the brain. It is also the entry point of blood vessels that pass blood to the retina. The optic cup is a depression at the centre of the disc. Orange or pink colouring of the OD is generally normal, whereas a disc region pallor indicates disease [4]. We will discuss the main structures of the retina and their significance in the next section.

As a result of the requirements posed by glaucoma diagnosis metrics, it is immaterial to correctly segment the optic nerve head in healthy individuals as well as in diseased cases. Furthermore, the extraction of the optic cup is based on the assumption of a correctly extracted disc region. Thus the algorithm has

to have two robust phases where the first one returns the segmented optic disc and the second one finds the cup region within the result of the previous phase. Not only is it difficult to extract the right boundary of the retinal structure but also to find an approximate optic disc region. This approximation is difficult due to the fact that the retinal fundus image can contain unwanted features as a result of a failure of the imaging technology or a pathologic case, see Figure 1 for difficult disc localisation examples. The segmentation of optic disc is a much more researched topic, and we are able to solve it with high accuracy. Although there has been a significant amount of work conducted on optic cup segmentation, this problem remains largely unsolved. The main challenges include differences between fundus images taken on different devices and rather small datasets often consisting of images taken only from a certain demographic [4], [5].

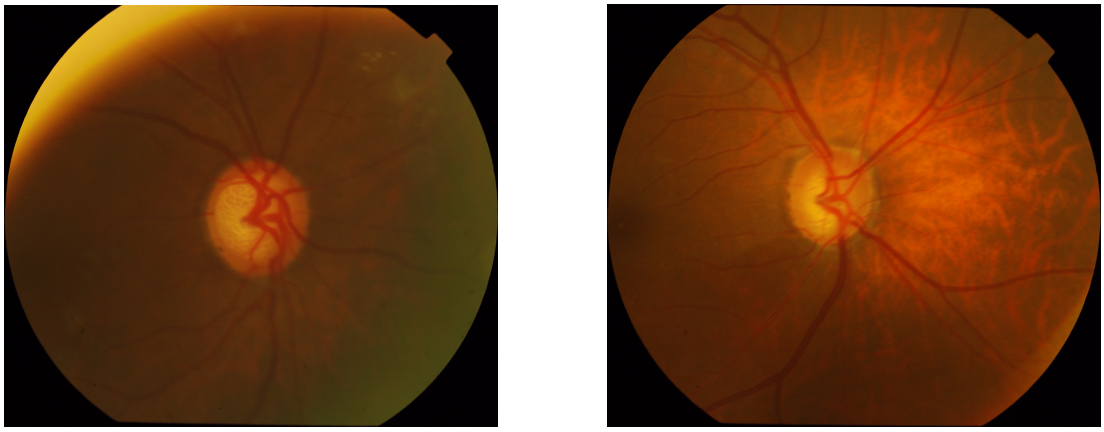

Figure 1: This figure shows examples of retinal fundus images, which can confuse algorithms searching for bright regions. Parts of the non-disc area with high-intensity values in the right image can match a circular or elliptical shape. That can confuse even algorithms searching for an approximate shape.

Retinal fundus image

A retinal fundus image is an image of the rear of an eye, known as the fundus, taken by a special fundus camera. It is used for the diagnosis of various eye diseases. Specialists take the images to check for different abnormalities. It contains various features which differ from human to human and also between normal and abnormal cases. Some of the structures visible in a fundus image are the central and peripheral retina, optic disc and macula. More information about the individual structures can be found in [4] where glaucoma and its relation to fundus images are described in greater depth. Furthermore, for a more comprehensive explanation behind fundus photography, see [6] which is the webpage for ophthalmic photographers' society. The available information is excerpted from [7].

An example of retinal fundus image can be seen in Figure 2. It shows the most important areas of a fundus image related to our work. These are the optic disc, the optic cup and blood vessels. Other areas such as the macula or fovea can be seen in the image as well. However, they are not very useful for the segmentation of the optic nerve head. The image also shows the boundaries of the optic disc

and cup marked by trained professionals as a part of the Drishti retinal fundus image dataset [8]. Now, we describe the optic nerve head in more detail since it is the primary focus of our work.

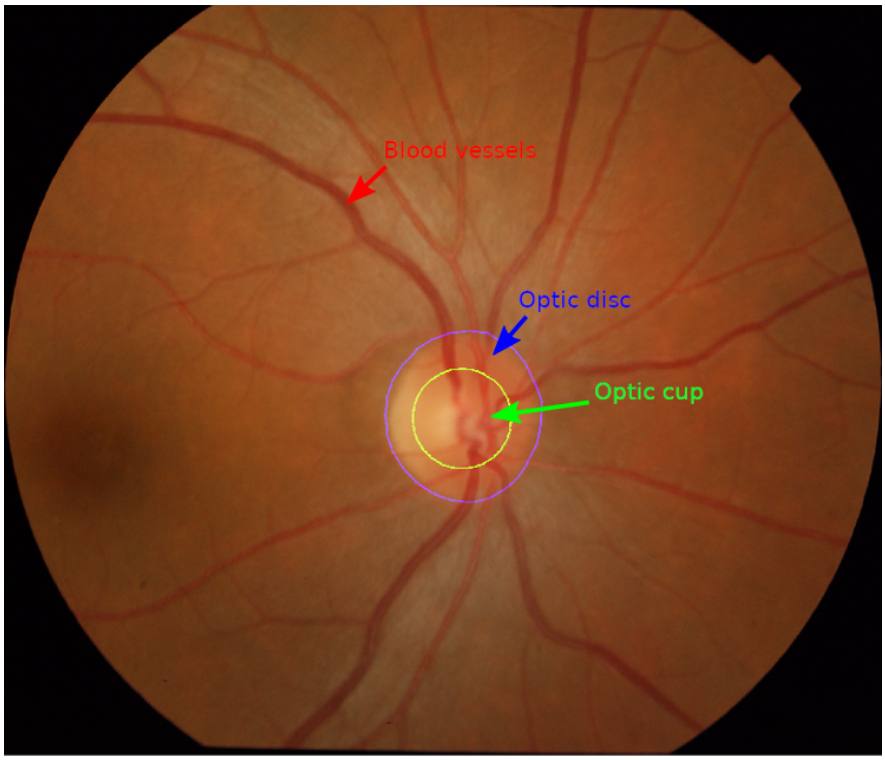

Figure 2: Example of a retinal fundus image with its main structures highlighted. The optic disc is marked with blue colour, and the optic cup is marked with green colour. Blood vessels are not highlighted. Nevertheless, there is an arrow pointing to a visible vessel. Trained professionals selected the optic disc and cup boundaries as part of the retinal fundus image dataset.

Optic disc

The optic disc is a bright central part of the retina. It is an entry point for the blood vessels outgoing into the retina. Its shape is approximately circular, although sometimes it can match an ellipse with high eccentricity due to the photographic projection. The size and shape vary from one person to another. The optic disc is often referred to as the blind spot since it does not contain rods and cones, which are photoreceptor cells. A normal optic disc is orange to pink in colour. A pale disc is an optic disc that varies in colour from pale pink or orange to white. It is an indication of diseased condition [4].

The optic disc is in general divided into three different regions: the optic cup, the neuroretinal rim and sometimes parapapillary atrophy (PPA), see [5]. Note that the cup is the central part of the optic disc, the neuroretinal rim is the area surrounding the cup and parapapillary atrophy surrounds the main region of the disc. We show an example of these features in Figure 3. In fact, for segmentation purposes, the desired area of the optic disc entirely excludes PPA. In addition, correctly selecting only the main region of the optic disc is one of the more considerable challenges of optic nerve head segmentation.

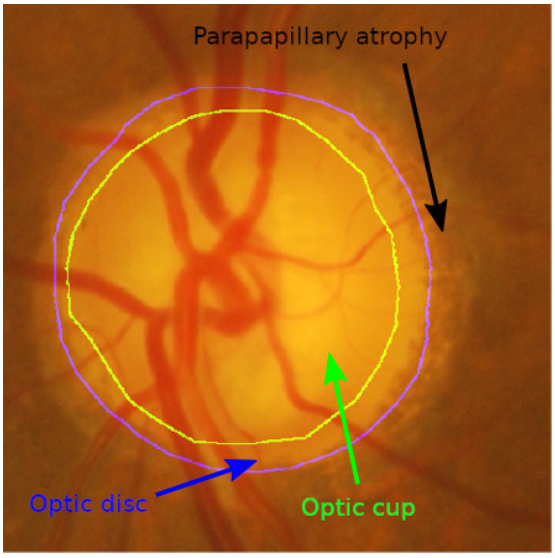

Figure 3: This figure shows an example of the optic nerve head area of a retinal fundus image with highlighted optic disc, optic cup and a general area of parapapillary atrophy. The disc and cup regions were selected by experts, the parapapillary atrophy region is chosen as an example and should not be taken as a clinician's selection.

Optic cup

An optic cup is a bright area in the centre of the optic disc. It is a depression where bundled retinal nerve fibres exit to the brain. Generally, the optic cup is small compared to the whole disc, and it enlarges due to dying nerve fibres. That is a result of increased pressure in the eye, or a loss of blood flow to the optic nerve [9]. In Figure 2 we can see optic cup region approximately half the size of the optic disc. We can compare that to Figure 3 where the optic cup is significantly larger with respect to the optic disc.

1. Related Work

There has been extensive work done on optic nerve head segmentation. Various methods proposed in the literature use different approaches to solving object segmentation. Among the applied algorithms, there are thresholding, clustering, level set and active shape modelling. A comprehensive list of the various approaches is available in [4] and also in [5]. The second survey does not distinguish between lists of algorithms belonging to a certain methodology. Nevertheless, it contains references to a large variety of different articles proposing algorithms for the segmentation of optic disc or optic cup. We focus mainly on the proposed approaches directly related to our work and mention others only for completeness.

Based on the extracted feature, we can divide the investigated problem into two main parts. Some algorithms work only with the disc whereas others work only with the cup. However, the most common approach is to segment both retinal features. There has been significantly more work done on the segmentation of the optic disc than the optic cup. In recent years, the focus has shifted more towards the cup since there are already methods capable of segmenting the optic disc with a high degree of accuracy.

The most common starting point of optic nerve head segmentation algorithms is optic disc localisation and region of interest selection. In [10], the approximate optic disc area coverage is described to be between 13% and 20%. Intensity-based OD localisation was applied as the first step of the algorithm. A simple fringe removal based method was used in [11]. Further, a de-hazing algorithm followed by Otsu's thresholding [12] composed the optic disc localisation step in [13]. The presence of artefacts caused by inadequate image acquisition was noted in [14] and border masking was used to overcome it. Optic disc localisation by wavelet transform and ellipse fitting was proposed in [15]. Region of interest detection methods give several candidates for the location of the optic disc. The right one is selected based on the highest intensity or similarity to a circle.

Thresholding is a simple segmentation approach extensively used in the optic disc and sometimes optic cup extraction. In [16], adaptive thresholding based on Gaussian window was applied to red and green colour channels for the extraction of the optic disc and optic cup. Similarly, the red and green channels were used for OD, and OC segmentation in [17]. The threshold was computed based on statistical features of the processed image improved by background suppression and histogram smoothing. Further, in [13], the value channel of HSV was used together with adaptive thresholding for optic disc segmentation. The best connected component from the binarised image was selected based on its area and eccentricity. Following that, an algorithm using the blue colour channel for OC thresholding was proposed in [18]. The noise in binarised images was removed using morphological operations, and a convex hull was computed to approximate the smooth boundary of the optic disc and cup.

An iterative multi-threshold approach that uses information from the red, green and blue channels was introduced in [19]. In each iteration, the colour distribution of the selected object is analysed, and a new threshold is computed based on the mean pixel value of the object. An improved multi-threshold method based on histogram analysis was described in [20]. An additional preprocessing

step was blood vessel removal from the source image for accurate segmentation.

Clustering and classification of pixels and superpixels is a second popular method in image segmentation. It was successfully applied to the problem of optic nerve head extraction. Superpixels are acquired using the SLIC (Simple Linear Iterative Clustering) algorithm. Unsupervised classification of superpixels for optic cup extraction was proposed in [21]. A low-rank representation of the data was computed, and an adaptive clustering on non-vessel superpixels was used for classification. The article reports good results, but it does not take into consideration imprecise segmentation of the optic disc. Next, a hard class assignment of edges by the k-means algorithm was used in [22] to extract the boundary of the optic disc and cup. For the improvement of hard class assignment results, fuzzy c-means classification of morphologically processed retinal images was proposed in [23]. Morphological operations were used for vessel noise suppression. Soft assignment of fuzzy logic was considered advantageous in both optic disc and cup extraction. Different improvement of the hard assignment was introduced in [24]. A Gaussian mixture model was used to classify pixels from a region of interest.

Furthermore, in [25], a superpixel based classification approach was introduced for the segmentation of both optic disc and optic cup. Several features, including colour channel histograms and centre surround statistics computed from a Gaussian pyramid, were defined. Finally, support vector machine classifiers were used for the classification of superpixels. A multi-scale method using SVM was proposed in [26]. This approach focuses on optic cup segmentation, and it does not consider imprecise disc extraction. The class assignment was done by multiple trained models on different superpixel scales. The final prediction is made by combining the results of the individual models using a unique integration model.

There have been attempts at training convolutional neural networks for optic nerve head segmentation [27]. However, the general concern is that retinal image datasets are too small. Thus, it is not easy to train the models properly on available data.

Our approach aims to improve the region of interest detection by introducing a refinement phase. Further, we look to improve threshold-based algorithms by introducing an empirical objective function describing the similarity of a selected object and optic disc. Finally, our optic cup segmentation algorithm builds on the superpixel classification methods and improves them by ensemble learning with gradient boosted decision trees.

2. Methodology

Implementing a complex algorithm in image processing or other disciplines is rarely an endeavour where we implement all necessary tools from scratch. Our proposed approach is no different, and we applied several well-established algorithms and techniques to perform the required tasks. Although it is unnecessary to describe every tool in detail, we provide a brief description of each algorithm we decided to use in our work. These short summaries provide general information and references to literature with additional details.

In addition, we describe techniques that are not part of our final proposed algorithm. We applied these techniques during our investigation of the problem, and very often, they pose an alternative to the algorithm we ended up using. As a part of this work, we decided to write down a comparison of different algorithms used to solve the presented tasks. For instance, we tried different clustering algorithms with various levels of success. To provide a brief overview of the performance of these algorithms, we compare them against our final chosen approach in Chapter 4.

2.1 Thresholding algorithms

One of the most common approaches for the extraction of objects from images is thresholding. The basic idea of the algorithm is finding a threshold for some greyscale image which separates the foreground object from its background. This type of algorithm is used when the segmented objects are distinguishable from their surroundings based on intensity. It is an approach that requires a careful selection of the colour channel in which we search for the threshold. In general, it also performs better on images with higher contrast, so algorithms such as contrast stretching are often used in conjunction with threshold searching techniques, see [12].

There are several types of thresholding algorithms. Some analyse the image histogram to find peaks and valleys from which they select the most probable threshold. For instance, these algorithms can smooth the histogram and select the deepest valley or, in the case of a popular Otsu's method, select the threshold maximising inter-class variance, see [12]. We can observe the histogram of a well separable optic disc region in Figure 2.1. We computed the histogram in the figure from the red channel of the image. It clearly shows two prominent peaks. In addition, the right image in the figure shows the result of Otsu's thresholding in the red channel. We can take, for example, the deepest point of the valley in between the two peaks as an optic disc separating threshold. Unfortunately, most examples do not show a clear bimodal histogram, and it is not easy to apply simple thresholding algorithms to our task.

Another class of thresholding algorithms uses the information obtained from the thresholded image or differences between the original and binarised version. This class is called thresholding algorithms based on attribute similarity [12]. We can observe that the optic disc and cup have a clear elliptical shape, see Figure 2.1 for an example, and we are interested in the algorithm which can find objects with a specific shape.

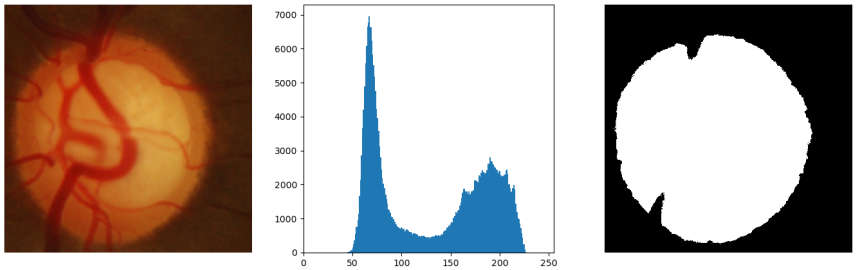

Figure 2.1: Optic disc area of a well separable fundus retinal image (on the left) with the histogram of its red channel (in the middle) and the result of thresholding by Otsu's method (on the right).

2.2 Simple linear iterative clustering

Simple linear iterative clustering (SLIC) is a method that separates the input image into a predefined number of superpixels based on their colour similarity and their distance in the image plane. Achanta et al. first introduced it in [28]. Superpixel is a general name for an object consisting of several related pixels. Superpixel-based classification methods have become increasingly popular in object segmentation due to a significant computational cost reduction compared to pixel-based approaches. The most popular algorithm for superpixel separation is SLIC. Its main principle is similar to K-means clustering. First, the algorithm selects N centres at regular grid points within the image. Then it assigns similar pixels to the groups defined by the centres and updates the centre positions. The algorithm clusters pixels of the image in a five-dimensional space $labxy$. The feature space comprises two main parts: lab and xy. The first one denotes the L, a and b channels of the $CIELAB$ colour space, and xy is the pixel position in the image plane. The distance measure introduced by the authors of SLIC ensures the compactness of created superpixels. It is denoted by D' in the following formula.

$$\begin{aligned} d_c &= \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2} \\ d_s &= \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \\ D' &= \sqrt{\left(\frac{d_c}{N_c}\right)^2 + \left(\frac{d_s}{N_s}\right)^2} \end{aligned} \tag{2.1}$$

Indices i and j in the formula denote a pair of pixels considered for grouping. The terms N_c and N_s are maximum colour and spatial distances expected within a given cluster. The maximum spatial distance should correspond to the sampling interval $N_s = S = \sqrt{N/K}$ where K is the desired number of superpixels. Determining the maximum colour distance is more difficult, and so the authors of [28] use constant $N_c = m$. We can write the simplified distance measure D as follows.

$$D = \sqrt{d_c^2 + \left(\frac{d_s}{S}\right)^2 m^2} \tag{2.2}$$

In our work, we use the algorithm implemented in the library scikit-image [29] for Python. In Figure 2.2 we can see an example of the SLIC algorithm applied on the optic disc region of a retinal fundus image. We can observe that the created superpixels preserve the shape of the optic disc and also follow the bright areas which belong to the optic cup.

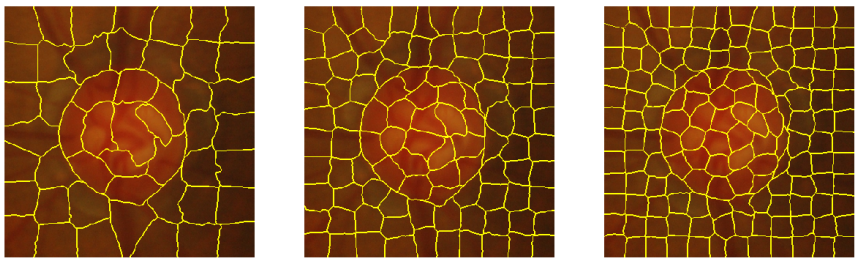

Figure 2.2: Superpixels generated by the SLIC algorithm on the same image with different values of K. We used $K = 48$, 96 and 144 for the left, centre and right example respectively.

2.3 Clustering algorithms

Similar to thresholding algorithms described in Section 2.1, clustering approaches separate data examples into multiple groups based on their features. The main difference is that clustering algorithms try to assign similar examples to the same class, whereas thresholding algorithms try to find a separating threshold for the two classes. We can describe a subset of thresholding algorithms as clustering into exactly two classes [12].

Given two data points, their similarity is defined by their distance. Clustering algorithms can use one of several standard functions as a measure of point dissimilarity. Let us denote Minkowski distance by *dist*. Then it is defined as follows.

$$dist = \left(\sum_{i=1}^{d} |x_{ji} - x_{ki}|^n\right)^{\frac{1}{n}} \tag{2.3}$$

Minkowski distance is probably the best-known metric, and given $n = 2$ we arrive at Euclidean distance. We are working with image pixels that carry information about intensity and position in a 2D plane. Minkowski distance is best suited for this type of data. For a comprehensive list of distance metrics used for clustering, refer to [30].

Let us assume that we have a distance metric. Now we can apply a clustering algorithm to our dataset, which groups together data examples based on the value of our selected metric. We can define several types of clustering algorithms that differ based on their approach to data grouping.

The first notable class is called algorithms based on partition. Two of the most famous examples belonging to this group are K-means and K-medoids. These algorithms are based around the idea that a centre of data points is the centre of the corresponding cluster [30].

The second class of clustering algorithms represents those based on hierarchy. This class is divided into two basic principles: agglomerative and divisive clustering. The basic idea of this group of algorithms is that they define a starting point, such as every data example is one cluster. Then they create a hierarchy on the data examples by merging selected clusters or dividing one. The decision which clusters to use for these operations is made based on the distance between them. For instance, it can be an average distance between each pair of points from the clusters given by our selected metric. Hierarchical algorithms often build the entire tree representing the data example hierarchy, and we can choose where to cut this tree to obtain the required number of clusters. An example of this class of algorithms is the Birch algorithm [30].

Fuzzy theory is the base for another set of algorithms. The basic idea of fuzzy logic is that a data point does not have a hard assignment. For instance 0 or 1 depending on whether it belongs to a cluster or not. Instead, the assignment is changed to continuous in the interval from 0 to 1. It is called a soft assignment, and it allows us to describe the belonging relationship between data points more reasonably. The examples of this class include Fuzzy c-means and Fuzzy c-shells [30] It is not easy to determine whether something belongs to our segmented object or the background. Therefore algorithms based on the fuzzy theory are of significant interest to us.

The fourth class of algorithms which we introduce is based on data distribution. The basic idea is that given an existing distribution in the original data, the points generated from the same distribution belong to the same cluster. One of the algorithms which belong to this class is the Gaussian mixture model, which assumes that data points were generated from a multivariate normal distribution [30].

The final class of algorithm which we present is based on density. The basic idea is that algorithms consider data points lying in a high-density region of the data space to belong to the same cluster. Typical examples of this class include DBSCAN and Mean-shift algorithms. Contrary to the previous classes we introduced, the algorithms from this class often do not need an exact number of clusters as a parameter. Instead, they induce the number of clusters from the data itself [30].

In the figure 2.3 we can observe the results of algorithms highlighted in this section. We applied them to a toy dataset and used parameters that give us reasonable classification. The toy dataset contains three clusters with different variance. That allows us to show how do the individual algorithms handle clusters with varying sizes.

The presented list of clustering algorithm types is by no means exhaustive. Further information on the individual types and a comprehensive list of

Figure 2.3: Results of clustering algorithms on a toy dataset. Red, blue and orange colours represent different clusters. Black represents points that do not have an assignment.

algorithms belonging to them can be found in [30].

In our experiments, we used implementations of the algorithms mentioned above from the Python scikit-learn library [31]. Next, we generated the toy dataset in the figure 2.3 using this library as well. Finally, the implementation of the Fuzzy c-means algorithm is from the Python library scikit-fuzzy [32].

2.4 Morphological operations

Morphology represents a general concept, and it is known in many scientific disciplines. It revolves around the analysis of the structure or relationship of objects. For instance, in biology, morphology works with the structure and relationship of organisms. In signal and image processing, we work with mathematical morphology, which is a theoretical model based on the lattice theory [33].

Mathematical morphology is a theory for the analysis of planar and spatial structures. It is suitable for shape analysis of objects, and due to a simple mathematical formalism, it can be used to create powerful image analysis tools. The key idea of morphological analysis of images is extracting information about the relation of an image and a simple object called structuring element. This small probe has a predefined shape, and we are checking whether it matches local shapes in the image. We can define a variety of operations. However, the most common morphological transformations are dilation, erosion and complex operations built on top of them. In the figure 2.4, we can observe the standard set of morphological operations applied to a thresholded image from the Drishti dataset [8]. A comprehensive overview of these operations can be found in [34] together with details about their mathematical definition. In the following subsections, we briefly describe the individual operations in the area of binary image processing. The definitions presented in this work can be extended to greyscale images. The resulting morphological operations are a powerful tool in many areas of image processing, e.g., removal of small unwanted structures from images. There is more detailed information about both binary and greyscale operation in [35].

We used the fast implementation of morphological operations from the scikit-image library for Python [29]. Although the algorithms are not complicated, a fast implementation is required to process large images efficiently.

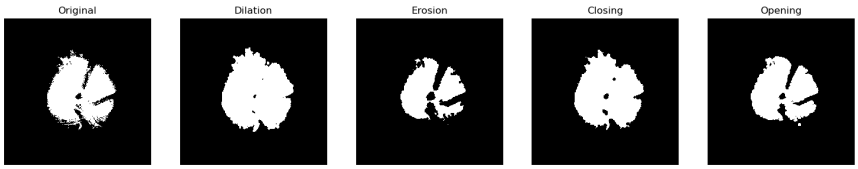

Figure 2.4: Image after the application of morphological dilation, erosion, closing and opening.

Morphological dilation

The definition of binary morphological dilation is given in Equation 2.4. The space E^N is Euclidian N-space [34].

$$A \oplus B = \{c \in E^N | c = a \ + b \text{ for some } a \in A \text{ and } b \in B\} \tag{2.4}$$

Dilation acts as a local maximum filter. It sets the value of the currently processed pixel to the maximum within the area given by the structuring element. In practice, it enlarges objects as it adds pixels on both their inner and outer boundary. Additionally, it removes holes in objects smaller than the structuring element. We can compare the effect of using circular structuring element to dilate an image in Figure 2.4.

Morphological erosion

The definition of binary morphological erosion is given in Equation 2.5. The space E^N is Euclidian N-space [34].

$$A \ominus B = \{x \in E^N | x + b \in A \text{ for every } b \in B\} \tag{2.5}$$

Erosion is the opposite operation to the dilation and acts as a local minimum filter. It sets the value of the currently processed pixel to the minimum within the area given by the structuring element. In practice, it shrinks objects by removing pixels from their boundary. In addition, it removes islands smaller than the structuring element. We can compare the result of erosion with the original image in Figure 2.4.

Morphological opening

The compound morphological opening operation consists of erosion followed by dilation. It removes narrow connections between regions, small islands and sharp peaks. The definition of the binary opening is written in Equation 2.6. An example of an image opened by a circular structuring element can be seen in Figure 2.4.

$$A \circ B = (B \ominus K) \oplus K \tag{2.6}$$

Morphological closing

The compound morphological closing operation consists of a dilation followed by erosion. It removes narrow gaps in objects and holes smaller than the structuring element. The definition of closing is given in Equation 2.7. An example of an image closed by a circular structuring element can be seen in the figure 2.4.

$$A \bullet B = (B \oplus K) \ominus K \tag{2.7}$$

Morphological gradient

The morphological gradient is the difference between the dilation and the erosion of an image. It can be used to find the pixel boundary of an object by applying a structuring element with a radius of 1. An example of gradient computed using circular structuring element can be seen in Figure 2.5. It is recommended to apply filtering to an image before computing the gradient because of the algorithm's high sensitivity to noise. Hence, we computed the gradient from the closed image that filled in the noisy boundary of the object.

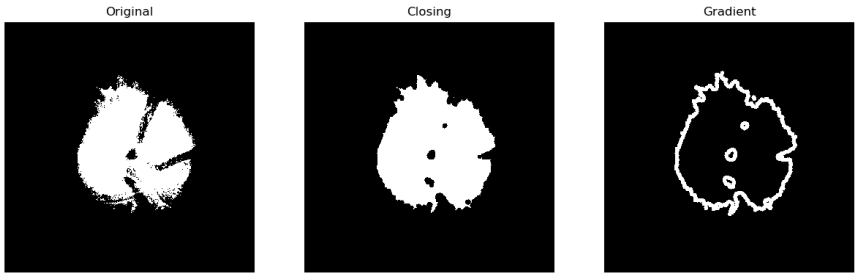

Figure 2.5: Thresholded image (left). The image after the application of morphological closing (middle). The morphological gradient computed from the closed image (right).

We use the fast implementation of morphological gradient from the OpenCV library [36] for Python.

Morphological top-hat transform

Top-hat transform operations are used to enhance features smaller than the structuring element. We distinguish two types of top-hat operation: the white and the black top-hat. The definition of the white top-hat is an image minus its morphological opening. It enhances bright spots and lines in the image. On the other hand, the definition of the black top-hat of an image is its morphological closing minus the image. It enhances the dark spots and lines in the image [33]. In Figure 2.6, we can see the result of top-hat operations. The image of the white top-hat returns noisy pixels and small islands, whereas the black top-hat returns the dark gaps in the object. Both operations return pixels around the boundary due to noise. They either respond to white pixels surrounded by black pixels or vice versa.

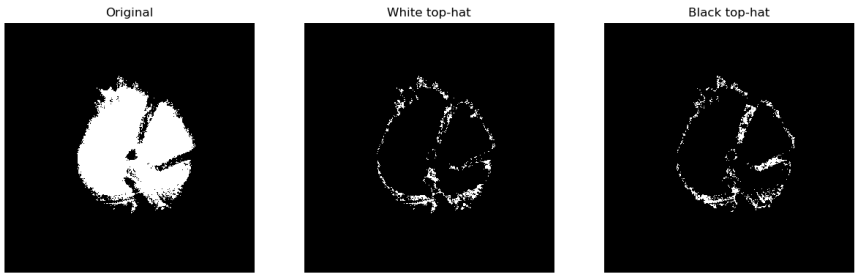

Figure 2.6: A binary image computed by thresholding (left). The result of morphological white top-hat (middle) and black top-hat (right) applied to the binary image.

2.5 Shape approximation algorithms

There are many cases when the shape of an object acquired through some segmentation technique is very far from a good approximation of our target. Sometimes, we can describe the sought shape in simple terms and apply post-processing to get a more desirable result. Our task is to extract elliptical objects from retinal images. Therefore, we focus on methods which produce convex shape. We present two algorithms that are useful in this regard. It is the computation of a convex hull and an ellipse fitting.

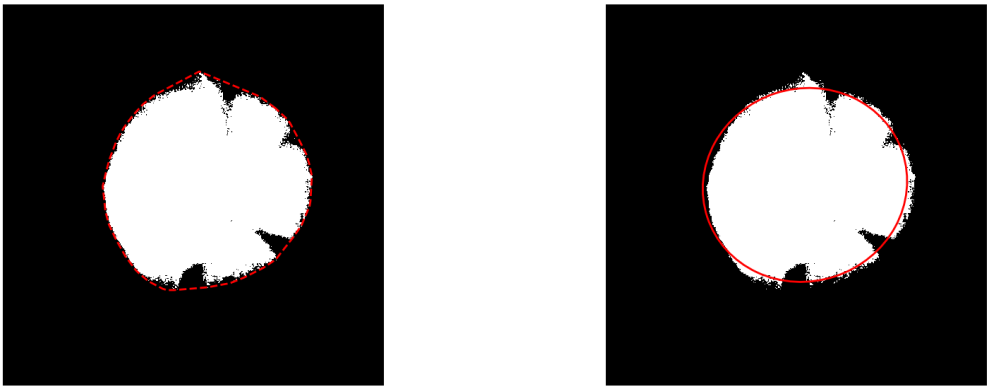

Figure 2.7: The left image shows the convex hull of the object selected by a threshold. The right image shows an ellipse fitted to the boundary of the object selected by a threshold.

Convex hull

The convex hull of an object is the smallest convex set that contains it. For a set of points in Euclidian space, it is the smallest convex set of points that contains the entire initial set. In a binary image, the convex hull of a group of pixels

is the smallest convex polygon that contains the entire shape [37]. An example of convex hull computed for an object extracted by thresholding can be seen in Figure 2.7 on the left. We use the implementation of the Quickhull algorithm from the Python library scipy [38].

Least squares ellipse fitting

In Section 2.1, we noted that the optic disc strongly resembles an ellipse. To improve the final segmented shape, we can apply an ellipse fitting algorithm to the boundary of our selected object to gain the best possible elliptical shape. The most common approach to ellipse fitting is using the least-squares method. Although it is prone to error on noisy or partially obscured data, it is sufficient for the optic disc boundary improvement task. The main reasons are that the optic disc in fundus retinal images is fully observable, and we can already extract a roughly elliptical shape. An example of an ellipse fitted to the boundary of a thresholded object can be seen in Figure 2.7 on the right. We used the least-squares ellipse fitting algorithm implemented in the OpenCV library ([36]) for Python.

2.6 Gaussian matched filters

One of the most common forms of pattern detection in signals are matched filters. They are nicely introduced in [39]. It presents mainly the definition and applications in the time domain of 1D signals. These filters enhance features in the input signal, which allows us to distinguish between, for instance, noise and radio signal. It can be easily observed that such filters can be extended into the 2D domain by applying a separable filter on a 2D signal. Furthermore, we do not need to restrict ourselves to continuous signals or the time domain. Therefore, we can use matched filters given by a discrete kernel on a matrix of pixels describing an image. The authors of [40] applied this technique to the detection of blood vessels after analysing their cross-section, which tends to resemble a Gaussian curve.

Whereas we can use morphological operations to improve the segmentation result of our algorithms, they are rarely a good starting point for the process itself. As proposed in [40], we can create a filter bank of rotated Gaussian kernels with different standard deviation for the individual axes. An example of such a bank can be seen in Figure 2.8. We use two as the ratio of standard deviation in the X-axis to the Y-axis. An actual application of the filter set needs a larger bank with kernel sizes scaled with respect to the size of the processed image.

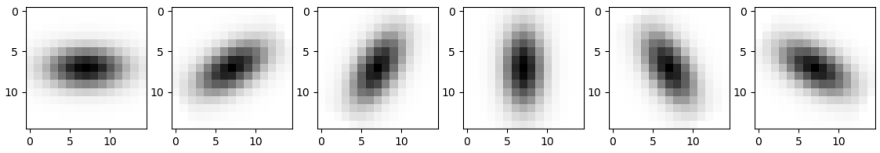

Figure 2.8: A bank of 6 rotated Gaussian matched filter kernels with angles linearly spaced in the range from 0 to 180 degrees. The kernels are square with the size of 15 pixels.

2.7 Gabor filters

We can define several more types of feature enhancing filters similarly to the previous section. One filter which is useful for detection of vessel-like structures in images is Gabor filter based on the human visual system [35]. It matches wave-like textures where it produces the strongest response. The filter is a Gaussian function modulated by a sine wave. It is composed out of a real and an imaginary part. We consider only the real part of the filter in our application. Similarly to the Gaussian matched filter, it is computed using oriented kernels. We have to produce a filter bank that considers all possible rotations of the texture we are trying to match. Considering sine wave with a longer period, we can observe that the kernels are composed of one valley and two adjacent hills. This shape resembles Laplacian of Gaussian in one direction, see Figure 2.9. It shows a simple Gabor filter bank that can be used to detect vessel-like structures in images. In the example, we use Gaussian with the ratio of standard deviation in the X-axis to the Y-axis equal to 1.5. Gabor filter bank in this form was used for blood vessel segmentation in [41]

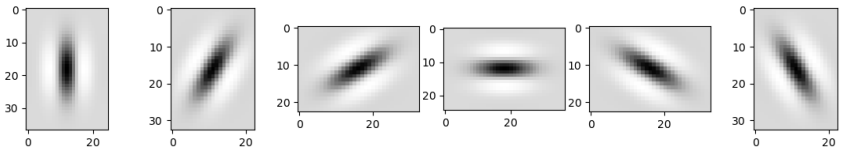

Figure 2.9: A bank of 6 rotated Gabor filter kernels with angles linearly spaced in the range from 0 to 180 degrees.

2.8 Support vector machine

We presented unsupervised classification in the form of clustering algorithms in Section 2.3. Let us consider that we have gold data available. Then, we can apply supervised techniques which can significantly improve our results. Many different supervised algorithms can be applied to our problem. Several of these algorithms are known to produce significantly better results.

The first algorithm is support vector machines, abbreviated to SVM. It is a supervised learning algorithm, which can be used for both classification and regression. It was first introduced in [42], and since then, it became one of the most popular and widely accepted machine learning algorithms. It is used in many areas, including glaucoma patient classification.

The algorithm searches for a hyperplane separating data points of two classes with the maximal margin. We assign the term margin to the perpendicular distance of the separating hyperplane and its closest data point. We can see an illustration of the found hyperplane and its margin in the figure 2.10. The optimisation problem of SVM can be described in two formulations. We call them the primary formulation, written in Equation 2.8, and the dual formulation,

[1]The image is taken from the Wikipedia entry on support vector machine `https://en.wikipedia.org/wiki/Support-vector_machine`.

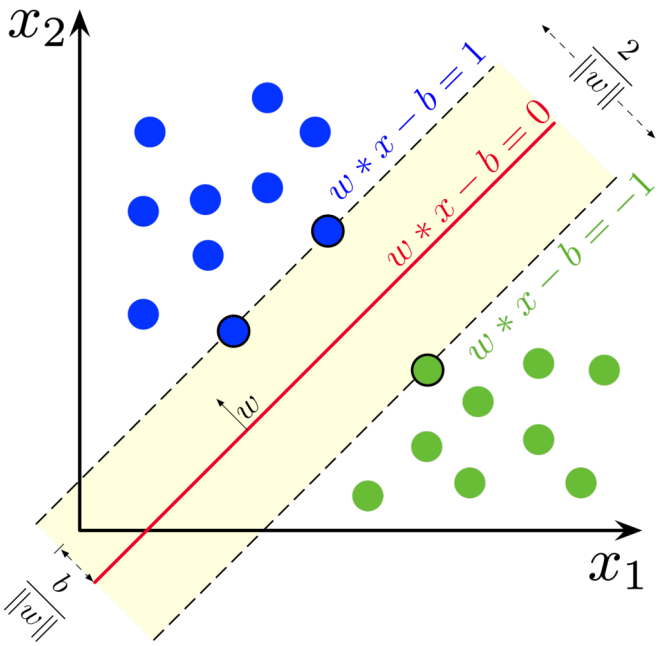

Figure 2.10: An example of the maximum margin search result of the SVM optimisation[1].

written in Equation 2.9. We also distinguish between a hard-margin and a soft-margin SVM. Equations 2.8 and 2.9 describe only the soft-margin algorithm which is more general. Hard-margin SVM assumes that our dataset is linearly separable and does not handle points that do not fulfil this assumption. This definition is generalised by introducing so-called slack variables, which allow data points to lie within the margin and even on the other side of the separating hyperplane. The algorithm with the slack variables is called soft-margin SVM, and every time we refer to the support vector machine algorithm, we mean the soft-margin definition [42]. Let us denote the prediction of SVM by $y(x_i) = wx_i + b$. Now, we define the primary formulation of SVM as

$$\underset{w,b}{\operatorname{argmin}} C \sum_i \xi_i + \frac{1}{2}||w||^2 \text{ given that } \xi_i \geq 0 \text{ and } t_i y(x_i) \geq 1 - \xi_i \tag{2.8}$$

where ξ_i is ith slack variable, w are the weights, and b are the biases of the model. Further, t_i marks the ground truth class of the ith example and $y(x_i)$ is the prediction of the model for the example x_i. Finally C is a regularisation constant that determines the trade-off between the accuracy of separation and generalisation. More general behaviour stems from allowing training examples to lie on the wrong side of the separating hyperplane. The symbols mentioned above apply to the dual formulation as well. We define the dual form as

$$L = \sum_i a_i - \frac{1}{2} \sum_i \sum_j a_i a_j t_i t_j K(x_i, x_j) \text{ given that} \tag{2.9}$$

$$\forall_i : C \geq a_i \geq 0 \text{ and } \sum_i a_i t_i = 0 \tag{2.10}$$

where L denotes the Lagrangian with multipliers a_i which we optimise and $K(x_i, x_j)$ is the kernel value computed from the data points x_i and x_j.

The SVM model remembers only the essential data points which define the decision boundary. They are called support vectors, and given a large dataset; there is usually only a small number of them. The low number of memorised examples makes the algorithm efficient, and it often performs significantly better than simple classifiers.

The model we defined until now searches for a linear separator - a hyperplane. In general, classification problems are not linearly separable, and it would be difficult to apply the algorithm to these problems. Therefore we use the kernel function K to transform our data into a higher dimensional space where it becomes linearly separable [42]. The most common kernels are polynomial and radial basis function (RBF). We define RBF as

$$K(x, y) = e^{-\gamma||x-y||^2} \tag{2.11}$$

where x and y are data points, and γ describes how far does the influence of a single example reach.

In conclusion, a soft-margin SVM with an appropriate kernel function can be used for classification of pixels or superpixels to determine the optic disc and cup regions.

2.9 Decision tree based algorithms

Decisions trees are one of the simplest supervised machine learning models. A decision tree is a flowchart-like tree structure, where each internal node represents a test on an attribute, each branch represents an outcome of the test, and each leaf node represents a prediction. We realise a prediction from a tuple of feature values by tracing a path from the root of the tree down to some leaf node. This final node represents the value assigned to the tuple. Training a decision tree means selecting a feature and an associated threshold for the given node. This process is repeated in the groups of data created by splitting the examples based on the threshold [43].

There are multiple well-known algorithms for decision tree learning. One of the simplest methods is the ID3 algorithm which tests each attribute at every node. The suitable property is generally selected based on information gain. Another algorithm is C4.5. It is capable of handling continuous attributes and missing values. Further, there is the CART algorithm which is based on binary splitting. In general, decision trees do not perform well compared to other machine learning models. However, they can be easily extended by ensemble methods to create robust and well-performing models [43].

Decision tree classifier is used in a variety of machine learning problems. Its main advantages are fast execution speed and a low amount of memory necessary for storing the tree. It is, however, prone to overfitting on the training dataset, which we can reduce by introducing methods such as pruning.

2.9.1 Random Forest

An extension of a traditional decision tree classifier that improves its generalisation at a minimal cost to its ability to learn the training data is a random forest (RF). It is an ensemble of trees where every tree gives a classification result, and the final class is selected based on some strategy. For instance, whichever class had the most votes is chosen. Each tree in the forest is constructed from a random subspace of the feature space of the data. Randomness allows the trees to complement each other in learning the training data and reduces the generalisation error. Random forest was first introduced in [44] and later improved and extended in [45]. We can see an illustration showing the basic structure of a random forest in Figure 2.11.

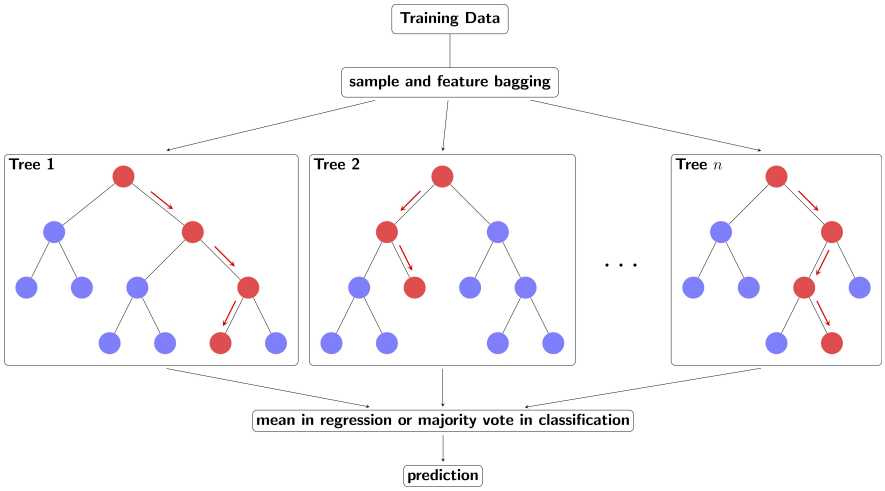

Figure 2.11: An illustration of a random forest. It can be used for both regression and classification without any significant changes[2].

2.9.2 Gradient boosted decision trees

The gradient boosted tree classifier trains a collection of trees similar to the random forest. The difference is that the RF classifier trains the trees independently, whereas the gradient boosted classifier trains the trees sequentially to correct the error of the previous ones. We can write a combined prediction of T trees as

$$y(x_i) = \sum_{t=1}^{T} y_t(x_i) \tag{2.12}$$

where $y_t(x_i)$ is the prediction of the tth tree for data point x_i. Next, we define the loss in the tth iteration, i.e. for the tth tree as

$$\ell^{(t)}(x_i) = \sum_i \left[\ell(t_i, y^{(t-1)}(x_i)) + y_t(x_i)\right] + \frac{1}{2}\lambda||W_t||^2 \tag{2.13}$$

[2]The image is taken from `https://tex.stackexchange.com/questions/503883/illustrating-the-random-forest-algorithm-in-tikz`.

where $\ell^{(t)}$ denotes the loss at the step t, x_i is a data point and t_i denotes its ground truth target. Next, $y^{(t-1)}(x_i)$ is the prediction of the classification forest for all the previous trees $1..(t-1)$, the full term $\ell(t_i, y^{(t-1)}(x_i))$ indicates the loss of the forest composed out of the first $t-1$ trees and $y_t(x_i)$ denotes the prediction for the tth tree. Finally, λ is the regularisation constant, and W_t denotes the parameters, i.e., leaf values of the tth tree. The minimisation of the loss happens in iterations similar to the gradient descent algorithm, however instead of improving a set of weights in each step, we improve the tth tree by selecting decision splits which minimise the loss. This new tree complements all of the previously created trees. For a more detailed description of the gradient boosted tree classifier see [46].

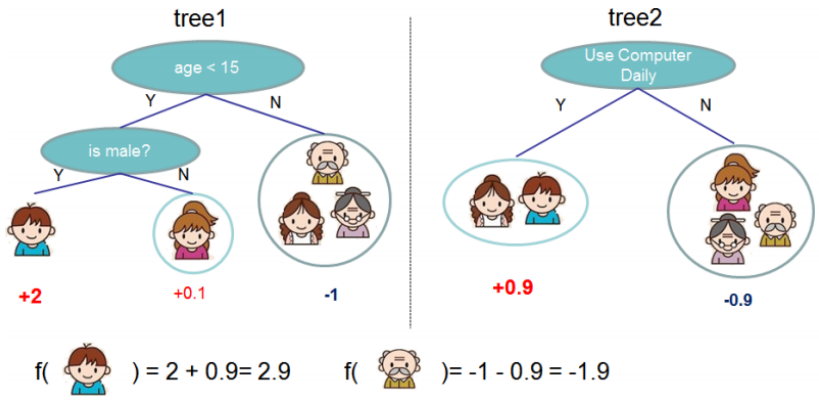

Figure 2.12: An illustration of the gradient boosted tree classifier. The image is taken from [46]

We do not need a special implementation of gradient boosted tree classifier, which can handle large amounts of data. Therefore, we use the implementation of the algorithm from the scikit-learn library for Python [31].

2.10 Evaluation criteria

Several metrics can be used to evaluate the quality of object segmentation. The metrics are related to each other. However, there is no easy way of converting values from one to another. A survey summarizing different approaches to optic nerve head segmentation has described all standard metrics which have been used in related works, see [4]. We describe the most important ones in this work for completeness. We also evaluate our algorithm using the described metrics to be more easily compared against previous work. In the following descriptions, we denote the ground truth object as A_g and the result of segmentation as A. Furthermore, symbols TP, FP and FN represent true positive, false positive and false negative areas. These values are defined at the pixel level as follows.

$$\begin{aligned} TP &= |A \cap A_g| \\ FP &= |A - A_g| \\ FN &= |A_g - A| \end{aligned} \tag{2.14}$$

Overlap area ratio

The most common criterium for quality of object segmentation is overlap area ratio which is also known as intersection over union, abbreviated to IoU. It describes the ratio of intersection between ground truth and our result to their union. We define it as

$$IoU = \frac{|A_g \cap A|}{|A_g \cup A|}. \tag{2.15}$$

The overlap area ratio assesses how well does the segmented area match the ground truth. It produces numbers ranging from 0 to 1, and a higher value of the ratio means better performance.

Non-overlap area ratio

A metric closely related to overlap area ratio is its complement. It is called non-overlap area ratio in [4], but it can be found under names such as an error rate or a segmentation error. We can define it as the complement to overlap area ratio such that their sum equals 1. We can write its formula as

$$err = 1 - \frac{|A_g \cap A|}{|A_g \cup A|}. \tag{2.16}$$

A non-overlap area ratio assesses the dissimilarity between the ground truth and the segmented area. It produces numbers ranging from 0 to 1, and a lower value means better performance.

Dice metric

The dice metric is very similar to the overlap area ratio. It describes the ratio of the intersection between the ground truth and the segmented area to their sum. We can write its formula as

$$dm = \frac{2 \cdot (|A_g \cap A|)}{|A_g| + |A|}. \tag{2.17}$$

Dice metric assesses how well does the segmented area match the ground truth. It produces numbers ranging from 0 to 1, and a higher value means better performance. The values produced by this metric are larger than values of overlap area ratio.

Recall

A standard metric to evaluate the quality of classification is recall, also known as sensitivity. It describes the ability of a model to identify diseased examples correctly. It is defined as the ratio of true positive to all actual positive examples. We can easily extend it to the segmentation case, where we denote each pixel as an example and compute the ratio of correctly segmented pixels. We can write its formula as

$$recall = \frac{TP}{TP + FN}. \tag{2.18}$$

Recall describes how well does the segmentation result encompass the ground truth area. It produces numbers ranging from 0 to 1, and higher values mean better performance. It is important to note that sensitivity does not consider false positive examples, and denoting the entire image as positive would result in a value equal to 1.

Precision

The second important metric for the evaluation of correctly classified examples is precision. Contrary to recall, it works with false positive examples rather than false negative ones. Therefore, it describes the ratio of true positive examples to all predicted positive examples. Similarly to the previous case, we can easily extend it to the segmentation case by denoting individual pixels as examples for classification. We can write its formula as

$$precision = \frac{TP}{TP + FP}. \tag{2.19}$$

Precision describes how well does the ground truth encompass our segmented area. It produces numbers ranging from 0 to 1, and higher values mean better performance. Similarly to the recall metric, it is important to note that it does not consider false negative examples. Therefore, depending on our definition of the ratio $\frac{0}{0}$, denoting the entire image as negative would result in the value equal to 1.

F-score

To alleviate the biases of precision and recall, we introduce their harmonic mean as another performance measure. This metric is commonly known as F-score, and the most used form is the F_1 score. A more general form of the metric is called F_β score where β factor is chosen so that recall is considered β times as important as precision. The F_1 score is defined as

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{2.20}$$

Further, we can define the F_β score as

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}. \tag{2.21}$$

The F_1 score is equal to the dice metric. We can prove this by substituting formulas for precision and recall into the F_1 formula. We obtain the dice metric by applying simple algebraic operations to the substituted form.

3. Proposed approach

Our algorithm combines several approaches to leverage the best properties out of each one. It includes an image preprocessing step, vessel extraction and region of interest search, which estimates the location of the optic nerve head. Then we extract the optic disc and apply additional image processing techniques to the OD area. The last step is optic cup extraction. A diagram showing all these steps with images of the intermediary results can be seen in Figure 3.1. We describe each of these steps in the following sections. In addition, we discuss the advantages and shortcomings of the techniques to provide better reasoning for our decisions.

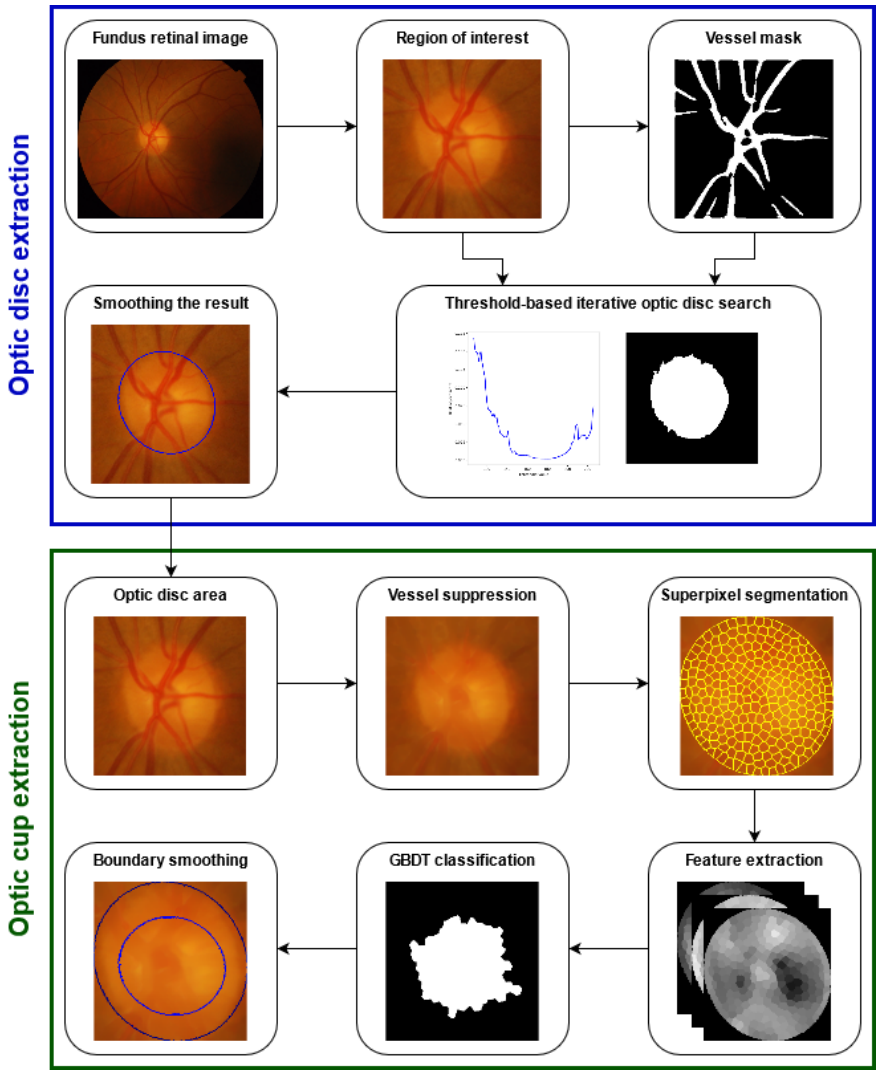

Figure 3.1: This figure shows a diagram of the combined optic disc and cup extraction algorithm. The parts for each retinal feature are separate, which is denoted by rectangles encompassing the smaller steps. We show the final results of both segmentation methods in their respective regions of interest rather than the full image.

3.1 Region of interest selection

The optic nerve head is a bright circular structure in a fundus retinal image that a human observer can identify without much difficulty. Teaching machines to see the obvious pattern is a much more daunting task. Primarily due to many visible retinal structures and disease-related image artefacts. For more details, see the section about retinal images in the introduction. Consequently, direct segmentation of the optic disc from the full retinal image is challenging. It is advantageous to find an approximate position of the optic nerve head and proceed from there. Localisation of the region of interest for the optic nerve head is a common preprocessing step for segmentation algorithms, see [4] and [5] for an overview. Several different methodologies can be used to extract the region of interest. In [47], a Hough transform is used to find the elliptical shape of the optic disc and an area around it is selected as RoI. An approach that considers 0.5% of the brightest pixels in the retinal image and selects a rectangular area around them was proposed in [48]. Another method proposed in [49] is based on a similar principle. The algorithm selects all pixels brighter than a predetermined threshold, and a region around the largest connected component is taken as RoI.

There is a large number of different algorithms and features that can be used for the detection of regions of interest. We could, for instance, use the fact that the optic disc is a bright circular area surrounded by darker background and compute Haar-like features where the maximum response is the optic disc. We propose an algorithm based on this idea which uses meta-information about retinal images to detect the region of interest in two subsequent phases. We can see a scheme of our RoI selection algorithm in Figure 3.2.

The first phase of the region selection

The optic nerve head is visible in all colour channels. The optic disc is best visible in the red channel, the optic cup in the green channel, and although the blue channel contains a significant amount of noise, OC is visible there. Based on that, we want to use information from all colour channels. Therefore, we want to choose coefficients c_r, c_g and c_b in Equation 3.1. The Green channel has the highest contrast, and therefore it is reasonable to set c_g to the highest value. In addition, setting c_b to the lowest value reduces the effect of noise. We can define greyscale intensity computation from an RGB image as

$$I = c_r \cdot R + c_g \cdot G + c_b \cdot B, \tag{3.1}$$

where R, G, B are the colour channels and I is the combined intensity. We decided to use colourimetric greyscale transform on the retinal image. The colour transformation is defined in Equation 3.2. It uses the principles of colourimetry to calculate the greyscale values to have the same luminance as the original colour image according to its colour space. Additionally, it gives the highest coefficient to the green channel and the lowest coefficient to the blue channel.

$$I_{cm} = 0.2126 \cdot R + 0.7152 \cdot G + 0.0722 \cdot B \tag{3.2}$$

The terms R, G and B in Equation 3.2 are the colour channels of the image in linear RGB colour space. Additional information about the greyscale conversion

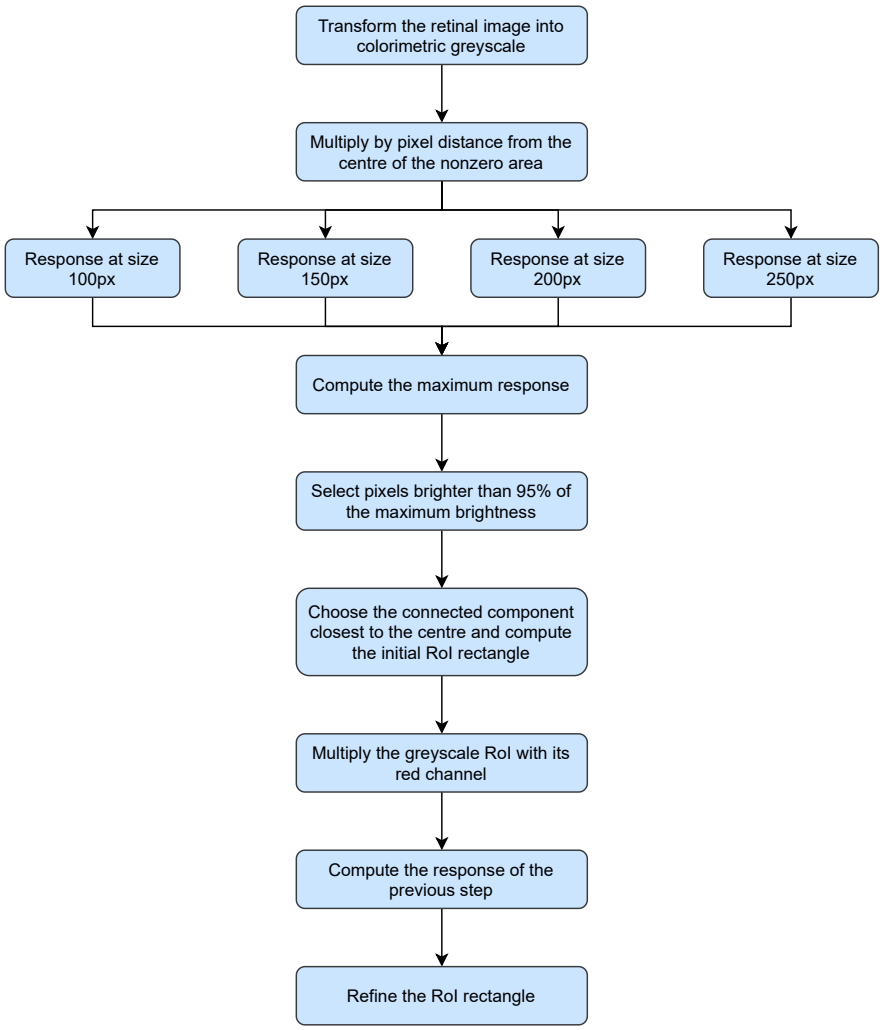

Figure 3.2: This figure shows a scheme of our region of interest selection algorithm. Some of the minor steps and details are omitted for clarity.

can be found in [50].

The next step of our RoI detection algorithm is scaling the transformed retinal image to a smaller size for improved performance. We use the width of $500px$ with heigh computed so that the aspect ratio of the image is preserved. We observed that changing this resolution to higher values does not affect the quality of the result. We consider $500px$ to be a good combination of performance and detail preservation. Before we locate the region of interest we have to resolve issues arising from bad retinal images. A failure of the imaging technology can cause very bright artefacts at the edges of the retina, which we need to suppress before any selection of bright pixels. An example of such artefact can be seen in Figure 3.3, image *(a)*. We are using prior information about the processed datasets, which places the optic disc close to the centre of the retinal image. The assumption about OD location is not a rule which applies to every retinal dataset. However, those used for optic nerve head segmentation usually fulfil this condition. Therefore, we use the distance from the centre of the retina for unwanted artefact suppression.

We start by computing the distance matrix of pixels with intensity higher than

1% of the maximum intensity in the image. The mask created by this threshold encompasses the circular retina and excludes the black border. We found that the values which fill the retinal image to form a rectangle are not always zero, but they are very close. Hence, we use an empirical threshold of 1%. Then we divide the distance matrix by its maximum value, which produces distances ranging from 0 to 1 with the lowest values at the edges of the retinal image. We can observe the source image and its distance matrix in Figure 3.3. The distances computed at this step are linear. The normalised distance matrix is shown in image *(b)*.

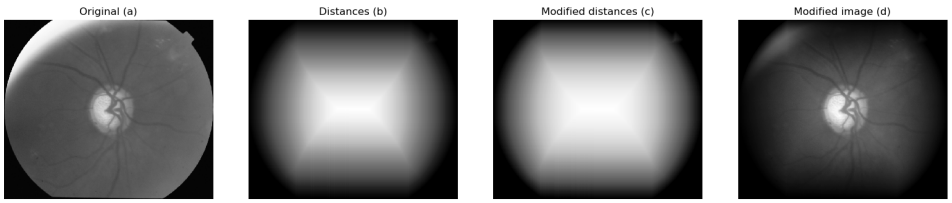

Figure 3.3: a) The original retinal image scaled down to $500px$, b) Linear distances of non-zero area of the retinal image, c) Distances modified by an exponential, d) The retinal image multiplied with the non-linear distances

Next, we apply a non-linear transformation to the linear distance matrix to obtain our pixel modification values. Note that pixels at the retina centre have distance values close to 1, and those at the border are nearing 0. We introduce non-linearity by computing an exponential out of negative distances. An important observation is that exponential out of any value ranging from -1 to 0 belongs to the range 0 to 1. Thus, we assign the right order of distances, i.e. the highest values in the centre, by subtracting them from 1. We define the precise formula as

$$d = (1 - \exp(-d_{linear})). \tag{3.3}$$

where d_{linear} is a linear distance of a pixel from the retinal border. In Figure 3.4, we can observe how does the distance modifier behave in the range from 0 to 1.

The suppression of artefacts appearing at the retinal border is done by multiplying the intensity values with the computed distance modifier. The non-linear distances and the final suppressed intensity image is shown in Figure 3.3.

In the next step of our algorithm, we search for bright circular areas in the retinal image and enhance them. We use convolution with a disc kernel modified by pixel distance from its centre, similar to the image modification done in the previous step. We use the same formula for the kernel computation as for the image distance, see Equation 3.3.

To properly distinguish between bright circular areas and monochrome regions, we subtract the mean value of the kernel from the kernel itself. We compute the mean only from the values within the original disc area. Then, we set the pixels outside of the disc to zero. This modification assures that the response of the monochrome region is zero. The result behaves similarly to Laplacian of Gaussian, which matches blobs in images. Our definition uses a sharp cut-off

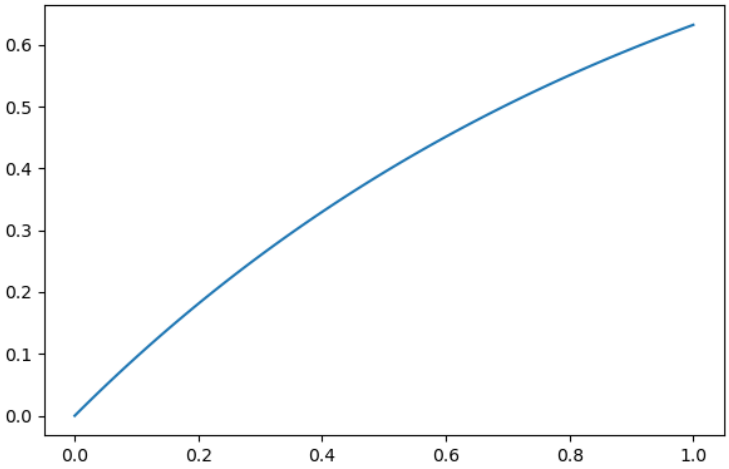

Figure 3.4: Plot of the non-linearity applied to pixel distances for values in the range from 0 to 1.

instead of a smooth transition which gives more importance to the darker surroundings of the bright region. The base disc shape, the distance modifier and final kernel can be seen in Figure 3.5. We also use mean subtraction in non-zero areas on the processed greyscale image to enhance the response of bright regions. The convolution can produce negative values in darker areas. However, we do not need to worry about the range of values because we are only interested in the maximum response.

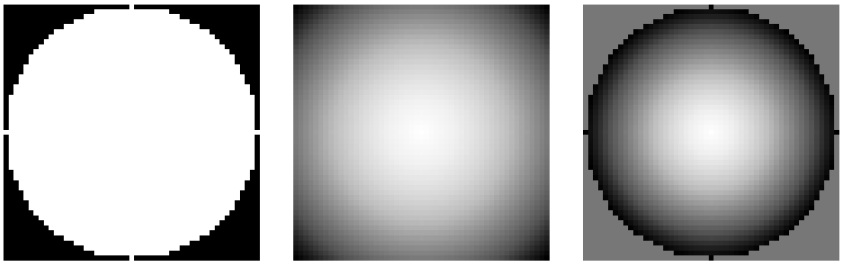

Figure 3.5: Basic disc kernel (left), the distance modifier computed according to the formula in Equation 3.3 (middle) and the final kernel combined from the distance modifier and the basic disc kernel.

We cannot predict the size of the optic disc before the region of interest selection. Therefore, we apply the convolution at different scales and select the maximum response. The resolutions used in our algorithm are $100px$, $150px$, $200px$ and $250px$ with the kernel width $25px$. This matches optic discs, which cover the area from $\frac{1}{16}$ to $\frac{1}{100}$ of the retinal image. In practice, the matched shapes are smaller due to the negative values of the kernel's border. Each response image is divided by the area of the kernel to ensure that computed values are comparable. The response is computed simply as $I = \frac{I_{current}}{a_{div}}$ where $I_{current}$ is the response image before kernel normalisation. We define the precise formula for

the division term as

$$a_{div} = (K_{width}\frac{250}{W_{current}})^2, \tag{3.4}$$

where K_{width} is the width of the kernel which is $25px$ in our case and $W_{current}$ denotes the width of the currently processed response image. An example of kernel responses at the respective resolutions is shown in Figure 3.6. Note that the apparent brightness of pixels in the response images does not represent the values that we compare while selecting the maximum response. That is due to intensity scaling for the display of images.

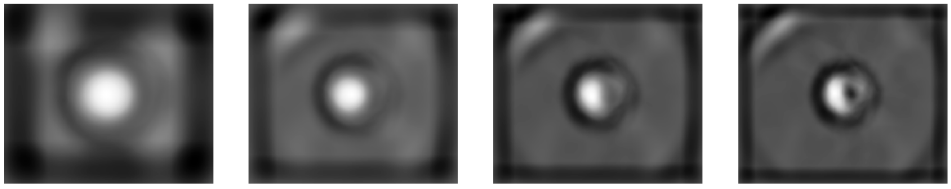

Figure 3.6: Responses returned by convolution with our disc matching kernel at resolutions $100px$, $150px$, $200px$ and $250px$ from left to right.

Now that we have a single maximum response image, we continue by selecting the pixels brighter than 0.95 times the maximum value in the image. Next, we use the prior information about optic discs being close to the centre of the retina and select the connected component, which is closest to the centre, as our approximate location of the optic nerve head. We define the centre of the region of interest as the mean pixel position of the chosen component. The final step of the first phase of our RoI selection algorithm is computing the width of the region. Let us assume that retinal images in our dataset are of approximately equal size. Then, we need to consider only the size of the optic disc in a particular image. We define the width of our selected region as

$$W_{RoI} = W_{data}(1 + r_1), \tag{3.5}$$

where W_{RoI} is the width of the region, W_{data} is a data dependent constant and $(1 + r_1)$ denotes a scaling factor computed based on the approximate size of the optic nerve head. We set $W_{data} = 100px$ for the Drishti dataset used in our experiments. We define the ratio r_1 as

$$r_1 = \frac{A_{bright}}{A_{size}}, \tag{3.6}$$

where the term A_{bright} is the number of pixels brighter than 0.95 times the maximum value from which we chose the location of RoI. The remaining term A_{size} denotes 5% of all pixels in the image. Following that, the factor r_1 is a ratio between the number of pixels surpassing a certain brightness threshold and the surface covering a certain percentage of the image. The core idea of this formula is that larger optic disc returns high response values in a larger area, whereas smaller discs have a sharper response. This property is quantified in the term r_1, and we observed that the values of 0.95 for the brightness threshold and 5% for

the area give the best results. We can see the partial results of the first phase of our algorithm in Figure 3.7. The middle image shows the maximum response, which is much sharper compared to the individual responses shown in 3.6.

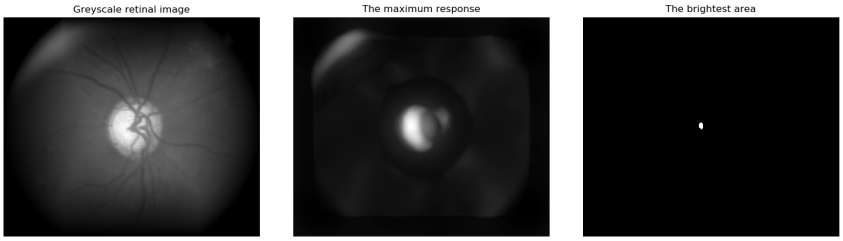

Figure 3.7: The greyscale image computed for the RoI selection (left), the maximum response of each pixel with respect to our modified disc kernel (middle) and pixels surpassing the threshold 0.95 of the highest response value.

Let us denote the centre coordinates computed from the brightest pixels as c_x and c_y. Then, we define the region of interest of the first phase as a square with the top left corner tl and bottom right corner br with coordinates computed as follows.

$$\begin{aligned} tl_x &= c_x - \frac{W_{RoI}}{2} \\ tl_y &= c_y - \frac{W_{RoI}}{2} \\ br_x &= c_x + \frac{W_{RoI}}{2} \\ br_y &= c_y + \frac{W_{RoI}}{2} \end{aligned} \tag{3.7}$$

The second phase of the region selection

The second phase of our algorithm starts by cropping the retinal image to the region selected in the first phase. The second phase aims to refine the region of interest to be smaller if possible and better centred on the optic disc. The steps of this part of the algorithm are very similar to the first phase.

Firstly, we compute the colourimetric greyscale transformation of the region. Then we multiply it with the red channel of the image. The majority of information about the optic disc is carried in the red channel, see [4]. Thus, the multiplication enhances the optic disc area. This step was not done in the first phase due to retinal and image defects, which might be present in the red channel. This concern becomes irrelevant after we crop the image.

Secondly, we scale the cropped region down so that its width is 100 pixels. Then, we convolve it with the kernel defined in the previous section. We set the width of the kernel to $40px$. We do not need to do a multi-scale response computation because all optic discs have a similar size in the cropped and scaled image. After analysing our dataset, we concluded that setting kernel width to 40% of the image width is a reasonable estimation of the disc sizes in the second phase.

Thirdly, we calculate the new location of the optic disc as the mean pixel position of pixels brighter than 0.9 times the highest value in the image. Compare that with the first phase, where we used the threshold of 0.95. We can use a lower threshold because the optic disc in the second phase covers a much larger region of the image than it did in the first one. Subsequently, we compute the equivalent ratio r_2 using the same formula as for r_1, see Equation 3.6. However, together with the brightness threshold, we also increase the area considered in the term A_{size} from 5% to 10%. The resulting formula is equivalent to the computation of r_1 if we assume a larger disc region. The cropped region of interest, its response and the brightest area used for localisation are shown in Figure 3.8.

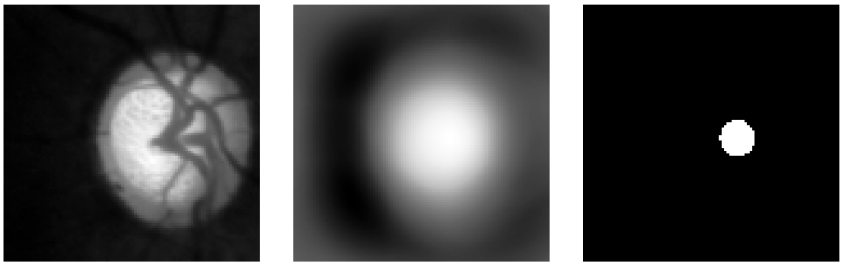

Figure 3.8: The greyscale image of the RoI from the first phase (left), the maximum response of each pixel with respect to the disc kernel (middle) and pixels surpassing the threshold 0.9 of the highest response value.

Finally, we have to compute the width of the refined region of interest. The goal of the second phase is to reduce the size of the region such that it still contains the entire optic disc. We introduce an empirical constant $\kappa = 0.3$, which defines the maximum scale reduction, i.e., we cannot make the region smaller than 70% of the original. This limitation helps us prevent cutting off parts of the optic disc due to a small r_2 factor. We chose the constant of 0.3 based on the observed results of the first phase computed on images from our dataset. In addition to the term r_2, we add a new distance factor ϵ, which penalises the size reduction for cases where the new approximate disc location is far from the previous one. We define the complete reduction factor ℓ as

$$\ell = (1 - \epsilon)^2(1 - r_2)\kappa, \tag{3.8}$$

where the term ϵ is computed as the distance between the approximate disc centres divided by half of the diagonal of the rectangular region. We subtract it from one to receive high values for short distances. We are working under the assumption that placing the new refined centre close to the original one means that the first approximation was already good. Then we can significantly reduce the size of the region because we were in the correct position. However, if the approximate centres are far apart, we are not confident in our centre localisation, and we keep the size of the region similar to the original. The new centre tends to be quite close to the previous one, and therefore we take the factor to the power of two, which enhances the penalisation effect of distant centres. Next, the term r_2 specifies how large is the new disc. Subtracted from one, it penalises the size reduction for large discs. Subsequently, we compute the width of the new region of interest as the width of the previous region multiplied by the factor

$(1 - \ell)$. Finally, we define the new region of interest as a square centred at the new approximate disc centre with the width obtained in the previous steps. We can see an example of the result produced by our algorithm in Figure 3.9. Note that the area of the refined region of interest reaches outside of the first phase rectangle.

Figure 3.9: This figure shows an example of the results computed using our RoI detection algorithm. Blue colour denotes the region extracted in the first phase of the algorithm, and red denotes the region from the second phase.

In conclusion, our region of interest detection algorithm is robust and also fast due to the low resolution of images, it processes. A disadvantage of the algorithm is that it has data-dependent constants, which should be specified for custom datasets individually. The algorithm performs well on fundus images that contain the whole retina because the default constants specified for our dataset consider that type of images. Modification of the constants is necessary only if we execute the algorithm on images where some form of region of interest was already selected or when the fundus image is significantly larger than the images from our dataset. The width of our images is around $2000px$, which is a standard resolution for retinal fundus images.

3.2 Vessel extraction and suppression

One of the most prominent features of a retinal fundus image is the blood vessel network. It is composed of vessels originating in the optic disc. Their convergence point is usually situated at one side of the optic cup, depending on whether we have an image of the left or right eye. The vessels split and spread as they reach the outer parts of the retina. This expansion results in a significant variance in vessel thickness. The goal of our work is to extract the area of the optic disc and the optic cup. Therefore we do not need to be able to extract vessels precisely. Correctly segmenting the major vessels in the optic disc region selected by our RoI detection algorithm is sufficient. As a result, we chose a simple, morphology-based extraction algorithm. We show an example of the results produced by our

algorithm in Figure 3.10. The vessel structure is incomplete, but we are interested in the central knot of vessels which is pretty accurate.

Figure 3.10: The result of our vessel segmentation algorithm on the region of interest extracted with 3.1. The left image shows the original image from the dataset and the right image shows the detected blood vessels.

A comprehensive survey of different vessel segmentation algorithms together with datasets used for their evaluation can be found in [51]. As a result of our requirements, we decided to implement a modified version of the algorithm presented in [52]. Instead of the application of three vessel-enhancing filters, we applied only two of them. We excluded the Frangi filter, which we deemed superfluous for our task. A general scheme of our vessel segmentation algorithm can be seen in Figure 3.11.

Let us discuss the individual steps of the algorithm in more detail. Firstly we select an appropriate colour channel for vessel extraction. The highest vessel contrast is observable in the green colour channel, which was used for feature extraction in several proposed approaches [51]. In other cases, a combination of colour channels with special coefficients is used to enhance the contrast. In [52], the authors estimated the values of channel coefficients from the training dataset and discovered that values $c_r = 0.1$, $c_g = 0.7$ and $c_b = 0.2$ are best performing for intensity computation defined as

$$I = c_r * R + c_g * G + c_b * B \tag{3.9}$$

where I is the grey level intensity intended for blood vessel extraction. Further, terms R, G and B denote the colour channel of RGB colour space. Finally, c_r, c_g and c_b are the channel coefficients used in the grey level conversion. Generally, the coefficients are constrained by $c_r + c_g + c_b = 1$, i.e., their sum has to be 1. In Figure 3.12 we can observe the optic disc region of interest split into individual colour channels. We decided to use the green channel for vessel segmentation due to its high contrast. We omitted the red channel due to low contrast in the optic disc region and the blue channel due to noise.

The second step of our algorithm estimates the blood vessel regions using matched filters. We scale the processed image down to $256px$ for faster computation. Similarly to [52], we apply Gaussian matched filtering described in Section

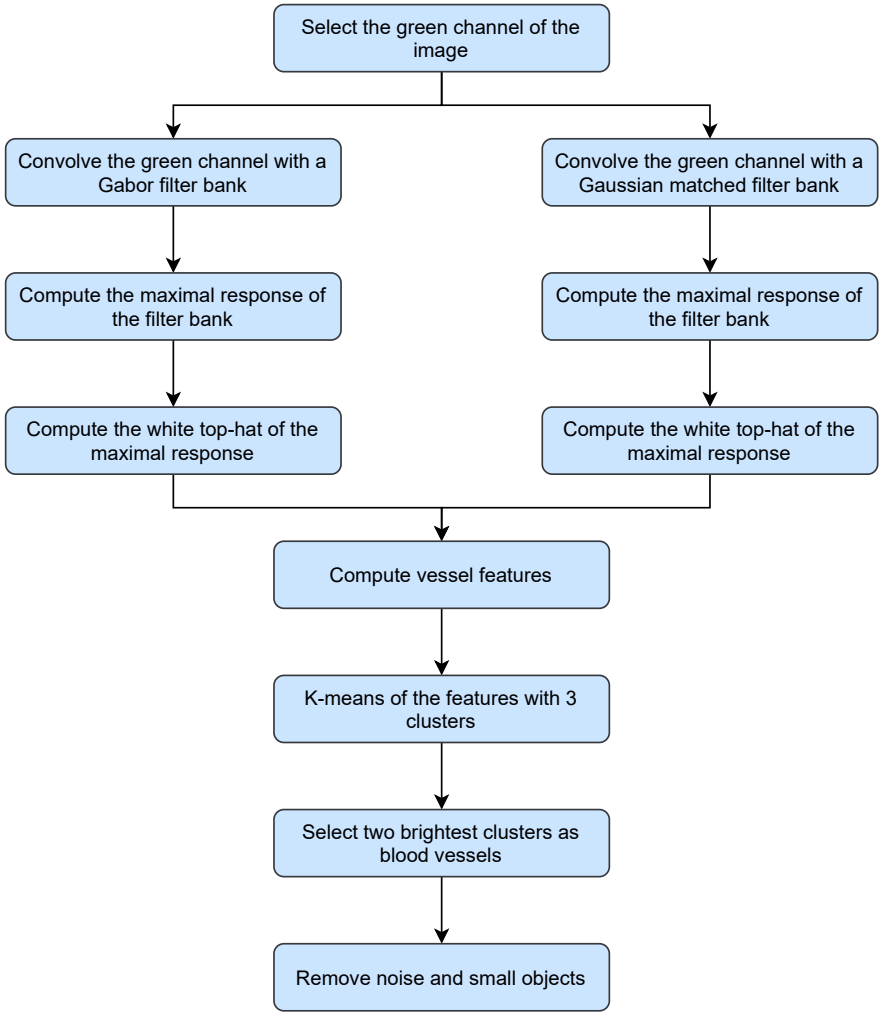

Figure 3.11: This figure shows a diagram of our vessel extraction algorithm.

2.6 and Gabor filtering describe in Section 2.7. We modify the kernels of these filters to ensure that they produce responses with the value 0 for monochrome areas spanning the entire kernel. We ensured this behaviour by subtracting the mean of the given kernel from the kernel itself. In some cases, blood vessels have thin bright stripes in their centre. These artefacts can cause incorrect segmentation of the vessel structure [52]. We found the bright stripes in our dataset as well. We consider them a general issue rather than a problem stemming from a faulty imaging device. An example of this artefact in an image from our dataset can be observed in Figure 3.13. To suppress these bright areas, we applied a morphological opening operation, see Section 2.4, with a disc-shaped structuring element. We use a disc with a radius of 5 pixels, representing about 2% of the image size after scaling it to $256px$.

The purpose of both Gaussian and Gabor filtering is very similar. They are supposed to enhance the vessel strands in the image. After that, we can apply further processing, which extracts the vessel region. As we noted in Section 2.6, the stretched and rotated Gaussian kernel responds to oriented strand-like areas. Therefore we convolve the image with a Gaussian filter bank of 12 rotated kernels with the angle of rotation linearly spaced in the range from 0 to 180 degrees. We consider 12 kernels to be a sufficient number for matching all vessel orientations.

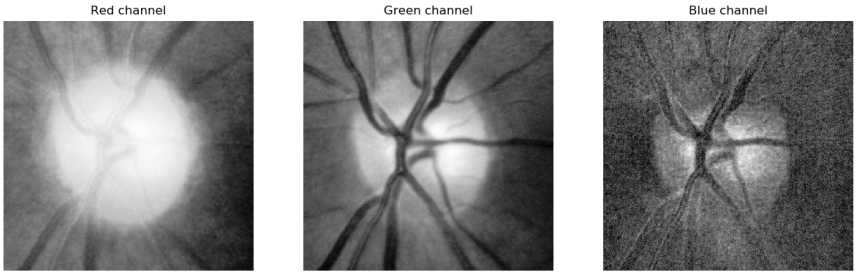

Figure 3.12: The region of interest separated into the RGB colour channels. The blood vessels are prominent in the green channel and they blend together with the optic disc in the red channel. The blue channel is too noisy for shape extraction.

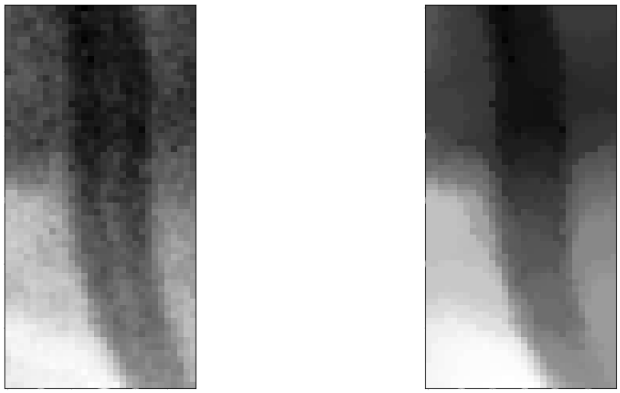

Figure 3.13: The left image shows a section of the colour channel intended for vessel extraction containing a bright stripe artefact. The right image shows the result of morphological opening applied to the left image.

Increasing the number of kernels does not improve the result. We take the maximum value for every pixel from the stack of response images computed for the set of kernels. The resulting pixel value represents the highest vessel-like response among the kernels, and the entire image represents enhanced vessel regions. Note that the response is still 0 for monochrome areas.

Similarly, we construct a Gabor filter bank of 12 rotated kernels with the angle of rotation linearly spaced in the range from 0 to 180 degrees. Then we apply the same enhancement process as for Gaussian filtering. We convolve the source image with the kernels and take the per-pixel maximum from the stack of responses. We show the computed responses for both types of filters in Figure 3.14.

The next step of the algorithm is further enhancement of the vessel strands by applying a morphological white top-hat operation as proposed in [52]. We use a disc structuring element with a radius of 13 pixels. White top-hat returns thin bright areas from the image, which match the vessel regions of the images produced by Gaussian and Gabor filtering. In Figure 3.15, we show the results of morphological white top-hat for Gabor and Gaussian filter responses.

Let us denote the results of white top-hat I_{Gauss} and I_{Gabor} for the Gaussian

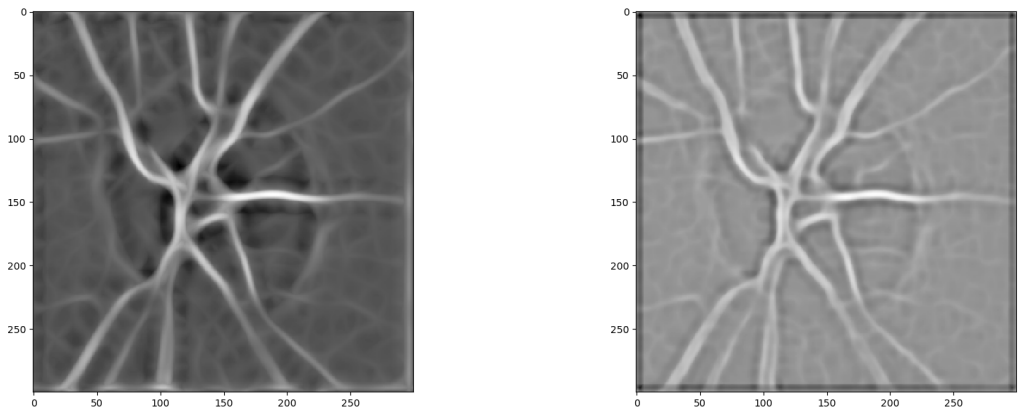

Figure 3.14: The left image shows the per-pixel maximum of the oriented Gabor filter responses. The right image shows the per-pixel maximum of oriented Gaussian matched filter responses.

and Gabor responses, respectively. We compute features for the clustering step of our algorithm from these two images by applying a polynomial feature transformation with the degree 2. More precisely, the set of features can be defined by the enumeration

$$F = \{I_{Gauss}, I_{Gabor}, I^2_{Gauss}, I^2_{Gabor}, I_{Gauss} * I_{Gabor}\}, \tag{3.10}$$

where F is the set of features, and $*$ denotes per-element multiplication. We decided to compute the feature set using polynomial transformation due to the apparent value in combining our feature images. We can observe that multiplication suppresses false and weak vessel responses. The full set of features is shown in Figure 3.16. The original feature images are included to add the weak vessel responses to the decision-making process. Finally, we stretch each feature from the set in Equation 3.10 to span the range from 0 to 1.

With the feature set computed in the previous steps, we can proceed to vessel classification. Generally, we would classify pixels into two groups: vessel and non-vessel. However, following the approach proposed in [52] and our feature set definition, we define three classes of vessels. Let us call them vessel pixels with high, medium and low probability. Then we assign the first two classes to the vessel region and the last class to the non-vessel region. The main reason for this assignment stems from the properties of the feature set. High values of all the features describe the class of vessel pixels with high probability. Next, high values of the original features and low values of multiplied features represent the class with medium probability. Finally, pixels with low values features represent the class with low probability.

The final step of our vessel extraction algorithm is the classification of pixels. The approach in [52] suggests using fuzzy c-means classifier instead of k-means. We discovered that classification of pixels using fuzzy c-means often selects false vessel regions and that k-means is marginally more reliable. The result of classification into three classes by the k-means algorithm can be seen in Figure 3.17. Each step of the algorithm is subjected to noisy input, which can cause small

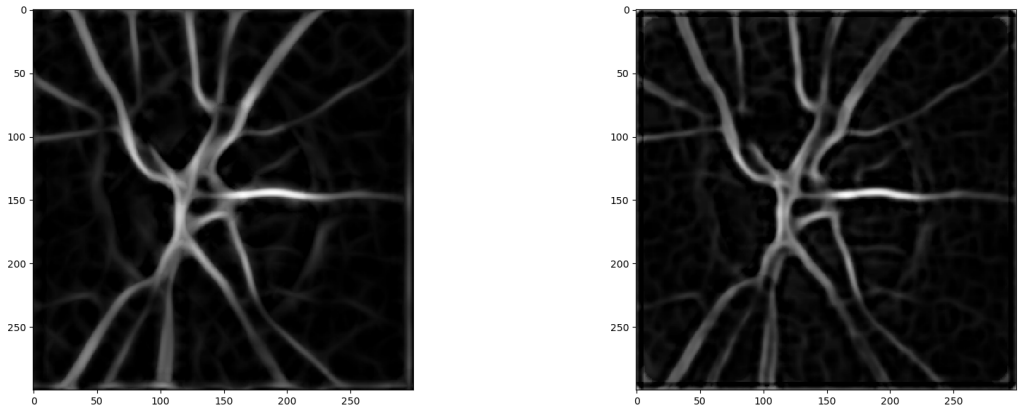

Figure 3.15: The results of morphological white top-hat operation applied to per-pixel maximum responses of rotated Gabor filters (the left image) and Gaussian matched filters (the right image).

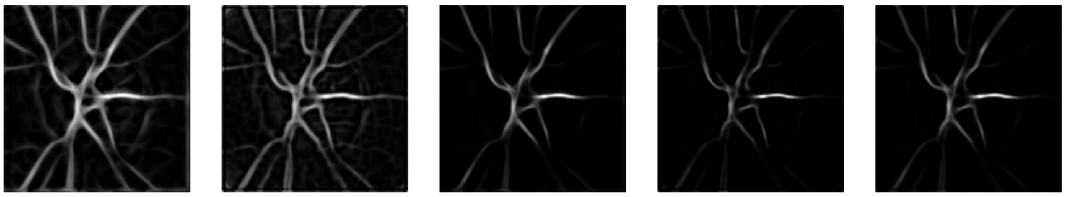

Figure 3.16: The set of features described in Equation 3.10 displayed in the order of their definition.

vessel-like islands to appear in the classified result. We filter out all objects smaller than 100 pixels from the result to resolve this issue.

Our vessel extraction algorithm has several shortcomings. Among the significant issues belong incomplete segmentation of thinner vessels visible in Figure 3.17 and false vessel areas appearing at the boundary of the optic disc and the optic cup. These problems would have to be resolved if the algorithm was applied to the vessel segmentation task. However, in our case, precise vascular regions are not necessary because we use them only for suppression. Thinner vessels do not cover large parts of the optic cup. Consequently, they do not affect segmentation substantially. Further, the false vessel regions at the optic disc and cup boundary are replaced by values that reflect the closer retinal structure. In our experiments, suppressing these regions did not move the boundary of either the optic disc or the optic cup. Nevertheless, we do not suppress vessels for our disc segmentation algorithm due to interference in the parapapillary atrophy of the retinal image and use it only during the optic cup extraction.

Vessel suppression

Until now, we talked about vessel segmentation, which is not useful for our overall optic nerve head extracting algorithm. We want to use the segmented vessel regions to suppress vessels. In this context, suppression means an approximation

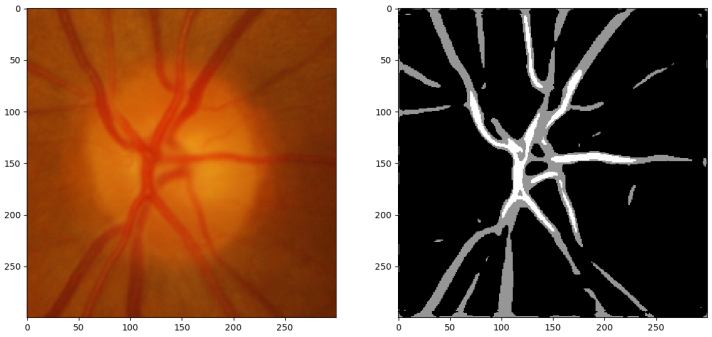

Figure 3.17: This figure shows the original image (left) and pixel labels produced by the k-means algorithm classifying into three classes.

of the pixel values in the vascular area from a retinal image without blood vessels. We display an example of a region of interest before and after vessel suppression in Figure 3.18.

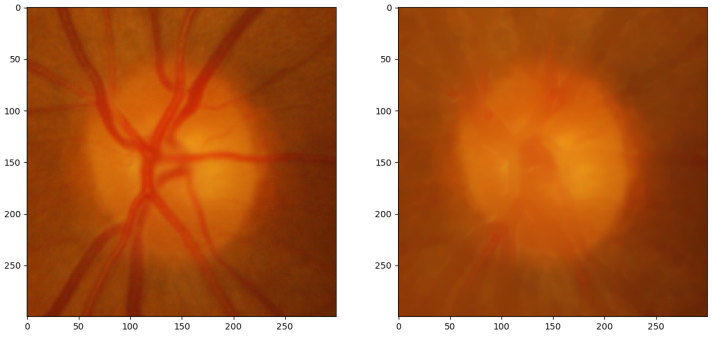

Figure 3.18: The original image (left) and an image with suppressed vessel regions (right).

We suppress the vessels in an image by replacing the colour values of vascular pixels with the median of non-vascular pixels in the pixel neighbourhood defined by a square window with the width of 30 pixels. The non-vascular pixels are left unchanged. We selected a large filtering window due to thicker vessel areas at the centre of the optic disc. Due to imprecise detection of blood vessel boundary, the resulting image can have dark lines around the suppressed areas. These artefacts result from the replacement value being significantly higher than the effective threshold by which we selected the vascular area. To remove these dark lines, we apply morphological grey level closing to the suppressed image.

3.3 Region of interest preprocessing

Retinal fundus images can vary a lot from one to another. These variations can be caused by the slightly different calibration of the imaging device, angle under which the retinal image was taken or slight movement resulting in blurry images. Like every other imaging technique, acquiring images can introduce noise, which reduces the quality of the image. The most crucial difference between retinal images is their contrast. Whereas some of them can have a very bright optic disc and dark background, others can have similar values in both regions. There are large variations within each dataset as every person has a slightly different retina, resulting in significant colour differences. Diseases such as glaucoma can cause additional issues resulting in low contrast of the image. These problems are even more apparent when we consider images from different datasets that were usually taken on particular devices. Therefore we want to improve the contrast of processed images and reduce noise introduced by the imaging process [4].

In Section 3.1 we located a region of interest containing the optic nerve head. Therefore, it is advantageous to apply preprocessing techniques on this region rather than on the full retinal image. For example, any contrast stretching on the full image would encounter problems with the black area surrounding the retina. In addition, the full image may contain various artefacts, which reduce the effectiveness of contrast stretching. An example can be observed in Figure 3.3, image *(a)*. The displayed retinal image contains values from the entire colour range, mainly the black border and bright region contain the extrema. Therefore, contrast stretching on this image would be ineffective. However, the selected region of interest excludes both unwanted areas and allows contrast-enhancing algorithms to improve the processed image.

We apply adaptive histogram equalisation (AHE) to enhance the contrast of images. An ordinary histogram equalisation uses the same transformation derived from the image histogram to transform all pixels. It works well when the distribution of pixel values is similar across the entire image. Areas that are bright or dark will not be sufficiently enhanced. Adaptive equalisation transforms pixels based on the histogram of their neighbourhood. A simple algorithm can compute the histogram from the square neighbourhood of a pixel and use the same transformation as global equalisation [53]. The effect of AHE on the optic disc region of interest in the individual RGB colour channels can be observed in Figure 3.19. We used the adaptive equalisation implementation available in scikit-image [29].

The second preprocessing technique which we apply to the equalised image is median filtering for noise suppression [35]. Adaptive equalisation tends to enhance noise in the image, and therefore we apply the noise reduction on the equalised image. We use a median filter of size 7 by 7 pixels. We found that this size of the filter gives the best results. An example of median filtering applied to the equalised image is shown in Figure 3.19.

These two techniques improve the contrast and reduce the noise in the image. This step is important to reduce differences between images from one or more datasets before proceeding to the object detection algorithm.

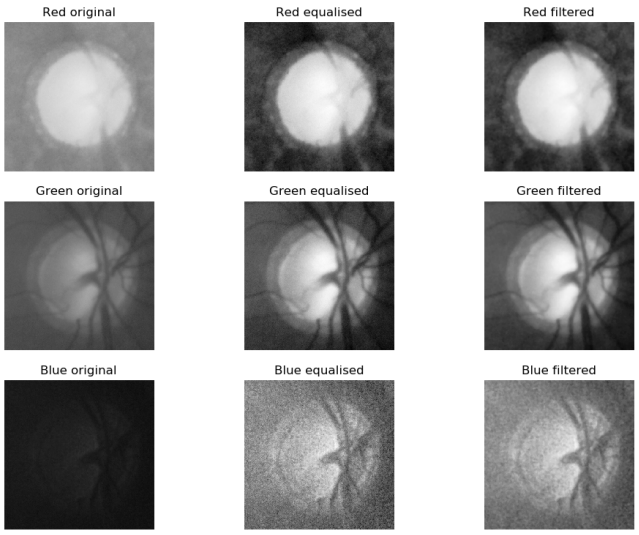

Figure 3.19: This figure shows the effect of adaptive histogram equalisation (middle) and median filtering (right) on the original image cropped to the region of interest (left).

3.4 Iterative optic disc thresholding algorithm

Optic nerve head segmentation can be divided into two major parts: the extraction of the optic disc and the optic cup. Most of the approaches in [4] and [5] propose a solution for both tasks. In these cases, the optic disc segmentation must be done first because the optic cup extraction depends on it. We propose an algorithm that also locates and extracts both retinal features. We present the overall diagram of the iterative algorithm in Figure 3.20. The scheme contains the region of interest selection and image filtering as the first step. It does not mean that these actions are done multiple times.

Let us assume that we have a region of interest computed using the algorithm in Section 3.1 from some retinal image. Furthermore, this region is preprocessed using the methods described in Section 3.3. Therefore, our starting point is a contrast stretched square area approximately centred on the optic disc with eliminated noise. Now, let us begin the description of our iterative optic disc segmentation algorithm.

The first task of optic disc segmentation is similar to the first step of the region of interest detection. It is an appropriate colour channel selection. Generally, the red channel from the standard RGB version of the image is taken since the optic disc has the highest contrast in it. We observed that the blue channel contains a distinct optic disc region when the noise is not dominant in some cases. Furthermore, various brightness and intensity channels show high contrast between the disc region and its background. For instance, the value (V) component of the HSV colour model is computed as the maximum of R, G and B values or the Y component of the YIQ colour space. We can also consider the L component of CIE L*a*b* or L*u*v* colour models. For more information on the individual

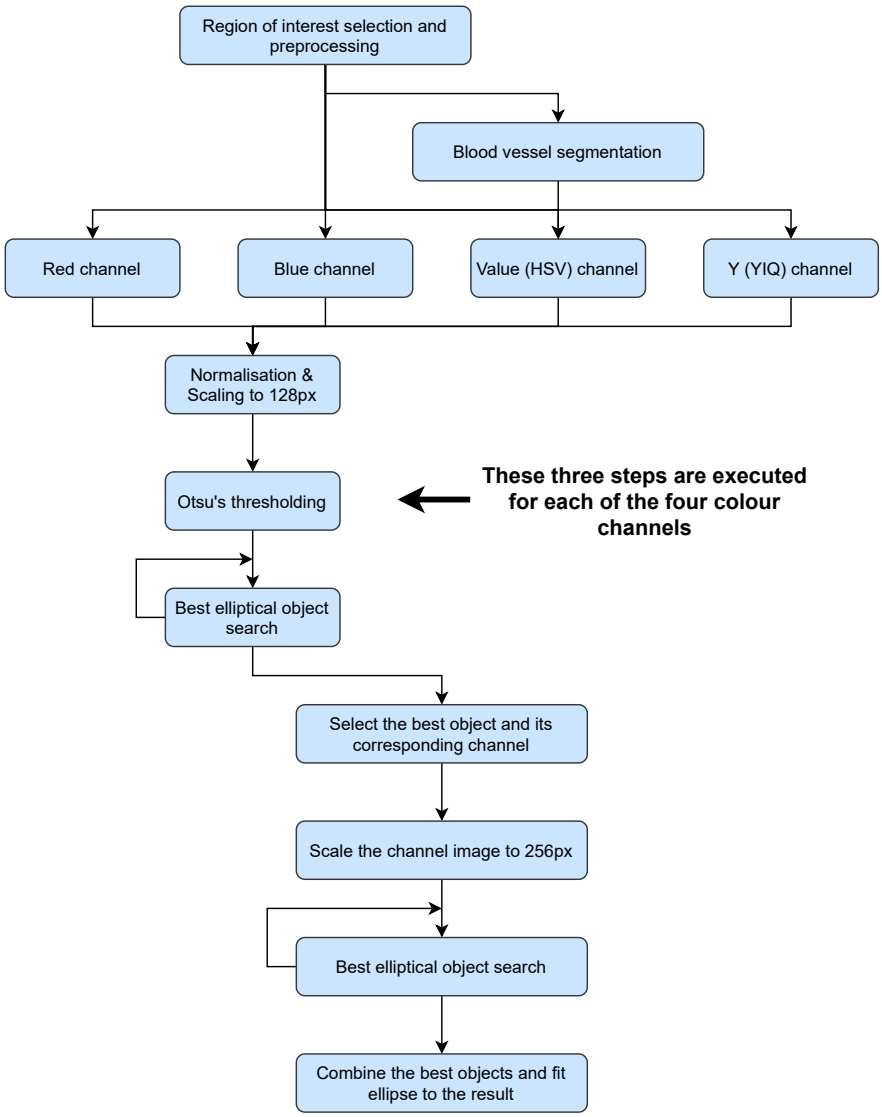

Figure 3.20: Scheme of our iterative thresholding algorithm for optic disc segmentation. Some of the minor steps are omitted for clarity.

colour spaces, see [54]. In Figure 3.21, we can see the individual colour channels of the transformed region of interest.

We decided to use colour channels R, B, V, and Y for the segmentation. Our algorithm can compute the quality of a given object, i.e., how much does it resemble the optic disc. Therefore, we do not combine these colour channels to produce an intensity image for segmentation. Instead, we execute the main part of the algorithm on each colour channel scaled down to a small resolution for improved performance. Each of these processes produces the best object it could detect in the colour channel together with its score. Based on the obtained scores, we choose the best colour channel for the segmentation and continue examining this colour channel. The initial run of our algorithm also gives us a starting point in the form of the best object found in the subroutine.

Before we start with the detection of objects in the selected colour channels, we execute the blood vessel extraction algorithm defined in Section 3.2. We noted in the referenced section that blood vessel suppression is not used for optic disc

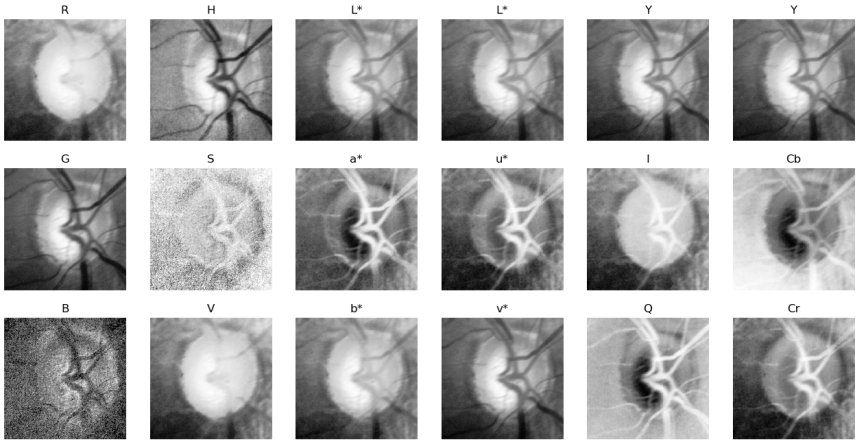

Figure 3.21: Channels of RGB, HSV, L*a*b*, L*u*v*, YIQ and YCbCr colour spaces transformed from the RGB image.

segmentation. That is due to our definition of object quality. Removing blood vessels from the image can cause small objects representing the optic cup to be marked as optic disc instead. Nevertheless, a blood vessel mask is used by the algorithm to evaluate the quality of detected objects. Therefore we compute it from the region of interest at the start of the algorithm. Then we use it in each subroutine, which produces an object. The scheme in Figure 3.20 shows the blood vessel segmentation slightly separated from the flow of the algorithm. This separation represents the fact that transformations and thresholding are not dependent on the vessel extraction, but parts of the algorithm use the vascular mask.

The central part of the algorithm can be split into two phases. The first, initial phase is different from the subsequent one because it runs on multiple colour channels and has a different starting point. The following phase builds on the results of the initial one. Each branch of the algorithm corresponding to a particular colour channel starts with the normalisation of the processed image. Normalisation ensures that all branches work with data consisting of values in the range from 0 to 1. Now, any operation working with intensity is going to produce comparable results between the colour channels.

Figure 3.22: From left to right: the result of Otsu's thresholding, the result of the first phase and the result of the second phase.

Let us consider the first phase of the algorithm. Its input is a normalised single-channel intensity image. Firstly, we scale the image down so that its width is $128px$. We do not require precise segmentation of the object, and this phase is running for multiple colour channels. Thus, we chose a small resolution of images for better performance. Optic discs at a low scale tend to lose details on their boundary. However, the ability to select an object resembling the optic disc is unchanged. Simply, if OD is well separable from its background in the original image, it will remain separable in the scaled-down version. There are certainly cases where the pixel interpolation causes the disappearance of a thin line separating the disc from some artefact. The issue of the unwanted region is resolved in the second phase. The second step of the initial phase is image analysis by Otsu's method for the selection of starting threshold, see Section 2.1. Otsu's thresholding performs well, and in most cases, it produces an object close to the ground truth OD. Nonetheless, there are cases where it cannot separate the object from its background even though a threshold exists. In addition, it tends to under-segment objects, i.e., the obtained object is smaller than the one we are trying to locate in the image. We apply it to the unscaled image because the computation requirements of Otsu's thresholding are relatively low, and we observed a notable improvement in the results. To improve the result of Otsu's method we run 3 iterations of threshold improving algorithm based on the segmented object's quality with the Otsu's threshold as its starting point. The search range of the algorithm is bounded by the values given in Equation 3.11. The formulas assume that the image is normalised. Otherwise, the additive term would have to be computed using the maximum and minimum value of the image. In Figure 3.22, we can compare the result of Otsu's method and the results of subsequent phases. Note that phase two has only marginal changes. It improves the result by adding details to the segmented object.

$$\begin{aligned} t_{min} &= t_{Otsu} - min(1.0 - t_{Otsu}, t_{Otsu}) \\ t_{max} &= t_{Otsu} + min(1.0 - t_{Otsu}, t_{Otsu}) \end{aligned} \tag{3.11}$$

Following the selection of the initial threshold and search boundaries, we can proceed to the iterative improvement of the returned threshold. Let us start by describing the general iteration scheme and search updates. A diagram displaying this part of the algorithm is shown in Figure 3.23. Endpoints of the search interval are denoted by t_{min} and t_{max}. Further, the starting threshold is denoted by t_{start}. If we find an object at a certain threshold and do not have additional information, we cannot decide whether we should search for a better object in the lower or upper intervals. Therefore, our algorithm finds the best object at three thresholds: t_{start}, $\frac{t_{min}+t_{start}}{2}$ and $\frac{t_{start}+t_{max}}{2}$. It chooses the best object out of the three propositions, and then it shifts the interval boundary so that the whole range is reduced to one half. Assuming that the best object lies in the lower half of the interval, we shift the upper boundary to the starting threshold $t_{max}^{new} \leftarrow t_{start}$. If the best object lies in the upper half of the interval, then the situation is analogical. Lastly, if the starting threshold is the best then we simply shift endpoints of the interval to the thresholds $\frac{t_{min}+t_{start}}{2}$ and $\frac{t_{start}+t_{max}}{2}$. The final step of a single iteration is the assignment of the new starting threshold. It is simply set to the midpoint between the new interval boundaries $t_{start} \leftarrow \frac{t_{min}^{new}+t_{max}^{new}}{2}$.

This set of steps is repeated for a small number of iterations. Note that the search interval is reduced by half in each iteration and, for a standard 8-bit greyscale image with 256 possible values for a pixel, it takes nine iterations to reach the precision of one intensity value. We do not need to run all nine iterations. In our experiments, we discovered that seven iterations are enough for convergence.

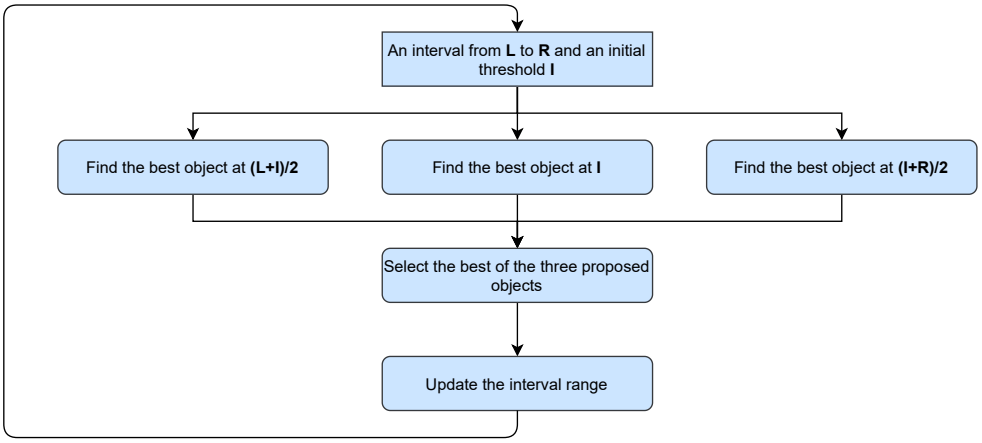

Figure 3.23: A scheme of the iterative process for the threshold improvement.

Until now, we described the algorithm by a general scheme saying that it finds the best object for a threshold and evaluates its quality, i.e. it assigns a score to objects which allows us to compare them. The subroutine for best object selection receives the source image and a threshold at which it should find the object. Firstly, we smooth the image by convolution with Gaussian kernel, and then we binarise the filtered image using the given threshold. Secondly, we remove small objects and holes which represent noise from the binarised image. Thirdly we select the remaining connected component which resembles an ellipse the most according to the similarity metric defined as

$$\phi = 1.4^e \sqrt{\frac{1}{N} \sum_{i=1}^{N} (d_i - \bar{d})^2}. \tag{3.12}$$

The formula of our ellipse metric given in Equation 3.12 is a standard root mean squared error (RMSE) of relative distances between object boundary pixels and the covariance ellipse calculated from object's pixel positions. It is multiplied by an eccentricity penalisation term 1.4^e where e is the ratio of the major and minor ellipse axes. We use exponentiation to penalise highly eccentric ellipses more significantly compared to circular ones. We opted for the base of the exponentiation such that for $e = 2$, the term is approximately equal to 2. We decided to use 1.4 instead of $\sqrt{2}$ for simplicity. The term N denotes the number of boundary pixels considered in the RMSE computation, the term d_i is a relative distance of the ith pixel to its closest point on the ellipse, and the term $\bar{d}$ denotes the mean of these distances. More precisely, d_i is computed as the pixel distance divided by the average of the major and minor axes of the ellipse. We define it as

$$d_i = \frac{d_i^{pixel}}{\frac{a+b}{2}}, \tag{3.13}$$

where a and b are the ellipse axes and d_i^{pixel} is the image plane distance of a point to the ellipse.

Lastly, we compute additional parameters of the selected object, which determine its quality as an optic disc candidate. We use these parameters to calculate the final score and return it together with the mask of the selected object. In Figure 3.24, we can see a visualisation of this sequence of steps.

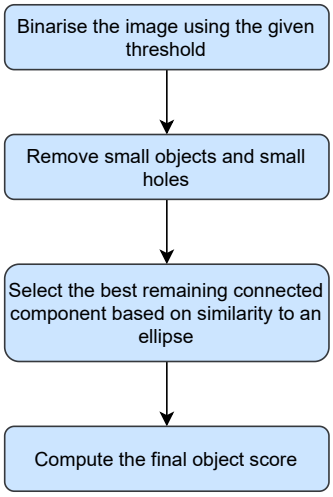

Figure 3.24: A scheme of the best object selection part of the algorithm.

The full optic disc score computation uses additional properties of selected objects. The base value is the ellipse similarity ϕ defined in Equation 3.12. In general, having a shape resembling an ellipse or a circle is not a good criterion for comparing different optic disc shapes. For instance, optic cups can be visible in the same intensity images as optic discs, and they can appear circular. An algorithm that decides purely on the similarity to an ellipse might select the optic cup area because it is more circular. Therefore, we define the optic disc quality metric as

$$\epsilon = \phi\nu\delta\lambda, \tag{3.14}$$

where the term ϕ is base value of ellipse similarity and the remaining factors determine how likely is the current object an optic disc. Each factor defines a condition that OD should fulfil and quantifies the success of the selected object in fulfilling it. Our objective is to minimise the function defined by ϵ to obtain the best object. The factor ν denotes vessel coverage description, δ penalises dark and distant objects, and λ denotes a penalisation term for large objects.

Now, let us consider the properties of optic discs within the region of interest acquired via the algorithm in Section 3.1. The optic disc is an elliptical shape positioned approximately in the centre of the RoI. Further, according to our data, it covers 25% or fewer pixels of the region. Next, it is an area where blood vessels converge and also where they obscure a significant section. Finally, the optic disc is a bright region on a relatively dark background, and it has no intersection with the boundary of the region of interest. An example of the score function evaluated for all thresholds of the red channel is shown in Figure 3.25. Notice that the function has a roughly convex shape. Nonetheless, it contains several local minima. In addition, the right image in the same figure shows the object mask

at the global minimum. The graph shows the score value only for thresholds in the viable range. Those higher and lower are considered invalid by the algorithm, and their score is set to a very high value so that they do not interfere with the real disc candidates.

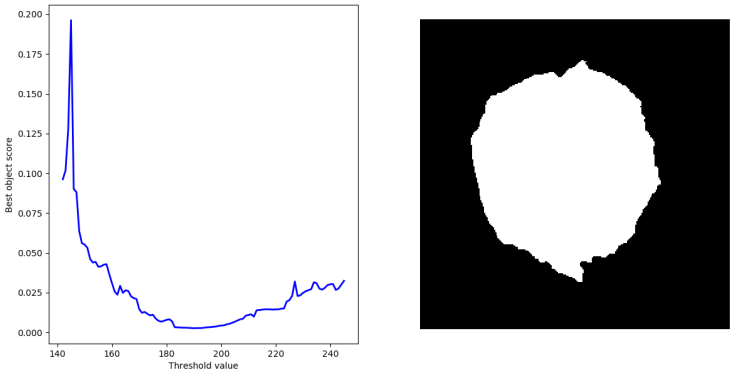

Figure 3.25: The left image shows the score function for the best object across all thresholds of the red channel. The right image shows the mask with the global minimum score.

The first important term in distinguishing small elliptical objects likely belonging to the optic cup from those which we are searching for is the vessel coverage term ν defined as

$$\nu = 1 - (\text{clamp}(\frac{|A_{obj} \cap A_{vessel}|}{|A_{vessel}| * 0.5}, 0, 1) * 0.75). \tag{3.15}$$

It considers the area of the selected object covered by vessels and assigns it a value in the range from 0.25 to 1. The value 0.25 means that the object fulfils the condition perfectly and 1 marks an object which does not intersect the vessel network at all. The core of the formula is $(\text{clamp}(\frac{|A_{obj} \cap A_{vessel}|}{|A_{vessel}| * 0.5}, 0, 1)$ where $\text{clamp}(x, 0, 1)$ is the standard mathematical clamp function which returns x for $x \in (0, 1)$, 0 for $x \leq 0$ and 1 for $x \geq 1$. The inner part $\frac{|A_{obj} \cap A_{vessel}|}{|A_{vessel}| * 0.5}$ describes the ratio of vessel coverage of the object to the vessel coverage of the region of interest. Terms A_{obj} and A_{vessel} denote the area covered by vessels for the selected object and the whole region, respectively. The vessel area in the denominator is multiplied by 0.5 to resolve the problem of objects spanning the entire RoI having a better vessel coverage score than OD candidates. Based on our available data, we discovered that the actual optic disc rarely contains less than half of the pixels marked as vascular by our algorithm. The average vessel coverage for ground truth optic discs in our training dataset is 66.6%, and the minimum is 43.9%. Based on these figures, we selected the threshold for the score clamping to be 50%. A lower value results in a linearly scaling penalisation. The constant 0.75 by which we multiply the clamped value describes how important is the vessel coverage information. A higher value of this term would increase penalisation for objects not intersecting the vessel structure. We observed that it is possible to select thresholds in specific images with an almost perfectly circular and bright optic cup during our data analysis. Further, the optic disc area has an

uneven boundary increasing its ellipse similarity score. Therefore, based on the variance of ϕ, we decided that an object which does not intersect vessels should be penalised by having its score four times as high as those that fulfil the vessel coverage condition. The calculated value is subtracted from one so that lower values are associated with better objects.

Vessel coverage is a good indicator for eliminating false disc-like objects which are too small. However, it cannot distinguish those too large. There are two types of objects more extensive than the optic disc, which our algorithm might consider. The first is visible parapapillary atrophy, which we described in the introduction. It appears as a bright region surrounding the optic disc with an uneven boundary due to the noisy texture present in this area. The second type of large object is caused by selecting a low threshold and obtaining an elliptical object by chance. We cannot avoid this problem in our algorithm due to a broad search for a suitable threshold. Based on our data examination, we concluded that these objects often share a significant part of their boundary with the region of interest. Therefore, we define an object boundary score λ as

$$\lambda = \frac{1}{(1 - \frac{|A_{boundary} \cap A_{obj}|}{|A_{boundary}|})^2 + \alpha}. \tag{3.16}$$

The formula computes the intersection of the object's area A_{obj} and the boundary of the examined image $A_{boundary}$. It divides the result by the number of pixels that lie on the image edges. This value is subtracted from one to obtain $\lambda = 1$ for an object which does not intersect the image boundary. We get a value of less than 1 otherwise. We take the result to the power of two to reinforce the effect. This exponentiation is based on observations where objects with large edge intersection were convex. This shape lowered their ellipse similarity score, and they were falsely selected as optic disc candidates. Finally, to ensure that lower values denote better objects, we invert the formula and add a smoothing constant α to the denominator. The added constant eliminates problems with division by zero, and we set it to 0.01. The boundary score is non-linear to not penalise small intersection severely but to ensure that objects spanning the whole image are not considered as optic disc candidates. Note that the non-linear scaling of λ without exponentiation has a value of two for an object which intersects half of all edge pixels. This penalisation can be overcome by a non-convex object sharing only a small part of the image border. Therefore, we introduce exponent into the formula to increase the penalty of objects which span nearly half of the image boundary.

The third score factor is denoted by δ and we define it as

$$\delta = 0.5(1 + \frac{d}{h})(1 - (D_{obj} - D_{neighbour})). \tag{3.17}$$

It describes two of general optic disc properties. Let us assume that we have an object described by a binary mask, and we want to evaluate δ on this object. The first described property, defined as $0.5(1 + \frac{d}{h})$, is a penalisation for objects which are far from the centre of the RoI. We assume that the second phase of our region of interest algorithm centres the square on the optic disc. The term d is the distance between centres of the region of interest and selected object. We divide this distance by half of the RoI diagonal. The second property

describes the contrast between an object and its background. We define it as $(1 - (D_{obj} - D_{neighbour})$, where D_{obj} denotes the median intensity value from the processed image restricted to the object area. Next, the term $D_{neighbour}$ denotes the median intensity value from the processed image restricted to the area surrounding the object. We compute the surrounding area as the difference between the dilated version of the object and the object itself. The dilation radius in the computation is 20% of the image width. Finally, we subtract the $D_{neighbour}$ from D_{obj}. This value is always greater than zero because pixels of the object have values above the threshold, and the neighbourhood pixels have values below it. Lastly, to assign a lower score to brighter objects, we subtract the calculated value from one. Multiplication of the distance and contrast terms produces the final factor δ.

The combined score ϵ is computed for every proposed object and defines an empirical function on the set of possible objects. We can find the optic disc by minimising this function. It is defined for every object. However, its global minimum is not necessarily the ground truth due to our empirical definition. Note that there does not exist a function that describes features of retinal images perfectly. Nevertheless, the minima of the function represent reasonable estimates of the true optic disc, and we are looking for them by applying a modified binary search defined in this chapter.

Following the application of our iterative algorithm to several colour channels in search of the minimum of our optic disc score, we obtain a proposed object in low resolution for each channel together with its score. We select the object with the minimum score and continue by processing only the colour channel of the best object. The concluded initial phase is followed by another phase which executes almost the same algorithm at a higher resolution. Instead of normalisation of the image, we reduce the intensity of all pixels selected as background in the previous phase by 10%. This modification assumes that previously obtained results are a good estimation, and we suppress the noise in the background. An additional difference of the second phase is that we start with a threshold that produced the best object in the previous phase. The starting threshold t_{start} and a new interval boundary for the following phase t_{min} and t_{max} are defined as

$$\begin{aligned} r &= \min(t^i_{max} - t^i_{best}, t^i_{best} - t^i_{min}) \\ t^{i+1}_{start} &\leftarrow t^i_{best} \\ t^{i+1}_{min} &\leftarrow t^i_{best} - \frac{1}{3}r \\ t^{i+1}_{max} &\leftarrow t^i_{best} + \frac{1}{3}r \end{aligned} \tag{3.18}$$

We reduce the starting interval of the previous phase to $\frac{1}{3}$ instead of starting with the last examined range of the previous phase so that the next phase can resolve minor errors of the previous one. This approach is general, and it can be extended to multiple phases. However, we found that the third phase no longer improves the result.

Each iteration reduces the size of the search interval by $\frac{1}{2}$, so the reduction between phases amounts to slightly more than one iteration. We consider this to be a sufficient reduction that narrows the interval and allows for minor corrections.

To conclude the iterative algorithm, we introduced a multi-channel technique that can adaptively select a suitable colour channel for optic disc segmentation. We defined an iterative algorithm that selects the best threshold based on an empirical objective function. Finally, we extended the base algorithm by an additional phase which processes images at higher resolution and improves the final result.

3.5 Optic disc region preprocessing

We applied a preprocessing step to the region of interest for improved optic disc segmentation. Let us assume that our iterative algorithm successfully extracted the disc area. Now, we crop the source image to the minimum bounding rectangle of the disc, which becomes the source image for the optic cup segmentation. This rectangular area has a different colour distribution than the region of interest. Furthermore, we have a mask that marks the approximate shape of the optic disc. Therefore, RoI preprocessing cannot be optimal for the new rectangular area. We have to apply a new set of filtering steps to improve the segmentation of the optic cup.

Contrary to the task of disc extraction, blood vessels obscure a large part of the optic cup. As a result, it is advantageous to remove the vessels from our cropped image. We use the vessel removal algorithm proposed in Section 3.2 which replaces values of vascular pixels by those derived from the neighbourhood of the processed pixel.

The main property of cup segmentation is intensity difference in specific colour channels, e.g., the green channel or saturation from HSV colour space as proposed by several approaches in [4]. Consequently, we improve the contrast in selected channels by adaptive equalisation before we extract features from them. Note that we do not apply AHE to the image before segmentation into superpixels. We observed that reduced performance with equalisation applied before running the SLIC algorithm.

3.6 Superpixel based classification for optic cup segmentation

Extraction of the optic cup is significantly more difficult due to blood vessels in the respective area. Contrary to the optic disc, the optic cup is largely obscured by the vessel structures. In addition, it often happens that the covered part is located only on one side, depending on the displayed eye. Vessel regions make it nearly impossible to extract the optic cup purely on the intensity information from any given colour channel. We can see an example of a significantly obscured optic cup in Figure 3.26. The figure shows a particular retinal image before and after vessel suppression. The obscured optic cup region regained its colour at the boundaries of vessels. However, the thicker section to the left of the centre cannot be replaced accurately by using intensities from its neighbourhood.

It is easy to observe that the thresholding algorithm we introduced in Section 3.4 will not work when applied to the problem of optic cup segmentation. There are two main issues connected to the application of iterative thresholding. The

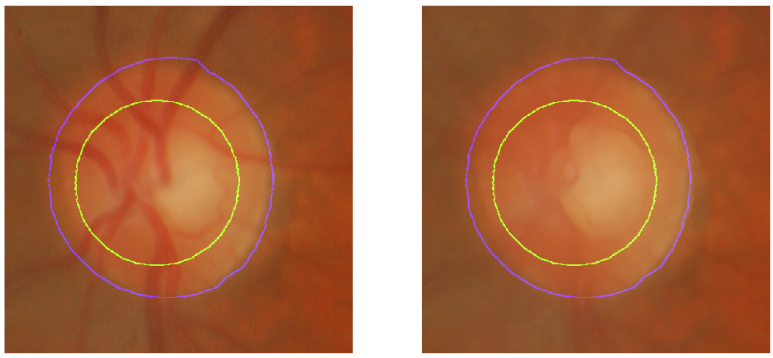

Figure 3.26: The source image with the optic disc and cup boundaries marked by blue and green respectively (left). An example of an obscured optic cup after vessel removal (right image). The suppression method fails to return the right intensity in the cup region due to vessel density.

first one is blood vessel coverage mentioned above, and the second problem is an undefined shape of the optic cup selected by a threshold. Whereas the disc tends to be an apparent ellipse with noise along its border, the best possible thresholded area of the cup can be non-convex. In general, the shape can have large fissures due to vessel interference. Additionally, the best possible segmentation by threshold might consist of several smaller objects that surround the obscured area. Selecting any single one of them will result in a significant error. Apart from that, these objects tend to be small and, therefore, have a lower ellipse similarity error, see Section 3.4.

Feature extraction

Segmentation of objects using machine learning models for classification revolves around the assignment of classes to individual pixels. We could extract features for each pixel and classify it as an object without considering its surroundings. However, there is a better alternative. Superpixels are small groups of pixels with similar properties that can be used to improve performance, and they naturally introduce pixel neighbourhood to the classification process. A comprehensive summary of superpixel advantages over simple pixels can be found in [28]. To divide the optic disc region into superpixels, we use the SLIC algorithm, which we introduced and described in Section 2.2.

After examining available data, we concluded that green channel from RGB colour space, saturation from HSV and a* from L*a*b* are the best candidates for intensity-based optic cup segmentation. The green channel is described as one with the highest contrast between the optic cup and neuroretinal rim in multiple proposed approaches [4]. In [55], it was noted that in many cases, the a* channel shows high contrast between the two retinal features. Further, the saturation channel was used as one of the features for optic cup segmentation by a classifier in [25]. During further investigation, we discovered that channels Y from the

YIQ colour space, L* from L*a*b* and blue from RGB, show high optic cup contrast as well. Note that we used the blue channel for optic disc segmentation as well. It can contain information about either the OD or OC. Therefore we decided to select the channels G, S, a*, Y, L and B as the base intensity feature channels for our algorithm. We computed the mean value of pixels belonging to each superpixel. The acquired values represent colour intensity features which we use in classification. In Figure 3.27, we can observe the intensity distribution in the respective colour channels together with the optic cup boundary marked by experts.

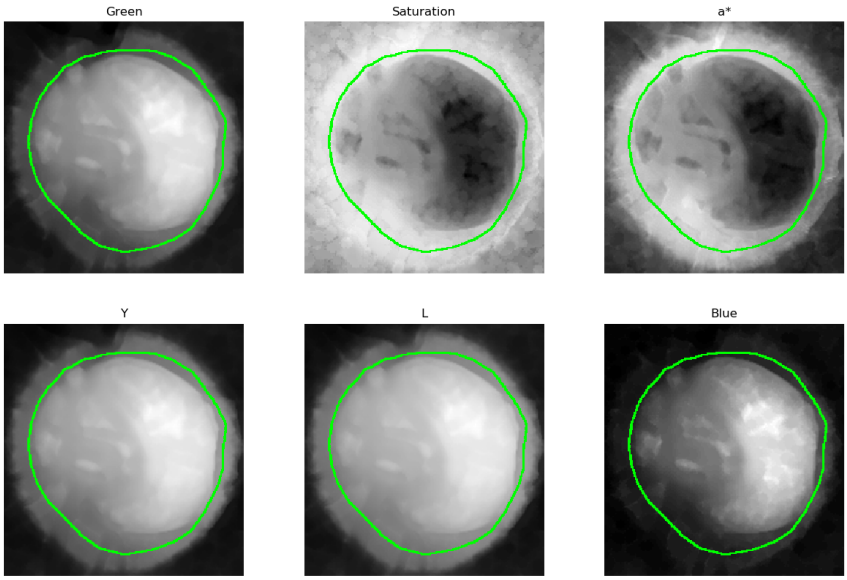

Figure 3.27: The main channels for intensity based extraction of the optic cup. From left ot right, top to bottom, they are G from RGB, S from HSV, a* from L*a*b*, Y from YIQ, L* from L*a*b* and B from RGB. Green boundary in the image shows the ground truth optic cup area.

The intensity information obtained in the previous step is less reliable than it was for the optic disc. Two exemplary problems are significantly different intensities in the blood vessel region obscuring parts of the cup and very low contrast between disc and cup in some of the examples. In these cases, intensity-based cup extraction fails and often selects pixels close to the optic disc boundary because of random intensity fluctuation in the area. To improve the cup segmentation, we introduce the distance from the approximate optic centre, given by the mean position of optic disc pixels, as a new feature denoted by d_i for ith superpixel. The feature is shown in Figure 3.28. It was used in [25] to stabilise the classification and reduce the above-mentioned negative effects of intensity-based segmentation.

Additionally, we introduce features that represent relation with respect to the lowest value in the image. This feature is computed for the three main intensity channels G, S and a*. The exact formula is defined as $f_{min} = (F - \min F)^2$ where F denotes a feature vector of superpixel values. Let us write the f_{min} value for

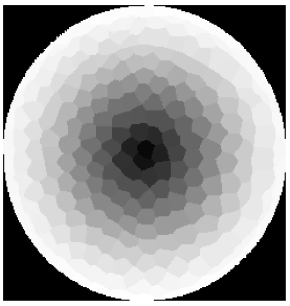

Figure 3.28: This figure shows the value of the linear distance from the approximate optic centre computed in each superpixel.

superpixel i and colour channel C as f_i^C. This feature enhances the contrast in the individual colour channels at the level of superpixels. The f_{min} features for colour channels green, saturation and a* are shown in Figure 3.29.

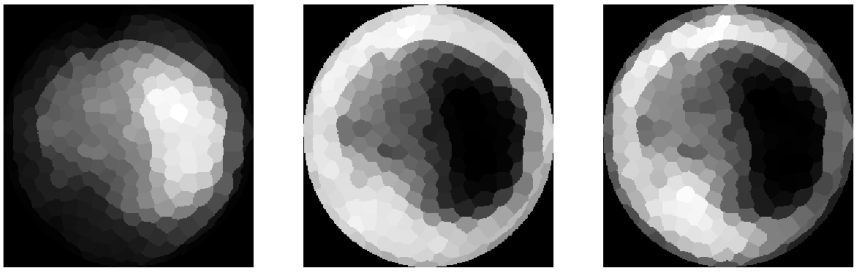

Figure 3.29: The f_{min} features: f^G (left), f^S (middle) and f^{a^*} (right).

The remaining set of features that we construct are based on centre surround statistics (CSS) introduced in [25]. We introduce two types of features. The first one follows the definition in [25] with slight modifications. We compute a Gaussian pyramid of the source image at resolutions: $256px$, $128px$, $96px$, $64px$, $48px$, $32px$, $24px$, $16px$. Each scale is filtered using Gaussian kernel with standard deviation of 5. We number these image from 0 to 7 and mark them as f_i. Then we compute the differences $f_{i,c} = |f_i - f_{i+c}|$ between numbers $i = \{1, 2, 3, 4\}$ and those obtained by adding offsets $c = \{3, 4\}$. The full set of differences contains $1-4$, $1-5$, $2-5$, $2-6$, $3-6$, $3-7$ and $4-7$. The final CSS features are the first two moments calculated within the individual superpixels, it is the mean and variance respectively. Let us denote them $\mu_{f_{i,c}}$ and $\sigma^2_{f_{i,c}}$. CSS features computed for the green channel of the image are shown in the figure 3.30.

The second type of centre surround features is based on the values of superpixels and their neighbours. The idea is derived from the centre surround statistics introduced in [25]. Let us assume that we have a set of superpixels with values representing a feature, for example, the green channel of an image. In addition, we have the centre positions of the superpixels. To compute the new type of centre surround features, we define the neighbourhood of a superpixel

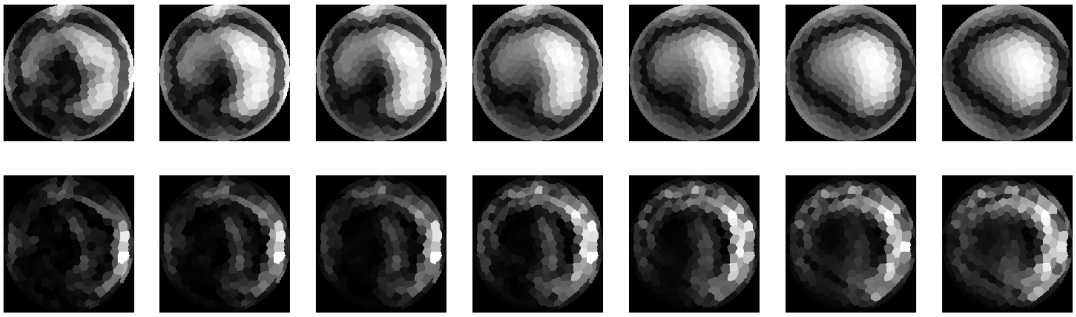

Figure 3.30: Centre surround statistics computed for the green channel of the image. The top row shows the mean features, and the bottom row shows variance.

as the set of its closest neighbours in the image plane. Instead of comparing values at a different resolution, we compare them at increasing neighbourhood sizes. The sets of closest neighbours are denoted by p and q. They represent the percentage of superpixels that are considered to be a part of the neighbourhood. For each pair of vectors p, q, we calculate their distance normalised versions $p \leftarrow pN(d_p; 0, d_{max}\delta_p)$ $q \leftarrow qN(d_q; 0, d_max\delta_q)$. The term $N(d_v; 0, d_{max}\delta_v)$ is a vector of values obtained by sampling from Gaussian distribution at points given by the distances from the currently processed superpixel, denoted by d_v. The distribution is centred at 0, and its standard deviation is the percentage determining the size of the vector v, marked δ_v, multiplied by the maximum distance of any superpixel d_{max}. Then, we compute the matrices $S_{p,q} = |p - q^T|$ and $D_{p,q} = |\frac{p}{q^T}|$ where $|A|$ denotes the per-element absolute value of matrix A. The final group centre surround (GCS) features are the first two moments of values present in the matrices. Let us denote the means: $\mu_{S_{p,q}}$, $\mu_{D_{p,q}}$ and variances: $\sigma^2_{S_{p,q}}$, $\sigma^2_{D_{p,q}}$. The sizes of groups used for the computation of vectors p and q are $5\% - 15\%$, $5\% - 20\%$, $10\% - 20\%$, $10\% - 25\%$ and $15\% - 25\%$ of the number of superpixels in the image. A showcase of this set of features computed for the green colour channel can be seen in Figure 3.31.

Both types of centre surround features are computed from a single colour channel. Therefore, we extract them from each of our primary cup segmentation channels G, S and a*. Three sets of CSS and GCS features give 42 and 60 values for a single superpixel, respectively. We extend this set by 6 features from the individual colour channels. Then, we add a relation to the minimum superpixel value f_{min} for each of the primary colour channels. Finally, we add the distance from the approximate optic centre to obtain the final set of features for superpixel i: $\{CSS_i, GCS_i, G_i, S_i, a^*_i, Y_i, L^*_i, B_i, f^G_i, f^S_i, f^{a^*}_i d_i\}$. In total, there are 112 values for decision making.

Training

Our gradient boosted tree classifier is trained using per-superpixel data created from our training dataset. We compute feature data on each image separated into 256 superpixels. We compute the labels using superpixel intersection with the ground truth mask. We use a threshold of 0.7 which specifies that superpixels with at least 70% pixels belonging to the ground truth mask are marked as part of the optic cup. Our dataset contains optic cup boundaries selected by multiple

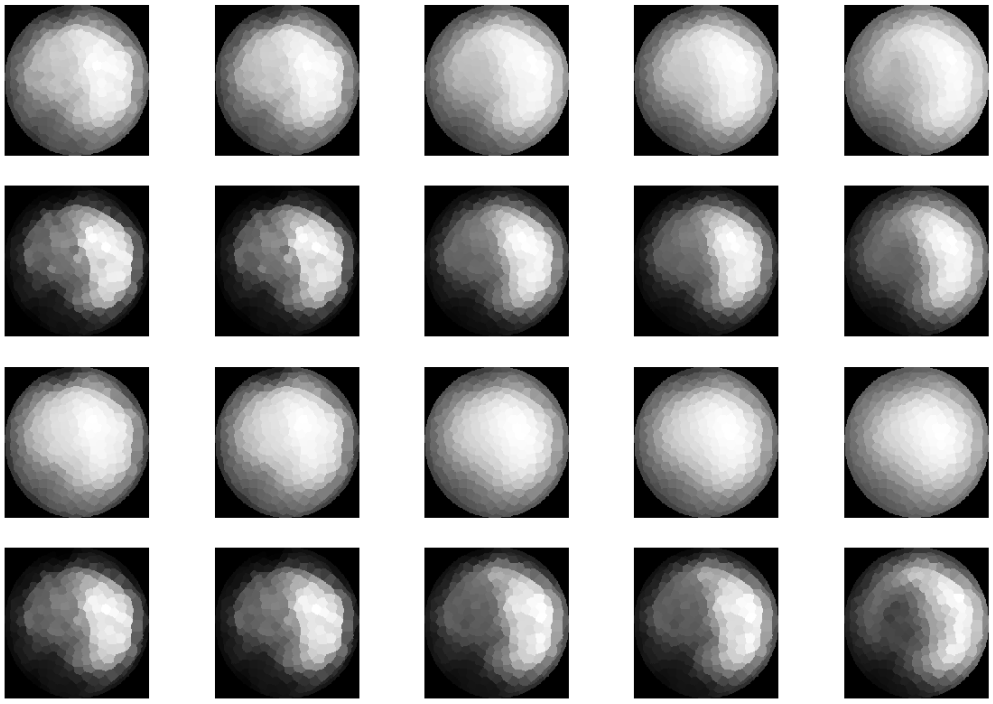

Figure 3.31: Group centre surround features computed for the green channel of the image. The rows from top to bottom signify the mean of the matrix $S_{p,q}$, the variance of the matrix $S_{p,q}$, the mean of the matrix $D_{p,q}$ and the variance of the matrix $D_{p,q}$.

clinicians. We define the ground truth area of the cup as the area chosen by the majority.

We can observe large differences between the number of superpixels that belong to the optic cup and those that belong to the neuroretinal rim. An example of two extremes is given in Figure 3.32. To balance the data quantity, we select an equal number of superpixels for the rim and cup from each image. The selection is realised by computing the minimum of the two counts and preserving the complete smaller set, and choosing random superpixels from the larger one. In general, it might be better to choose an equal number of training examples from each class at random. However, our training dataset is relatively small, and we do not want to reduce it further. Thus, we decided to use all superpixels from the smaller optic nerve head feature.

In addition, we noted a bias towards large optic cups within our dataset. To improve the performance of the model for smaller cup sizes, we applied selective data augmentation described in Section 3.6.

To select the parameters for our gradient boosted model, we used 5-fold cross-validation. The final training parameters of our classifier are 500 trees in the ensemble, learning rate of 0.005 and maximum depth of a tree set to 11. Let us discuss the selected parameters in the context of the highest cross-validation score. The depth of trees is the main parameter describing the ability to fit the training dataset. Note that our data contains 112 features. Building deeper trees causes over-fitting, and at a depth of around 21, this classifier becomes able to distinguish training superpixels perfectly. Over-fitting results in worse generalisation, and

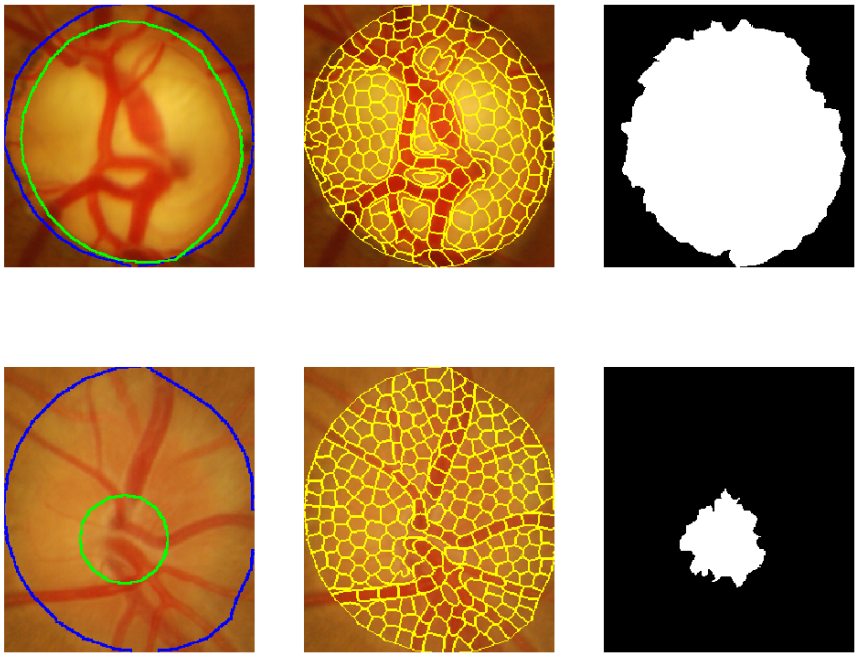

Figure 3.32: This figure shows two extreme examples of the optic cup size. The first row shows an image with a large optic cup (left), its decomposition into superpixels (middle) and a subset of superpixels labelled as a part of the optic cup (right). The second row shows the same examples for an image with a small cup. The ground truth optic disc and cup boundaries are marked in the first column with blue and green colours.

therefore we decided to limit the depth. On the other hand, setting the depth to lower numbers tends to under-fit on the dataset. Both training and validation results are worse due to the low capacity of the model. Furthermore, the number of trees in the ensemble and the learning rate is less important, and we observed only minor differences between models with tree count in the range from 200 to 2000. We examined the results of setting learning rate to values complementing the number of trees for every ensemble. It is important to note that the learning rate for gradient boosted tree classifier is, in fact, the weight of each tree in the final sum. For more information, see [46]. Therefore, it makes little sense to select a learning rate such that the sum of weights is less than one. By the sum of weights, we understand the learning rate multiplied by the number of trees. Validation results showed that a higher sum of weights performs slightly better, and the best results were acquired for the sum equal to 2.5. We investigated the performance of classifiers with the sum of weights ranging from 1 to 10. Next, opting for less than 200 trees in the ensemble resulted in a more significant under-fitting. Lastly, we train the gradient boosted decision tree classifier with subsampling. Only half of the data samples are used to train the individual trees

to reduce the generalisation error of the model.

Data augmentation

The training dataset which we used to train our classifier is composed of 50 images. It is a very low number for a machine learning task, and we immediately noticed a bias towards large optic cups. In Table 3.6, we show the distribution of optic cup sizes in our dataset. A denotes the percentage of optic disc covered by the optic cup. In [9], the authors mention that the optic cup is usually quite small compared to the entire optic disc. In any case, smaller cup sizes are undoubtedly underrepresented in our dataset.

Dataset	$A > 70\%$	$A > 60\%$	$A > 50\%$	$A > 40\%$	$A > 30\%$
Training	36.00%	62.00%	76.00%	88.00%	94.00%
Testing	37.25%	70.59%	80.39%	84.31%	92.16%

Table 3.1: This table shows the distribution of optic cup sizes with respect to the optic disc.

We decided to apply selective data augmentation to offset the low amount of images with a small cup. Generally, data are augmented using the same methods in the whole dataset and transformation between two images differ only in the randomly generated parameters. That would not help with the problematic distribution. Therefore, we evaluated a simple clustering algorithm on our dataset using features engineered to provide higher contrast between the optic cup and neuroretinal rim. More information about complex features can be found in Attachment A.2. The simple clustering algorithm failed to segment smaller optic cups due to the set of features computed to maximise performance on the training dataset.

Following that, the training dataset was manually split into four groups. The first group contained well-segmented images with an intersection over union score greater than 0.8. The second group was composed of images where the segmented result was good, but its boundary was distant from the ground truth in some places. These object had an IoU score in the range from 0.7 to 0.8. Images in the third group were segmented poorly, and either the position of the result did not match the ground truth, or the size was significantly different. The IoU score for this group was in the range from 0.6 to 0.7. Finally, the fourth group contained images with inferior segmentation results with an IoU score of less than 0.6. Investigation of the poorly segmented images led us to conclude that the latter groups were composed of images with comparatively smaller optic cups and those that were positioned notably off-centre. Note that the last two groups had fewer examples than the first two because we constructed features to improve the performance of the model on the whole training dataset. This type of feature engineering resulted in the introduction of dataset bias to the feature set.

To reduce the bias of our training set, we add additional augmentation to images from the latter groups. In detail, for every image from the training dataset, we compute its horizontally mirrored image and add it to the final training set. Further, we also add a vertically mirrored version for images from the second,

third and fourth group. The last two groups are also augmented by one random rotation of the original image. Finally, images from the last group are augmented by adding a second random rotation of the original image as well as one randomly rotated version for their mirrored counterparts. Each random rotation is taken uniformly from the range 5 to 85 degrees. The last group is the most represented one, excluding the well-segmented images from the first one. Therefore, we decided to add significantly more augmentations for this group compared to the other ones.

All possible augmentations of an image are shown in Figure 3.33. Note that we compute the optic disc mask only for the original and flip or rotate it with the source image. Also, the optic cup processing pipeline works with a small region around the optic disc so the clipped corners of rotated images cannot interfere with it.

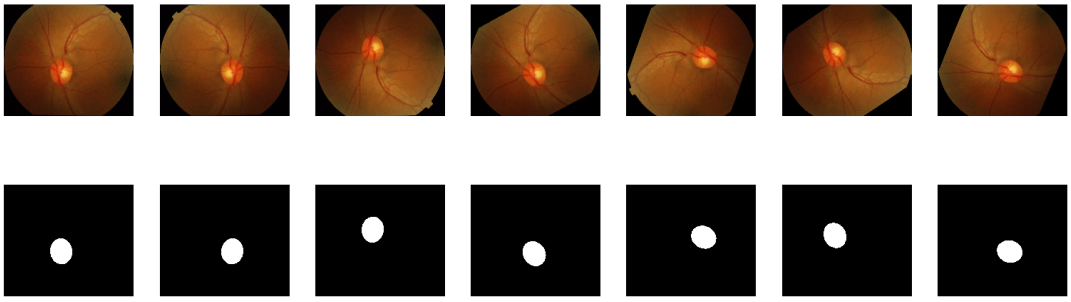

Figure 3.33: All possible augmentations applied to one of the training images with its optic disc mask. From left to right, they are the original, horizontal flip, vertical flip, two rotations of the original, rotation of the horizontally flipped version, and the last is a rotation of the vertically flipped version.

In summary, we selectively augmented training images based on information gathered from a simple clustering model. We split the training set into four groups defined by the performance of the afore-mentioned model. Then, we increased the number of augmentations for the worse performing ones.

Prediction

Prediction of the optic cup region is realised in several steps. Firstly, we apply the optic disc region preprocessing. Then we segment the image into superpixels and compute features on the source image using the same parameters as for training. Note that no data augmentation is done during prediction. Then, we classify acquired superpixels using our gradient boosted classifier. We assign a class to every pixel in the image based on the superpixel which contains it. This step creates a binary mask which is our initial object prediction. Superpixel classification results in masks with an uneven boundary. To smooth out the result obtained from the classifier, we compute its convex hull. Then we interpolate the CH using B-spline interpolation and, at last, fit an ellipse to the interpolated result. The smoothing starts with a convex hull due to the possibility that blood vessels obscure a large part of the optic cup, and the segmented result can be non-convex with a depression at the vessel root. The resulting polygon can contain long sides around the non-convex parts of the initial boundary. Thus we apply

interpolation to extrude the straight lines of the convex hull slightly. Finally, we fit an ellipse to the interpolated boundary because optic cups have a clear elliptical shape. We denote all pixels within the ellipse as part of the cup and everything else as background. The resulting mask is returned as the final prediction of our model.

4. Evaluation

All training and evaluation algorithms were executed on Windows 10 system, Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz, 16 GB RAM. The algorithms are implemented in Python programming language using image processing and machine learning libraries including scikit-image [29], scikit-learn [31] and scipy [38].

In the following text we write the results of our algorithm in the form $\bar{x} \pm \sigma$ where $\bar{x}$ denotes the mean value from the entire dataset and σ marks the standard deviation. The variance of the results is computed using an empirical biased estimate given by the formula $\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x - \bar{x})^2$. The standard deviation is computed simply as the square root of variance.

In addition to the main algorithm described in Chapter 3, we introduce its alterations which were considered during our investigation of the segmentation problem. In each section marking part of the algorithm, we describe the performance of the proposed approach. We present previous versions of the algorithm and a comprehensive comparison of the results achieved using the old and new version. All versions of our algorithm are evaluated primarily on the Drishti dataset, which has a prior separation into a training and a testing set. Therefore, we do not add randomness as a factor in data evaluation.

Drishti dataset

Drishti retinal dataset contains 101 retinal fundus images split into training and testing set with 50 and 51 images, respectively. Each image has a soft map with a ground truth optic disc and optic cup provided in PNG image format. As a result, researchers can compare their segmentation techniques with the gold labels marked by experts. In addition, the dataset contains CDR values for evaluation of calculated CDR values with the defined ones. The dataset was created by researchers of the International Institute of Information Technology, India, together with experts from Aravind eye hospital, Madurai, India. Images were collected from visitors to the hospital with their consent. Selected patients were 40-80 years of age with a roughly equal number of males and females. The dataset contains images of varying brightness and contrast. Bad quality images in terms of contrast or OD positioning were discarded [8]. The ground truth retinal features marked by expert clinicians are provided in the form of soft mask images. Each soft mask contains regions selected by multiple experts, and we consider the pixels marked by the majority as the ground truth.

Dataset bias

We analysed the distribution of optic cup sizes in Section 3.6. The results show that our dataset consists of retinal images with large optic cups. In detail, 36% of optic cups in the training dataset are larger than 70% of the disc area, whereas only 12% are smaller than 40% of the disc area. More detailed information is written in Table 3.6. We introduced custom data augmentation to reduce the

dataset bias. However, it is easily observable that our classifiers still learned the bias. Note the high recall and relatively low precision of results in Table 4.4. The only true solution to this problem is a larger and more diverse training dataset. Therefore, it is likely that any approach will be biased towards the optic cup size distribution of its training dataset. In addition, images from a single set are usually taken on one or more similar devices. Each device introduces bias of the imaging technique. An algorithm trained on a particular dataset can analyse images taken by a specific set of devices. To improve the performance of our algorithm on unseen type of retinal images, we applied contrast-enhancing techniques and automatic colour channel selection.

Optic disc segmentation results

Our optic disc segmentation algorithm performs well in both the training and testing datasets. There is no image in which it fails to extract the region of interest, and there are only minor errors in the segmentation of the final optic disc shape. There is one image in both training and testing dataset which produces undesirable results due to a noisy region surrounding the optic disc. The results of our algorithm evaluated using common segmentation metrics introduced in Section 2.10 are shown in Table 4.1. Both precision and recall values are similar, and therefore, there is no bias towards segmenting small or large objects. Further, a low standard deviation means that the results are close to the mean. The minimum intersection over union values are 0.6943 and 0.6014 for the training and testing datasets respectively.

	Drishti training dataset	Drishti testing dataset
IoU	0.9237 ± 0.0499	0.9198 ± 0.0560
Dice metric/F1 score	0.9596 ± 0.0289	0.9572 ± 0.0344
Recall	0.9762 ± 0.0266	0.9708 ± 0.0290
Precision	0.9464 ± 0.0572	0.9475 ± 0.0618

Table 4.1: The performance of our proposed optic disc segmentation algorithm on the Drishti dataset.

These results indicate that our iterative thresholding algorithm can determine the position and shape of the optic disc reliably. We noted a lower algorithm performance on images where the processed colour channel contains high values between the disc and the edge of the region of interest. The iterative algorithm falsely selects them as part of the optic disc. In some cases, they form an elliptical shape which results in convergence towards the wrong threshold. The main reason is that the better alternative in the algorithm contains non-convexities or deep depressions, and thus, has a higher score. These artefacts show up as local minima in our empirical function, and an unlucky initial selection of the examined threshold cannot escape them. An example of an image where the algorithm failed to converge to the correct shape is shown in Figure 4.1. Note that ellipse fitting partially resolved the error introduced by failed convergence.

Additionally, due to the low variety of our dataset, few images are containing severe parapapillary atrophy. Consequently, our algorithm does not explicitly

Figure 4.1: The region of interest together with the final ellipse (left). The selected object before smoothing together with the final ellipse.

resolve the related problems. A subsequent iteration of an image with elliptical PPA can converge to the wrong threshold. We are working under the assumption that the PPA region has a noisy boundary, and it will have a poor value of our objective function. Despite that, the algorithm can deal even with more complicated examples. In some cases, it managed to correctly extract the disc region even though there is a clear circular PPA region surrounding the OD. An example of such image is in Figure 4.2. The success of segmentation can be attributed to low noise on the optic disc boundary in this case.

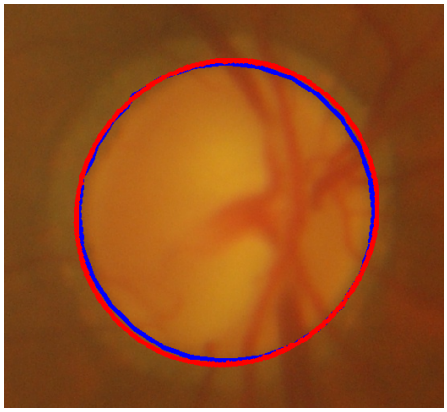

Figure 4.2: This figure shows a correctly extracted optic disc from an image with a clear circular PPA region. Blue denotes the ground truth and red the segmented ellipse.

One of the disadvantages of our algorithm is that it is pretty slow. It takes an average of 18.92 seconds to extract an optic disc. The most time-consuming parts of the algorithm are the first and second phase with around 5 and 7 seconds needed for their computation. The remaining time is split between RoI detection, vessel extraction and image pre-processing.

Our results can be compared with other approaches. An active shape modelling approached evaluated on the Drishti dataset presented in [56] reached F_1 score of 0.9077. The thresholding method with PPA elimination introduced in [11] reports the non-overlap area ratio of 0.1. This metric is computed as the

complement of the intersection over union metric, which evaluates to 0.0802 on our testing dataset. Then, a clustering approach based on fuzzy c-means and morphological operations in [23] achieved an F_1 score of 0.927 on a dataset consisting of 27 images. Next, in [14], the authors report F_1 score of 0.96 using a total of 98 images. We achieved the same results up to a rounding error. Finally, an overlap area ratio of 0.94 is reported on a dataset of 50 images by the authors of [20]. In conclusion, our algorithm achieves comparable results. Even though there are algorithms that outperform ours, the iterative thresholding approach with an objective function we designed shows promising results.

Algorithm modifications

The proposed algorithm consists of multiple steps. Some of these steps are optional, and others have alternatives that we considered during development. In this section we present results of some modifications and alternatives to the proposed approach.

The first interesting alternative of our algorithm is its single phased version. In order to extract the best possible optic disc area, we execute our algorithm in two phases with increasing resolution of processed images. We discussed this decision in the algorithm definition, see Section 3.4. The core idea is an iterative improvement of the threshold and consequently the segmented object. Instead of two phases, we could execute the algorithm only with a single phase at a predetermined image resolution. The first phase of our algorithm processes images at a resolution of $128px$, whereas the second phase at $256px$. Therefore, we decided to examine a single-phase version of the algorithm in these two resolutions. We increased the number of iterations for the single-phase version so that the final threshold search interval is approximately the same as for two phases. We can see in Table 4.2 that all three algorithms achieve similar results. Based on the intersection over union metric, the single-phase version running at a resolution of $128px$ performs the best. We noted that the higher resolution version of the algorithm and the two-phase versions perform better on images with uneven boundary. Nevertheless, the low-resolution result is already an excellent approximation of the true optic disc. Furthermore, the single-phase version processing $256px$ images takes about three times longer to compute than the proposed two-phase version. On the other hand, the single-phase algorithm processing $128px$ images is about 30% faster than the two-phase version. A higher resolution algorithm improves the results on more complicated examples, but the difference is marginal. Therefore, without further optimisation, the single phase version is better because of its segmentation speed. The main algorithm describes th two-phase version because we decide to present the more accurate version rather than the fastest.

Another interesting modification of the algorithm is at its post-processing step. The boundary of a ground truth optic disc is very smooth and often forms a perfect ellipse. Therefore, it is advantageous to apply boundary smoothing to the object acquired through our thresholding algorithm. In Section 3.4, we selected ellipse fitting of the raw object based on the performance of this method on the training dataset. There are three alternatives, taking the convex hull of the object, interpolating the convex hull and fitting an ellipse to the convex hull

IoU	Drishti training dataset	Drishti testing dataset
Two phases (128+256)	0.9237 ± 0.0499	0.9198 ± 0.0560
Single phase (128)	0.9235 ± 0.0564	0.9217 ± 0.0561
Single phase (256)	0.9275 ± 0.0492	0.9205 ± 0.0565

Table 4.2: Comparison of the performance of our optic disc segmentation algorithm with a single phase and the proposed two phases. The numbers in brackets signify the pixel resolution of processed images.

rather than the object. Due to the natural smoothness of the optic disc, we exclude basic CH because it is polygonal. We compare the remaining techniques in Table 4.3. It shows that our chosen method performs best on both training and testing datasets. The optic disc is usually well separable, and other retinal features do not obscure it, and therefore, the base algorithm can extract the approximate shape quite well. Consequently, computing a convex hull of this shape might result in an object which is too large. Further, fitting an ellipse to the convex hull reinforces this problem. Conversely, fitting an ellipse to the boundary of the extracted object assumes that we have the right size and only smooths the boundary.

IoU	Drishti training dataset	Drishti testing dataset
Convex hull	0.9178 ± 0.0553	0.9142 ± 0.0614
Ellipse fitting	0.9237 ± 0.0499	0.9198 ± 0.0560
Ellipse fitting of CH	0.9078 ± 0.0633	0.9078 ± 0.0675

Table 4.3: The performance of different object boundary smoothing techniques for optic disc segmentation.

To summarise the algorithm modifications, it is essential to consider what kind of objects the base algorithm produces and to apply an appropriate post-processing techniques. Furthermore, it is important to consider possible alternatives and extensions. In our case, introducing additional phases to our base algorithm resulted in better performance on the worse examples at the cost of a significant increase in execution time.

Optic cup segmentation results

Our optic cup extraction algorithm performs well on most of the training and testing datasets. There are examples in both sets where the algorithm fails to achieve an intersection over union greater than 0.5. The problematic images contain small optic cups, which the model cannot distinguish due to the prevalence of larger cups in our training dataset, as we noted before. Conversely, the algorithm performs well for mid to large cup sizes. The results of evaluation on the Drishti dataset using the standard metrics introduced in Section 2.10 are shown in Table 4.4. The two images, which proved to be very challenging for the optic disc extraction algorithm, ended up having an intersection over union of less than

0.2. The selected discs contain a large part of the background, and the ground truth optic cup is the smallest in the whole dataset.

	Drishti training dataset	Drishti testing dataset
IoU	0.8654 ± 0.1357	0.7857 ± 0.1594
Dice metric/F_1 score	0.9203 ± 0.1050	0.8687 ± 0.1287
Recall	0.9910 ± 0.0156	0.9154 ± 0.0700
Precision	0.8737 ± 0.1403	0.8651 ± 0.1935

Table 4.4: The performance of our proposed optic cup segmentation approach evaluated using the standard criteria.

The time it takes to get a prediction from features is only a moment in every presented model. However, we apply complex image pre-processing steps and feature extraction, which significantly increase the execution time. We measured how long it takes to evaluate the optic cup extraction algorithm, and we discovered that it takes an average of 13.63 seconds. This value does not take any optic disc processing into account. The complete algorithm, as shown in Figure 3.1 takes an average of 32.21 seconds. The algorithm is relatively slow, but the evaluation shows that gradient boosted decision trees can perform better than other classification algorithms for the task of superpixel-based optic cup extraction.

Let us compare our algorithm to other methods proposed in the literature. A low rank superpixel representation used in [21] reached non-overlap area ratio of 0.244 on the *ORIGA* dataset consisting of 650 images. However, the optic disc extraction is not part of the algorithm. An approach based on fuzzy c-means clustering introduced in [23] achieved F_1 score of 0.892 and precision 0.999 on a dataset consisting of 27 fundus retinal images. Further, an automatic thresholding algorithm with morphological operations presented in [14] reached F_1 score of 0.88 using a total of 98 images. In conclusion, our algorithm achieves comparable results to other approaches in the literature. Even though some of the algorithms outperform ours, gradient boosted decision trees show promising results.

Alternative classification models

Gradient boosted decision trees is an algorithm we chose based on the performance of several supervised algorithms. Our initial approach used support vector machines which we introduced in Section 2.8. SVM was successfully used in [25] for high dimensional optic cup and disc classification. The authors propose the use of linear SVM. We found that for our dataset, SVM with the radial basis function kernel performs better. However, both versions of the algorithm underperform for a high dimensional feature space used for decision tree learning. We evaluated the supervised algorithms on the Drishti testing dataset. We present the results in Table 4.5. Their performance is compared using the intersection over union metric.

Investigating supervised algorithms was our second course of action. Before that, we examined the performance of unsupervised clustering techniques. General description of these algorithms is available in Section 2.3. Our main discov-

Test	Random forest	SVM-Linear	SVM-RBF
IoU	0.7758 ± 0.1608	0.7531 ± 0.1575	0.7569 ± 0.1575
Dice score/F1 score	0.8621 ± 0.1305	0.8475 ± 0.1315	0.8501 ± 0.1304
Recall	0.9122 ± 0.0717	0.8824 ± 0.0829	0.8887 ± 0.0806
Precision	0.8588 ± 0.1995	0.8651 ± 0.2074	0.8624 ± 0.2054

Table 4.5: The comparison of supervised machine learning algorithms applied to superpixel classification for the optic cup segmentation.

ery is that clustering algorithms perform significantly worse on simple features for supervised learning. Therefore we applied more complex feature engineering intending to normalise cluster sizes. The optic cup is not well separable by clustering. All pixel and superpixel values span the entire available range, and there is no separation margin. We show a plot of superpixels with green and a* values with ground truth separation of superpixels in Figure 4.3. The left image shows the result of clustering using the fuzzy c-means algorithm. It is easily observable that the default distribution of values within these colour channels is insufficient for accurate segmentation. The feature modifications used for the classification via clustering algorithms are described in Attachment A.2.

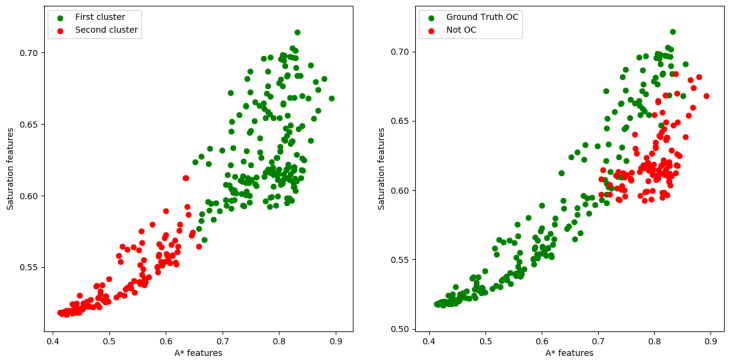

Figure 4.3: Two clusters classified by the fuzzy c-means algorithm in the dataset created from the green and a* channels of the source image (left). The ground truth class assignment of the same dataset (right).

The performance of the different clustering techniques on the Drishti testing dataset can be seen in Table 4.6. We chose only algorithms which receive a predetermined number of clusters. Those that infer everything from the data failed to perform consistently on the whole training dataset, and therefore, we excluded them from the evaluation. Their main problem stems from image dependent parameters of the algorithms. This dependency causes the requirement to search the parameter space of the algorithm for every image. We could not find any correlation between global image describing values such as its mean or variance and the required parameters.

We examined five main algorithms representing the types introduced in Section 2.3. We opted for an agglomerative approach for the hierarchical clustering type with Ward's method for merging clusters. We define two classes of super-

pixels: with a low and high probability of belonging to the optic cup. Clustering produces assignment into these two classes, and we select the closest one to the approximate optic centre as the optic cup. We introduced distance from the optic centre and vessel neighbourhood modifier to the clustered features. Descriptions of these modifications are written in Attachment A.2. The result of our feature engineering is that the disc area often splits into two parts: the bright part of the optic cup together with vessels close to the centre and the neuroretinal rim. The second class often contains superpixels that belong to the cup due to the low contrast between the rim and these superpixels. Following the extraction of the base object, we smooth its boundary. The selected cluster might be composed only of the brightest part of the cup and centremost vessel area. This superpixel group might represent an object that is too small or composes only a part of the optic cup. Therefore, similarly to the main algorithm, we fit an ellipse to the convex hull computed from the boundary of the predicted object.

Test	IoU	Recall	Precision
K-means	0.6693 ± 0.1698	0.8574 ± 0.1265	0.7753 ± 0.2150
Birch	0.6451 ± 0.1720	0.8436 ± 0.1250	0.7788 ± 0.2351
Fuzzy c-means	0.6703 ± 0.1697	0.8592 ± 0.1242	0.7736 ± 0.2131
Hierarchical clustering	0.6432 ± 0.1729	0.8417 ± 0.1294	0.7789 ± 0.2352
Gaussian mixture	0.6215 ± 0.1695	0.7602 ± 0.1637	0.8286 ± 0.2363

Table 4.6: The comparison of unsupervised clustering algorithms applied to superpixel classification for optic cup extraction.

The best performing algorithms are fuzzy c-means and k-means. Hierarchical clustering and birch algorithm do not perform very well on this task. Finally, it is surprising that the Gaussian mixture model showed the worst results due to size differences between clusters belonging to the optic cup and neuroretinal rim. Feature engineering normalises the cluster sizes to an extent. However, a notable difference is still present. It seems that the density of points in the feature space is not a distinguishing factor, and GMM cannot estimate distributions with the correct standard deviation.

Let us summarise the selection of machine learning algorithms for the problem of optic cup segmentation. Supervised learning algorithms perform significantly better compared to unsupervised ones. Clustering techniques often produce substantially smaller or larger area and require a significant amount of feature engineering. Supervised algorithms can learn more information from basic features, and they tend to extract large objects. Maximum margin classifiers, i.e. SVM with linear and RBF kernel, produce better results on a smaller number of features. However, for a large feature space, decision tree ensembles perform better, and we found that gradient boosted tree classifier is the best performing algorithm on our dataset.

Algorithm modifications

Our optic cup extraction algorithm is composed of several steps which have their alternatives and modifications. In this section, we present some of the alternative

techniques we considered throughout development. We compare their results with the proposed algorithm and discuss the reason for higher or lower performance.

The first modification which we used before the introduction of blood vessel suppression is a combined classifier. It consists of two gradient boosted decision trees where one is trained on vessel areas and the other on non-vessel areas. Without blood vessel suppression, the combined classifier produces better results. However, the classification of superpixels with removed vessels brings the two approaches close to each other. We observed that the combined classifier could fit the training data more accurately, whereas the single classifier was better at generalisation. Note that evaluation on suppressed vessels does not affect the performance of the combined classifier as the replaced vessel values signalise the optic cup region. This property of vessel superpixels makes the segmentation easier. Nevertheless, the combined algorithm showed very similar results to the single model approach, and thus, we opted for the simpler alternative.

Another significant modification is the boundary smoothing technique which we discussed in the optic disc section as well. The result of model prediction for optic cup segmentation is a group of superpixels. They can compose one or several connected components. In Section 3.6 we noted that we select the largest component and apply ellipse fitting to the convex hull of its boundary. Instead of the selected technique, we can use the convex hull of the object, its interpolation, or apply ellipse fitting directly to the object's boundary. It is easy to see that ground truth optic cups are never polygonal. Therefore, we consider only the interpolated version of the convex hull. The performance of these three approaches is shown in Table 4.7. It shows the intersection over union metric computed on the whole training and testing datasets. Note that the value for our selected approach on the training dataset is lower than the other ones. It is caused by the fact that we did not select the smoothing technique based on this metric. Instead, as noted in Section 3.6, we chose to reduce the error introduced by blood vessels. The evaluation of both datasets shows that the selected technique is better at generalising. That is most likely due to the ability of the decision tree ensemble to fit the training data well, even in vessel areas. That results in lower performance of the technique, which tries to fix errors in these areas.

IoU	Drishti training dataset	Drishti testing dataset
Convex hull	0.8921 ± 0.1269	0.7704 ± 0.1445
Ellipse fitting	0.8951 ± 0.1226	0.7690 ± 0.1390
Ellipse fitting of CH	0.8654 ± 0.1357	0.7857 ± 0.1594

Table 4.7: The performance of different object boundary smoothing techniques for optic cup segmentation.

In conclusion, it is essential to consider combined classifiers and partial problem solutions since they can improve the result in many cases. We were successful in applying separate classifiers to the problem of the vessel and non-vessel areas. The introduction of blood vessel suppression improved the performance of single classifiers, which in some cases, e.g. gradient boosted decision trees, surpassed the combination. Further, the post-processing of selected objects is an important step. It fills in missing information based on empirical observations and improves

the performance of the model.

Conclusion

This thesis provides an overview of segmentation challenges and solutions with respect to the task of optic nerve head extraction from fundus retinal images as well as a new hybrid approach utilising gradient boosted decision trees. Firstly, we introduced the problem and described the properties of retinal images. Secondly, we established definitions of various algorithms used in the process of segmentation and classification. Next, we introduced a comprehensive list of performance criteria used for object comparison which allow us to compare different algorithms in a single place. Following that we proposed a hybrid optic nerve head segmentation approach. It consists of two separate algorithms. The first one extracts optic disc using modified binary search of intensity thresholds based on the quality of the selected object. The second part of the proposed approach uses gradient boosted learning to extract the optic cup. Both parts collectively create a single pipeline which has one input, i.e., a retinal fundus image, and returns masks of the optic disc and cup. The entire combined algorithm is composed of smaller sections which solve their individual tasks. Finally, we evaluated the performance of our approach together with its several modifications and wrote down the exact results using all previously described evaluation criteria. Not only did we evaluate the proposed algorithm and its versions which include or exclude certain steps, but also a set of different classification models which can be used to solve the problem. To complete the overview of these algorithms, we also provide a summary of their respective challenges and a description of observed behaviour which results in a better or worse performance. Now, let us discuss the enumerated achievements in more detail.

The first algorithm which we introduced is a intensity based region of interest search. We examined the challenges of the optic nerve head localisation and proposed a robust solution. Our region of interest searching routine disregards artefacts caused by imaging devices as well as bright noisy regions which signalise defects in the retina. To enhance the initial estimation of the optic disc location, we added a secondary phase to RoI selection which centres the square region on the approximate optic centre. Consequently, we introduced a reusable intensity based region of interest detection algorithm which produces high quality centred regions.

Additionally, we implemented a simple blood vessel extraction algorithm. Despite its simplicity it fully satisfies our requirements and it is capable of replacing vessel structures in the selected region of interest. However, it cannot be used for standalone vessel segmentation as it is prone to extracting falsely positive areas. Nevertheless, it is suitable for general blood vessel suppression.

Following that, we combined the previous algorithms with a wide binary to define an iterative thresholding approach for optic disc segmentation. To describe the disc likeness we introduced a score based on ellipse similarity. The function is defined for all objects obtainable within a specified region of interest and the closest approximation to ground truth optic disc lies in a local minimum. It is an empirical function based on our available dataset. Together with the iterative search it creates an easily understandable and versatile optic disc extraction algorithm.

The second part of the pipeline which we implemented is segmentation of the optic cup using gradient boosted decision trees. This classifier is capable of handling small and large feature spaces compared to support vector machines which often fail for high dimensional problems. In addition, it uses ensemble techniques innately. These advantages make it a great candidate for the optic cup classification.

Furthermore, we evaluated our segmentation algorithm on the Drishti testing dataset using all previously defined performance criteria. This ensures that our algorithm is comparable with other segmentation approaches. It is unlikely that other algorithms use a metric, which we did not present. Additionally, we present an overlap area ratio for objects segmented using alternative parts of the algorithm, which we considered during development. This completes an overview of different techniques we examined and evaluated.

Finally, to evaluate the performance of gradient boosted tree classifier with respect to other classification algorithms, we evaluate their performance. The results and a related discussion give an overview of advantages and disadvantages of different algorithms for the task of object segmentation.

In conclusion, we fulfilled all of our goals and we explored several additional techniques and algorithms. The full pipeline of our approach allows us to extract the optic nerve head and the applied image processing techniques improve the final result. All of the considered algorithms are evaluated and compared against each other.

Future Work

The segmentation approach proposed in this work fulfils the role of an optic nerve head extraction algorithm. It is fully featured in the sense that it receives a fundus retinal image as its input, and it can produce masks for both an optic disc and an optic cup. Despite that, there are several shortcomings and points that could be improved in the continuation of this work. Let us discuss the main points worth investigating in extensions of our approach. Firstly, our algorithms are written in Python, and there are several optimisations that could be done to improve their performance.

Secondly, our approach was trained and evaluated on a relatively small dataset with a limited representation of certain optic nerve head features. For instance, the dataset contains only a few retinal fundus images with significant parapapillary atrophy. Subsequently, the optic disc segmentation algorithm does not handle this retinal defect explicitly. In cases where the PPA region has an elliptical shape, the algorithm might converge towards it rather than to the true shape. It is worth investigating parapapillary atrophy in future work as it is the main reason for incorrect segmentation of the optic disc shape. The limitations of our training dataset affect the optic cup segmentation algorithm as well. Small optic cups are under-represented in the dataset, and large ones make up a significant portion. Even though we applied a custom data augmentation to reduce the cup size bias, the algorithm has lower performance on these images. Therefore, expanding the dataset and training the algorithm to distinguish a larger variety of optic cups is a great future improvement of this work.

Following the second point, extensions of our approach should be trained and evaluated on larger datasets. We implemented steps that should improve performance on images taken with different devices. However, the quality of these improvements needs to be evaluated. Large scale evaluation is another continuation point.

Finally, the vessel extraction algorithm can be improved as the current version often selects edges of the optic disc as vessels. It is not an essential improvement because vessel suppression replaces pixels with the values from their neighbourhood which does not deform the optic nerve head shape.

In conclusion, we implemented the full optic nerve head segmentation pipeline with multiple extension features. There are, however, several advanced points where it can be improved, and lastly, a large scale evaluation can be done in the future.

Bibliography

[1] Debjit Bhowmik, K.P. Sampath Kumar, Lokesh Deb, and A.S. Dutta Shravan Paswan. Glaucoma -a eye disorder its causes, risk factor, prevention and medication. *Pharma Innovation*, 1:66–81, 2012.

[2] GRF. Optic nerve cupping. https://www.glaucoma.org/glaucoma/glaucoma-facts-and-stats.php, 2017. Accessed: 2021-04-19.

[3] Harry Quigley and A.T. Broman. The number of people with glaucoma worldwide in 2010 and 2020. *The British journal of ophthalmology*, 90:262–7, 04 2006.

[4] Niharika Thakur and Mamta Juneja. Survey on segmentation and classification approaches of optic cup and optic disc for diagnosis of glaucoma. *Biomedical Signal Processing and Control*, 42:162–189, 2018.

[5] Ahmed Almazroa, Ritambhar Burman, Kaamran Raahemifar, and Vasudevan Lakshminarayanan. Optic disc and optic cup segmentation methodologies for glaucoma image detection: A survey. *Journal of Ophthalmology*, 2015, 11 2015.

[6] THE OPHTHALMIC PHOTOGRAPHERS' SOCIETY INC. Fundus photography overview. https://www.opsweb.org/page/fundusphotography, 2021. Accessed: 2021-04-18.

[7] John Hipwell. Ophthalmic photography—a textbook of retinal photography, angiography, and electronic imaging. *British Journal of Ophthalmology*, 81(12):1115–1115, 1997.

[8] Jayanthi Sivaswamy, Subbaiah Krishnadas, Arunava Chakravarty, Gopal Joshi, and Ujjwal. A comprehensive retinal image dataset for the assessment of glaucoma from the optic nerve head analysis. *JSM Biomed Imaging Data Pap*, 2, 01 2015.

[9] Harry Quigley MD. Scott Burk MD. PhD., John S. Cohen MD. Optic nerve cupping. https://www.glaucoma.org/glaucoma/optic-nerve-cupping.php, 2017. Accessed: 2021-04-19.

[10] Farnoosh Ghadiri, Robert Bergevin, and Masoud Shafiee. An adaptive thresholding approach for automatic optic disk segmentation, 2017.

[11] Jun Cheng, Jiang Liu, Damon Wing Kee Wong, Fengshou Yin, Carol Cheung, Mani Baskaran, Tin Aung, and Tien Yin Wong. Automatic optic disc segmentation with peripapillary atrophy elimination. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 6224–6227, 2011.

[12] Bulent Sankur and M. Sezgin. Image thresholding techniques: A survey over categories. *Pattern Recognition*, 34:1573–1583, 01 2001.

[13] Tariq M. Khan, Mehwish Mehmood, Syed S. Naqvi, and Muhammad Fasih Uddin Butt. A region growing and local adaptive thresholding-based optic disc detection. *PLOS ONE*, 15(1):1–16, 01 2020.

[14] Anindita Septiarini, Agus Harjoko, Reza Pulungan, and Retno Ekantini. Optic disc and cup segmentation by automatic thresholding with morphological operation for glaucoma evaluation. *Signal, Image and Video Processing*, 11, 07 2017.

[15] P. M. D. S. Pallawala, Wynne Hsu, Mong Li Lee, and Kah-Guan Au Eong. Automated optic disc localization and contour detection using ellipse fitting and wavelet transform. In Tomás Pajdla and Jiří Matas, editors, *Computer Vision - ECCV 2004*, pages 139–151, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[16] Ashish Issac, M. Parthasarthi, and Malay Kishore Dutta. An adaptive threshold based algorithm for optic disc and cup segmentation in fundus images. In *2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN)*, pages 143–147, 2015.

[17] Ashish Issac, M. Partha Sarathi, and Malay Kishore Dutta. An adaptive threshold based image processing technique for improved glaucoma detection and classification. *Computer Methods and Programs in Biomedicine*, 122(2):229–244, 2015.

[18] Tehmina Khalil, Muhammad Usman Akram, Samina Khalid, and Amina Jameel. Improved automated detection of glaucoma from fundus image using hybrid structural and textural features. *IET Image Processing*, 11(9):693–700, 2017.

[19] NM Noor, NEA Khalid, and NM Ariff. Optic cup and disc color channel multi-thresholding segmentation. In *2013 IEEE International Conference on Control System, Computing and Engineering*, pages 530–534, 2013.

[20] Mila Kankanala and Sanjeev Kubakaddi. Automatic segmentation of optic disc using modified multi-level thresholding. In *2014 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 000125–000130, 2014.

[21] Yanwu Xu, Lixin Duan, Stephen Lin, Xiangyu Chen, Damon Wing Kee Wong, Tien Yin Wong, and Jiang Liu. Optic cup segmentation for glaucoma detection using low-rank superpixel representation. In Polina Golland, Nobuhiko Hata, Christian Barillot, Joachim Hornegger, and Robert Howe, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014*, pages 788–795, Cham, 2014. Springer International Publishing.

[22] Dr Rao, R. Gayathri, and R. Sunitha. A novel approach for design and analysis of diabetic retinopathy glaucoma detection using cup to disk ration and ann. *Procedia Materials Science*, 10:446–454, 12 2015.

[23] Noor Elaiza Abdul Khalid, Noorhayati Mohamed Noor, and Norharyati Md. Ariff. Fuzzy c-means (fcm) for optic cup and disc segmentation with morphological operation. *Procedia Computer Science*, 42:255–262, 2014. Medical and Rehabilitation Robotics and Instrumentation (MRRI2013).

[24] N.M. Tan, J. Liu, D.W.K. Wong, F. Yin, J.H. Lim, and T.Y. Wong. Mixture model-based approach for optic cup segmentation. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pages 4817–4820, 2010.

[25] Jun Cheng, Jiang Liu, Yanwu Xu, Fengshou Yin, Damon Wong, Ngan-Meng Tan, Dacheng Tao, Ching-yu Cheng, Tin Aung, and T-Y Wong. Superpixel classification based optic disc and optic cup segmentation for glaucoma screening. *IEEE transactions on medical imaging*, 32, 02 2013.

[26] Ngan-Meng Tan, Yanwu Xu, Wooi Boon Goh, and Jiang Liu. Robust multiscale superpixel classification for optic cup localization. *Computerized Medical Imaging and Graphics*, 40:182–193, 2015.

[27] Julian Zilly, Joachim M. Buhmann, and Dwarikanath Mahapatra. Glaucoma detection using entropy sampling and ensemble learning for automatic optic cup and disc segmentation. *Computerized Medical Imaging and Graphics*, 55:28–41, 2017. Special Issue on Ophthalmic Medical Image Analysis.

[28] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurélien Lucchi, Pascal Fua, and Sabine Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34, 05 2012.

[29] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: Image processing in python. *PeerJ 2:e453*, 2014.

[30] Dongkuan Xu and Yingjie Tian. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2, 08 2015.

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[32] Josh Warner, Jason Sexauer, scikit fuzzy, twmeggs, alexsavio, Aishwarya Unnikrishnan, Guilherme Castelão, Felipe Arruda Pontes, Tobias Uelwer, pd2f, laurazh, Fernando Batista, alexbuy, Wouter Van den Broeck, William Song, The Gitter Badger, Roberto Abdelkader Martínez Pérez, James F. Power, Himanshu Mishra, Guillem Orellana Trullols, Axel Hörteborn, and 99991. Jdwarner/scikit-fuzzy: Scikit-fuzzy version 0.4.2, November 2019.

[33] J. Serra. *Image Analysis and Mathematical Morphology*. Number zv. 1 in Image Analysis and Mathematical Morphology. Academic Press, 1984.

[34] Robert M. Haralick, Stanley R. Sternberg, and Xinhua Zhuang. Image analysis using mathematical morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(4):532–550, 1987.

[35] Elena Šikudová, Zuzana Černeková, Wanda Benešová, Zuzana Haladová, and Júlia Kučerová. *Počítačové videnie Detekcia a rozpoznávanie objektov*. Wikina,Praha, Praha, 2013.

[36] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[37] Franco P. Preparata and Michael I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, Berlin, Heidelberg, 1985.

[38] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[39] G. Turin. An introduction to matched filters. *IRE Transactions on Information Theory*, 6(3):311–329, 1960.

[40] Subhasis Chaudhuri, Shankar Chatterjee, Norman Katz, Mark Nelson, and Michael Goldbaum. Detection of blood vessels in retinal images using two-dimensional matched filter. *Medical Imaging, IEEE Transactions on*, 8:263 – 269, 10 1989.

[41] Hugo Aguirre-Ramos, Juan Gabriel Avina-Cervantes, Ivan Cruz-Aceves, José Ruiz-Pinales, and Sergio Ledesma. Blood vessel segmentation in retinal fundus images using gabor filters, fractional derivatives, and expectation maximization. *Applied Mathematics and Computation*, 339:568–587, 2018.

[42] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, September 1995.

[43] H. Sharma and S. Kumar. A survey on decision tree algorithms of classification in data mining. 2016.

[44] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282 vol.1, 1995.

[45] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001.

[46] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754, 2016.

[47] Arunava Chakravarty and Jayanthi Sivaswamy. Coupled sparse dictionary for depth-based cup segmentation from single color fundus image. In Polina Golland, Nobuhiko Hata, Christian Barillot, Joachim Hornegger, and Robert Howe, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014*, pages 747–754, Cham, 2014. Springer International Publishing.

[48] J. Liu, D. W. K. Wong, J.H. Lim, X. Jia, F. Yin, H. Li, W. Xiong, and T. Y. Wong. Optic cup and disk extraction from retinal fundus images for determination of cup-to-disc ratio. In *2008 3rd IEEE Conference on Industrial Electronics and Applications*, pages 1828–1832, 2008.

[49] Gopal Datt Joshi, Jayanthi Sivaswamy, Kundan Karan, and S. R. Krishnadas. Optic disk and cup boundary detection using regional information. In *2010 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 948–951, 2010.

[50] Charles A. Poynton. Rehabilitation of gamma. In Bernice E. Rogowitz and Thrasyvoulos N. Pappas, editors, *Human Vision and Electronic Imaging III*, volume 3299, pages 232 – 249. International Society for Optics and Photonics, SPIE, 1998.

[51] M.M. Fraz, P. Remagnino, A. Hoppe, B. Uyyanonvara, A.R. Rudnicka, C.G. Owen, and S.A. Barman. Blood vessel segmentation methodologies in retinal images – a survey. *Computer Methods and Programs in Biomedicine*, 108(1):407–433, 2012.

[52] Zafer Yavuz and Cemal Köse. Blood vessel extraction in color retinal fundus images with enhancement filtering and unsupervised classification. *Journal of Healthcare Engineering*, 2017:1–12, 08 2017.

[53] Stephen M. Pizer, E. Philip Amburn, John D. Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart ter Haar Romeny, John B. Zimmerman, and Karel Zuiderveld. Adaptive histogram equalization and its variations. *Computer Vision, Graphics, and Image Processing*, 39(3):355–368, 1987.

[54] G. Hasting and Alan Rubin. Colour spaces - a review of historic and modern colour models*. *African Vision and Eye Health*, 71, 12 2012.

[55] Gopal Datt Joshi, Jayanthi Sivaswamy, Kundan Karan, and S. R. Krishnadas. Optic disk and cup boundary detection using regional information. In *2010 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 948–951, 2010.

[56] J. R. Harish Kumar, Aditya Kumar Pediredla, and Chandra Sekhar Seelamantula. Active discs for automated optic disc segmentation. In *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 225–229, 2015.

List of Figures

List of Tables

List of Abbreviations

ONH Optic nerve head

OD Optic disc

OC Optic cup

GRF Glaucoma research foundation

U.S. United States

CDR Cup-to-Disc Ratio

ISNT Inferior Superior Nasal Temporal

DDLS Disc Damage Likelihood Scale

GRI Glaucoma Risk Index

PPA Parapapillary Atrophy

SLIC Simple Linear Iterative Clustering

SVM Support Vector Machine

RF Random Forest

RoI Region of Interest

AHE Adaptive Histogram Equalisation

RMSE Root Mean Square Error

CSS Centre Surround Statistics

GCS Group Centre Surround

CH Convex Hull

RBF Radial Basis Function

GMM Gaussian Mixture Model

A. Attachments

A.1 Python scripts with the implementation

The algorithms proposed in this work as well as their modifications presented in the evaluation chapter 4 are submitted together with the text. The implementation is in the programming language Python and its image processing and machine learning libraries. In order to allow an easy examination of the source code and testing of its functionality, we present the code structure and a simple manual.

A.1.1 Python environment

There are many different versions of Python and its libraries. Unfortunately, not all of them are compatible. Firstly, let us present the environment in which we developed the algorithms. We used `Python` version 3.7.6 64bit with libraries `numpy` version 1.18.1, `scipy` version 1.4.1, `scikit-image` version 0.17.2, `scikit-learn` version 0.23.2 and OpenCV for python (`cv2`) version 4.2.0. Images were loaded into scripts using the library `Pillow` version 7.0.0 and all plots, images and graphs were drawn using the library `matplotlib` version 3.1.3. Further, Fuzzy c-means algorithm used in scripts for clustering algorithms is from the library `scikit-fuzzy` version 0.4.2. We recommend using this Python setup to ensure identical performance.

A.1.2 Code structure

We provide the scripts with the implementation of the algorithm in a simple directory hierarchy. The root directory is called `onh_main` and it contains everything needed for the execution of our algorithm except for data. Fundus retinal images from public datasets cannot be freely shared and therefore we cannot provide example images.

The main directory `main` contains scripts which handle the execution of different parts of the algorithm. Then there is a directory `onh_lib` which is structured as a python module and the scripts will only work if executed from a directory which has access to the library `onh_lib`. Following that, there is a directory `BACKUP` which contains copies of generated data and trained models in case the files in the root directory are overwritten or removed.

A.1.3 User documentation

The scripts available in the directory `onh_main` can be executed on their own without any parameters, provided that the default data files which they refer to exist. If not, the each of them has a set of parameters which allow us to select data and to parameters of the algorithm.

This is perfectly fine for using `roi.py` or `vessels.py`, but becomes more difficult with `od.py` and `oc.py`. Both segmentation algorithms require labels

as well as the source image. We can use the scripts situated in the root directory `onh_main` in the following way. We execute them using the command `Python "script.py"` where `"script.py"` can be one of `roi.py`, `vessels.py`, `od.py` or `oc.py`. All four scripts have an argument `--img` which denotes the source image to be processed. For example,

`--img="./data/drishtiGS_002.png"`.

The scripts for segmentation of optic disc and cup require labels in the form `--disc_label="..."` and `--cup_label="..."`. The script `od.py` requires the disc label and `oc.py` requires all three. These arguments accept string in the form of path from the current working directory, e.g. `onh_main`, to the respective files. Labels are needed because the scripts evaluate performance of the model.

Additionally, each of those scripts contains a long list of properties which can be changed to alter the execution of the algorithm. We recommend looking at the scripts themselves. It is most likely easier to edit the script itself rather than write long lines of commands in the console. The last notable property belongs to `oc.py` and it is `--model_file`. It accepts a path to a model file. The default value is the gradient boosted classifier present in the root directory, but there are also files with trained SVM, linear-SVM and random forest.

In conclusion, running the algorithms is very easy, but modifying the parameters becomes very verbose. Thus the best way of examining the algorithm is editing the parameters in the main scripts which is faster and more efficient.

A.2 Feature engineering

Optic cup segmentation techniques presented in this thesis work with superpixels and features describing them. Supervised machine learning techniques are able to infer information from basic features extracted directly from colour channels. Now, let us consider unsupervised algorithms, specifically clustering. The basic features do not form natural clusters and the distribution of values belonging to the optic cup and neuroretinal rim is not similar. Therefore we have to apply feature transformation and enhancement to make the classes more distinct.

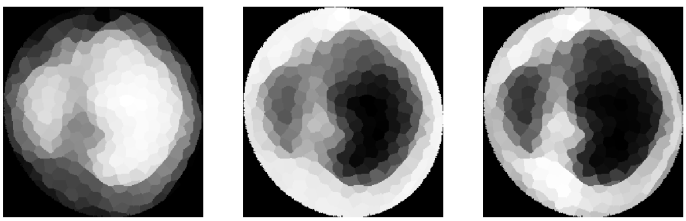

Figure A.1: Green superpixel features (left), saturation superpixel features (middle), a* superpixel features (right).

Our complex features are based on the three colour channels which show the highest contrast for optic cup area. These are the green, saturation and a* channels from RGB, HSV and L*a*b* colour spaces respectively. They are displayed in the figure A.1. Let us denote the green channel by G, saturation by S and a* by A.

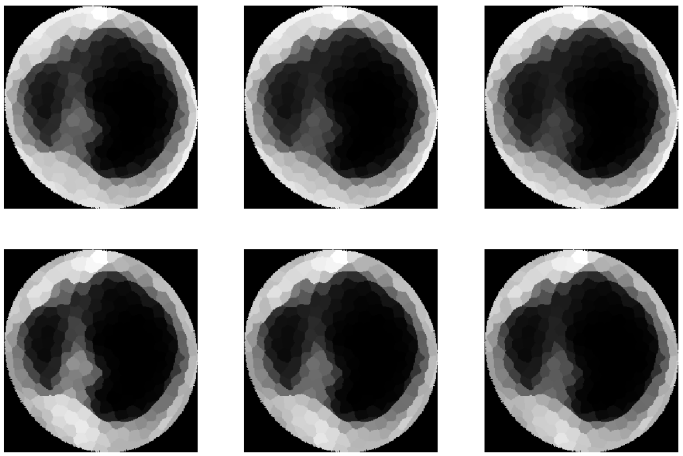

Figure A.2: The top row shows improvement of saturation features S and the bottom row of a* features A. The first column shows the result of multiplication of the source channel by $1-G$, the second column shows these features multiplied by our vessel neighbourhood modifier and the third column shows final features multiplied by distance factor.

We calculate the final features as a combination of saturation and a* with the green channel, distance from the approximate optic centre and vessel neighbourhood modifier. Let us denote the final features C_S and C_A, the vessel modifier κ and the distance modifier d. Then, the formulas for C_S and C_A are given in A.1. The individual steps of the computation are visualised in the figure A.2. The first row of images shows $S \cdot (1-G)$, $S \cdot (1-G) \cdot \kappa$ and $S \cdot (1-G) \cdot \kappa \cdot \frac{2-\delta}{2}$. The second row is analogical.

$$C_S = S \cdot (1-G) \cdot \kappa \cdot \frac{2-\delta}{2} \tag{A.1}$$

$$C_A = A \cdot (1-G) \cdot \kappa \cdot \frac{2-\delta}{2} \tag{A.2}$$

We define the modifier δ as an exponentiated negative distance from the optic centre $\delta = \exp(-2 \cdot \frac{d}{h})$ where d is the pixel distance and h is half length of the region of interest diagonal. Furthermore, we define the vessel neighbourhood modifier κ as $\kappa = \exp(-v \cdot \delta_{normalised})$ where v denotes the fraction of pixels surrounding the given superpixel which belong to blood vessels and $\delta_{normalised}$ are δ values normalised across all superpixels in an image.

In conclusion, we defined a set of features which help normalise the sizes of clusters present in the superpixel data created from retinal images. The features consider distance from the optic centre as well as the intensity information from multiple channels.