TECHNISCHE UNIVERSITÄT WIEN

DIPLOMARBEIT

Automatische Quantifizierung der Ausrichtung torischer Intraokularlinsen basierend auf Visual Computing und Deep Learning

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Biomedical Engineering

eingereicht von

Ing. Christoph Baumann, BSc.
Matrikelnummer 11907101

ausgeführt am Institut für Stochastik und Wirtschaftsmathematik
an der Fakultät für Mathematik und Geoinformation
der Technischen Universität Wien

Betreuung
Betreuer: Privatdoz. Dipl-Ing. Dr. Zsolt Saffer

Wien, 19. Jänner 2025

Christoph Baumann | Zsolt Saffer

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

TECHNISCHE UNIVERSITÄT WIEN

DIPLOMA THESIS

Automatic quantification of toric intraocular lens orientation based on visual computing and deep learning

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Biomedical Engineering

by

Ing. Christoph Baumann, BSc.
Registration Number 11907101

conducted at the Institute of Statistics and Mathematical Methods in Economics
to the Faculty of Mathematics and Geoinformation
at the TU Wien

Advisor
Advisor: Privatdoz. Dipl-Ing. Dr. Zsolt Saffer

Vienna, January 19, 2025

Christoph Baumann | Zsolt Saffer

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Ing. Christoph Baumann, BSc.

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 19. Jänner 2025

Christoph Baumann

Danksagung

Ich danke meinem Betreuer Privatdoz. Dipl.-Ing. Dr. Zsolt Saffer für seine Unterstützung beim Verfassen dieser Arbeit – sein Feedback war stets hilfreich und inspirierte mich, ein besserer wissenschaftlicher Autor zu werden. Er nahm mich bereitwillig als seinen Diplomanden auf, obwohl ihm die vorherigen Umstände nicht vertraut waren. Er widmete mir Stunden seiner Zeit, selbst wenn sein Zeitplan voll war – nicht jeder würde bis spät an der Universität bleiben, nur um eine Arbeit zu korrigieren.

Ich möchte meine Dankbarkeit Dr. Marcus Lisy aussprechen, der mir meinem Betreuer empfahl und mir wertvolle medizinische Einblicke gab, die ich ohne ihn nicht gehabt hätte.

Ebenso möchte ich meiner Familie danken, die mich mit Anregungen unterstützte und die gesamte Arbeit Korrektur gelesen hat. Die Diskussionen mit meinem Vater Wolfgang gaben mir eine Richtung, wenn kein klarer Weg erkennbar war. Die Gespräche mit meiner Mutter Barbara halfen mir, jegliche Zweifel an mir selbst und meinen Fähigkeiten zu überwinden. Das Zusammenleben mit meinen Brüdern Johannes und Thomas machte das Schreiben einfacher – sie leisteten mir Gesellschaft und hatten stets ein offenes Ohr für mich. Ich bin dankbar, sie zu haben; sie könnten nicht besser sein.

Zusammen mit meinem engen Freund Peter Blohm begann ich, diese Arbeit zu entwickeln. Sein anfängliches Interesse am Thema half, einen Einstieg zu finden, und seine – beinahe genialen – Erkenntnisse machten die Lösungsfindung erheblich einfacher. Die Zeit, die er in die Vorbereitung dieser Arbeit investierte, werde ich nicht vergessen. Einen besseren Freund kann man sich nicht wünschen.

Ich möchte meiner Verlobten Amila danken, die mich in jeder erdenklichen Weise unterstützt hat. Ihre konstruktiven Vorschläge und Ideen waren stets hilfreich und sie war mein Kompass auf der Reise des Schreibens dieser Arbeit. Wenn sie in meiner Nähe war, fiel mir das Schreiben und Programmieren viel einfacher und jegliche Unsicherheit oder Zögern verschwanden. Ihre unerschütterliche Unterstützung hielt das gesamte Jahr über an, das ich für diese Arbeit benötigte, und ließ keinen Moment nach. Selbst wenn ich bis weit nach Mitternacht schrieb, war sie für mich da und zeigte mir den Weg nach vorn. Ohne sie wäre diese Arbeit nicht fertiggestellt worden.

Acknowledgements

I thank my advisor Privatdoz. Dipl-Ing. Dr. Zsolt Saffer for aiding me to write this thesis - the given feedback was always useful and it helped me to become a better scientific writer. He happily accepted me as his advisee, despite being unfamiliar with the previous circumstances. He gave me hours of his time even when his schedule was busy - not everyone would stay at university late just to correct a thesis.

I want to express my gratitude to Dr. Marcus Lisy for connecting me with my advisor and giving me practical medical insight, which I would not have without him.

I also want to thank my family for supporting me with valuable insights and proof-reading the whole thesis. The discussions with my father Wolfgang gave me a sense of direction, when no clear path was evident. Talking to my mother Barbara helped me to remove any doubts I had in myself and in my abilities. Living with my brothers Johannes and Thomas made writing easier - they kept me company and always had an open ear for me. I am thankful for having them, they could not be any better.

Together with my close friend Peter Blohm we started developing this thesis. His interest in the topic in the beginning aided in finding a starting point and his - close to genius - insights made finding a solution much easier. The time he put into preparing this thesis is not forgotten and one cannot ask for a better friend.

I want to thank my fiancee Amila for supporting me in every way imaginable. Her constructive insights and fresh ideas were always helpful and she was my compass through the journey of writing this thesis. When she was around, writing and programming was much easier and any hesitation or uncertainty disappeared. Her unwavering support was consistent for the whole year it took me to write this thesis and did not falter for even a second. Even if writing took me way after midnight, she was there for me and pointed out the way ahead. Without her this thesis would not have been completed.

Kurzfassung

Bei der Durchführung einer Kataraktoperation wird die natürliche Linse durch eine künstliche ersetzt, wodurch die Möglichkeit besteht, eine Hornhautverkrümmung mithilfe torischer Intraokularlinsen zu korrigieren. Die Wirksamkeit dieser Linsen hängt von ihrer präzisen Ausrichtung ab. Allerdings kann sich die torische Intraokularlinse in den Wochen/Monaten nach der Operation drehen. Diese postoperative Rotation verringert die Korrekturwirkung der torischen Intraokularlinse erheblich.

Derzeitige Methoden zur Beurteilung der Rotation sind manuell, zeitaufwendig und oft ungenau. Diese Arbeit entwickelt und evaluiert zwei automatisierte Methoden zur Beurteilung der Rotation — eine auf Computer Vision basierende sowie eine Deep-Learning-basierte Methode — mit dem Ziel, die Rotation der Linse präzise zu bestimmen. Die Methoden wurden anhand eines Datensatzes von Bildern von 130 Patienten getestet und ihre Genauigkeit mit manuellen Messungen von Ärzten verglichen. Die Deep-Learning-basierte Methode erreichte eine Vorhersagegenauigkeit von weniger als 6 Grad für mehr als 82% der Bilder. Diese Methode basiert auf einem standardisierten ResNet18-Modell, das zusätzlich explizit extrahierte Merkmale nutzt.

Dieser Ansatz kann in einem Klassifikator-Setting mit einem entsprechend gewählten Schwellenwert auf den postoperativen Intervallen von 1 Stunde bis zu 6 Monaten nach der Operation etwa 50% der angenommenen pathologischen Fälle mit einer Linsenrotation über 2 Grad erkennen. Dieses vorläufige Ergebnis muss mit einem größeren Patientendatensatz weiter validiert und spezifiziert werden. Dies zeigt jedoch bereits die Machbarkeit einer vollständigen Automatisierung des Beurteilungsprozesses der Linsenrotation mit angemessener Genauigkeit.

Darüber hinaus legen die Ergebnisse nahe, dass die Deep-Learning-Methode weiter verbessert werden kann und ein hohes Potenzial besitzt, die Vorhersagegenauigkeit und damit die praktische Anwendbarkeit weiter zu steigern.

Abstract

When cataract surgery is performed, the natural lens is replaced with an artificial one, which offers the opportunity to correct astigmatism using toric intraocular lenses. The effectiveness of these lenses depends on their precise orientation. However the toric intraocular lens can be rotated in the following weeks/months after the operation, due to currently not clarified reasons. This postoperative rotation significantly reduces the corrective power of the toric intraocular lens.

State of the art methods for assessing rotation are manual, time-consuming and often inaccurate. This thesis develops and evaluates two automated candidate assessing methods — using a computer vision-based and a deep learning-based approach — in order to determine lens rotation accurately. The methods were tested on a dataset of images of 130 patients and their accuracy was compared to manual measurements performed by physicians. The deep learning-based approach achieved a prediction accuracy of less than 6 degrees for more than 82% of the images. This is based on a standardized ResNet18 model, which also uses explicitly extracted features. This approach can detect approximately 50% of the assumed pathological cases, having a lens rotation above 2 degrees, when used in a classificator setting with a properly set threshold on the postoperative intervals between 1 hour up to 6 months after the operation on the considered image set. This preliminary result must be justified and specified on an image set with a higher number of patients. This demonstrates the feasibility of fully automating the lens rotation assessment process with appropriate accuracy.

In addition, the results suggest that the deep learning method can be further improved and has high potential to improve the prediction accuracy and therefore the practical usability of it.

Contents

CHAPTER 1

Introduction

Astigmatism is a phenomenon in the human eye that significantly impacts vision experience by inducing blurred vision. It is normally caused by an asymmetric distribution of the refractive power of the cornea of the eye. This leads to uneven focus of the incoming parallel light rays, causing them to not converge at a single point on the retina [SSF13]. When treating cataracts, astigmatism can be reduced by implanting special artificial lenses, called toric intraocular lenses (toric IOLs). These lenses remove the refractional error introduced by the cornea, again focusing the light on a single point. Due to the asymmetric nature of the toric IOLs, movements of the lens can worsen its effectiveness, again inducing severe blurred vision. Especially rotation of the IOL is critical; one degree of rotation can already cause 3.3% reduction in the correctional effect, completely losing the benefit at 30 degrees of rotation [SMS94].

Currently, there are no fully automatic solutions for measuring this rotation, based on optical microscope images. The task can be conceptualized as an issue of image classification or regression. For this class of problem, specialized artificial neural nets (ANNs) - namely convolutional neural networks (CNN), a type of deep learning (DL) - have been shown to be effective. The goal of this thesis is to develop a fully automated algorithm based on DL that can detect and measure the rotation of a toric IOL between several given images taken with a biomicroscope.

1.1 Medical context

This section introduces the main medical problems relevant for this thesis and several medical imaging techniques. It should give an overview of commonly used tools and specify the medical side of the problem.

1.1.1 Medical imaging

For a better understanding of how 2D- and 3D-images in computer vision and machine learning are generated, common medical imaging techniques used in ophthalmology (a medical discipline dealing with diseases of the eye) are introduced. In the following, we explain their use in different medical fields and give a brief description on them:

- **Digital photographs** are used to store images mainly for documentation purposes. Taking photographs and comparing them at different points in time has been widely adopted in fields like dermatology, where it is crucial to observe the development of possible melanomas over time [QTJ+15]. These kinds of image are also available when examining patients with a biomicroscope in an eye examination.
- **Ultrasound** utilizes acoustic waves with a frequency of more than 20kHz to visualize parts of the body, that cannot be properly examined by other means. This includes visualizing soft and deep-seated tissues, but also flow rate measurements and 3D-images can be generated [Lei07]. In ophthalmology, ultrasound of the eye is performed in situations where the optical path to the anatomy of interest is obfuscated, for example due to hemorrhaging or a clouded lens [Sou09].
- **Computed tomograpahy (CT)** is a method to generate a 3D representation of a scanned body part. It is based on X-rays taken at different angles, which are combined to form a 3D model. It is commonly used in preventative care, for example detecting tumors in the lung, brain, or eyes [Gar02].
- **Magnetic resonance imaging (MRI)** is based on strong magnetic fields and radio pulses, to generate 3D images of examined body parts. It is slower than CT imaging, but has certain advantages, such as a lower radiation dose and a higher image quality. In ophthalmology it is used to diagnose orbital lesions or retinoblastomas [ISD12].
- **Optical coherence tomography (OCT)** can be used to obtain micrometer-resolution 3D images of tissues, by utilizing interferometry of light [HSL+91]. In ophthalmology, it is used to visualize the front third of the eye or the retina at the back of the eye. Therefore, it can be used in the diagnosis of a multitude of diseases, like glaucomas [GT13] or macular degeneration [KPL+12].

Due to the optical and anatomical properties of the eye, digital images of slit lamp microscopy and optical coherence tomography are the main ways of gathering information when implanting intraocular lenses [LWK+17]. Techniques such as magnetic resonance imaging and ultrasound can also provide insight in certain circumstances but are used mainly to identify different diseases of the optical apparatus, for example, the detection and description of retinoblastomas [dJdGN+14].

Slit lamp microscopy is one of the simplest and most powerful ways to diagnose medical conditions in ophthalmology. For one, it is used to identify a variety of diseases; on

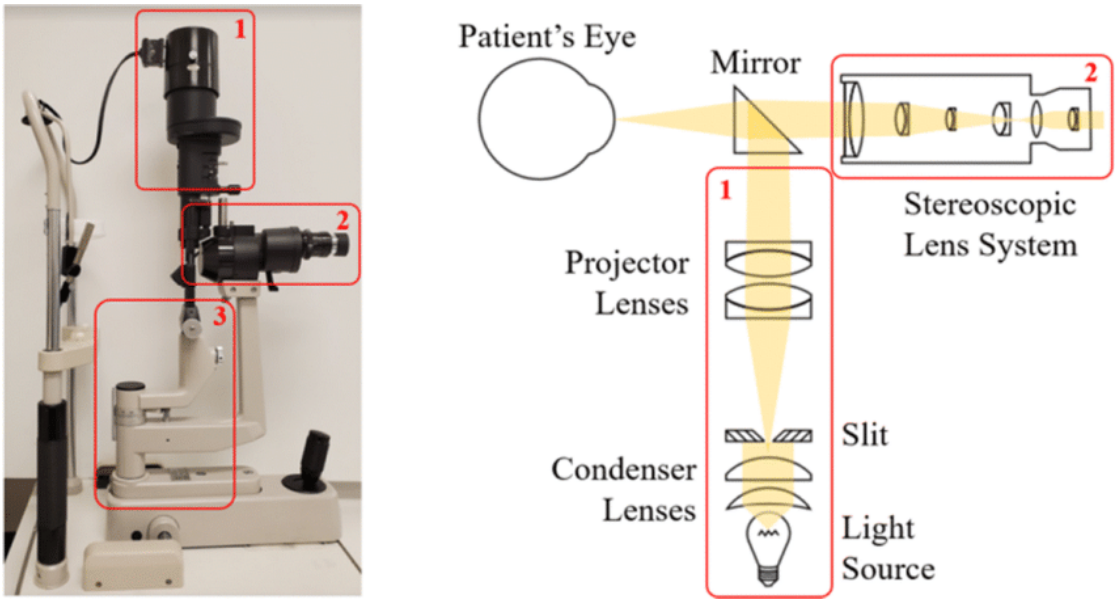

Figure 1.1: Biomicroscope with optical path. 1: Light source. 2: Examiner side ocular. 3: Adjustable base. Source: [TPT+20]

the other hand, it produces digital images that can be compared over long periods of time, enabling advanced diagnostic methods. Understanding the device used to take the pictures provides deeper insight into the applicability of slit-lamp microscopy in ophthalmology. The instrument enabling this method is called a biomicroscope, pictured in Figure 1.1. It is made of a bright light source attached to a microscope. The light source can be focused or de-focused by the operator (to a thin line, hence the name slit lamp) depending on the performed procedure. It is located above or below the microscope and the light path is guided directly to the patient's eye with a semitransparent mirror. The light is then reflected back towards the mirror, where it is passed through the microscope and to the examiner. There exists a multitude of optical adapters that allow one to take pictures with an attached camera sensor simultaneously while performing routine inspections with a conventional ocular. This enables a basic assessment of the extent of the disease while documenting it at the same time. Depending on the configuration of the slit lamp device, different parts of the eye can be examined. For example, the retina and optic nerve in the back of the eye can be evaluated with fundoscopy. In a different configuration, the front part of the eye can be observed, which is especially useful when diagnosing a cataract. Usually, these kinds of image are retroilluminated, meaning that the illumination of the lens is provided by the reflection of emitted light on the fundus (back) of the eye. Additional ambient light can be introduced to also slightly illuminate the iris and the sclera. An example of such an image can be seen in Figure 1.2.

1.1.2 The problem of toric lens rotation

To better understand the problems that arise with toric IOLs, the underlying medical condition and treatment methods are explained.

Cataracts are diseases in the field of optometry that cloud the lens and make it less transparent, causing a significant reduction in the quality of vision and therefore the quality of life of the patient. The most widely available remedy is cataract operation, a surgical method that is performed mainly on a day-patient basis (119,987 surgeries in Austria in 2021 [Eur23]). In this procedure, the clouded lens is removed and an artificial intraocular lens (IOL) is implanted. An implanted lens can be seen in Figure 1.2. If a *toric* IOL is implanted, not only is the lens clouding removed but the preoperative corneal astigmatism of the patient can be corrected, resulting in improved postoperative vision. Due to the patient-specific magnitudes of astigmatism and possible other medical factors, a individualized evaluation of the severity of astigmatism is needed, which is performed using optical and tomography-based methods. They produce a set of measurements from which the appropriate power of the toric IOL can be determined [FR20].

Due to the asymmetric nature of astigmatism, the orientation of the toric IOL significantly affects the residual astigmatism. The lens movement can be caused by intraoperative misalignment or postoperative rotation due to trauma or physical strain. Especially rotation is critical, because, as previously mentioned, each degree of rotation can cause a 3.3% reduction in the correctional effect of the toric IOL [SMS94]. Thus, physicians need an accurate measurement method to detect these rotations and react to them in an appropriate way.

1.2 Problem statement

Commonly used methods for measuring the rotation of toric IOLs are still semi-automated [SSS+21] or the evaluation is performed completely manually with the help of a slit lamp. The proposed semi-automated methods require high-quality images and a significant amount of manual labor to find the change in rotation. In addition, it takes a lot of time and skill to operate the slit lamp microscope itself to capture sharp and well-illuminated images that include all necessary details. Furthermore, commonly used manual methods using the slit lamp only have an accuracy of 5 degrees, which significantly reduces the quality of the measurements, although there are newer methods that reduce the measurement error to around 0.5 degrees [MSK11].

Currently, there is no fully automated method available that determines the change in rotation of the lens over time after toric IOL implantation. There is a need for developing such an algorithm in order to demonstrate the rotational stability of newly developed toric IOLs making patient care more efficient and enabling further research based on large volumes of processed images. Such research can target to investigate the dependency of change in rotation on type of toric IOL and other relevant factors. In addition, manual

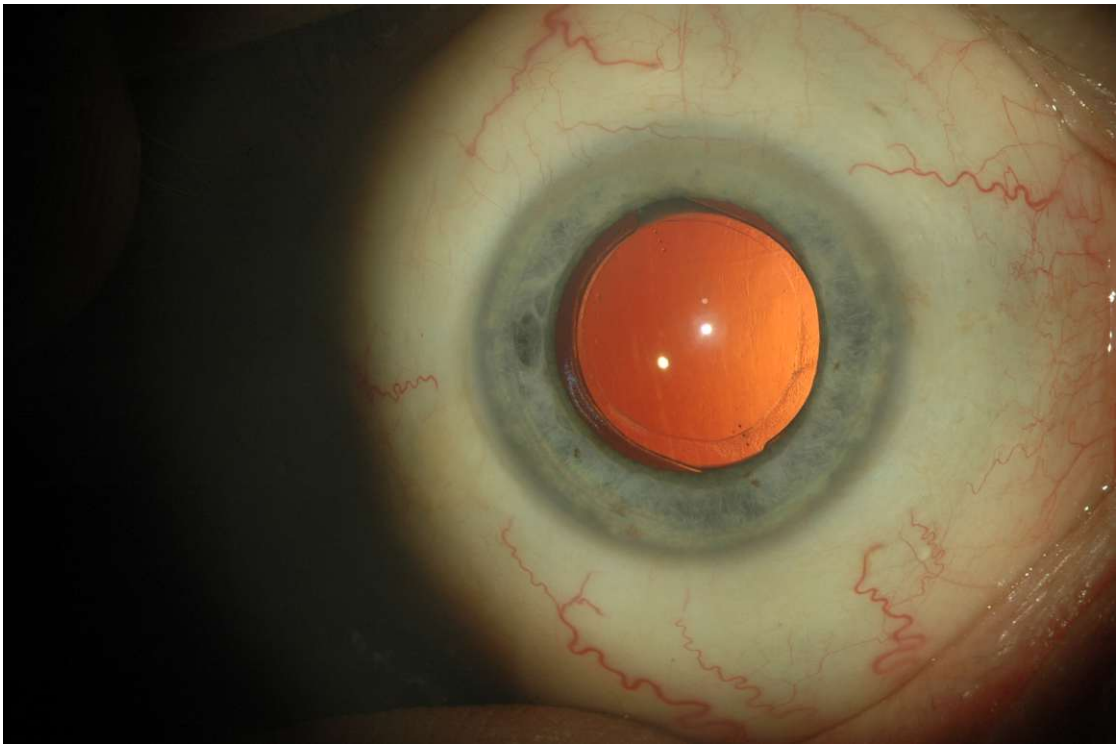

Figure 1.2: Slit-lamp photograph of eye with implanted IOL, 4 months after implantation

measurements can introduce human errors and inconsistencies, depending on the level of expertise of the grader and image quality.

1.3 The goal of the thesis

The aim of the work is to allow full automatic processing of retroilluminated slit lamp photographs to detect and measure the grade of postoperative toric IOL rotation. Novel algorithms are aimed to be developed that automatically generate fast and reliable results with high accuracy, comparable to the best available traditional methods, with the additional benefit of providing very high repeatability between tests. In addition, automatic methods eliminate the need for manual evaluation performed by medical personnel and save valuable time for physicians. While for medical diagnosis the rotation angle is the main interest, additional information such as the position of the center of the lens can be beneficial for further medical and research purposes.

As these measurements are repeated to assess the change in lens orientation, the technical aspects of the measurement method are not the only relevant factor. The test-retest reliability of these methods also depends on the consistency of the labeling of the patient photos performed by medical personnel and the quality of patient photos [SSS+21]. Automated methods can help reduce these variances and are also able to achieve more

robust measurements.

An overarching goal is to establish preventive measures by identifying candidate causes of rotation of toric intraocular lenses. This can be done by checking the correlation between the hypothesized candidate causes and the size of the rotation of the toric intraocular lenses. The findings can then be used to take preventive measures. However, the target of this thesis is not to identify these candidate causes themselves. Instead, it aims to provide the necessary tools to automatically assess the rotation of the toric lens.

1.4 Research questions

The research questions are as follows:

- RQ1: Can the predictive accuracy of a fully automated computer vision based method for assessment of the toric IOL rotation without deep learning achieve a level of quality equivalent to those of a manual data analysis method as performed by medical professionals?
- RQ2: Which deep learning (DL) architecture performs best for automatic assessment of toric IOL rotation?
- RQ3: Can a deep learning dominated algorithm outperform a traditional computer vision algorithm in terms of predictive accuracy for the assessment of the toric IOL rotation?

All in all, the thesis results in a set of images automatically labeled with the lens rotation in degrees by a computer vision and DL algorithm that can determine lens positioning of retro-illuminated images. Investigating methods for quality awareness is a key-interest, in order to enable doctors to rely on predictions and be aware of when manual supervision of a certain measurement is needed.

1.5 Approach

Two approaches are explored as possible processing algorithms: A computer vision (CV) based processing pipeline and a DL based algorithm. The dataset used for DL based algorithms can be augmented, which enables the improvement of the predictive accuracy of the DL based methods. The validation part of the thesis gives an indication on how well the algorithms perform and how different methods compare to each other with respect to run-time and accuracy. The workflow of this thesis can be described by the following steps:

1. **Literature review**: Background information is gathered on possible approaches and existing methods, to serve as a theoretical basis.

2. **Purely computer vision based assesment**: A computer vision algorithm is developed that automatically detects the difference of the angles and the rotational position of a given image pair.

3. **Label generation**: A computer vision algorithm is developed that automatically detects lens angle and position for any given image in the data set. This algorithm is based on the previously conceived computer vision algorithm.

4. **Creation of the labeled images**: The labeling algorithm is used on a large dataset to generate a set of labeled images needed for training and testing of DL approaches. The algorithm enables automatic processing, so there is no need for labeling by medical personnel, which saves valuable time of doctors and reduces variances. These images can additionally be augmented to simulate nonoptimal image conditions, such as obfuscation of parts of the image.

5. **DL dominated algorithm**: A combination of visual computing pre- and post-processing algorithms with a DL algorithm for calculating the lens position on an image is devised. The DL algorithm is based on a convolutional neural net (CNN) architecture, designed and implemented by training and testing it on labeled images. The DL algorithm is only used to detect the lens angle of a single image automatically; it has to be post-processed with a CV algorithm to detect the lens angle difference of any image pair.

6. **Comparison of the methods**: The labeling algorithm and the DL-dominated method are compared to the semi-automated method [SSS+21]. The benchmark test, including all developed algorithms, is evaluated by the help of applying selected regression metrics.

1.6 Structure of the work

First, the relevant medical background is introduced, where the focus is on physical phenomena, relevant anatomy, structure of artificial lenses, and existing methods used in cataract surgery and ophthalmology. In Chapter 3, the visual computing algorithms are introduced. The chapter is structured according to the order in which they arise in the experiments. The next chapter gives a background on CNNs. Chapter 5 deals with the experiment design and discusses all the necessary factors needed to establish the four algorithm flows. In addition, it provides the specification of the concrete experiments based on the four algorithms introduced in the previous chapter. Finally, the results of the experiments are presented and discussed, and the research questions are answered. The thesis ends with conclusions and possible future steps, with additional content in the appendix.

CHAPTER 2

Medical background

To clarify the challenges faced in this work, we give an introduction to the medical background. In particular, studies addressing medical image techniques are explored and surgical procedures are explained, including relevant anatomical structures and phenomena. Moreover, a review of existing research works on the assessment of IOL rotations is given.

2.1 Visual computing in medicine

First, existing conventional algorithms in the field of visual computing are briefly discussed. This should provide an overview of methods that are similar to the algorithms that are used in this thesis. They contain valuable input to understanding the problem of IOL rotation detection and establishing possible solutions for it.

2.1.1 Visual computing algorithms in various medical areas

There are several applications of computer vision algorithms that do not use machine learning in medicine. For example, a common problem that has been solved is the automated counting of various types of blood cells. It uses methods such as corner detection and feature extraction to gather information on microscopic images of a blood sample. Most importantly, it can count the number of cells present in a given sample in very little time, which makes time-consuming counting by hand obsolete [IPS20] [CSS+20].

The presence of visual computing methods does not always have to be immediately noticeable; for example, de-noising and contrast enhancement algorithms are used to refine the image quality of X-rays and improve general interpretability. There are novel approaches that utilize mathematical and CV methods to achieve this goal while reducing the potentially harmful radiation dose in X-Ray imaging [LLC+20][AASIK23].

It could even be considered that the implementation of the Radon transform that generates a 3D model from individual images taken by a CT machine is a simple visual computing algorithm. The basis for this algorithm, a mathematical function, was developed before the emergence of CT machines as a mathematical concept without practical application. Only decades later, it was discovered that it can be used to "overlay" X-Ray images taken from various angles to generate a 3D image [Kal00][Rad17].

These algorithms that rely only on computer vision provide a first feeling of what can be achieved with medical imaging. They illustrate common methodologies, such as edge/circle detection and contrast enhancement, which serve as a basis for avenues that can be utilized when looking for a solution to this thesis' problem.

2.1.2 Visual computing algorithms in ophthalmology

There have been advances in ophthalmology using visual computing algorithms that do not use machine learning. Automatic evaluation of the quality of retinal images is an application that can be useful in determining whether the image quality of an image is high enough to be used in diagnostics [LGB+01]. Another use of visual computing algorithms is to improve low-quality retinal images that have been poorly illuminated [ZWB23]. This shows that CV techniques have been successfully used to solve problems in the field of ophthalmology.

2.2 Deep learning (DL) for image processing in medicine

In recent years, many machine learning and deep learning based algorithms have been used for image processing in medicine. The core concepts of such methods are introduced in later chapters; this section provides an overview of common DL based applications and clarify their relevance.

2.2.1 DL algorithms for image processing in various medical areas

Segmentation and image classification have been used to detect liver lesions [LBK+18] or to detect tuberculosis on chest radiographs [RKK+20]. These methods rely on deep learning to identify regions of unusual or known-to-be-malignant tissue. They are often used in combination with manual examinations, to combine computer knowledge with practical expertise of medical specialists.

There are also use-cases for image processing over multiple appointments, for example monitoring malignant melanoma on skin photographs [GB19]. The most important aspect of a proper diagnosis is tracking the changes in nevi over time. Such trackings were already used before the development of advanced deep learning algorithms. Hence, such tracking-based detection enables the training of DL based algorithms resulting in high accuracy. Such algorithms are already used in real-world settings.

Some authors combine already existing computer vision algorithms with deep learning approaches, such as more sophisticated counting of different types of blood cells. Such combined algorithms work with different types of cells that are automatically marked and classified, further improving the speed of cell count assessments and increasing the amount of information gathered [CTH22].

All of these algorithms show that DL methods can outperform existing CV methods, and their combinations can lead to improved solutions. In addition, DL-based algorithms can be built on already existing data, indicating that physicians have little to no additional manual work to enable their development.

2.2.2 DL algorithms for image processing in ophthamology

Various methods from the field of artificial intelligence and machine learning are already being investigated for various uses in ophthalmology. Many algorithms use fundus images to determine diseases. An example is the automatic detection of glaucoma disease in images of the eye fundus, where a specialized deep learning algorithm classifies those images to contain signs of glaucoma or not [Abb17]. Another case in which retinal fundus images are used is the detection and classification of diabetic retinopathy. In this study, the algorithm employed classifies if it suspects retinopathy or not, similarly to the previously mentioned article [GL17].

Diseases such as macular degeneration can also be detected based on fundus images, but researchers are also exploring methods that use OCT. This implies that certain diseases can be detected with multiple different methods based on varying image types. This field has been explored in multiple studies and current algorithms based on convolutinal neural networks (CNNs) yield results with high diagnostic accuracy [LBL17][LSW+23].

However, there are not only deep learning methods for diagnosing patients, but also algorithms that help to determine the right treatment options for these diseases. For example, in cataract surgery the optical properties of the implanted artificial lens have to be calculated so that they properly restore the patient's vision. This procedure is known as the IOL power calculation, where a multitude of methods have been explored and compared. Machine learning and deep learning based methods are one of the most reliable methods in certain cases, e.g., researching formulas for IOL power calculations [SCG24] [Grz21].

All of these papers lead to the conclusion that DL-based approaches are beneficial for the field of ophthalmology, being employed a variety of use cases. To remain at the forefront of academic research, it is crucial to consider DL-based methodologies to enhance both the quality of diagnosis and the quality of treatment.

2.3 DL based image classification for cataract

There have been a multitude of papers exploring deep learning methods to detect and classify cataracts based on fundus images. Almost all researched works use a form of

convolutional neural networks to process scaled photographs. These networks are often custom-designed and have different performance characteristics and goals. They are a good example of how deep learning can be used to reduce the workload on physicians and perform studies on large sets of data that might not be properly processable manually [ZNX+19] [DZQY17] [KCA+22]. This topic is very close to the task discussed in this thesis; thus, their content is of high relevance and serves as a source of possible methodologies that can be applied in the algorithms developed.

2.4 Surgical background for implanting IOLs

Not only are existing methods concerning DL and CV important, but also the commonly used surgical routines can provide insight into the topic. In this section, the medical details of how cataracts are treated and how an IOL is implanted in order to correct astigmatism are discussed. In addition, an overview of the anatomy of the eye and the relevant type of IOL used in this thesis is specified.

2.4.1 Surgical procedure to treat cataracts

The underlying medical procedure used to treat cataracts is surgery, in which the clouded lens is removed and instead an artificial lens is implanted. For the commonly used sutureless procedure, one or more small incisions are made on the outer side of the pupil, where the surgical instruments are inserted. The lens capsule is then opened with a scalpel, resulting in a circular opening, also visible in the slit lamp photograph (see Figure 2.1). The original lens is then emulsified with the help of ultrasonic waves and removed with a suction tool. Afterwards, the folded artificial lens is inserted into the opened lens capsule, where it unfolds itself. If the IOL is implanted to correct astigmatism, then in the last step, the artificial intraocular lens is moved, so it is centered and the rotation corresponds to the preoperatively determined rotation value [Mal08]. This procedure can cause irritation and bleeding of the conjunctiva, possibly obfuscating the underlying sclera vessels. This can lead to problems when assessing the lens rotation, because the view of important landmarks - namely corkscrew vessels - can be blocked by irritated vessels of the conjunctiva.

The lens can move after implantation, for example, due to trauma or infection, reducing visual clarity for the patient. If the patient and the physician deem rotation being too large, additional surgery is needed to correct the position of the IOL [SKBK22]. Rotation is particularly problematic, because small changes can already cause deterioration of optical performance and worsen patient satisfaction, due to asymmetric optical properties of toric IOLs [SMS94]. To avoid unnecessary surgeries, methods are needed to accurately assess rotational changes over time.

When the eye of a patient is examined in follow-up appointments, the pupil is usually enlarged with drugs such as tropicamide or phenylephrine [SSS+21] to improve lens visibility. Without any of the drugs, the pupil has a smaller diameter, and the markings

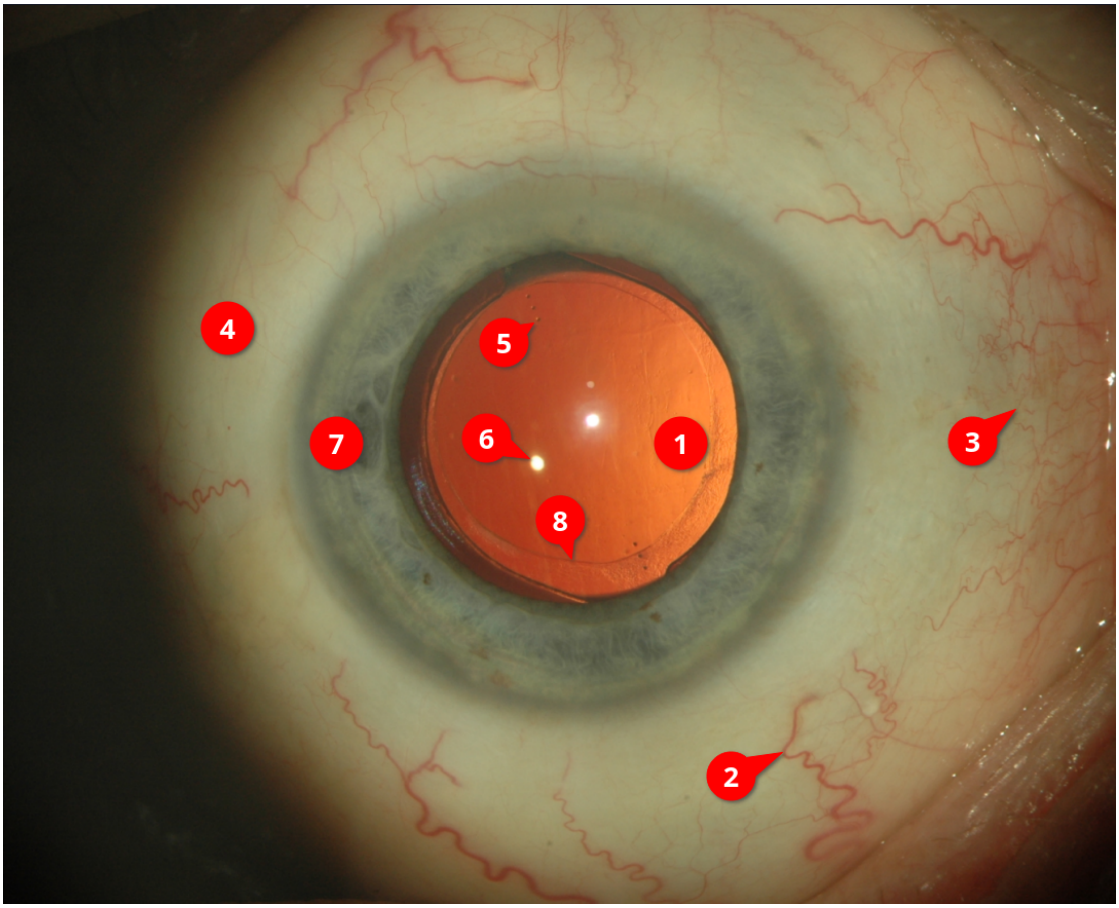

Figure 2.1: Anatomy of the eye, containing following parts: (1): The considered type of artificial toric intra-ocular lens (2): Immovable vessel of the sclera (3): Movable vessel of the conjunctiva (4): Sclera (5): Markings on the surface of the artifical IOL (6): Purkinje reflex (7): Iris (8) Border of the opening of the lens capsule

on the IOL may not be fully visible. In addition, the change in diameter can affect algorithms that take into account the relation of pupil size to eyeball.

2.4.2 Relevant anatomy of the eye after implanting IOL

There are several anatomical structures that are important for measuring IOL rotation: The pupil containing the toric IOL, surrounded by the iris and the sclera („white of the eye“) containing immovable blood vessels, and lastly the transparent conjunctiva overlaid on the sclera containing small movable blood vessels. A detailed view can be seen in Figure 2.1.

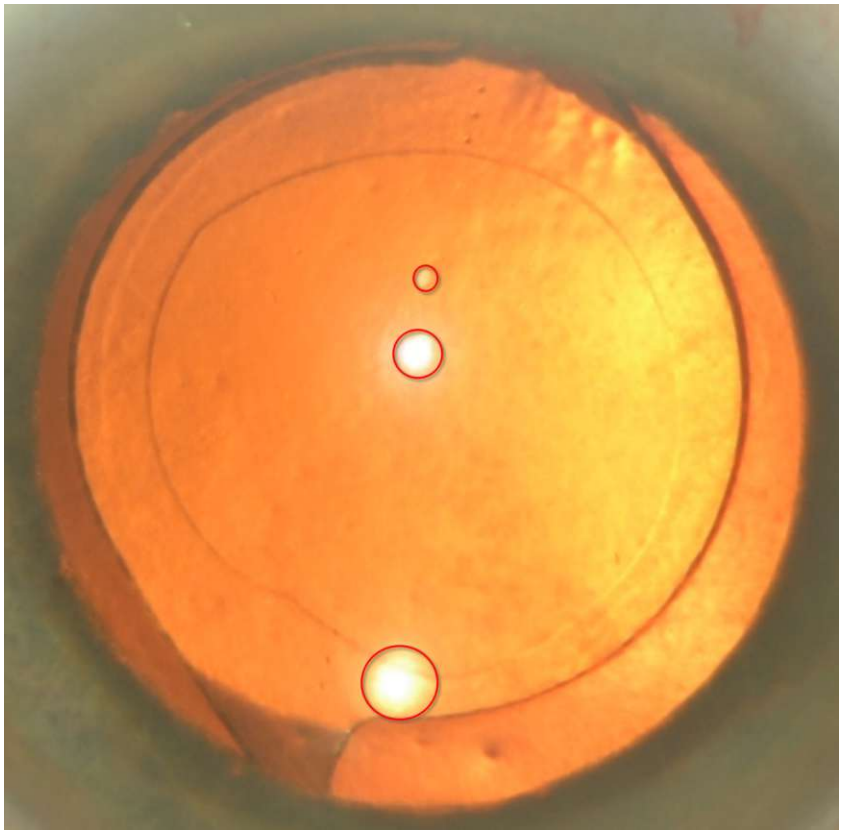

Figure 2.2: Purkinje reflections (marked red)

2.4.3 Purkinje reflections

A physical phenomenon used in various algorithms is the reflection of the microscope-light-source in different parts of the eye: There can be up to four very bright spots in a slit lamp photograph of the eye, called Purkinje reflections (see Figure 2.2). Each reflection is created at an optical boundary and has a different intensity and location, depending on the origin of the reflection (see Figure 2.3). This includes reflections at the front and back of the lens and the cornea. At least one point is always located and visible within the circumference of the pupil, if viewed from the front [GK16]. This fact can be utilized to improve algorithm performance when searching for the location and size of the pupil.

2.4.4 Structure of IOLs

In this study, the single-piece Clareon Toric IOL (Alcon, Fort Worth, TX) with three dot markings on opposite sides of the lens is used exclusively (see Figure 2.4). In general, this type of artificial toric intraocular lens has two main components: The optical lens itself and the so-called haptics, which are used to secure the lens in place. Usually there are

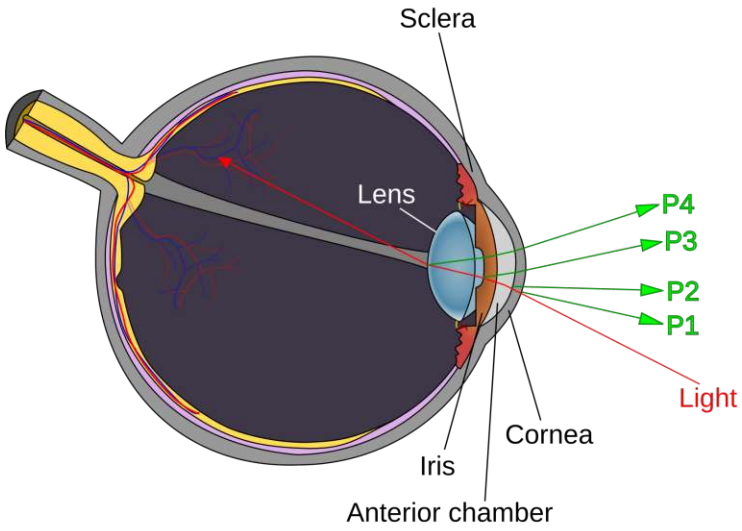

Figure 2.3: Diagram of purkinje reflections. Source:[Com15]

also markings located on the lens, which indicate the current position and rotation. The haptics and markings have a fixed position relative to each other. The haptics and the lens itself can have different shapes and materials, depending on the manufacturer and the desired characteristics [ZLC+16]. The haptics themselves are either only partially or not visible in slitlamp photographs of the eye.

2.5 Manual method for determining lens rotation

A semi-automated method, that determines angular rotation over time, is based on a custom manual method that uses the "Rotix" program [SSS+21]. The process of determining the difference of lens rotation is described as follows:

> „To evaluate the true rotation of the IOL, all pictures recorded at the follow-ups are imported into the custom-made software “Rotix.” The Rotix program [...] automatically calculates the axis between 2 lines drawn by the observer. First, the observer defines 2 nonmovable critical landmarks on the sclera. [...] Then, a connecting line is drawn between 2 landmarks. In a second step, the observer defines the IOL axis. This second line can either be drawn between the toric axis markings or between the axilla or shoulders of the haptic base of the IOL. The axilla is defined as the transition zone between the optic and the haptic. In IOLs where the transition zone is rather flat and no clear distinction between optic/haptic transition can be made, a stencil of the IOL should be overlaid on the IOL via Photoshop. At each follow-up, the same critical landmarks identified at the end of surgery are compared to the IOL axis. The

Figure 2.4: Clareon Toric IOL (Alcon, Fort Worth, TX) Source:[Alc24]

> software automatically calculates the axis difference between the 2 connecting lines [...]. The software then automatically calculates the difference between the axes of follow-up 1 and follow-up 2, 3, et cetera.“[SSS+21]

"Follow-ups" in the context of the quotation refer to sequential photographic images of the same eye that have been taken at appointments over the course of 6 months. This method has been used to generate rotational differences as labels for the dataset, which serves as a comparison basis for this study (see Section 5.5 for more details).

There are also other methods present, which utilize optical coherence tomography, relying on more specialized equipment, such as the IOLMaster 700 (Carl Zeiss, Germany). These methods give accurate results; however, the equipment needed is expensive and is not always available or used in follow-ups to assess rotational stability [OOS+21].

2.6 Review of research on assessment of IOLs rotation

Current research, which compares the rotation of the toric IOL for different lenses, uses mainly methods based on slitlamp photographs [KFSN14], (enhanced) manual detection [MSK11], or semi-automated methods [SSS+21]. Typically, IOL orientation is evaluated at several timepoints: For example, postoperatively or on day 1, month 1, month 3 and month 6 [KFSN14]. This results in an evaluation of at least 4 images per patient, sometimes with several hundred patients needed for a comparable study. Due to the fact that they are done manually/semi-automated [Wen21][WB10][SSM+20], a huge workload is created for the medical personnel.

There exists research that measures toric IOL rotation and other metrics with shallow feed-forward neural nets, but these algorithms need additional tomographic data to work efficiently [LSC+23]. Computer vision/machine learning approaches that only use image data from optical photographs have not yet been implemented. There are propositions to use convolutional neural networks, but there are no specific studies that implement this kind of software [LMY24].

CHAPTER 3

Visual computing background

In this thesis, a computer vision algorithm is used that combines out-of-the-box components with a custom algorithm that is interfaced with pre- and post-processing steps to detect the angular rotation of an artificial toric IOL. This section aims to explain the pre-existing components, provide an overview of the methods used, and describe how physiological phenomena are utilized.

3.1 Preprocessing

Preprocessing can be divided into two parts: First, general preparations are needed to assign identifiers to individual images and videos and second, bring all data into a common image format. The second step includes extracting images from intra-operational video footage and splitting the names of all files into its components. Afterwards, an image-specific preprocessing step is performed to create a copy of the images in a different color space and resize them to a common resolution. In the following, we will describe the second preprocessing step.

3.1.1 Extracting images from video

In the given data set, the footage of the freshly operated eye (footage tagged with the timepoint "OP") is given as a video between 10 seconds and 2 minutes and 52 seconds long. The videos come with a frame rate of 25 image frames per second (FPS). With the proposed algorithms, only pictures can be processed; therefore, several frames per video must be extracted and saved as image files.

The main challenge in this process is the presence of artifacts created by quick movements of the eye (see Figure 3.1) in the video. Presumably, they are present due to the so-called interlacing[Sch30], or are introduced as deinterlacing artifacts[DHB98]. They are an issue for the algorithm used, because important landmarks for detecting the lens angle can be

distorted and made unrecognizable. If there is little to no difference between frames, no interlacing artifacts should be visible. To extract images that have as few artifacts as possible, a heuristic method is employed to detect the presence of significant movements. This is done by converting each frame to the HSV-space (see Section 3.1.3) and building the difference in brightness (encoded in the V-channel) for any given frame in the video (c) to its previous frame (p) and the next frame (n):

$$\begin{aligned} \Delta p_{i,j} &= |p_{i,j} - c_{i,j}| \\ \Delta n_{i,j} &= |n_{i,j} - c_{i,j}| \end{aligned} \tag{3.1}$$

i and j represent each individual pixel in the image. If little movement is present, the difference in brightness is close to zero, and if large movements are present, the difference in brightness is large. The sum of the total changes indicates the amount of movement compared to the previous and next frame, which is described by the variable s:

$$s = \sum_{i,j} \Delta p_{i,j} + \Delta n_{i,j} \tag{3.2}$$

The images with the smallest sum of changes in brightness s are chosen as potential candidates with fewer artifacts. For these candidates, it is checked whether the pupil mask can be calculated (see Chapter 3.2) to ensure that the pupil is not obstructed by surgical tools, which block the view of the lens in some frames. Two different lens-finding algorithms were utilized to calculate the possible position and size of the pupil. If both algorithms agree that the center is approximately at the same position (max. 100 pixels apart), the frame is considered to show an unobstructed view of the lens and thus it is usable.

The five images that have the lowest score s and pass this test are chosen as samples for further processing. If any of those five images has a score that is ten times higher than the with the lowest s, they are discarded and not used. This ensures that the images are of similar quality. A limitation of the algorithm can be other video artifacts like blocking, which introduces changes in brightness that could be interpreted as movement. Thus, images could have been removed that had little to no interlacing artifacts included.

3.1.2 Splitting the file names

The next preparation step is the extraction of the patient ID, information on the location of the eye (OS - oculus sinister - left eye, OD - oculus dexter - right eye), sequence number, and point in time of picture taken. This information is stored inside the filename, encoded in the format "[ID]_[OS|OD]_[Time]_[Sequence number].tif". For example the file "001_OD_1w_1.tif" is interpreted as patient with ID 001, right eye, photo taken at 1 week after operation, first picture taken at that point in time. The patient IDs are integer values that range from 001 to 130. The point in time where the follow-up

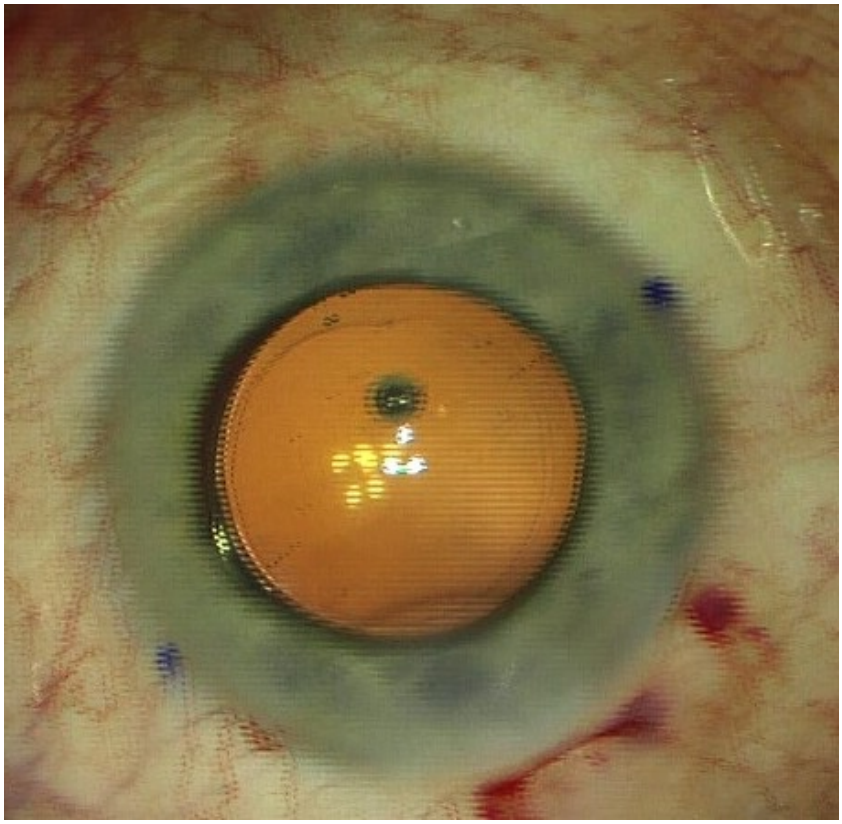

Figure 3.1: Cropped video frame with interlacing artifacts

appointment was performed, also called a "timepoint" in this thesis, can be one of the following:

- OP - intra-operational picture
- 1h - taken 1 hour after operation
- 1w - taken 1 week after operation
- 1m - taken 1 month after operation
- 4m - taken 4 months after operation
- 6m - taken 6 months after operation

This information is extracted by removing the extension of the file name and splitting the rest of the string at the "_" character.

3.1.3 Preprocessing steps for individual images

First, a conversion to the HSV color space is performed for each image. Except for possible rounding errors, the conversion does not reduce or alter the given information; it simply changes the color space in which the image is represented. Instead of having a channel for each primary color (**R**ed, **G**reen, **B**lue), the information is stored as **H**ue, **S**aturation and **V**alue. This is useful in later steps because image features such as the pupil and the blood vessels are more distinguishable from the background in the HSV color space. The conversion for each pixel happens under the following formula:

$$V \leftarrow max(R, G, B) \tag{3.3}$$

$$S \leftarrow \begin{cases} \frac{V-\min(R,G,B)}{V} & \text{if } V \neq 0, \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

$$H \leftarrow \begin{cases} \frac{60(G-B)}{V-\min(R,G,B)} & \text{if } V = R \\ \frac{120+60(B-R)}{V-\min(R,G,B)} & \text{if } V = G \\ \frac{240+60(R-G)}{V-\min(R,G,B)} & \text{if } V = B \\ 0 & \text{if } R = G = B \end{cases} \tag{3.5}$$

All images are rescaled to match a common resolution. Especially images extracted from videos need this resizing, because the videos are provided in a resolution of 1280x720px, whereas almost all other images have a resolution of 1720x1143px. The rescaling is done with the opencv function "resize", using bicubic interpolation [Key81]. The ratio of scaling sr is calculated by dividing the smaller components of the resolutions:

$$sr = 1143/720 = 1.5875 \tag{3.6}$$

This ratio is then used to resize the image, resulting in a resolution of 2032x1143px. To match the common resolution of 1720x1143px, both from left and right sides 156px are cut off, 312px in total.

3.2 Masking pupil and vessel area

For some steps of the algorithm, only certain regions of the photo are important; namely, the pupil of the eye and the white of the eye, which therefore must be extracted. They compose a disjunct partition of the eye, thus this extraction means a separation. The separation is performed by means of two masks (one for the pupil, one for the sclera), which have the same dimension as the base image. They indicate when a pixel is part of the pupil or sclera (=white of the eye) with a boolean value at each position. Calculating the mask for the pupil accurately is important due to multiple reasons. The mask of the sclera is based directly on the pupil mask, and the center point and radius of the pupil are used in several calculations. Two algorithms for finding the pupil are defined: A color-based segmentation algorithm and another one that applies a specialized type of the

Hough transform, called Circle-Hough-Transform (CHT) [Hou60][Bal81]. Both of these methods yield three outputs: The center of the pupil (given by x and y coordinates), the radius of the pupil, and a mask that covers the pupil, represented as a logical array. These methods are chosen, because of the very specific anatomical structures of the eye: For one, the healthy iris and the pupil have an almost circular shape [Kol95], which makes Circle-Hough transform based detection and segmentation possible. This transform is designed to find circles in an image. In addition, the retina is reflecting parts of the incoming light back, giving it a yellow to orange hue. This phenomenon partially depends on the light source used, but is mainly dependent on the blood vessels in front of the retina and the specific pigments (including melanin) present in the retinal tissue [YZL21]. This enables detection and segmentation with the help of color/hue thresholding [BGK15]. The purkinje reflexes are used in both algorithms to verify that the pupil has been found.

3.2.1 Pupil masking based on Circle-Hough-Transform

The Circle-Hough-Transform is based on knowledge of the approximate diameter of the pupil, measured in pixels. Therefore, the method with the given parameters is only applicable to this specific dataset and has to be adjusted when processing images taken with a different resolution or zoom level. The Python library "opencv" has an out-of-the-box method called "HoughCircles", which implements the circle detection. To ensure good performance, several parameters need to be set appropriately:

- **minRadius and maxRadius**: This represents the interval of possible pupil radii. It is set to $I = [180, 316]$.
- **param1 and param2**: These parameters represent the edge detection thresholds, as described in the Canny algorithm [Can86]. The lower value always represents the lower bound for the Canny edge detection, and the higher value the higher bound. The values have been set at $p1 = 50$ and $p2 = 87$
- **dp**: Represents the inverse ratio of the resolution used internally to the resolution of the image. It is set to $dp = 1.3$, meaning that the resolution of the original image is reduced by this factor. This increases the robustness of the algorithm by merging circular structures that are located close to each other.
- **minDist**: The minimum distance between circles. Set to the same value as minRadius, so that only a minimum amount of overlap is allowed.

The function is applied to the V channel of the HSV image, resulting in a set of possible circles, represented as pairs of centers (coordinates with components c_x and c_y) and radius r. The first entry is kept as the result, because the "HoughCircles" function sorts the result by confidence in circularity. If no circle was found, the image is considered unprocessable. The mask of the circle found, M^h, a logical array with the same dimensions as the original image, is then generated with the following formula:

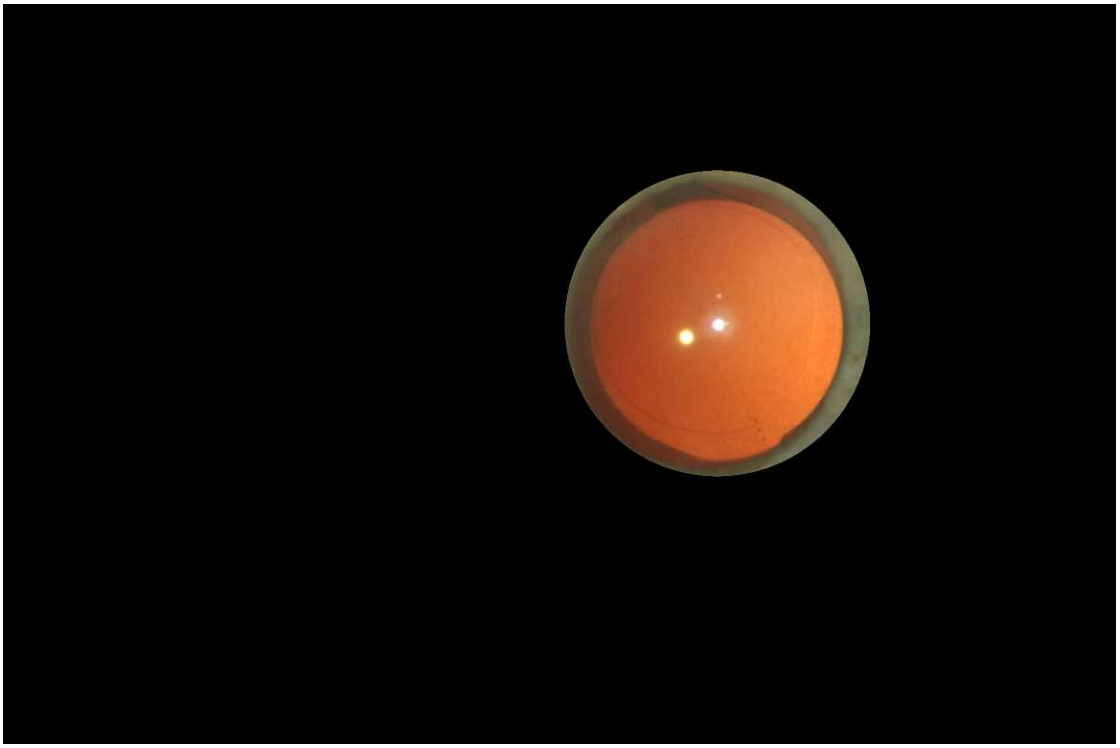

Figure 3.2: Pupil mask overlaid on original image generated by Hough Transform

$$M^h_{x,y} := \begin{cases} \text{True}, & \text{if}(x - c_x)^2 + (y - c_y)^2 \leq r^2 \\ \text{False}, & \text{otherwise} \end{cases} \tag{3.7}$$

An example of a resulting mask can be seen in the Figure 3.2.

3.2.2 Pupil masking based on color

A method for detecting the pupil independently of the size of the picture is based on the orange-to-yellow hue of the pupil. The parameters and its description for this function can be found in Table 3.1. As first step, a Boolean mask of the image in the HSV space (given as HSV) is created with the following formula:

$$M_{ij} := \begin{cases} \text{True}, & \text{if } hsv\_lower < HSV_{ij} < hsv\_upper \\ \text{False}, & \text{otherwise} \end{cases} \tag{3.8}$$

Possible artifacts of the generated image mask are removed with the help of morphological operations: First, the mask is eroded with a circular kernel sized 5×5 and then dilated with a circular kernel sized 30×30. Then, the contours of this mask are detected, and approximately circular shapes are extracted based on these contours. This extraction is

Parameter name	Value (H,S,V)	Description
hsv_lower	(5,100,100)	Lower bound for color detection
hsv_upper	(25,255,255)	Upper bound for color detection
hsv_avg_lower	(4,66,66)	Value for accepting the proposed circle - lower bound of average pixel value
hsv_avg_upper	(25,255,255)	Value to accept the proposed circle - upper bound of the average value of pixels in the solution
hsv_max_std	(20,78,100)	Value for accepting the proposed circle - highest standard deviation for pixels values in the calculated circle
bright_spot_min	254	Lowest value for detecting Purkinje Reflections

Table 3.1: Parameters for detecting pupil size by color

performed with the "findContours" method of the "opencv" library, which returns a list of possible contours. They are then processed with the "minEnclosingCircles" function to find a circle around the detected contours. Afterwards, the circular shapes are filtered by the following criteria:

- The pupil is always fully on the image; Circles, that are not fully inside the image are discarded
- The pupil has to have a minimum size; all circles that have a diameter that is smaller then 10% of the largest image dimension are discarded
- The pupil always has a bright orange hue; If there are too many other colors inside the circle, it is discarded. This is checked by calculating the average HSV channel values and the standard deviation. If they are not in between the parameters hsv_avg_lower, hsv_avg_upper and hsv_max_std they are discarded.
- The pupil contains purkinje reflections (bright white spots); There must be a few pixels that have a saturation and brightness of more than bright_spot_min.

If multiple masks pass those criteria, the best mask is chosen by taking into account that the purkinje reflections must be approximately near the center of the circle. The mask is chosen, where the center has the least distance to the average position of pixels that have a saturation and brightness of more than bright_spot_min. The resulting center and radius are used to calculate the mask M^c as shown in equation 3.7.

3.2.3 Combination of CHT and color based masking

The algorithms mentioned above can be combined to improve the results. One method would be to calculate the mask with both algorithms and compare the output of each. Only masks with high overlap are used (intersectional method). Another way would be

to calculate the mask with one algorithm and only if it produces no result, the second algorithm is used (union method). The intersectional method would produce masks that have a very high probability of marking only the pupil. The union method can be useful if one algorithm yields high-quality results but cannot be applied properly to each image. The second algorithm would act as a back-up in this case. In this thesis, whenever mask generation was applied, the union method was used. The CHT algorithm was called first, because of the suspected higher quality of masks.

3.2.4 Masking the sclera

The sclera mask, which marks the white of the eye around the pupil, is calculated by utilizing the anatomy of the eye: The approximate diameter of the eyeball, and thus the sclera, is 25mm [BGV14]. The diameter of the dilated pupil under the influence of commonly used drugs, such as tropicamide, is approximately 8mm [LGVSC17]. This fact is used to calculate an approximate mask and the corresponding center c_x and c_y and radius r for the sclera, based on a previously computed pupil mask M^c or M^h. The radius of the sclera mask r_s is calculated by multiplying the radius of the pupil mask r by an extension factor $f_e = 2.1$. This factor is lower than the ratio of the sclera to the pupil ($= 25/8 = 3.125$), because parts of the sclera are obfuscated by anatomical structures, such as the eyelids. The mask M^s is then calculated as in equation 3.7, using the sclera radius r_s and the center of the pupil mask c_x and c_y. Additionally, the pupil mask itself is subtracted from M_s, resulting in an annulus-shaped mask, as seen in Figure 3.3.

3.3 Finding potential points and triplets

Figure 3.4 shows an explanation of finding points and triplets in flowchart form. The goal of this step is to find groups of points that have the same coordinates as the dot-markers of the image (see (5) in image 2.1). Specifically, three points on one side of the IOL are searched for; only in the next steps these so-called triplets are combined to find the orientation of the lens.

3.3.1 Finding potential marker points

Feature extraction is based on identifying key points on an image, starting with detecting edges on the image using image filters such as Canny or Sobel. The generated edges can be linked, resulting in corners that are the basis of the features. The quality of the corners can then be measured and its surroundings can be described with a response function [HS+88]. Various algorithms, such as SIFT[Low04b], SURF[BETV08], ORB[RRKB11] and KAZE[ABD12] address this process differently, employing varying techniques to detect, measure, and describe features in images. In this thesis, advanced algorithms are utilized that support local image descriptors: Specifically, the KAZE algorithm and its accelerated version AKAZE [AS11] are used. They are chosen because of their superior speed and focus on detecting natural boundaries while simultaneously reducing noise, enabling the detection of small features in the image. In addition, parameters such as

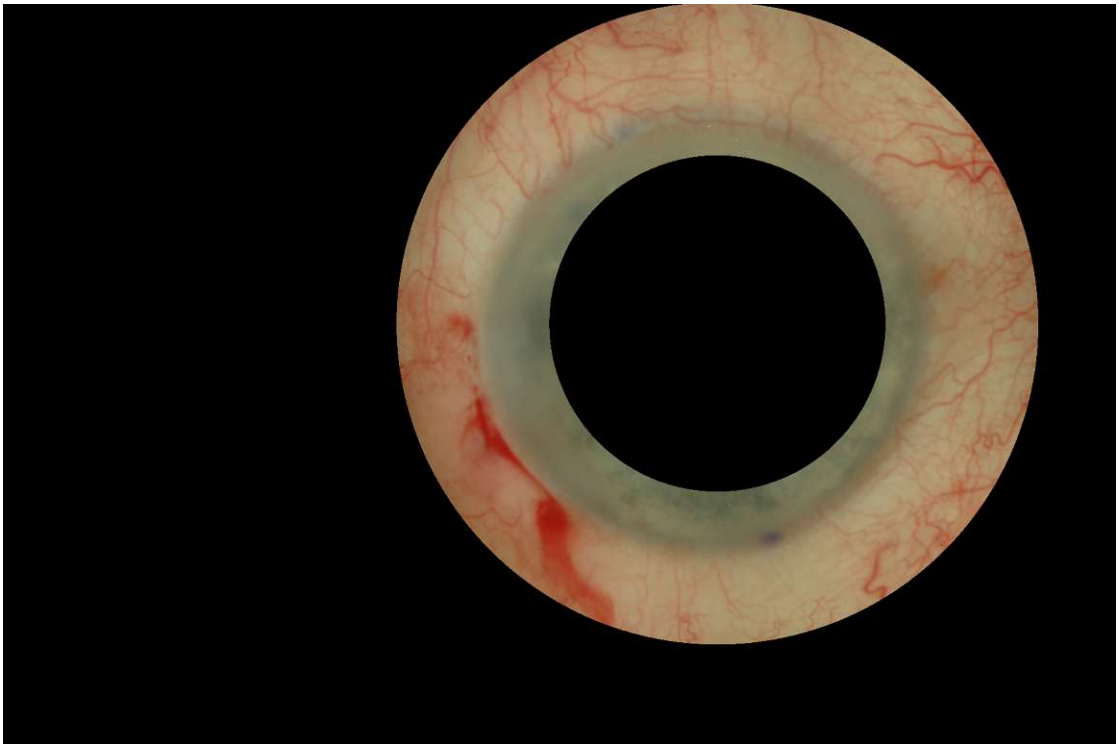

Figure 3.3: Sclera mask overlaid on original image

the number of octaves (refers to the logarithmic scale spaces created by progressively downsampling the image to detect features across multiple resolutions) and diffusivity are adaptable, enabling the fine-tuning of the results.

This algorithm has a single HSV image as input. For the image, the possible coordinates of the markers are calculated using the AKAZE algorithm. The number of octaves and layers was set to one, resulting in simple diffusion on the full-scale image. Other algorithms that detect points on an image could be used as well, and might even lead to better results. However, the AKAZE algorithm with the given settings performs well enough and has usable results. The detection itself is performed on the "V" channel of the HSV image, because the contrast present between the background and marker spots is high on this channel. The maximum amount of points returned by AKAZE (max_points) must be set, as well as the sensitivity to the features (feature_threshold). An example image together with the resulting marker points can be seen in Figure 3.5. It is visible that the AKAZE algorithm produces many points that are not the markers of the lens. All of these points at other locations can be considered noise and have to be removed by applying pertinent information.

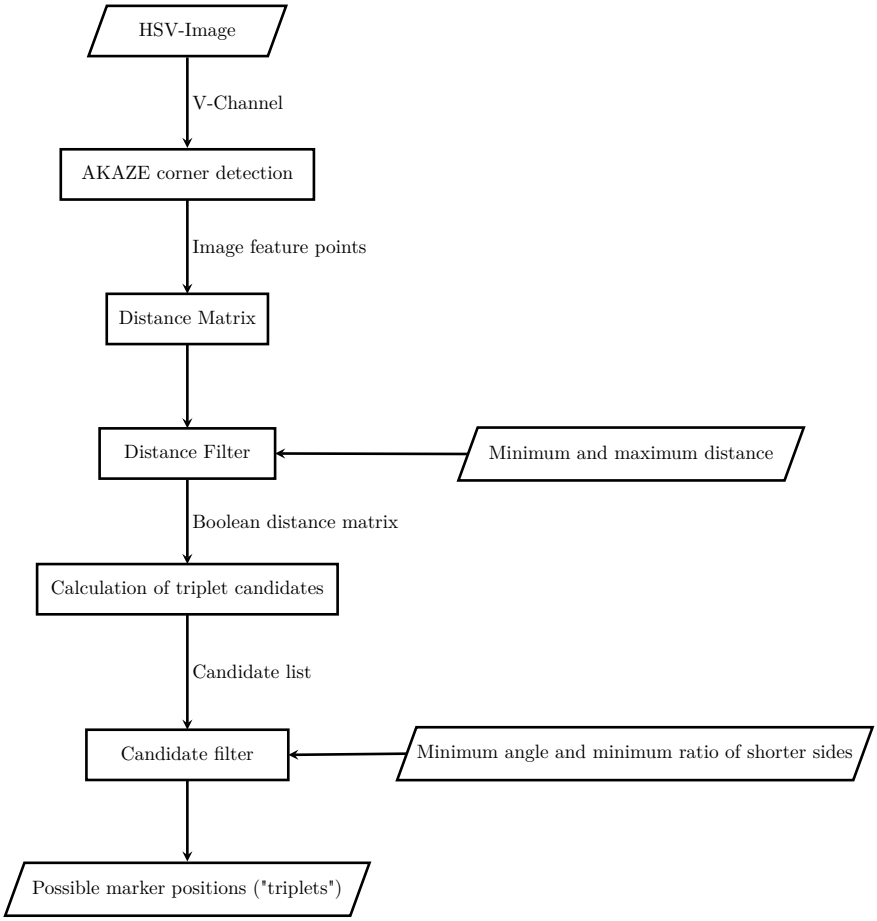

Figure 3.4: Part 1 of Process flow for single image processing: Triplet generation

3.3.2 Identifying triplet candidates

For removing points considered noise, a distance matrix is built up, whose elements represent the distance between all candidate points. The distance between each point pairs is calculated by applying the algorithm distance s.

$$dist_{i,j} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \tag{3.9}$$

where x_i, y_i and x_i and y_i are the x and y coordinates of the points i and j, respectively.

This matrix is then transformed into a Boolean matrix, D whose i and j components represent the connectivity between the respective points. An element has the value "True" if the distance of the corresponding two points is in between a maximum (max) and a minimum (min) distance. An interval of distances is used due to the imprecision of the AKAZE algorithm.

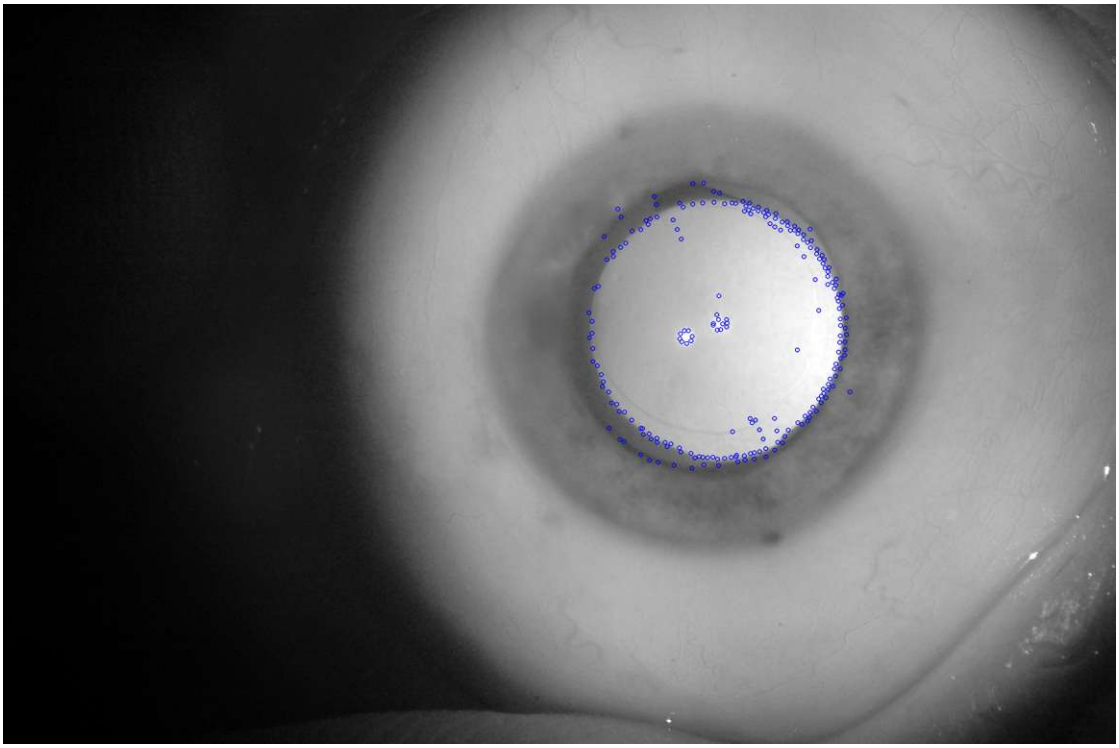

Figure 3.5: V-Channel of HSV-image with possible marker positions (blue)

$$D_{i,j} := \begin{cases} \text{True}, & \text{if } min < dist_{ij} < max \\ \text{False}, & \text{otherwise} \end{cases} \tag{3.10}$$

The next step is to calculate groups of three points having the correct appearance to be a potential part of the markings on each side of the lens. Such possible three-point groups are designated as "triplets" in this thesis. These triplets always have a constant distance from each other and are located on a straight line.

Triplets can be constructed by accessing all points that are one step apart, which is done by extracting the indices where the value is "True" in D:

$$i, j := \text{indices where } D_{ij} = \text{True} \tag{3.11}$$

In addition, a third point is added to the triplet if it is in range of either of those points. These points are found by the operation:

$$k := \text{indices where } D_{jk} \vee D_{ik} = \text{True} \tag{3.12}$$

If three such points exist, it is checked if all points have different positions and the two smaller distances have approximately the same value to ensure that a triplet is eligible

min	11.3
max	20
min_triplet_angle	170°
max_points	217
leg_ratio	0.4
feature_threshold	$3e-9$

Table 3.2: Parameters for triplet generation

(the maximum relative difference is defined by leg_ratio). In addition, it has to be checked if all points are located on a straight line. This is done by applying the law of cosines and calculating the largest angle. a, b, and c are the Euclidean distances between the points i, j and k:

$$\eta := arccos(\frac{a^2 + b^2 - c^2}{2ab}) \tag{3.13}$$

If it is above a certain threshold ($\eta > leg\_ratio$), the triplets form a line, and it is assumed that it is straight enough for further processing. All triplets that are verified this way are considered to have the proper geometric properties to be part of the real markers and are therefore used in the next calculation steps as potential candidates.

The values of the parameters applied in this thesis can be found in Table 3.2.

3.4 Calculating potential line candidates

To further reduce the amount of noise and find the correct triplets, the specific arrangement of the six markers is considered. All must be located on a straight line, and the groups of three points must be at least a minimum distance apart. The input to this algorithm is multiple triplets, given as sets of three x-y coordinates. The result consists of combinations of triplets (called triplet pairs) that have the required characteristics to describe the position of the markers. An overview of the calculation of the possible line candidate is given in Figure 3.6.

All triplets are compared pairwise; for two triplets to form an eligible line, the triplets must have at least a minimum distance (given by the parameter min_big_distance). The average position of the points between the triplets is calculated and used to build a distance matrix that describes the average distance between each triplet. From this matrix, possible candidates can be found by checking if the distance is large enough. These candidates are then further filtered by checking if the regression lines through each individual triplet have approximately the same slope/angle (given by the parameter angle_similarity). If that is the case, the total loss of a triplet pair is calculated by summing the squared orthogonal distance of each point to the regression line of the other

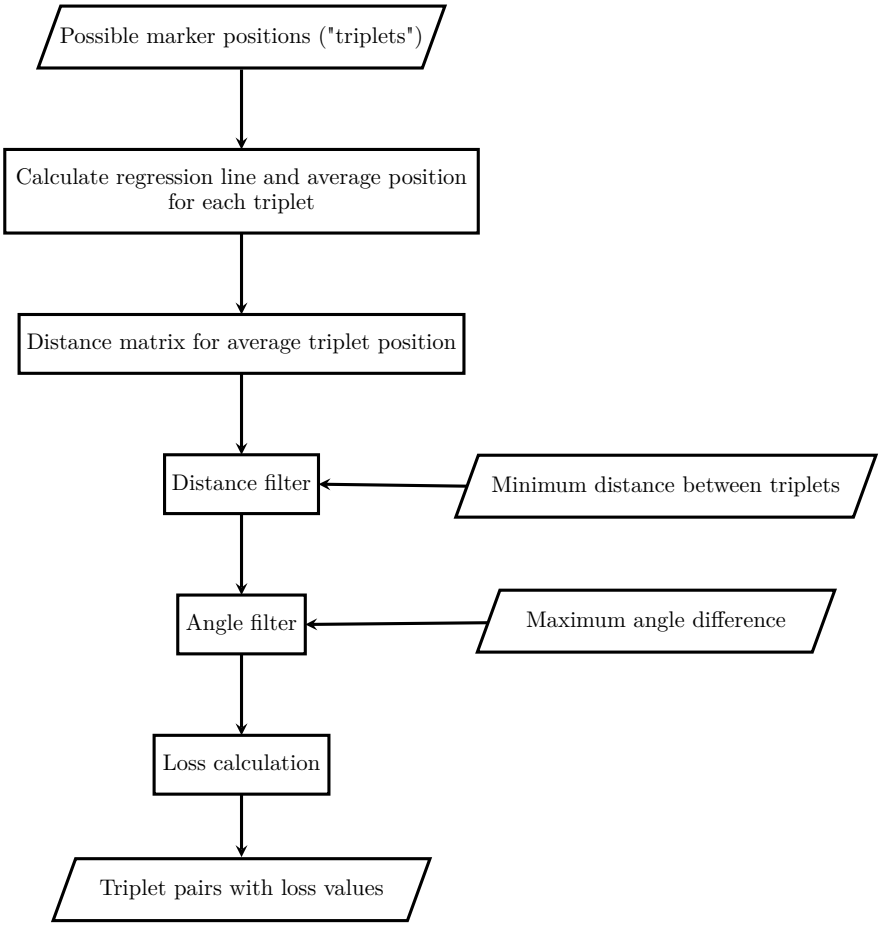

Figure 3.6: Part 2 of Process flow for single image processing: Triplet matching

triplet. This describes how well a line fits the given points. The squared orthogonal distance is defined as

$$\delta_{\perp}((x,y),s,i) := \frac{(y - s * x - i)^2}{s^2 + 1} \tag{3.14}$$

where s and i represent the respective slope and intercept of a line. The approximate center of the lens is denoted as c, which is composed of a x and y coordinate (result from mask calculation, see Section 3.1).

In addition to that, the orthogonal distance of the center point of the lens to each one of the regression lines is also added to this loss (see equation 3.15). Then the total loss ($\mathcal{L}$) for a given triplet pair a and b (each consisting of three unique points) is calculated as:

$$\mathcal{L} := \sum_{k=1}^{3} \delta_{\perp}(a_k, s_b, i_b) + \sum_{k=1}^{3} \delta_{\perp}(b_k, s_a, i_a) + \delta_{\perp}(c, s_b, i_b) + \delta_{\perp}(c, s_a, i_a) \tag{3.15}$$

min_big_distance	290
angle_similarity	6.8°
loss_cutoff	8400

Table 3.3: Parameters for triplet matching

s_a and i_a represent the slope and intercept of the regression line for triplet a, and s_b and i_b represent the slope and intercept of the regression line for triplet b.

This results in a list of possible triplet pairs with their respective loss. If the loss exceeds a cutoff value, it is ignored and not used in further processing (cutoff value given by loss_cutoff). The used parameter values can be found in Table 3.3

3.5 Determination of the best line

The goal is to find the line chosen from a set of possible candidate triplet pairs that has the highest probability of correctly describing the position of the lens. The slope and intercept of a pair can always be constructed by calculating the linear regression for the six points that make up the triplet pair. There are several ways to determine the line that is closest to the markings. Those methods are described in this section.

3.5.1 Choosing line by best regression

The simplest way to choose the best triplet pair is to take the pair of triplets with the lowest loss (from calculation 3.15). A regression line is fitted through all six points, resulting in a single line describing the position of the lens. The steps taken can be found in Figure 3.7.

3.5.2 Choosing line by best regression over multiple images

Due to the fact that there is more than one image captured at each timepoint, a comparison algorithm can be derived which utilizes the triplet pair information of all images. All images taken at the same time must be very similar, and their lens rotation α across them must be constant, because the images were taken only seconds after each other. However the eye can move between images taken after each other, resulting in an image rotation β. Therefore this image rotation (see Section 3.6) must be eliminated by applying a image synchronization $\Delta\beta$ to get comparable images with respect to the position of the markers. Triplets that really do describe the lens markers can be found in all of those images at the same coordinates - triplet pairs that only occur in a single image are more likely to be random noise. Filtering those reduces the influence of random noise on the final result and enhances the reliability of the resultant data. The algorithm that accomplishes this is described in more detail in this subsection.

First, all images taken at the same timepoint (e.g. all images of Patient 001 at 1h) have to be processed with the previous steps, so that they all have possible triplet pairs.

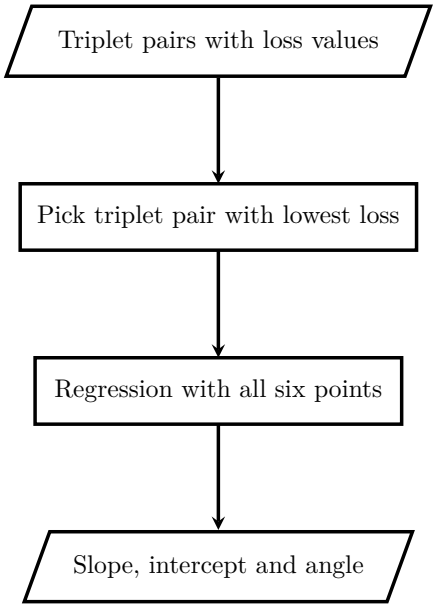

Figure 3.7: Part 3 of Process flow for single image processing: Choosing best line

Triplet pairs with a very high loss are discarded, and only those with a small loss are considered. Then an image synchronization is performed for all images, so that they are oriented consistently. This image alignment is done relative to an arbitrarily chosen reference image from the pool of images taken at that timepoint to simplify calculation. This results in a transformation matrix for each image that is used to transform the points into the same coordinate space, simulating the effect that the lenses are always located at the same coordinates.

The transformed points are then scored; all points that are within a certain distance of each other (bounded by a parameter called *binning_size*) are considered to describe the same feature in the image. If a point is detected in multiple images at the same location, it is a real image feature and less likely random noise, and its score will be higher than points that have only been found in a few or only in a single image. In the end, the sum of the scores of each triplet pair is calculated, and the triplet pair with the highest score is taken as a result. If there are multiple triplet pairs with the same score, the pair with the lowest loss is taken. Finally, a regression line is fitted through the triplet pair, which serves as the result.

3.6 Image alignment/registration

Image registration is the process of aligning the images of a patient to each other so that there are no influences such as head rotation or movement present. This alignment is represented by a transformation matrix that describes translation, scale, and rotation. Other kinds of affine transformations, such as shearing or reflection, are not important,

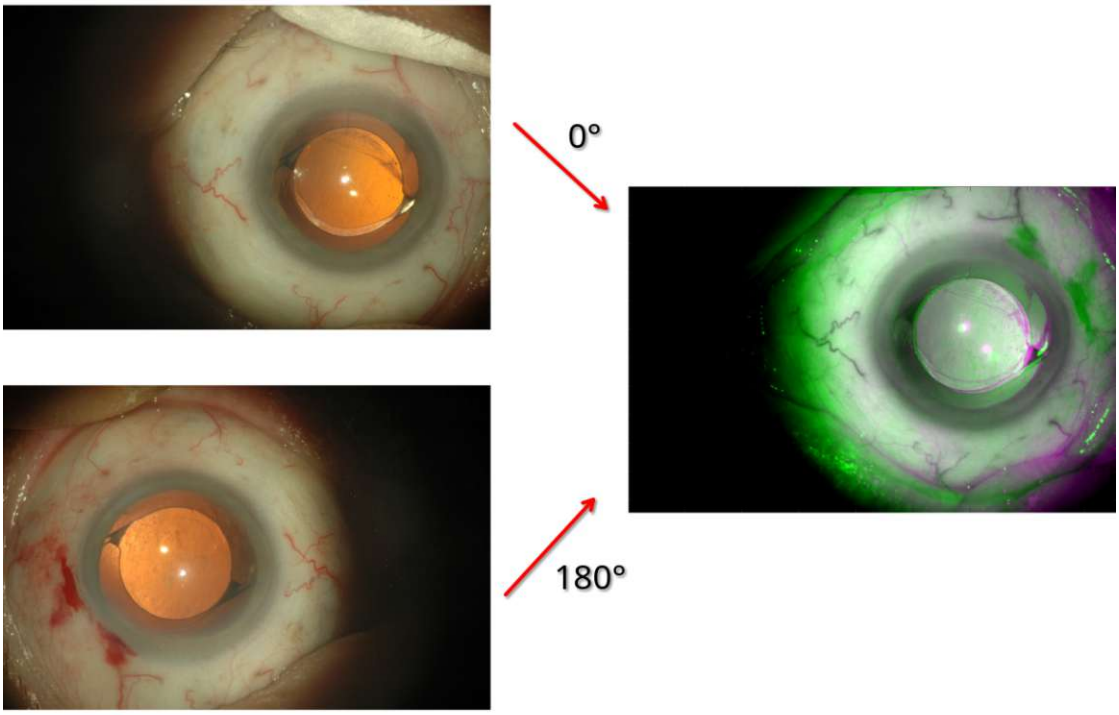

Figure 3.8: Image alignment example. Lower image needs to be rotated 180° to match the upper image. The overlayed images can be seen on the right (green: lower image; purple: upper image

because the images were always taken with the same camera equipment in the same setting. The main difference is the positioning of the head of the patient (slight rotation and translation possible) and the distance of the camera ocular to the position of the eye (slight scale differences). An example of an image registration can be seen in Figure 3.8.

Similarly to the manual method (see Section 2.5), the registration algorithm is designed to identify blood vessels that are used as landmarks to match the same vessel between two pictures. This is done by matching key-points of the images that are generated with the help of KAZE[ABD12] and AKAZE[AS11]. The same settings as previously discussed were used, the number of octaves and layers was set to one, resulting in simple diffusion on the full-scale image. The detection itself is performed on the "G" channel of the RGB image because it proveides high contrast present between the background and blood vessels. The maximum amount of points returned by AKAZE (max_points) must be set, as well as the sensitivity of the features (feature_threshold). For calculating the feature-correspondence of a pair of pictures of the same eye, the FLANN (Fast Library for Approximate Nearest Neighbors) algorithm is used [ML09]. This algorithm has a parameter, called the "Lowe's ratio", which serves as a measure to control false positive feature matches [Low04a].

Lowe's ratio	0.85
max_points	500
leg_ratio	0.4
feature_threshold	$1e-10$
t_{ecc}	0.3

Table 3.4: Parameters for choosing the best line

The registration is always performed for a pair of images. The previously generated sclera mask is used to focus the algorithm on features on the sclera, so that the registration is independent of the lens position or any other image features that might skew the result. The AKAZE algorithm is applied first on both images, with their respective sclera mask applied. The resulting keypoints and descriptors are then matched with the FLANN algorithm. This results in pairs of points that are considered to describe the same feature in each image. These pairs of points are passed to an estimation function (built into the opencv library, called estimateAffinePartial2D), which calculates the affine transformation matrix. This function excludes wrong matches based on a RANSAC detection of outliers [FB81] and refines the matrix based on the Levenberg-Marquardt optimization method. This results in a transformation matrix in the form

$$t := \begin{pmatrix} t_{0|0} & t_{0|1} & t_{0|2} \\ t_{1|0} & t_{1|1} & t_{1|2} \end{pmatrix} \tag{3.16}$$

The elements $t_{0|0}$, $t_{0|1}$, $t_{1|0}$ and $t_{1|1}$ encode rotation and scale, $t_{0|2}$ and $t_{1|2}$ translation. It is then checked, if the result is usable by calculating the similarity between the pictures after applying the transformation matrix to one of the images. For this purpose, the enhanced correlation coefficient *ecc* is used [GE08]. If the absolute value of *ecc* is close to one, the images are very similar. If it is close to zero, almost no similarities are found. The threshold t_{ecc}, which must be exceeded for an image pair to pass, has to be set rather low (approx. to 0.3), so that slight differences in the image are allowed. This is necessary, because images taken at different timepoints might differ in details, but should look very similar in general.

3.7 Calculation of image rotation difference

The calculation of the rotational difference is the final step in the processing pipeline. As input, it needs the marker angle γ of each image and the transformation matrix between image pairs. It yields the pairwise difference of lens angles for each timepoint. Due to the registration step, the lens angle difference $\Delta\alpha$ can be easily computed by subtracting the marker angles of each image and adding the registration rotation. For each pair of pictures, the image rotation $\Delta\beta$ is extracted from the registration matrix using the

formula:

$$\Delta\beta = atan(\frac{t_{1|0}}{t_{0|0}}) \tag{3.17}$$

given the transformation matrix t.

3.7.1 Application of image rotation difference to calculate lens angle difference

In this thesis, we look for the lens angle difference $\Delta\alpha$. Since α, the angle of the lens, cannot be directly measured, it must be derived indirectly via measuring γ, called marker angle. It is defined as

$$\gamma := \alpha + \beta \tag{3.18}$$

where α is the lens angle and β is the rotation of the whole image (caused by head rotation of the patient). Since the absolute head rotation of a patient is not known, only the difference in head position between two images can be calculated (as explained in Section 3.7). Therefore, the lens angle difference can be derived in the following way for an image pair I1 and I2:

$$\begin{aligned} \Delta\alpha &:= \alpha_{I2} - \alpha_{I1} \\ \Delta\beta &:= \beta_{I2} - \beta_{I1} \\ \Delta\alpha &= \gamma_{I2} - \beta_{I2} - (\gamma_{I1} - \beta_{I1}) \\ \Delta\alpha &= \gamma_{I2} - \gamma_{I1} - \beta_{I2} + \beta_{I1} \\ \Delta\alpha &:= \gamma_{I2} - \gamma_{I1} - \Delta\beta \end{aligned} \tag{3.19}$$

γ_{I1} and γ_{I2} are the rotational results of the images from previous steps, which can be calculated by applying the *atan* function to the slope of the regression line. The resulting angle $\Delta\alpha$ is the value of interest for physicians.

Alternatively, a calculated line can also be rotated by applying the transformation matrix t to the calculated regression line of the image $I2$. This method has the benefit that not only the angle is transformed, but also the intercept and slope itself are transformed onto the compared image. For this method, two points on the line have to be chosen. The points are then multiplied by the transformation matrix, resulting in the transformed points. These points can then be used to calculate the rotated slope and the new intercept of the line, by solving the linear equation system $y = x * s + i$. The angle γ^t_{I2} is derived by converting the slope to the required angle, using the atan function. Lastly, the rotational difference can be calculated directly with the formula:

$$\Delta\alpha = \gamma^t_{I2} - \gamma_{I1} \tag{3.20}$$

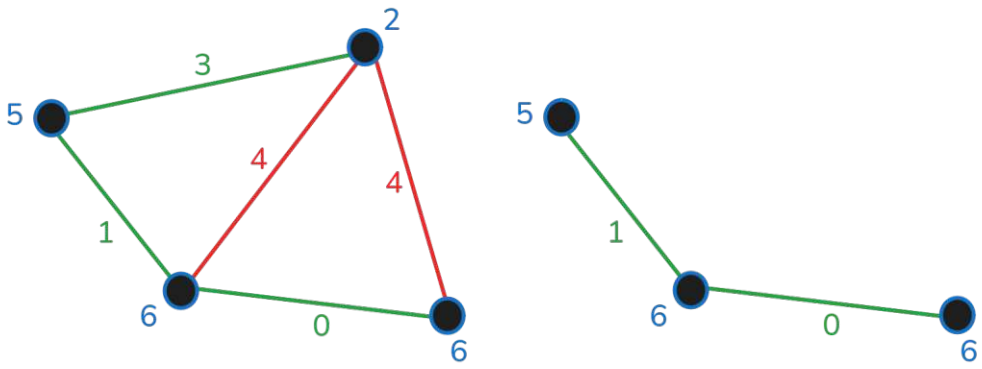

Figure 3.9: Example graph before (left) and after processing (right). Blue: γ measurements; Green: Calculated $\Delta\alpha$ below threshold t_1; Red: Calculated $\Delta\alpha$ above threshold t_1

3.8 Cleaning results

The cleaning algorithm can be used to enhance the results after processing all images using any algorithm. It needs all angular differences between image pairs ($\Delta\alpha$) as input. The algorithm removes entries that seem to be wrong by utilizing different heuristic methods, which are described in this section. Two different approaches are discussed, named intra-timepoint and inter-timepoint cleaning.

These methods can be best explained by viewing the given data as a simple graph made up of nodes and edges. Each node represents a single image (with a measurement α_I), the undirected edges represent a measurement ($\Delta\alpha$, calculated with the equation 3.19 or 3.20) between a pair of images.

3.8.1 Intra-timepoint cleaning

The base assumption is that all measurements $\Delta\alpha$ should be zero when comparing images of a patient that have been taken at the same time. If a measurement is over a certain threshold $\Delta\alpha > t_1$, it is considered faulty. This value is set to 3° in execution. Due to the fact that $\Delta\alpha$ is generated by comparing the rotations of two images (γ_{I1} and γ_{I2}), a node with many faulty measurements attached to it has probably been wrongly evaluated. Therefore, the algorithm iteratively removes all nodes (measured γ values), deleting the node with the most faulty edges connected to it in each iteration until no faulty connections are left. If a node is deleted, all connected edges (correct and faulty ones) are also removed, thus ignoring the measurements $\Delta\alpha$, which are based on a faulty measurement γ. When there is an equal number of suspected faulty edges connected to two different nodes, the one that will be deleted is chosen randomly. An iteration of this process on an example data set can be seen in Figure 3.9.

3.8.2 Inter-timepoint cleaning

A different method for cleaning the result is a generalized method of the previously described cleaning algorithm. Instead of relying only on measurements taken at the same timepoint, measurements taken at different timepoints are also considered. The fact that the lens rotation $\Delta\alpha$ for a patient between two appointments is constant is utilized. The main problem is that there is no previously known value for the "true" rotation between timepoints. Thus, it must be estimated by taking the average μ of the $\Delta\alpha$ values between all pairs of timepoints. Calculating the standard deviation σ of these sets, the accuracy of the guess can be assessed: A standard deviation higher than a certain threshold t_2 indicates that the calculated $\Delta\alpha$ values do not agree on a certain estimation. This value t_2 is set to 1.1 in execution. Therefore, all values where $\Delta\alpha - \mu \geq t_2$ are considered faulty. Then all images with the most faulty measurements are iteratively deleted, as described in Section 3.8.1.

3.8.3 Comparing measurements to given labels

The problem of not knowing the "true" value could also be circumvented by using the labels provided by medical professionals (denoted as $\phi_{\Delta\alpha}$). The algorithm would work like the one mentioned in Section 3.8.2, with the exception that the mean μ would not have to be calculated; the value of μ is determined by manual measurements ($\mu = \phi_{\Delta\alpha}$). Then, all values could be deemed faulty that are too far away from this chosen "true" measurement. This value is set to 1° in execution. The drawback of this method is the need for manually labeled values, which might not be readily available.

CHAPTER 4

Convolutional neural networks background

In this chapter an introduction to a special type of an Artificial Neural Network (ANN), known as a Convolutional Neural Network (CNN) is given. CNNs have been continuously researched since their introduction in 1990 and have been applied to various applications, especially in picture classification, and improved in numerous scientific works [HHHJ90][LBBH98][GWK+18]. In the following, we briefly discuss their typical architecture, design, and limitations. After the relevant evaluation metrics are reviewed and a description of the specific network used in this thesis is given. At the end, a detailed explanation of the necessary data preparation is provided.

4.1 Typical architecture of CNNs

In general, ANNs are algorithms that process data (like images) by combining it with pretrained parameters, called weights. This combination includes multiplication, summation, and the application of a nonlinear activation function, but more complex algorithms can also be applied. These calculations are performed multiple times in sequence, always reusing previously calculated information. In the end, there is a set of output variables that can be interpreted depending on the task that the net should solve. ANNs usually consist of layers, representing architecture units realizing some abstraction level. In the following, we briefly discuss typical layers of a CNN.

4.1.1 Convolutional layer

CNNs are specialized ANNs that are based on utilizing so-called convolutions as mathematical operations. The application of convolutions in a CNN can be interpreted by the help of a window (called kernel k) that is moved over the input with a given step

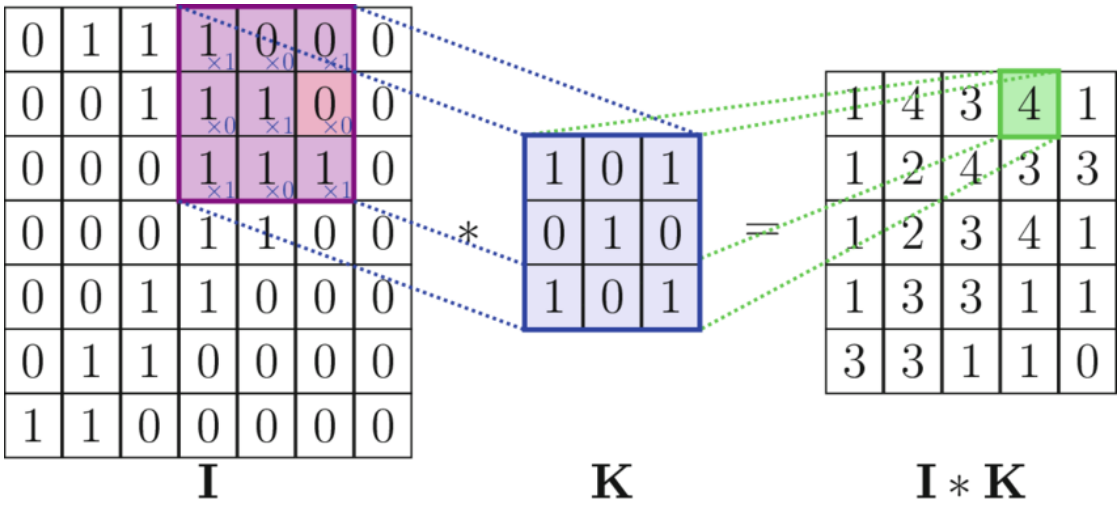

Figure 4.1: Visualization of a single step in convolution. Source: [VCB+23]

size (known as a "stride"). At each step, all elements of the inputs that are inside the window are multiplied by the corresponding weight of the window. The resulting values are then summed, so that a singular new value is generated. The individual result values are inserted in the same order as read from the input [DM11]. A visualization of this convolution for one step can be seen in Figure 4.1. The values of kernel k are trainable parameters, meaning that they act as weights that can be changed in training with algorithms mentioned below. Usually, a group of kernels of the same size, applied to the same input make up a convolutional layer. For each kernel of this layer, an output is generated. In many CNN architectures, multiple convolutional layers are used subsequently, always using the output of the previous layer as their input.

4.1.2 Fully connected layer

Another core part of CNNs are the fully connected layers. These layers consist of a fixed number of neural units, each of which has the full output of the previous layer as input. For each node, there exists a weight for each input to it, which is multiplied by the input itself - all the resulting values are summed and passed through an activation function, which is explained below [VCB+23]. A schematic of a fully connected layer can be seen in Figure 4.2 - the formula used in this image can be found in equation 4.1. The inputs are labelled with I, the activation function is f and the weights are written as w_{ij}, where i is the index of the input and j the index of the node.

$$O_k = f(\sum_{i=1}^{n} w_{ik} * I_i) \tag{4.1}$$

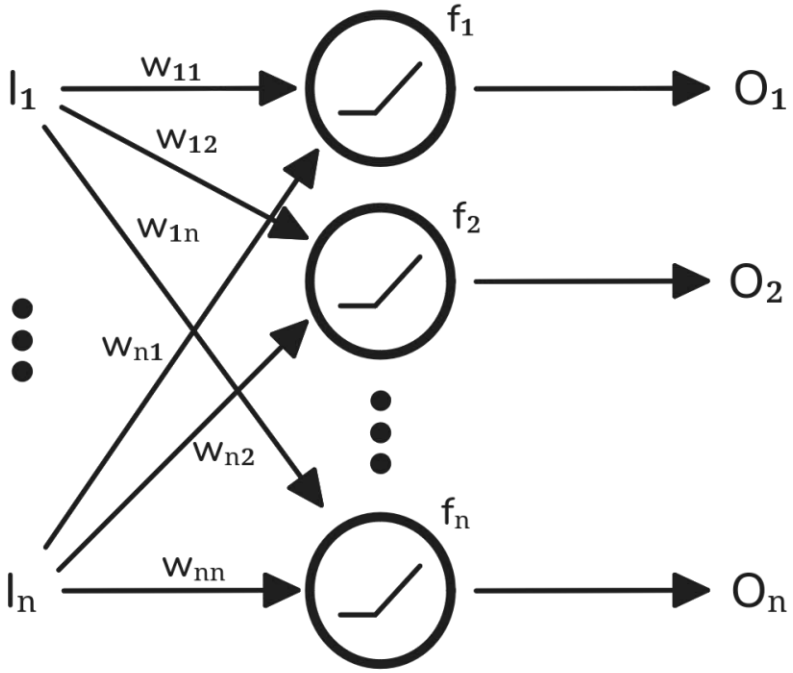

Figure 4.2: Fully connected layer with ReLu activation function

4.1.3 Output layer

The output layer generates the final output for an ANN. This means that after it, no additional layers are present. For classification tasks, the output layer consists of multiple neural units, while for regression tasks, there is only one output. In this thesis, we use a CNN with a single output variable that is interpreted as the angle of a lens in an image.

4.1.4 Activation function

Usually, an activation function is used to introduce non-linearity in the neural network. This change enables ANNs to realize more complex discrimination functions and thus model more complex patterns of a given dataset. Without nonlinearity, only a small subset of problems can be solved. A commonly used activation function is ReLu, which mitigates issues such as the problem of vanishing gradients by setting all negative output values to zero. This enables learning for CNN architectures consisting of many layers.

4.1.5 Pooling layer

Max-pooling and average-pooling are multidimensional downsampling methods, which reduce the resolution of a given input. Similarly to convolutional layers, both methods require a window that determines the size of the region that is processed. This window is moved over the input with a given step size (known also as a "stride"), which can be

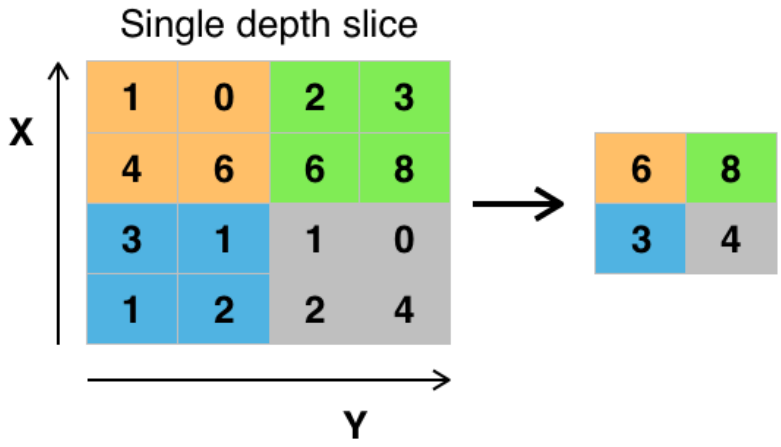

Figure 4.3: Example for max-pooling with a window size of 2 and a stride of 2. Source: [Aph15a]

adjusted according to the required output resolution. The difference between the two methods is that, in each step, max-pooling takes the maximum value in a given region, and average-pooling calculates the average of all values currently in the window. This value is then used as an output for the current window position. Spatial information is preserved, because each result is inserted into the output matrix in the same order as the input is processed [VCB+23]. An example of max-pooling can be seen in Figure 4.3.

4.1.6 Flatten layer

To convert a matrix of higher dimension to a one-dimensional vector, a flatten layer is used. It usually is located between the output of a convolutional layer and the input of a dense layer. The reason for this is that most software libraries have generic implementations for those types of layers, which require specific input shapes; due to the multidimensional outputs of convolutional layers and the requirement for one-dimensional inputs for dense layers, the flatten layer is needed. Therefore, this layer is of technical nature and does not change the contents of the inputs, but only the arrangement of it.

4.1.7 Dropout layer and Batch normalization

A dropout layer realizes a method to temporarily disable a percentage of neural units so that they are not effective in the network. These neural units are chosen newly in each training iteration, resulting in the usage of all neural units with with enough training iterations, but not all at the same time. The dropout layer is effective only for training the network. This layer is used to improve the generalization ability of the CNN model by preventing overfitting [SHK+14].

Another method that can be used to improve training performance is batch normalization.

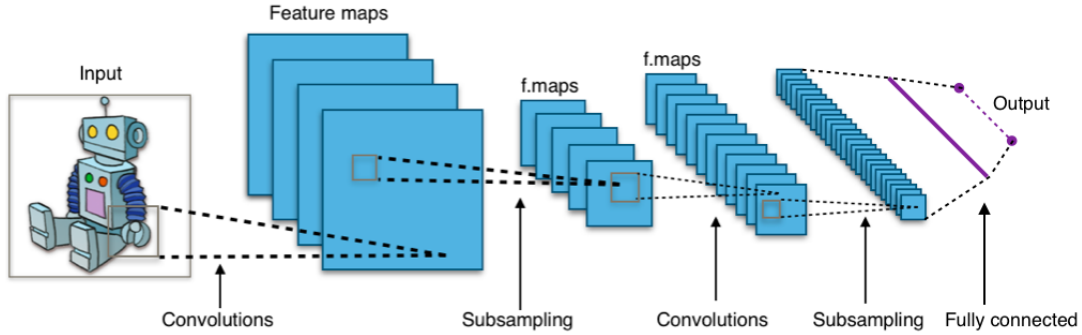

Figure 4.4: Typical image processing CNN structure. Pooling corresponds to subsampling in the picture. Additional batch-normaliation layers and dropout layers are not shown. Source: [Aph15b]

This layer is designed to process inputs in groups called batches, enabling the simultaneous handling of multiple images. Within each batch, the mean and variance of the inputs are calculated. These values are used to normalize the data, transforming it to have a mean of zero and a standard deviation of one. This normalization ensures that the input values are on a consistent scale. After normalization, the data is rescaled and shifted using parameters that are learned during training. These parameters allow the model to restore its representational power, ensuring that it can still capture the complexities of the data after normalization.

By normalizing the inputs in this way, the layer reduces a phenomenon known as covariance shift, where changes in the input distribution can disrupt the learning process. This reduction in shift reduces the need to constantly adapt to varying input distributions. As a result, the training process is better regularized, meaning it becomes less prone to overfitting and noise. In addition, this method improves the efficiency of training, leading to faster and more stable convergence of the model parameters [IS15].

4.2 Design of the CNN algorithm

A typical CNN architecture usually consists of a combination of the layers mentioned above. Their structure has to be carefully chosen on the basis of the task that has to be solved. An overview of how commonly used CNNs are generally structured can be seen in Figure 4.4.

To properly use a CNN or ANN for solving a given task, two things have to be done: In training, the weights of are gradually changed so that the network is better able to calculate the results closer to the real value. In testing, the performance of the trained net is evaluated on data that the network has never seen before. There is an optional third part called validation, which is used to change parameters that are not directly used in the network, but still changes the networks' behavior and performance. This is known as hyperparameter tuning, which, for example, is an adjustment of the number of neural

units in each layer. After performing these tasks, the generated net with the calculated weights can be used on any input data and is usually interfaced with automatic pre- and postprocessing algorithms.

Training an ANN is the way to gradually increase the accuracy of the network by adjusting the weights. The basis for this procedure is a training set, consisting of input data i and target data ϕ. In the ideal case, an ANN should calculate the exact same value as the target ϕ given a certain input i. When creating a new ANN, the weights are often randomly initialized and in most cases the output of an ANN o will not be close to the desired target ϕ. To achieve a closer match, the weights must be gradually adjusted so that a given input produces an output closer to ϕ. This weight change can be performed with different optimization methods explained below; although the methods are commonly based on the results of a procedure called backpropagation, which uses gradients to change each weight parameter, based on the difference in ϕ and o [Roj96]. This training process is repeated over the same training set in so-called epochs and adjusts the weights based on the error of each iteration.

To measure how large the difference between an output o and the target ϕ (the error) is, a loss function is defined. There are many different types of loss function for regression tasks, such as the mean absolute error MAE or the mean squared error MSE. In this thesis, the Huber loss was taken, which combines aspects of MAE and MSE. More specifically, this function has a parameter δ, where $|o - t| > \delta$ is scored with MAE and all other differences are scored with MSE. This makes it more robust to outliers and can improve training performance [CEL+23].

The weight change in each iteration is determined by an optimization function. The goal of these functions is to try to change the weights in such a way that the error gets as small as possible in as little time as possible. A widely used optimization function is called Adam [KB14]. This optimization method builds on the results of backpropagation; on the basis of these results, it calculates the amount of change in individual weights.

4.3 Limitations of CNN

There are limitations of ANNs that must be taken into account when planning the structure of an ANN. Here is a list of the ones relevant to this thesis:

- **Necessary amount of data**: For an ANN to learn important features properly, large amounts of data are needed. This can easily surpass the available information, especially if the ANN is large in size and many trainable weights are present. If it is not possible to collect enough data, more data have to be generated by augmenting the already existing training set. The consequence of having little data is poor accuracy or overfitting.
- **Overfitting**: Usually, overfitting can be observed when test accuracy is significantly lower than the training accuracy. It implies that the network learned how to identify

specific images of the training data, but is not able to generalize the "knowledge" of the the training due to samples that it has never seen before. This renders the result unusable, because the network cannot be used on new data. There are several ways to reduce the impact of overfitting, for example with dropout layers or introduction of noise in an augmented dataset.

- **Computational cost**: Nowadays, even consumer GPUs are able to train relatively large models in a short amount of time. Nevertheless, it is always easy to worsen the performance to the point of uselessness by increasing the network size, because the computational cost does not scale linearly with network size. In addition, more data points or more complex inputs also significantly worsen performance. Therefore, special care has to be taken in the design of a network and in the preparation of training data. When done correctly, training time can be reduced significantly (e.g. to a few hours instead of days or weeks).

- **Imbalanced target distribution**: When performing regression tasks, it can happen that the distribution of training data lacks uniformity. The predominance of certain values results in poor generalization of the trained network to data that alters from this distribution. This can be alleviated with methods like resampling, like domain specific augmentations.

All of these factors cannot be viewed independently; if, for example, overfitting occurs, increasing the amount of data might lead to training times that are no longer acceptable. Therefore, when designing a CNN these limitations have to be considered carefully and balanced properly.

Designing a CNN, an ANN with a specific type of architecture, introduces several further limitations:

- **Poor generalization to out-of-distribution data**: If data is supplied that does not come from the distribution the network was trained on, the CNN can perform considerably worse. This can for example include different lighting conditions or differing image quality. This can only be fixed by including images from other sources - increasing the variety of image data provided alleviates this problem.

- **Limited ability to capture spatial relationships**: CNNs focus on local features through their convolutional filters. They are not designed to capture global context or features that are far apart in given input images. This problem can be alleviated by performing hyper-parameter tuning and therefore customizing the given CNN model by altering its structure, which can improve the ability to capture global spatial relationships.

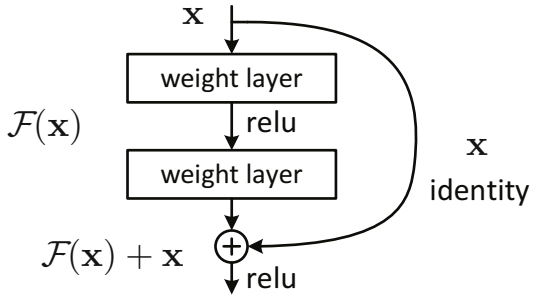

Figure 4.5: Utiliezd block in ResNet18. Source: [HZRS15]

layer name	output size	18-layer
conv1	112×112	7×7, 64, stride 2
conv2_x	56×56	3×3 max pool, stride 2
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$
	1×1	average pool, 1-d fc
FLOPs		1.8×10^9

Table 4.1: ResNet18 architecture. Building blocks are shown in brackets, with the numbers of blocks stacked. Downsampling is performed by conv3_1, conv4_1, and conv5_1 with a stride of 2. Source: [HZRS15].

4.4 CNN-Model: ResNet

In this thesis, the smallest standard model of the ResNet family has been used as CNN model. It is called ResNet18 [HZRS15]. Its structure can be seen in Table 4.1. The mentioned building block for ResNet18 consists of two convolutional layers, with the ReLu function applied after the first layer. After the second layer, the input of the block is added to the output, which is called the residual connection. Subsequently, ReLu is performed, and downsampling in the form of MaxPool occurs. A diagram of the block can be seen in Figure 4.5

Compared to the original model, two things have been adapted to fit the task at hand. First, the output was changed to contain only one output variable. This is due to the goal of training the net to perform regression, which in turn means that this output variable will be the slope / angle of the lens. Additionally, in different experiments, one to four input channel are used, which encode the RGB image with an additional layer encoding important lens features. Due to ResNet18 having only three channels as input by default, the first convolutional layer "conv1" has to be changed to accommodate the appropriate number of channels as input.

4.5 Data preparation

The utilized ResNet CNN needs images with a resolution of 224x224px. Additionally, only the lens is of interest for the CNN, features like the sclera are not important for this step. The location of the pupil is extracted with the help of the pupil mask, as described in Section 3.2 and the whole image is cut so that only the pupil is visible. Due to the fact that the size of the lens slightly differs in each photograph, a resizing of the image has to be carried out. Additionally, augmentation and encoding of additional information ("important" points) is performed, which is described in the following sections.

The result of the preprocessing is either a 4-channel or 3-channel ".png" images with corresponding labels. The data is split into a training set ($\sim 65\%$ of images), a test set ($\sim 20\%$ of images) and a validation set ($\sim 15\%$ of images). To ensure that the test and validation sets contain pictures that the CNN was not trained on, the data is split with respect to ID. The data of each patient are always fully contained in one of the sets.

4.5.1 Augmentation

Augmentation is needed, so that the model can properly learn how to identify important landmarks to determine the lens rotation. Without augmentation, the diversity in given data can possibly be too low and therefore the performance of the CNN is not ideal. This can be seen in the top plot in Figure 4.6 - certain marker angles are not present in the source data. To fix this issue rotation augmentation is applied. Image rotation is needed, so that every possible marker position is represented in the dataset. Without it, certain marker positions might be overrepresented, because artificial lenses are often implanted in a particular orientation. Therefore, a rotation sampled from a uniform distribution in the interval of $]0°, 180°[$ is applied to the image and the corresponding label. The difference in distribution can be seen in Figure 4.6.

4.5.2 Encoding additional information

Certain information known from the CV algorithm can be utilized, in order to focus the neural net on potentially interesting points. These interesting points are derived from the result of the AKAZE algorithm of Section 3.3 and can be seen in Figure 3.5.

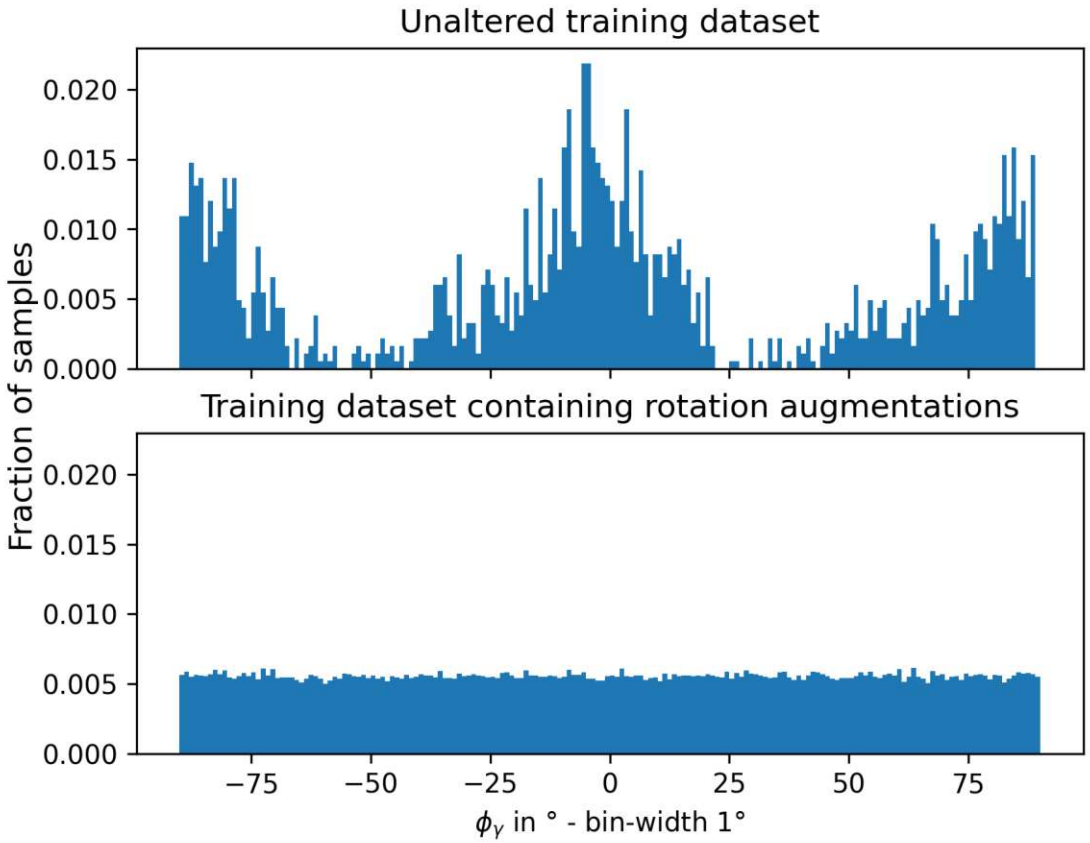

Figure 4.6: Density histogram comparing rotation distribution ϕ_γ with and without augmentation

The possible marker spots generated with AKAZE serve as feasible candidates for lens markers, which, in turn, are used to determine rotation.

To make this information usable for CNNs, an additional channel is generated for each image. This channel has the same dimension as the image itself (224x224px). It encodes regions of interest and can have values ranging from 0-255 (8 bit), where 255 represents regions of high interest, and 0 represents regions that should be ignored. It is either stored alongside the original RGB color channels as a fourth channel, or it can be encoded into the RGB channels themselves. For the latter, each RGB value is divided by 255 and multiplied by the corresponding mask value. This can have unwanted side-effects due to removing information on the original channels, therefore a fourth channel is used instead.

The points of interest are encoded onto the additional channel as x-y coordinates. Due to each marker on the lens having a certain diameter, it makes sense to mark all pixels in a region around each point as interesting. This can be achieved by using a scaled Gaussian kernel; this kernel has the value one in the center and falls off with a given intensity. The

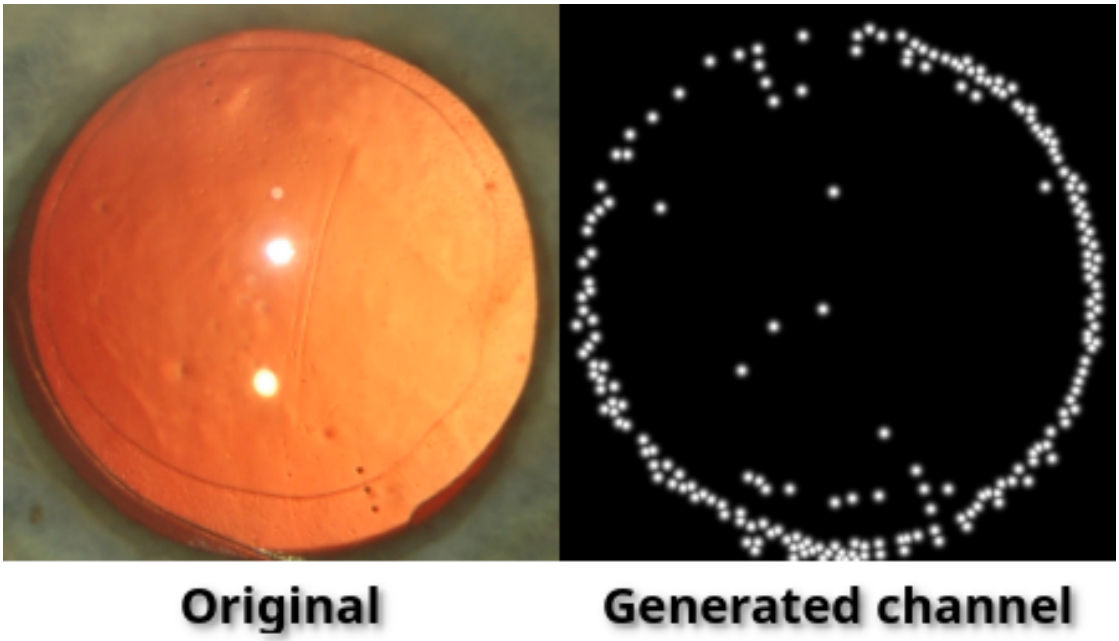

Figure 4.7: Original image next to generated mask.

kernel can be adjusted by providing a standard deviation σ and an odd-numbered size s. All kernel values can be multiplied by a value p, which describes the intensity of the pixel with the highest value, normally the center pixel. All other pixels have lower intensities and decrease in value the further away they are from the center. This kernel is generated for each point of interest and added to an all-zero mask *poi*. If two kernels overlap, the larger value is taken as the intensity of the pixel. The peak value p can be set uniformly to 255. The result of the algorithms can be seen in Figure 4.7.

A possible improvement is to take into account that the chosen corner detection algorithm AKAZE creates many points of interest on the rim of the lens, which are not as interesting for detecting the relevant markings on the lens. Therefore these points have to be eliminated their intensity has to be reduced: It is assumed that most points that are on the rim of the lens are almost at the border of the lens image, because of the way masking is performed. Based on this knowledge, a distance matrix is created between all points of interest. Then, a histogram is created from this distance matrix. All peaks on the histogram that are larger than a certain threshold are then calculated and the peak with the largest distance is taken as the approximate radius of the lens. All points that are close to this distance are considered to be part of the lens rim and therefore not as important. They get a peak value $p = 0$. Linear interpolation is used to increase the value to $p = 255$ over a small distance.

CHAPTER 5

Experiments

The goal in this chapter is to show how the previously discussed methods are combined to form full algorithms, which serve as a basis for the experiments conducted. The focus is put on explaining which steps are taken in each algorithm, therefore giving a clear picture on their structure, inputs and outputs. Additional information is provided on limitations of the algorithms, utilized libraries, and specific parameters used in the respective algorithms. Experiments are defined by stating their goals and the questions that are answered. The name of the experiments consists of the same letter as the corresponding base algorithm.

5.1 Algorithm A: Computer vision based marker angle calculation

The basis for this computer vision (CV) algorithm is the detection of the markings on the lens: Each lens has exactly six dots, sitting on a straight line in a very specific pattern (see Figure 2.1). This algorithm predicts the angle of the straight line relative to a horizontal line. This angle is denoted by γ and named marker angle subsequently. The basic idea is to find those dots and fit a straight line through them. The main obstacle is the noise present in the image: Due to the markings being very small, they are not easily distinguishable from other points considered as random background noise for any automated method. Therefore, an algorithm had to be devised that could exclude noise from the calculation and rank the found lines by quality. This algorithm is described in this section and can be seen in Figure 5.1. An approach was chosen that processes each image in isolation. Other approaches, such as working with multiple images at once could also be considered, but are not used due to their expected suboptimal performance.

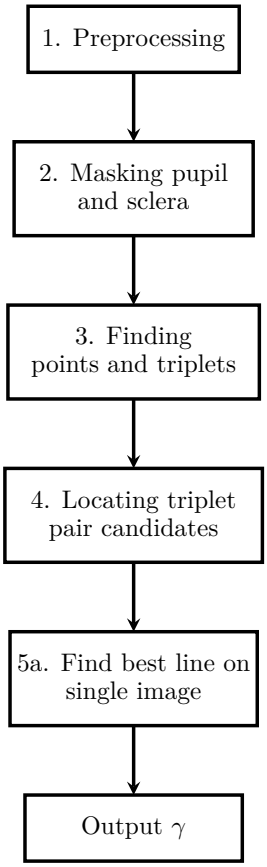

Figure 5.1: Algorithm A: Calculating γ based on single images

5.1.1 Structure

This section introduces the processing steps of this algorithm. The numbering of the steps is equal to the numbering seen in Figure 5.1. Algorithm A requires unaltered videos and images as an input and yields a marker angle γ for each individual input. If in any step processing fails, a result is returned that indicates that the given image cannot be processed.

Step 1 | Preprocessing

- Input: Raw images and videos
- Output: Identifiable images with common size and in HSV space
- Detailed description: Section 3.1

Preprocessing takes care of parsing all data into a common format and adding proper identifiers. The input consists of videos and images, with different resolutions, which

are converted into the same file format and common image resolution. Subsequently, the images are also converted to the HSV color space; both RGB and HSV color space representations are stored and used in later steps.

Step 2 | Masking pupil and sclera

- Input: Preprocessed images
- Output: Preprocessed images with corresponding binary pupil and sclera masks
- Detailed description: Section 3.2

The pupil- and the sclera mask are generated in this step. The pupil mask describes the location and size of the pupil; it consists of a center and a radius, which are used to generate a circular binary array indicating which pixels are part of the pupil. The sclera mask is calculated based on the pupil mask and describes the vessel area, i.e. which part of the image contains the sclera ("white of the eye"). The results are stored alongside their corresponding image. In this thesis, the method combining both CHT and color-based masking is applied.

Step 3 | Finding points and triplets

- Input: HSV images with corresponding pupil mask
- Output: Triplets
- Detailed description: Section 3.3

Based on the image content inside the pupil, points that indicate possible locations of the lens markers are calculated. These points are then filtered by geometric criteria, such as distance and alignment, and used to form groups of points having exactly three members, called triplets.

Step 4 | Locating triplet pair candidates

- Input: Triplets
- Output: Triplet pair candidates, that could describe the real lens markers with corresponding loss values
- Detailed description: Section 3.4

The triplets are screened to find pairs that could potentially describe the six lens markers. These lens markers must be located on a straight line, so the main part of this step is to discern which pairs of triplets also lie approximately on a straight line. In addition, further filter criteria, like a minimum distance between triplets, are applied to focus the results on fewer possible combinations. For each triplet pair, a loss value is calculated based on the position of the two triplets compared to the position of the center.

Step 5a | Find best line on single image

- Input: Triplet pair candidates, that could describe the real lens markers with corresponding loss values
- Output: A line in the form of intercept, slope and marker angle γ
- Detailed description: Section 3.5.1

The triplet pair with the lowest loss value is dedicated to be the best triplet pair. Subsequently, the slope and intercept of the line formed by the six distinct points that comprise the best pair in relation to the center of the pupil, are calculated. The marker angle γ can be calculated by applying the atan function to the slope of this line.

5.1.2 Experiment

This experiment aims to investigate the accuracy of the γ prediction, by using algorithm A. One overreaching goal is to assess whether algorithm A is suitable for estimating γ to serve as a basis for the calculation of the lens angle difference $\Delta\alpha$ (see Equation 3.19). In addition, it is examined whether the prediction accuracy of algorithm A is high enough to be used for automatic generation of γ labels, which are necessary to train a CNN model (see algorithm B).

To assess the feasibility of the given algorithm, the calculated marker angle γ is compared to the manually generated target angles ϕ_γ. For this analysis, the difference between γ and ϕ_γ is calculated according to formula 6.4. The resulting value, called marker angle error, is denoted by ϵ_γ. It is checked with the help of various histograms how this error is distributed. Key metrics, such as the mean absolute error (MAE) are calculated on a timepoint basis, to identify timepoints that have a lower error then others. Lastly, it is checked if the precision of the algorithm is dependent on the measured value γ.

5.2 Algorithm B: DL dominated marker angle calculation

The main part of Algorithm B is a CNN model (modified ResNet18), which can calculate the marker angle γ based on a 3 channel RGB image input. These images only contain the pupil of the eye - other parts of the images are removed in preprocessing steps. Optionally explicit feature extraction is also applied, resulting in features that are put in

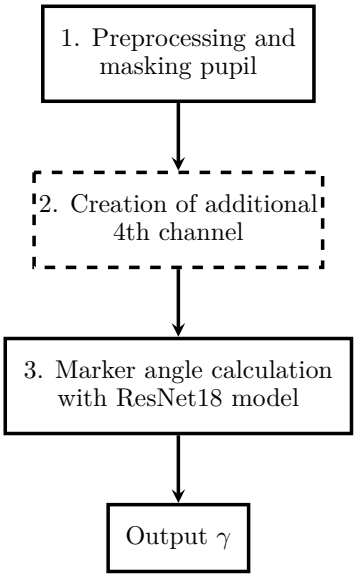

Figure 5.2: Algorithm B: Calculating γ with a DL based algorithm based on single images. Step 2 is needed when using an additional 4th channel.

an additional 4th channel. This explicit feature extraction is based on CV methods for finding potential marker points, which was already used in algorithm A. Using explicitly extracted features reduces the complexity of the applied CNN model, since the CNN is exempted from the task of identifying the necessary features. Thus, using explicit features makes it possible to perform experiments on hardware with lower processing capabilities.

5.2.1 Structure

This section introduces the processing phases of this algorithm. The numbering of the steps is equal to the numbering seen in Figure 5.2. Algorithm B requires unaltered videos and images as an input and yields a marker angle γ for each individual input. If in any step processing fails, a result is returned that indicates that the given image cannot be processed.

Step 1 | Preprocessing and masking pupil

- Input: Raw images and videos
- Output: Preprocessed images with corresponding binary pupil mask
- Detailed description: Sections 3.1 to 3.2 and Section 4.5 for CNN specific preprocessing

This step consists of the steps one and two of Algorithm A. In the course of additional data preparation, the pupil is cropped and the image size is set to 224x224px by rescaling.

Step 2 | Creation of additional 4th channel (optional)

- Input: Preprocessed images with corresponding binary pupil mask
- Output: Preprocessed images in a 224x224px resolution and additional 4th channel
- Detailed description: Section 4.5

This step utilizes CV methods (as used in step 3 of algorithm A) to generate candidate marker points. For each image, these points are then encoded onto an additional 4th channel, which serves as an additional input to the CNN.

Step 3 | Marker angle calculation with ResNet18 model

- Input: Preprocessed images in a 224x224px resolution and optional additional channel
- Output: Marker angle γ for each image
- Detailed description: Sections 4 and 5.7.

The CNN model is trained with manually generated lables called ϕ_γ. The model has to be pre-trained before it can be used to determine marker angles. The used CNN training approaches are explained in Subsection 5.7. The ResNet18 is choosen to be used as CNN model.

5.2.2 Experiment

The goal of the experiments of algorithm B is to investigate the accuracy of γ prediction by using algorithm B. Several different versions of algorithm B are carried out to determine the version with the highest prediction accuracy. For this reason, the different algorithm versions are compared in terms of their MAE value on a timepoint basis. The method with the lowest MAE is then selected for γ prediction for the calculation of the lens angle difference $\Delta\alpha$ in algorithm D. The tested algorithm versions with correspondingly adapted and trained ResNet18 models are as follows:

- **(1)** 3-channel input data: No preprocessing, RGB image as input
- **(2)** 4-channel input data: 4th channel is also fed into CNN

- **(3)** 4-channel input data: A modified 4th channel is fed into the CNN, which is created from the 4th channel by removing points on the rim of the lens as described in Section 4.5
- **(4)** 2-channel input data: One randomly chosen color channel and the additional 4th channel are used
- **(5)** 1-channel input data: Only the additional 4th channel is used

Similarly to algorithm A, the calculated marker angles γ are compared to the manually generated marker angles ϕ_γ; the difference between γ and ϕ_γ, the error ϵ_γ, is calculated as defined in algorithm A. The same key metrics, histograms and MAE are used again on a timepoint basis. This enables to select timepoints with lower marker angle errors. It is also checked, if the algorithm is dependent on the measured value γ.

5.3 Algorithm C: Computer vision based image rotation calculation

Image registration or synchronization is the process of aligning pairs of images to each other, so that externally induced rotations (caused by slight head movement) are eliminated when comparing images to each other. This process can also be called "image rotation". The amount of image rotation between the images is expressed by image rotation angle $\Delta\beta$, which represents the angle at which one image has to be rotated so that it becomes the same position as the other image. The anatomical structure that serve as concrete reference points are corkscrew vessels present in the sclera. They are visible in all images and remain relatively constant in appearance and location over multiple appointments. Therefore, this algorithm focuses on finding these vessels in the sclera and aligning them. Possible hurdles are the presence of irritated conjunctiva vessels or other objects that block the direct observation of the underlying corkscrew vessels.

5.3.1 Structure

This section introduces the processing steps of this algorithm. The numbering of the steps is equal to the numbering seen in Figure 5.3. Algorithm C requires unaltered videos and images as an input and results in an image rotation angle $\Delta\beta$ for each pair of images. If the rotation synchronization fails, no result is returned and the pair is deemed not synchronizable with this algorithm.

Step 1 | Preprocessing and masking sclera

- Input: Raw images and videos
- Output: Preprocessed images with corresponding binary sclera mask

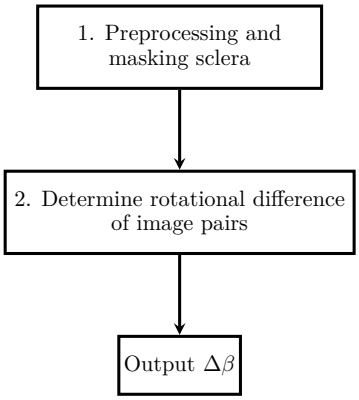

Figure 5.3: Algorithm C: Calculating $\Delta\beta$ with a CV based algorithm, based on pairs of images.

- Detailed description: Section 3.1 and 3.2

This algorithm needs the preprocessed and masked images as input; therefore, Steps 1 and 2 of algorithm A have to be applied to the raw images and videos, to make them usable for this algorithm.

Step 2 | Determine rotational difference of image pairs

- Input: Preprocessed and masked image pairs of the same patient
- Output: Transformation matrix for the image pair describing the rotation between them, being equivalent with specifying the image rotation angle $\Delta\beta$
- Detailed description: Section 3.6

The determination of rotational difference of image pairs is based on location and the appearance of the sclera. Notable landmarks of the sclera (usually large corkscrew vessels) are determined and matched if their local surroundings are similar enough. This results in pairs of matching points that can be used to calculate a transformation matrix, which describes how a set of points of one image can be moved to have the same coordinates as the same points on the other image. Thus, the transformation matrix includes the same information as the image rotation angle $\Delta\beta$. If no match is found between two images, or the synchronization drops below a certain quality, no output is generated and the pair is marked as unprocessable.

5.3.2 Experiment

The goal of this experiment is to evaluate the accuracy of the computer vision based image rotation calculation algorithm C. This is done by comparing the calculated image

rotation values $\Delta\beta$ to the manually generated labels $\phi_{\Delta\beta}$, i.e. by assessing the image rotation angle error $\epsilon_{\Delta\beta}$. Note that the labels $\phi_{\Delta\beta}$ are computed from ϕ_γ and $\phi_{\Delta\alpha}$ as

$$\phi_{\Delta\beta} = (\phi_{\gamma I2} - \phi_{\gamma I1}) - \phi_{\Delta\alpha} \tag{5.1}$$

5.4 Algorithm D: DL dominated lens angle difference calculation

Algorithm D combines the best version of algorithm B with the image synchronization method of algorithm C, to calculate the lens angle difference $\Delta\alpha$ for an image pair. The CNN model works on any image in isolation and produces the marker angle γ for each input. The lens angle difference $\Delta\alpha$ is computed from the equation

$$\Delta\gamma = \Delta\alpha + \Delta\beta \tag{5.2}$$

in the last step of the algorithm as

$$\Delta\alpha = \Delta\gamma - \Delta\beta = (\gamma_{I2} - \gamma_{I1}) - \Delta\beta \tag{5.3}$$

5.4.1 Structure

A flowchart describing this algorithm can be seen in Figure 5.4. It is a combination of algorithm B and C, with an additional last step to calculate the lens angle difference $\Delta\alpha$. This algorithm takes the unaltered videos and images as input and results in the lens angle difference $\Delta\alpha$. Cleaning is an optional step that can improve the accuracy of $\Delta\alpha$ results after all calculations have been done. It works by removing images with hypothesized faulty measurements. However, it is not investigated further in this thesis.

Step 1 | Preprocessing and masking pupil and sclera

- Input: Raw images and videos
- Output: Preprocessed images with corresponding binary pupil and sclera masks
- Detailed description: Sections 3.1 to 3.2 and Section 4.5 for CNN model specific preprocessing

This step consists of the Steps 1 and 2 of Algorithm B and and Step 1 of Algorithm C. In addition to cropping the images to a 224x224px resolution, a copy of the original image is created that is used in Step 3 (rotation calculation). The cropped version is passed to the CNN used in Step 2.

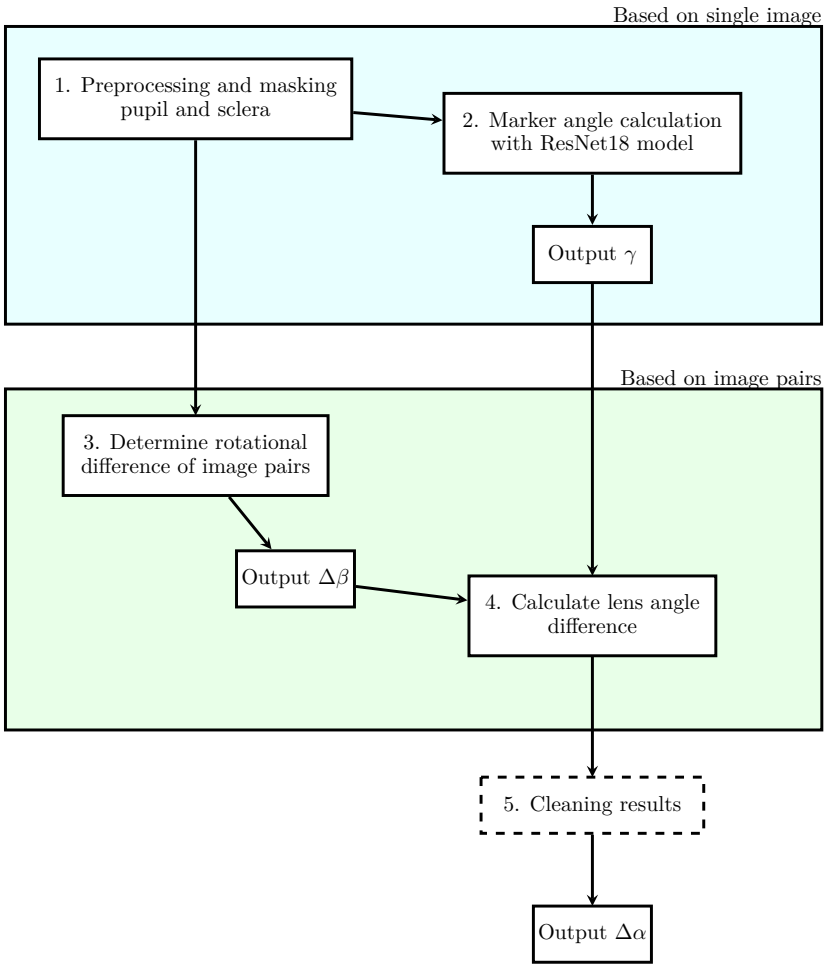

Figure 5.4: Algorithm D: Calculating $\Delta\alpha$ with a DL based algorithm. Steps 1 and 2 are based on single images, steps 3 and 4 are based on pairs of images. Step 5 (cleaning) is optional.

Step 2 | Marker angle calculation with ResNet18 model

- Input: Preprocessed images in a 224x224px resolution and optional additional channel
- Output: Marker angle γ for each image
- Detailed description: Sections 4 and 5.7.

This step is equal to Step 3 of algorithm B. The trained and adapted ResNet18 CNN, that performs best in algorithm B, is used to calculate the marker angle γ on test images.

Step 3 | Determine rotational difference of image pairs

- Input: Preprocessed and masked image pairs of the same patient
- Output: Transformation matrix for each image pair describing the rotation, being equivalent with specifying the image rotation angle $\Delta\beta$
- Detailed description: Section 3.6

This step is equal to Step 2 of Algorithm C.

Step 4 | Calculate lens angle difference

- Input: Transformation matrix (specifying $\Delta\beta$) and calculated marker angles γ_{I1} and γ_{I2}
- Output: Lens angle differences $\Delta\alpha$ on image pairs
- Detailed description: Section 3.7

Images are always compared pairwise; $\Delta\alpha$ can be computed from γ_{I1} and γ_{I2} of the individual images and $\Delta\beta$ by using formula 5.3.

Step 5 | Cleaning by comparing images (optional)

- Input: Lens angle differences $\Delta\alpha$ on image pairs and marker angles γ_{I1} and γ_{I2} of individual images
- Output: Cleaned lens angle differences
- Detailed description: Section 3.8.1

Cleaning is the process of removing wrongly calculated angle differences by applying pertinent information about them. One method, named intra-timepoint cleaning, relies on the fact that the rotation of a lens does not change much in the course of the same examination; therefore, all images that were taken at the same timepoint should have the values of lens angles close to each other. If α of some images differs significantly from their mean value, then they are removed.

5.4.2 Experiment

The experiment uses algorithm D without cleaning, to calculate $\Delta\alpha$. The goal of this algorithm is to evaluate the prediction accuracy of the DL dominated lens angle difference calculation.

The quality of the lens angle difference $\Delta\alpha$ is evaluated by comparing it to the corresponding labels $\phi_{\Delta\alpha}$. The resulting error is called $\epsilon_{\Delta\alpha}$, which is calculated based on the Formula 6.4. Moreover, it is analyzed if certain combinations of timepoints have a higher accuracy than others. Additionally, a patient specific analysis is performed, to explore possible weak points and limitations of the given algorithm.

5.5 Data

5.5.1 Images and videos

The data used in the experiments consist of pictures and videos of 130 patients. The videos were taken directly after the surgery and are saved in an interlaced 1280x720 pixel format with a color depth of 24 bits. The pictures were taken some time after the surgery (1 hour, 1 week, 1 month, 4 months/6 months) and are present in a 1720x1143 TIF format with a color depth of 24 bits. For every patient and timepoint, there are 1 to 5 unique images available. The intra-operational videos vary in length, most of them being approximately 1 minute long. The pictures were taken with a Nikon D70s camera, connected to a special macroscopic objective.

The data contains images of the human eye (left and right), lit with a slit-lamp. All of the images give an unobstructed view of the sclera and the dilated pupil containing the artificial lens (for an example, see Figure 1.2). In the videos, several surgical tools are visible that at some point obstruct the view of the eye and its features.

5.5.2 Labels

The manually measured angle difference labels provided by medical professionals (denoted by $\phi_{\Delta\alpha}$) are given in a comma-separated format, where each line represents a patient. The structure of the table of labels can be seen in Table 5.1. In each line the ID of the patient and the lens angle difference for all possible timepoints are given. The timepoint at operation, 1 hour, 1 week, 1 month, 6 months after the operation is denoted by OP, 1h, 1w, 1m, 6m respectively. For example, the values in 1m_to_6m describes the lens angle difference for an image pair at 1 months and 6 months after the operation. Non-sucessive timepoint combinations, such as 1h to 1m, are not directly present in the label file, but they can be calculated by summing values that follow each other. For example, 1h to 1m can be calculated by adding the value for 1h_to_1w and 1w_to_1m.

Another type of label that was manually created by the author (not by medical professionals) for this work are the lens marker labels (denoted by ϕ_γ). They are only used in evaluating the prediction accuracy of algorithms A and B. Each image has its own label,

ID	OP_to_1h	1h_to_1w	1w_to_1m	1m_to_6m	...
001	1.54	0.98	0.90	0.19	...
002	0.25	0.57	0.66	0.96	...
⋮	⋮	⋮	⋮	⋮	

Table 5.1: $\phi_{\Delta\alpha}$ values as given by medical professionals (partial data shown).

as it can be seen in Table 5.2. The ϕ_γ labels were generated with a custom program that allows a human grader to pick two points on an image. The script automatically calculates the slope, intercept, and angle of a line through the two points. These two points must be the markers of the lens, preferably, points on opposite sides of the lens. The results are then stored in a .csv file. Note, that the coordinate system used to calculate the lens angles have its origin in the top left corner with the y axis being positive in the down direction. Therefore, if the angles have to be interpreted in a standard Cartesian coordinate system, the sign of the angles in the .csv file must be multiplied by -1.

Image name	Angle in °
$001\_OD\_OP\_1.tif$	61.35
$001\_OD\_1h\_1.tif$	−18.58
$002\_OS\_OP\_1.tif$	90.0
⋮	⋮

Table 5.2: ϕ_γ values generated by manual process (partial data shown).

The labels discussed previously can be used to derive the labels that describe the image rotation angle (denoted by $\phi_{\Delta\beta}$). This calculation is given in Equation 5.1. Based on this equation, a label can be created for each image pair I1 and I2. A sample of the calculated values is visible in Table 5.3.

Image 1	Image 2	Angle in °
$001\_OD\_OP\_1.tif$	$001\_OD\_1h\_1.tif$	−78.29
$001\_OD\_OP\_1.tif$	$001\_OD\_1h\_2.tif$	−76.8
$001\_OD\_1h\_1.tif$	$001\_OD\_1m\_1.tif$	3.61
⋮	⋮	⋮

Table 5.3: $\phi_{\Delta\beta}$ labels calculated with formula 5.1 (partial data shown).

5.5.3 Data Evaluation

This section provides an analysis of the labels of the medical professionals $\phi_{\Delta\alpha}$. This gives an insight in the magnitude of the lens angle difference between the timepoint pairs and enables to determine the interval where the rotation of the lens is most probable.

The distribution for the lens angle difference $\phi_{\Delta\alpha}$ can be seen in Figures 5.5 and 5.6. It can be observed, that for all successive timepoint pairs, 90% of the labels are inside an interval from $-2°$ to $2°$, indicating that the lenses of most patients have not rotated significantly. This interval gets smaller the later a timepoint combination is (e.g. OP_1h has 90% of measurements inside an interval of approx. $-1.9°$ to $1.9°$, whereas 1m_6m has an interval of $-0.83°$ to $0.83°$). This indicates that the lens is more susceptible to rotation in the interval after the operation, than at later timepoints. The highest lens angle difference arises at the timepoint pair "1h_1w". This timepoint pair has very few images pairs (approx. 3.8%) that have a label that is outside the interval of $-3°$ to $3°$. For all subsequent successive timepoint pairs, all labels are inside this interval.

The histograms shown in Figure 5.6 can be understood as being derived by summing the labels of successive timepoint pairs. For example, 1h_6m can be constructed by summing the values for each patient of the successive timepoint pairs 1h_1w, 1w_1m and 1m_6m. Due to this summing, the non-successive timepoints have a higher variance. In addition, all non-successive combinations that use 1h_1w in their summing, have at least 4.6% of the labels outside of the interval of $-3°$ to $3°$.

Timepoint pair	Mean	Mean of absolute values
OP_1h	0.05	0.86
OP_1w	-0.34	1.22
OP_1m	-0.38	1.30
OP_6m	-0.38	1.33
1h_1w	-0.39	1.06
1h_1m	-0.40	1.27
1h_6m	-0.43	1.22
1w_1m	-0.03	0.47
1w_6m	-0.03	0.49
1m_6m	-0.00	0.38

Table 5.4: $\phi_{\Delta\alpha}$ distribution for timepoint pairs

In Table 5.4 it is shown that the difference between OP and 1h is small; so the interval where the rotation of the lenses is most probable starts later than OP. It can be concluded from table 5.4 that the interval with the highest rotation of lens is "1h_6m".

5.5.4 Limitations

The data provided includes only pictures of eyes with a certain type of lens. Images of other lens types with other types of markers and lens haptics are not present in the data

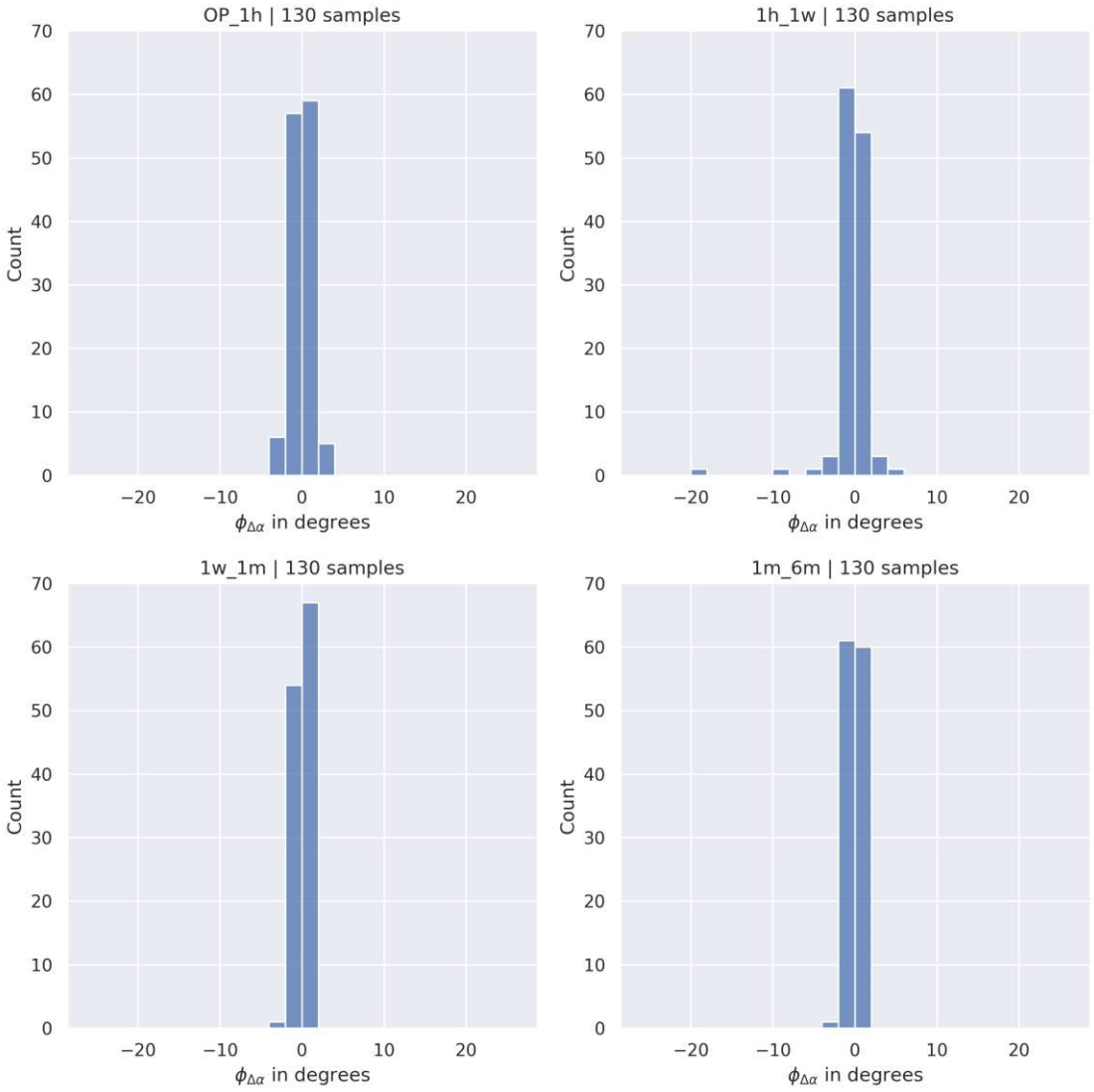

Figure 5.5: Distribution of label data $\phi_{\Delta\alpha}$ per successive timepoint combination - bin-width 2°

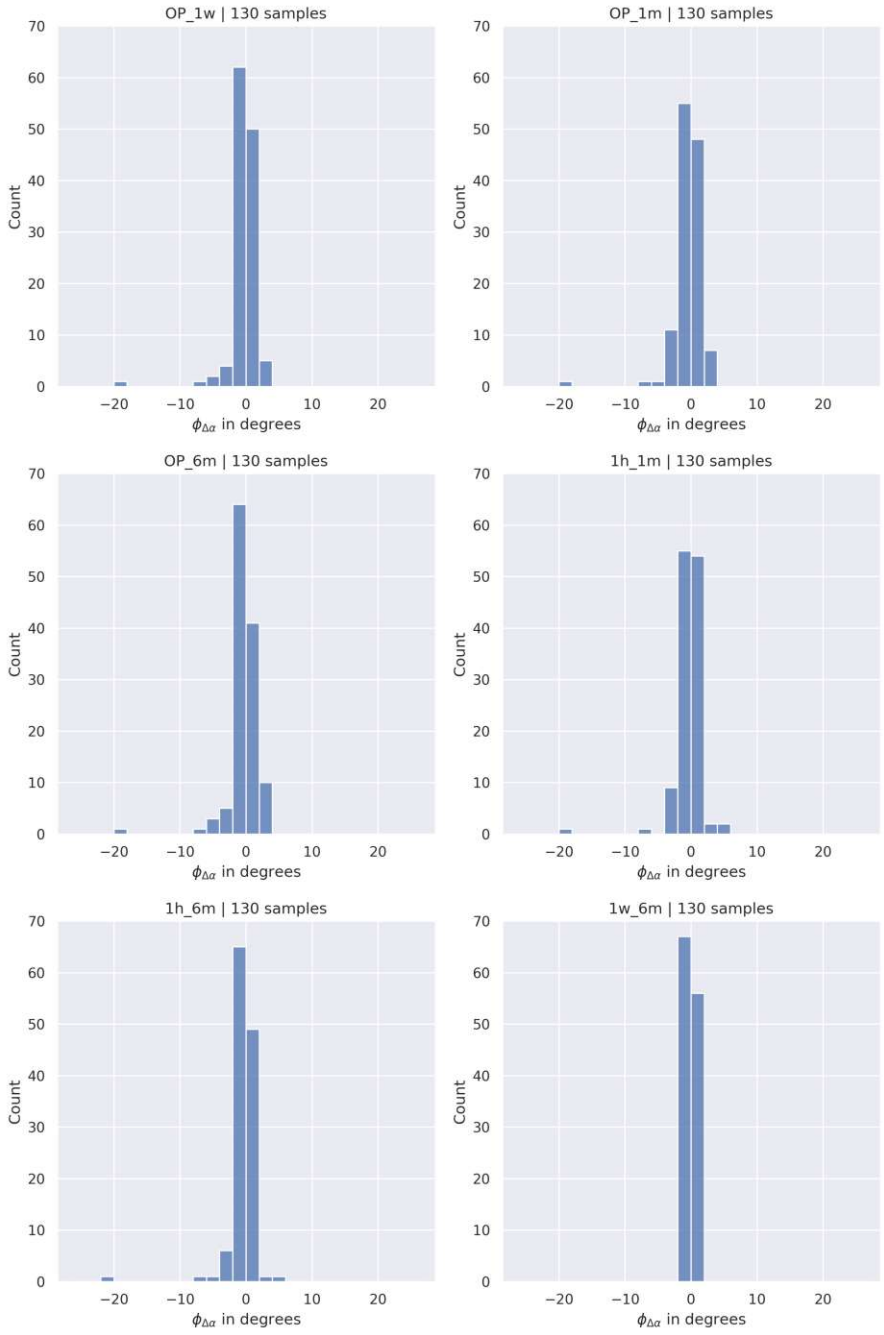

Figure 5.6: Distribution of label data $\phi_{\Delta\alpha}$ per non-successive timepoint combination - bin-width 2°

set. In addition, the same microscope and camera were used for all provided images not taken during surgery (marked "1h", "1w", "1m", "6m"). They were shot at the same resolution with the same lighting and camera settings. The intra-operational videos (timepoint "OP") were shot under different lighting conditions and camera settings than all other pictures. Furthermore, some manual measurements could not be performed due to poor visibility of lens features or image quality. They are set to a NaN value in the label file.

5.6 Parameters for visual computing steps

In this section, all parameters for steps that utilized a visual computing algorithm are listed. For each of them, the Table 5.5 provides a short content description and a reference to the section/table, in which it is explained.

Step	Context	Reference
Masking	Used in finding pupil and sclera mask	Section 3.2 and Table 3.1
Triplet finding	Used in calculating possible triplets	Table 3.2
Line candidate calculation	Used for filtering triplet candidates	Table 3.3
Rotations synchronization of image pairs	Used for aligning a pair of images to each other	Table 3.4
Cleaning	Used in enhancing result in post-processing	Section 3.8

Table 5.5: Parameters for visual computing processing steps

5.7 CNN model parameters

A CNN model is used in Algorithm B and D as part of the processing pipeline. Specifically, a modified ResNet18 network was selected (see Section 4.4). To extract the image area containing the pupil and prepare the training set and test set, several preprocessing steps are performed, ensuring that the images and targets are prepared in the correct format (see Section 4.5). The parameters under which the experiments were performed can be seen in Table 5.6. The preprocessing and cleaning parameters are identical to those used in the CV algorithm.

5.8 Python libraries used

The most important Python libraries and their versions are listed in Table 5.7.

Parameter	Description
CNN Network Architecture	modified ResNet18
Batch Size	Fixed at 128 for all experiments, due to hardware limitations.
Optimizer Type	Adam used for all experiments.
Loss function	Huber loss with a δ of 10.
Training iterations	50
Number of augmentations	128 per image

Table 5.6: CNN parameters

Library name	Version	Description
pytorch	2.3.1+rocm6.0	Machine learning library
torchvision	0.18.1+rocm6.0	Computer vision based machine learning library
numpy	2.1.0	Library for efficiently processing matrices and arrays
matplotlib	3.9.1	Plotting library
opencv-python	4.10.0.84	Computer vision library
pandas	2.2.2	Data analysis and management library
scipy	1.14.0	Fundamental optimization and interpolation library

Table 5.7: Essential python libraries used

5.9 Hardware

The hardware used for conducting the experiments is specified as:

- CPU: AMD Ryzen 7 5800X
- RAM: 32GB DDR4 SDRAM
- GPU: ASUS DUAL RX 6700XT
- SSD: 128GB+ M.2 SSD

CHAPTER 6

Results and discussion

This chapter is structured according to the algorithms and the corresponding experiments. After showing the results of the experiments, a comparison between the algorithms is given, and questions related to those comparisons are answered.

A test set of the CNN model including 23 patients is used to interpret the results of experiments A, B and D, so that they can be compared to each other. These patients were randomly chosen; the patient's data is consistent across all experiments and consists of the same images and labels in all experiments. Utilizing the same set of patients enables one to compare the performance of the algorithm variance, i.e. the experiments. The number of images present per patient at the different appointments (timepoints) is given in Figure 6.1. Additionally, the data of all 130 is evaluated for experiment A and C.

The following experiments use the lens angle difference error $\epsilon_{\Delta\alpha}$, which is given by

$$\epsilon_{\Delta\alpha} := |\Delta\alpha - \phi_{\Delta\alpha}| \tag{6.1}$$

The image rotation error $\epsilon_{\Delta\beta}$ describes the error at measuring the image rotation angle and is given by

$$\epsilon_{\Delta\beta} := |\Delta\beta - \phi_{\Delta\beta}| \tag{6.2}$$

Finally, the marker angle error ϵ_{γ} describes the error when measuring the marker angle and is given by

$$\epsilon_{\gamma} := |\gamma - \phi_{\gamma}| \tag{6.3}$$

	OP	1h	1w	1m	6m
015	5	5	3	3	3
017	5	3	3	3	3
026	5	3	2	3	3
027	5	4	3	2	3
028	1	3	3	3	5
034	0	3	3	3	3
037	0	5	3	7	5
054	2	4	3	4	3
063	0	5	3	2	3
071	5	4	3	3	3
073	1	5	2	3	4
077	0	2	2	3	3
080	5	5	3	4	2
081	5	2	4	3	4
089	1	4	3	0	2
102	5	3	3	4	3
109	5	3	3	3	2
111	0	4	2	3	2
120	5	4	3	3	4
121	5	3	5	4	4
122	5	3	3	3	3
123	1	0	3	4	2
124	5	4	2	3	3

Figure 6.1: Image count per patient per timepoint for limited test-set

Standardization of angle difference from $-90°$ to $90°$

To evaluate the error of an output, a specific metric for the given problem has to be introduced. Due to dealing with lines located on a circle, a standard method has to be adjusted. In this thesis, the absolute error is taken as the basis for an output o and a target ϕ:

$$\epsilon = |o - \phi| \tag{6.4}$$

The problem is that this simple difference between o and ϕ is not always the smallest possible angular difference between intersecting lines. Due to the fact that o and ϕ are angular values that represent the slope of a line across the lens (a circle), there exist mirrored angles that can be interpreted as describing the same line (see Figure 6.2). To calculate the smallest possible angle, these mirrored angles have to be calculated by shifting them by 180 degrees:

$$\begin{aligned} o_m &= (o + 180°) \mod 360° \\ \phi_m &= (\phi + 180°) \mod 360° \end{aligned} \tag{6.5}$$

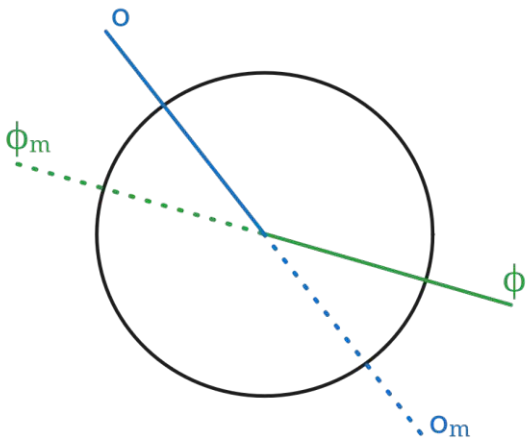

Figure 6.2: Calculated angle o and labelled angle ϕ and the corresponding mirrored versions o_m and ϕ_m

To obtain the smallest possible angular difference, all possible combinations would have to be explored. This can be simplified by not checking combinations that would yield already calculated values, so the following formula is applied:

$$\epsilon = min(|o - \phi|, |o - \phi_m|, |\phi - o_m|, |o_m - \phi_m|) \tag{6.6}$$

This implies that the absolute error can be at most 90°, because this is the maximum angle difference of two intersecting lines. The result of formula 6.6 can be used as input for the Huber-loss function.

Evaluation metric

To score a multitude of measurements at once, the mean absolute error (MAE) is introduced. It is defined in equation 6.7, where n denotes the number of samples and epsilon is the error defined above.

$$MAE := \frac{1}{n} \sum_i \epsilon_i \tag{6.7}$$

Another tool to evaluate the different algorithm are histograms, which denote the distribution of the values and errors. In combination, MAE and histograms can describe distributions in detail and provide a straightforward number to compare the different algorithms. A key benefit of these evaluation metrics is that they work with any kind of distribution, in contrast to other commonly used evaluation values like t-values or z-scores, which require a underlying normal distribution. In addition, the metrics can be

easily understood in the medical context, since absolute deviations are used to determine the change in lens rotation.

6.1 Algorithm A: Computer vision based marker angle calculation

The prediction accuracy was assessed by applying algorithm A to all labeled images. The evaluation was performed individually for each timepoint. The fraction of images with an error ϵ_γ below 3° can be found in Table 6.1, alongside the number of images processed and the corresponding mean absolute error. The histograms describing the distribution of γ per timepoint can be seen in Figure 6.3.

	OP	**1h**	**1w**	**1m**	**6m**
$< 3°$	0.34	0.41	0.57	0.54	0.43
$> 3°$	0.66	0.59	0.43	0.46	0.57
Images	372	384	370	366	343
MAE	29.96	26.06	20.75	19.86	21.32

Table 6.1: Ratio of ϵ_γ errors smaller and larger than 3°, image count and MAE (mean absolute error) for experiment A

It can be observed, that the error ϵ_γ is centered around a MAE of 27.14 (=MAE of all images), with the timepoint having the smallest MAE at 1m and the largest at OP. The distributions in Figure 6.3 look very similar, all having some of the errors below 3° and a large number of errors uniformly distributed above that threshold.

Figure 6.4 shows γ with their respective label ϕ_γ for all images in all timepoints. If the two values coincide, their error ϵ_γ is low and they are located at the image diagonal. If a bias towards a certain angle existed (e.g. the algorithm would have a systematically higher error for a certain marker rotation), the line would not be straight, but have a deviation at certain intervals; due to the fact that this line is (disregarding the noise) very straight, no bias can be observed. One noticeable fact is the gap between 25° and 50° and −40° and −55° on the y-axis. The number of labels is significantly lower in these intervals, therefore a bias in this area cannot be ruled out.

The CNN model test-set consists of the images of 23 randomly selected patients. These patients together with the number of available images of different timepoints can be seen in Figure 6.1. To enable comparison with algorithm B, Table 6.2 describes the fraction of images with an ϵ_γ error below 3° for the CNN model test-set. Therefore, the counts are lower and the results slightly differ from the analysis above.

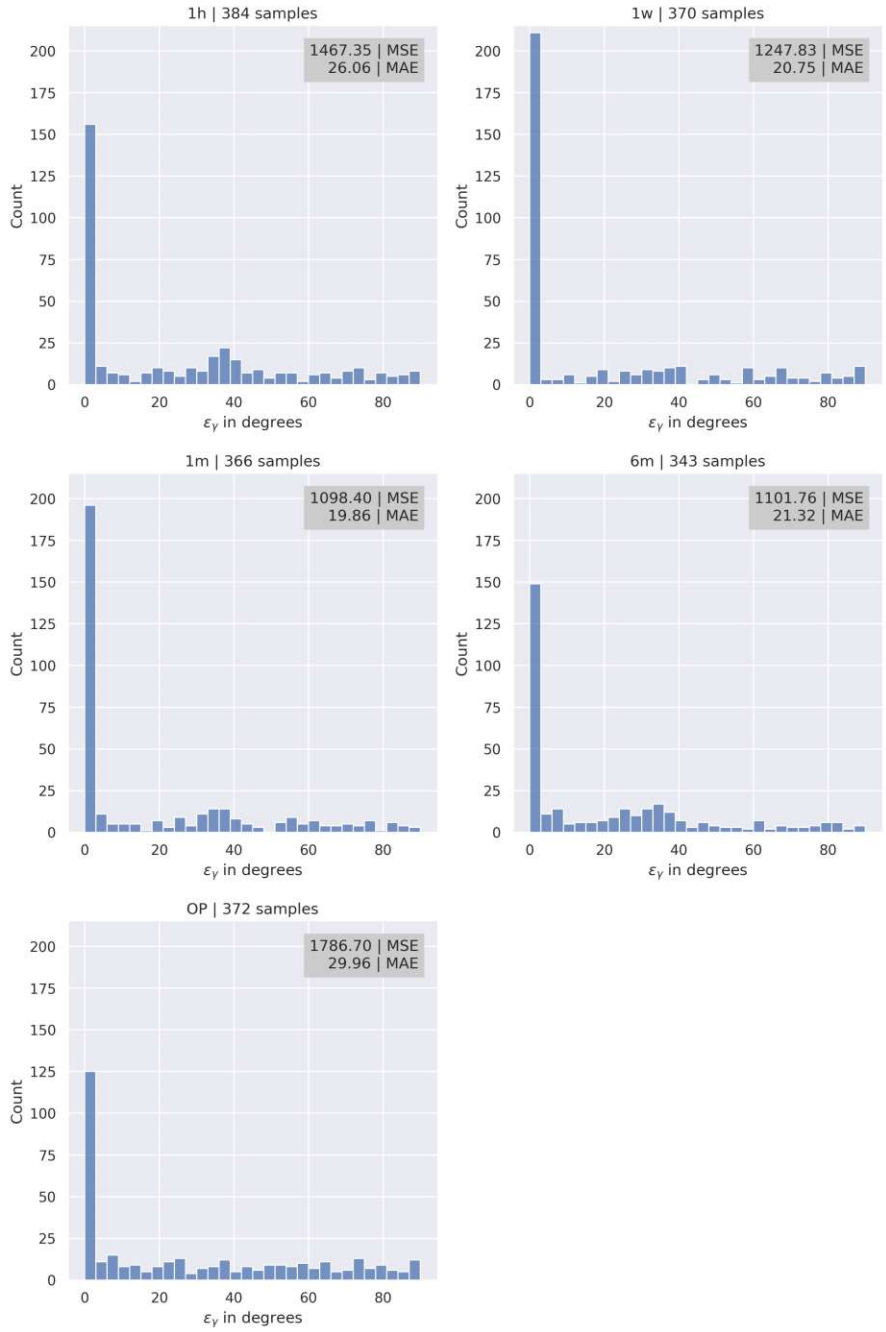

Figure 6.3: ϵ_γ distribution for experiment A per timepoint - bin-width 3°

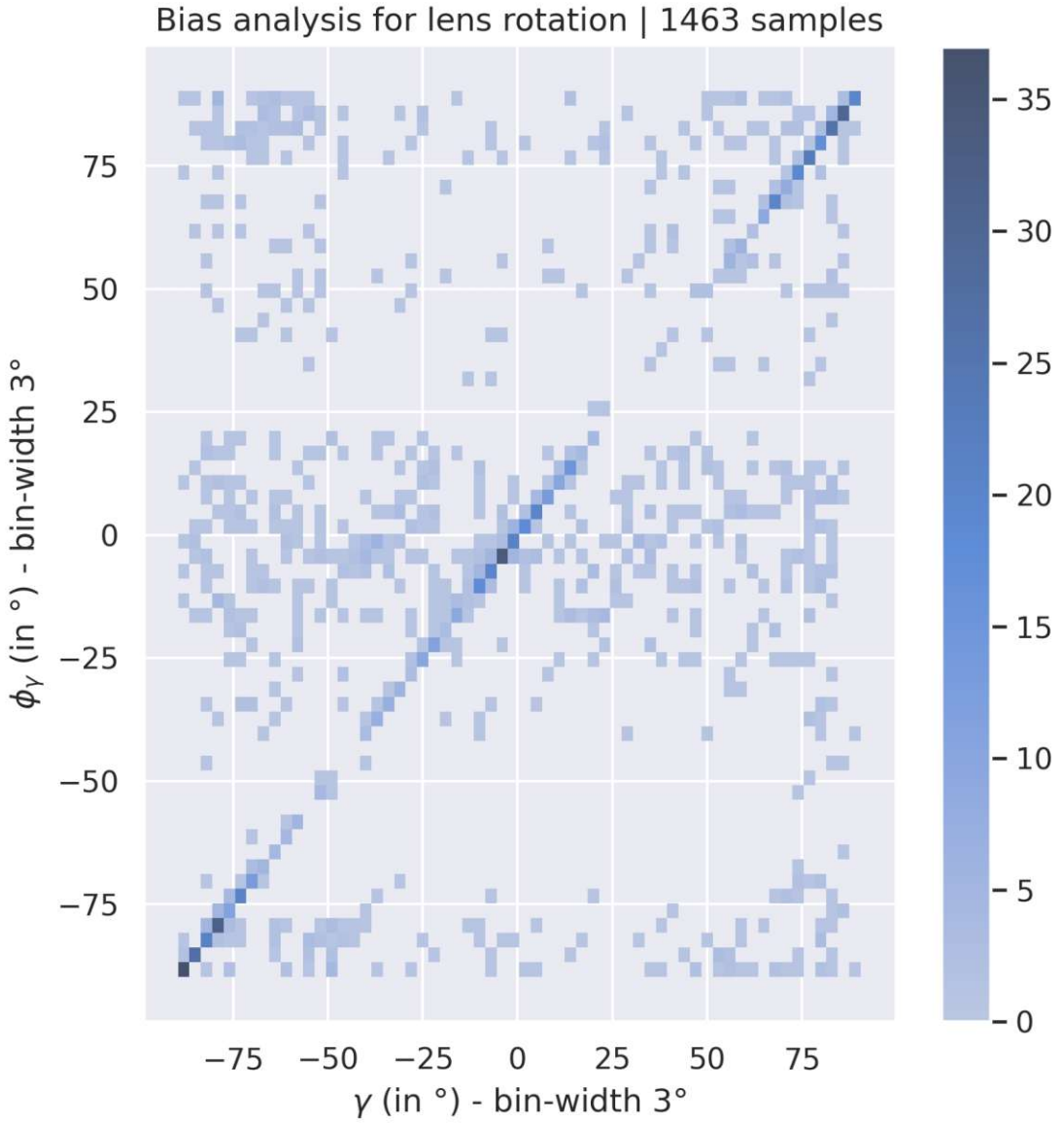

Figure 6.4: 2D histogram comparing the calculated marker angle γ and the corresponding label ϕ_γ for experiment A

	OP	**1h**	**1w**	**1m**	**6m**
$< 3°$	0.46	0.44	0.38	0.39	0.40
$> 3°$	0.54	0.56	0.62	0.61	0.60
Images	70	70	65	72	63
MAE	25.28	23.39	30.88	31.71	24.30

Table 6.2: Ratio of ϵ_γ errors smaller and larger than 3°, image count and MAE (mean absolute error) for experiment A, for limited test set to enable comparison with algorithm B

6.2 Algorithm B: DL dominated marker angle calculation

To assess the prediction accuracy of algorithm B all images contained in the test-set of the CNN model were processed. This is necessary, because the other images were used to train the CNN. Testing the model on the training set does not yield representative results, due to the network potentially recognizing the images and exactly reproducing the wanted labels. As in algorithm A, the evaluation is performed individually for each timepoint.

First, the best performing algorithm version has to be established. The most straightforward way is to compare the mean absolute errors of the algorithm versions to each other. The one, where the error is lowest consistently across all timepoints, is considered the best one and will be further analyzed. As seen in Table 6.3, the algorithm version (3) has the lowest MAE at most timepoints. Therefore, a model with a 4-channel input, that consists of a RGB image and an additional modified 4th channel is used for further evaluation in this section and for evaluating Algorithm D. The additional 4th channel is created by finding feature points with a CV algorithm and processed by removing certain points that are on the rim of the lens, as described in Section 4.5. Table 6.3 also shows that timepoint OP consistently has a much higher MAE, compared to all other timepoints. This is an indication that the images of OP are different from images at all other timepoints; this can be explained by the fact, that OP images were extracted from a video taken during surgery and therefore have different lighting conditions and worse image quality. This drop in quality can potentially be caused by a lower resolution of the videos and temporal artifacts, caused by the video compression algorithms. Due to this inconsistency, all images from OP cannot be directly compared to images at all other timepoints and have to be analyzed separately. Therefore, they are excluded from further analysis in Algorithm B.

The fraction of measurements with an error ϵ_γ below 3° and 4° can be found in Table 6.4, alongside the number of images processed and the corresponding mean absolute error, per timepoint. The histograms describing the distribution of γ per timepoint can be seen in Figures 6.5 and 6.6.

It can be observed, that the error ϵ_γ is approximately in the magnitude of 3°. Images of timepoint 1h have a slightly elevated MAE of 3.46, all other images have a consistent

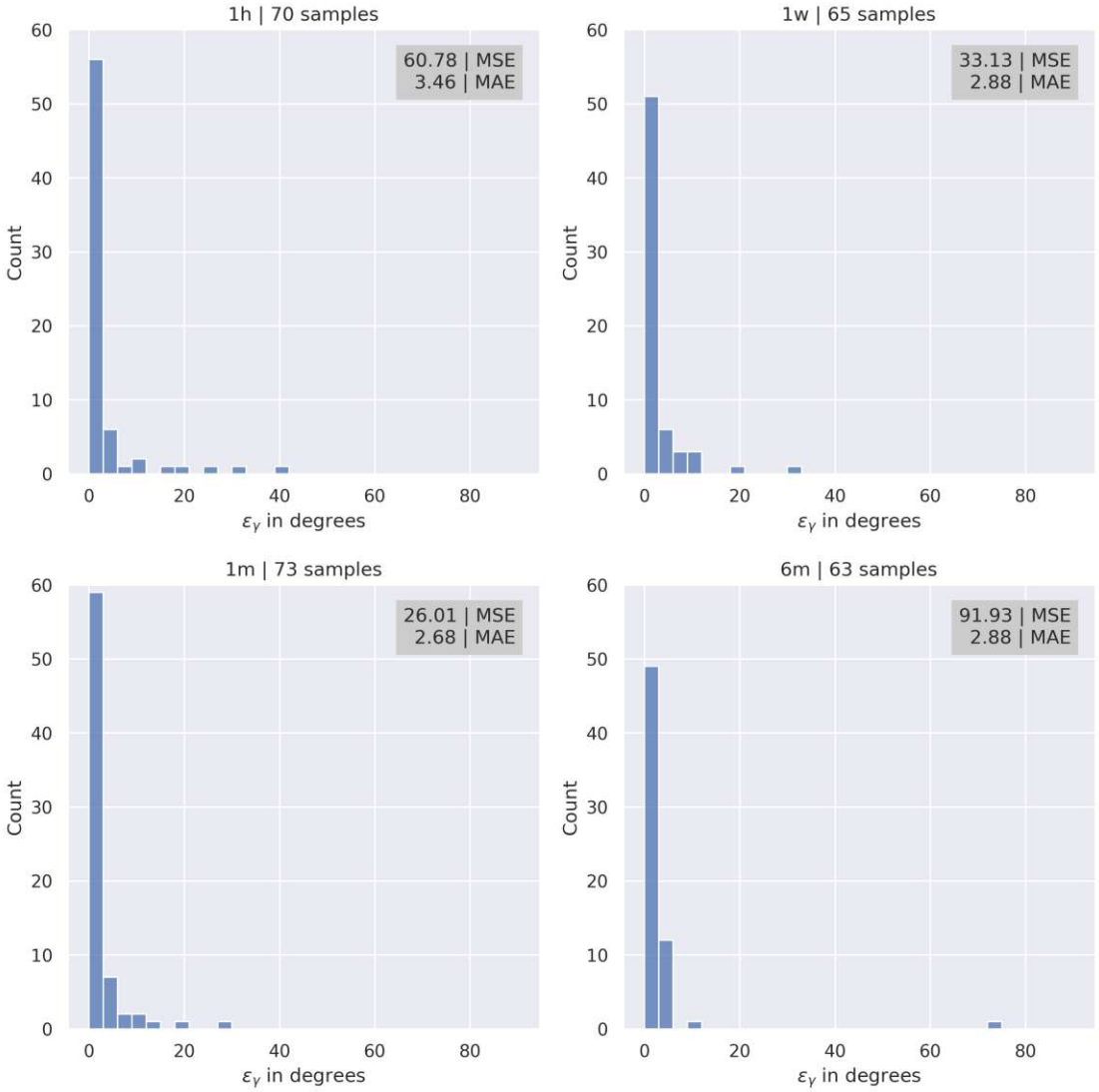

Figure 6.5: ϵ_γ distribution for experiment B, method (3) per timepoint - bin-width 3°

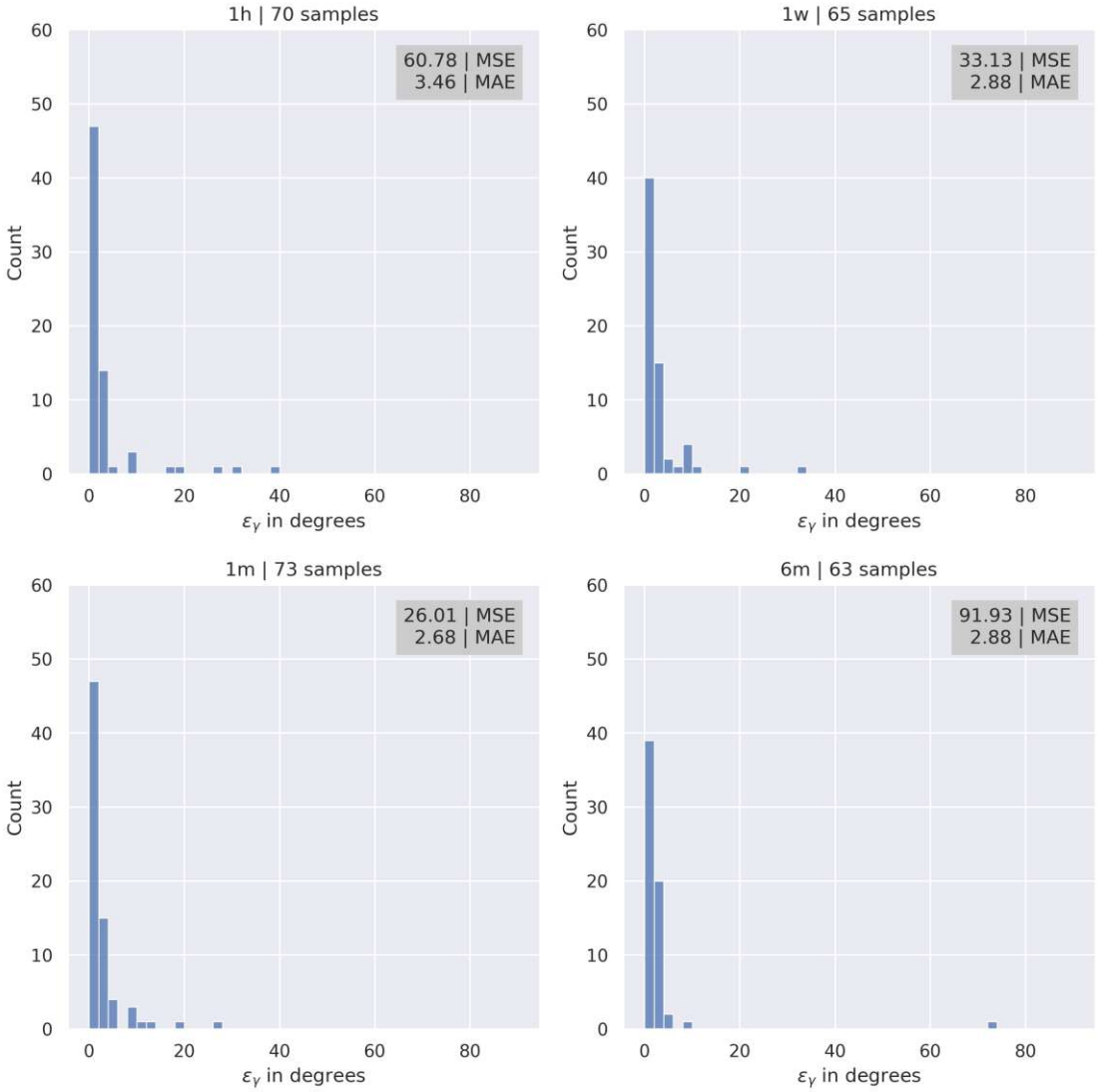

Figure 6.6: ϵ_γ distribution for experiment B, method (3) per timepoint - bin-width 2°

Method	OP	1h	1w	1m	6m	Total
(1)	53.31	29.63	26.98	24.92	25.49	32.27
(2)	19.06	3.97	4.86	2.24	4.04	6.92
(3)	10.73	3.46	2.88	2.68	2.88	4.58
(4)	13.02	7.24	6.44	5.1	4.5	7.33
(5)	22.11	6.8	7.14	7.23	7.1	10.19

Table 6.3: MAE values for differently prepared training data and varying ResNet18 structure. The methods referenced can be found in Section 5.2

	1h	**1w**	**1m**	**6m**
$< 3°$	0.8	0.78	0.81	0.78
$> 3°$	0.2	0.22	0.19	0.22
$< 4°$	0.87	0.85	0.85	0.94
$> 4°$	0.13	0.15	0.15	0.06
Images	70	65	73	63
MAE	3.46	2.88	2.68	2.88

Table 6.4: Ratio of ϵ_γ errors smaller and larger than 3° and 4°, image count and MAE for experiment B, method (3)

MAE of 2.68 to 2.88. Table 6.4 shows that approximately 80% of images have an error below 3° for all timepoints and approximately 85% to 94% of images have an error below 4°, depending on the timepoint.

Figure 6.7 shows the measurements γ with their respective label ϕ_γ. For all images of the test set for all timepoints except OP, the error ϵ_γ is low and they are located at the image diagonal. In general, the observed line is straight, but the region where γ ranges from $-90°$ to $-75°$ shows some irregularities. The measurements there are not located on a straight line, but have a higher variance. Due to the wrapping of the angles from $-90°$ to 90°, a larger amount of images have a measurement of slightly lower than 90°, instead of $-90°$. This could point to a systematic issue with marker angles that are measured at that wrapping boundary - due to the error still being rather low, this artifact can be ignored, a further investigation is not needed in this thesis. Furthermore, a gap between 50° and 75° and $-50°$ and $-75°$ on the y-axis is visible. The number of labels is significantly lower in these intervals, therefore a bias in this area cannot be ruled out.

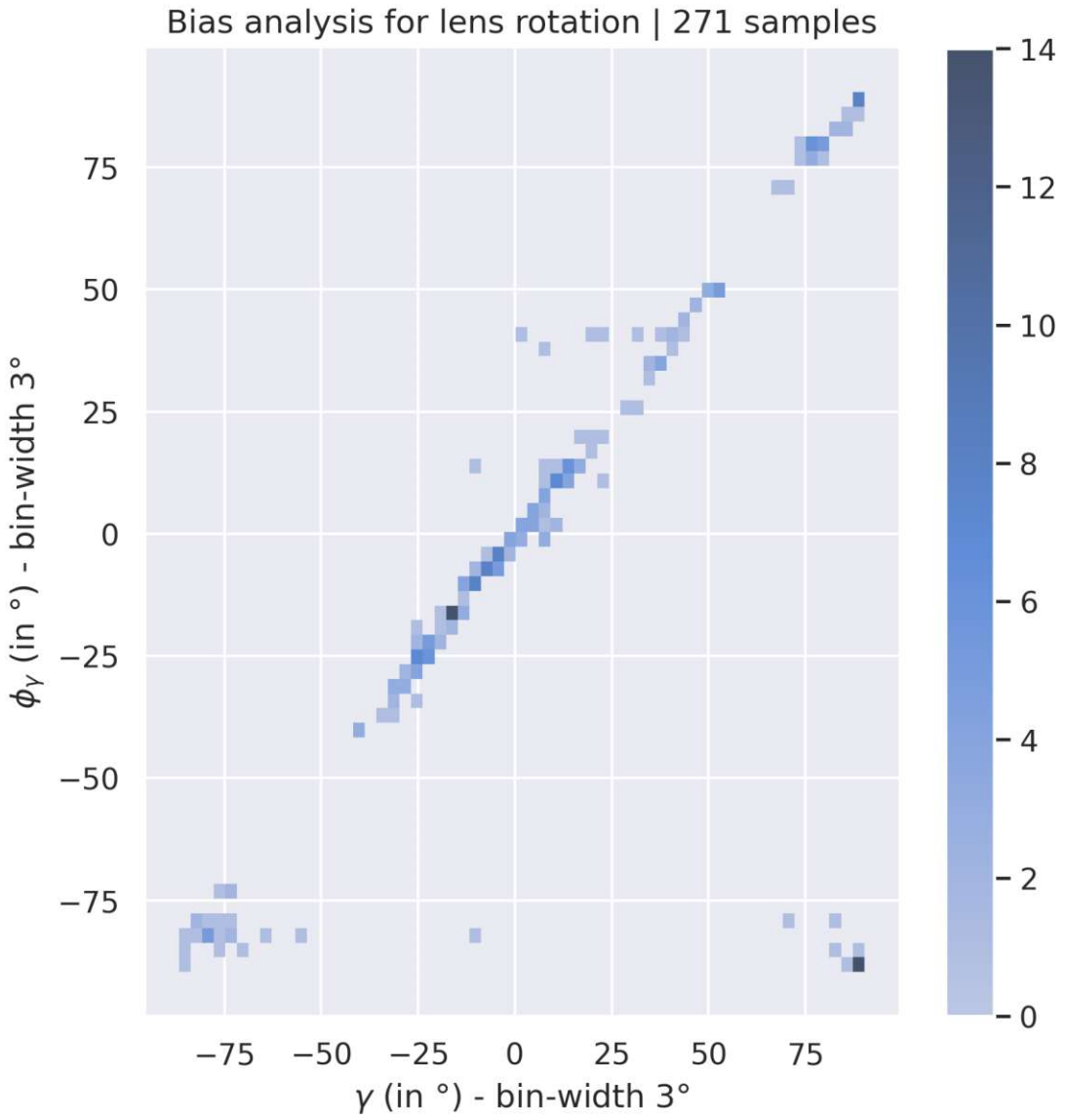

Figure 6.7: 2D histogram comparing the calculated total lens angle γ and the corresponding label ϕ_γ for experiment B, method (3) without images from timepoint OP - bin-width 3°

6.3 Algorithm C: Computer vision based image rotation calculation

Due to investigating pairs of images, the number of measurements created by this algorithm is considerably larger than the previous algorithms. In theory, there exist 13726 unique image pairs - for 7923 a $\Delta\beta$ value was calculated.

1h_1w	1h_1m	1h_6m	1w_1m	1w_6m	1m_6m
1.18	1.5	1.7	0.65	0.89	0.75

Table 6.5: MAE for timepoint pairs without OP

OP_1h	OP_1w	OP_1m	OP_6m
2.02	3.45	2.75	3.3

Table 6.6: MAE for timepoint pairs containing OP

The MAE of $\epsilon_{\Delta\beta}$ can be seen in Table 6.5 and 6.6 for timepoint pairs with and without OP. When comparing Table 6.5 and Table 6.6 the MAE for all timepoint pairs containing OP are larger than all other pairs. Similar to algorithm B, this is an indication that the images of OP are different from images at all other timepoints; this can be explained by the fact, that OP images were extracted from a video taken during operation and have a different lighting condition and worse image quality. This drop in quality can potentially be caused by a lower resolution of the videos and temporal artifacts, caused by the video compression algorithms. This reduces the image clarity of important landmarks and hinders proper functioning of algorithm C. Due to this inconsistency, the timepoint pairs containing OP cannot be compared to all other timepoint pairs. Therefore, they are excluded from any further analysis in Algorithm C. The histograms showing the distribution of $\epsilon_{\Delta\beta}$ can be found in Figure 6.8. The fraction of $\epsilon_{\Delta\beta}$ values below 2° are listed in Table 6.7 for the different timepoint pairs.

	1h_1w	1h_1m	1h_6m	1w_1m	1w_6m	1m_6m
$\epsilon_{\Delta\beta} < 2°$	0.89	0.78	0.79	0.96	0.95	0.97
Images	798	783	699	943	883	836

Table 6.7: $\epsilon_{\Delta\beta} < 2°$ for Algorithm C

All investigated timepoint pairs have a consistent processable count between 699 and 943 and a MAE in the interval of 0.65 to 1.7. The MAE is slightly elevated for timepoint pairs containing "1h", the percentage of images where $\epsilon_{\Delta\beta} < 2°$ is reduced to around 80%. This is also confirmed by the histograms, where those combinations are more spread out than those not containing "1h". A cause for this phenomenon can be inaccuracies in the labels for timepoint 1h in either ϕ_γ or $\phi_{\Delta\alpha}$ from which the $\phi_{\Delta\beta}$ labels are derived.

It can be concluded that if an image pair can be processed and it does not contain images from the "OP" pool, the algorithm produces $\Delta\beta$ measurements that are very

close to the labels $\phi_{\Delta\beta}$. All in all, more than 95% of image pairs can be processed with a corresponding error $\epsilon_{\Delta\beta}$ lower than 2° for the timepoint pairs "1w_1m", "1w_6m" and "1m_6m".

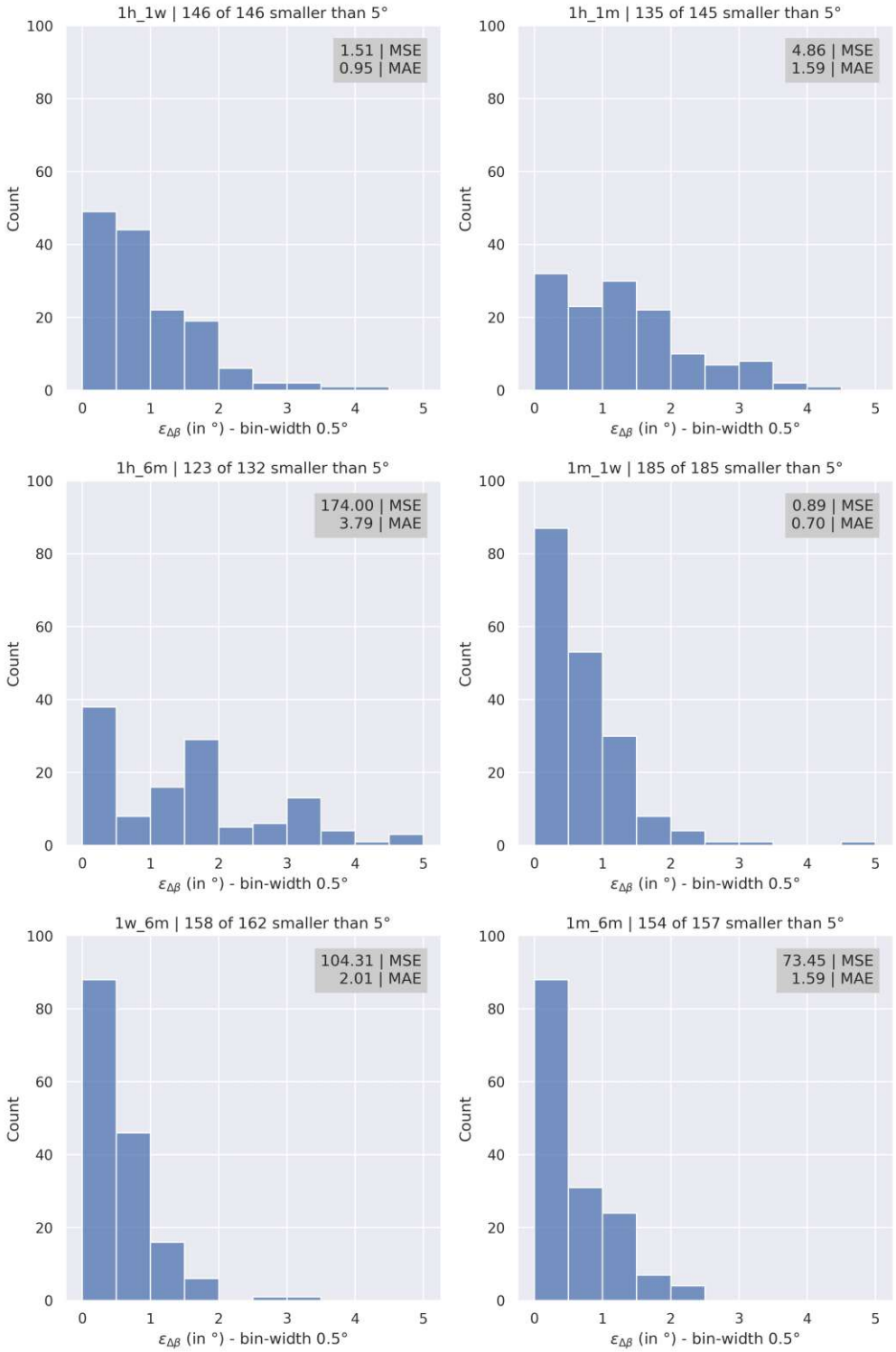

Figure 6.8: $\epsilon_{\Delta\beta}$ distribution for experiment C - timepoint pairs

6.4 Algorithm D: DL dominated angle difference calculation

Similar to algorithm C, pairs of images are investigated in this experiment. In theory there exist 2619 unique image pairs in the test set - for 1487 a $\Delta\alpha$ value was calculated. For reasons explained in algorithms B and C, images that were made during timepoint OP are not analyzed for this algorithm. The MAE of $\epsilon_{\Delta\alpha}$ for different timepoint pairs are listed in Table 6.8.

1h_1w	1h_1m	1h_6m	1w_1m	1w_6m	1m_6m
3.49	5.67	3.52	3.84	3.13	3.08

Table 6.8: MAE for timepoint pairs

	1h_1w	1h_1m	1h_6m	1w_1m	1w_6m	1m_6m
$\epsilon_{\Delta\alpha} < 6°$	0.82	0.79	0.89	0.82	0.88	0.87
Images	146	145	132	185	162	157

Table 6.9: $\epsilon_{\Delta\alpha} < 6°$ with image counts for Algorithm D

Table 6.8 show that all timepoint pairs have a consistent processable count between 132 and 185 and a MAE in the interval of 3.08 to 5.67. The histograms of the investigated timepoint pairs can be seen in Figure 6.9. The fraction of $\epsilon_{\Delta\alpha}$ values below 6° is shown in Table 6.9 for the different timepoint pairs. The MAE is slightly elevated for the timepoint combination "1h_1m", the percentage of images where $\epsilon_{\Delta\alpha} < 6°$ is reduced by 5 to 10%. This is also confirmed by the histograms, where pairs including timepoint "1h" are more spread out than those not containing "1h".

In Figures 6.10 and 6.11 a patient specific analysis can be seen. Figure 6.10 classifies $\epsilon_{\Delta\alpha}$ in three ranges and indicates cases with several irregularities. Figure 6.11 shows error causes for several timepoint pairs, mainly for cases in which $\epsilon_{\Delta\alpha}$ is high. As it can be observed, the main causes of the error is an inaccurate marker angle calculation (ϵ_γ), especially if the corresponding $\epsilon_{\Delta\alpha}$ error is large. $\Delta\alpha$ values that are closer to 3° usually have a larger image rotation error $\epsilon_{\Delta\beta}$, or it is caused by both γ and $\Delta\beta$ being slightly inaccurate. In conclusion, 79% to 89% of the measured lens angle differences $\Delta\alpha$ are 6° or closer to the labels $\phi_{\Delta\alpha}$ provided by medical professionals, depending on the timepoint pair.

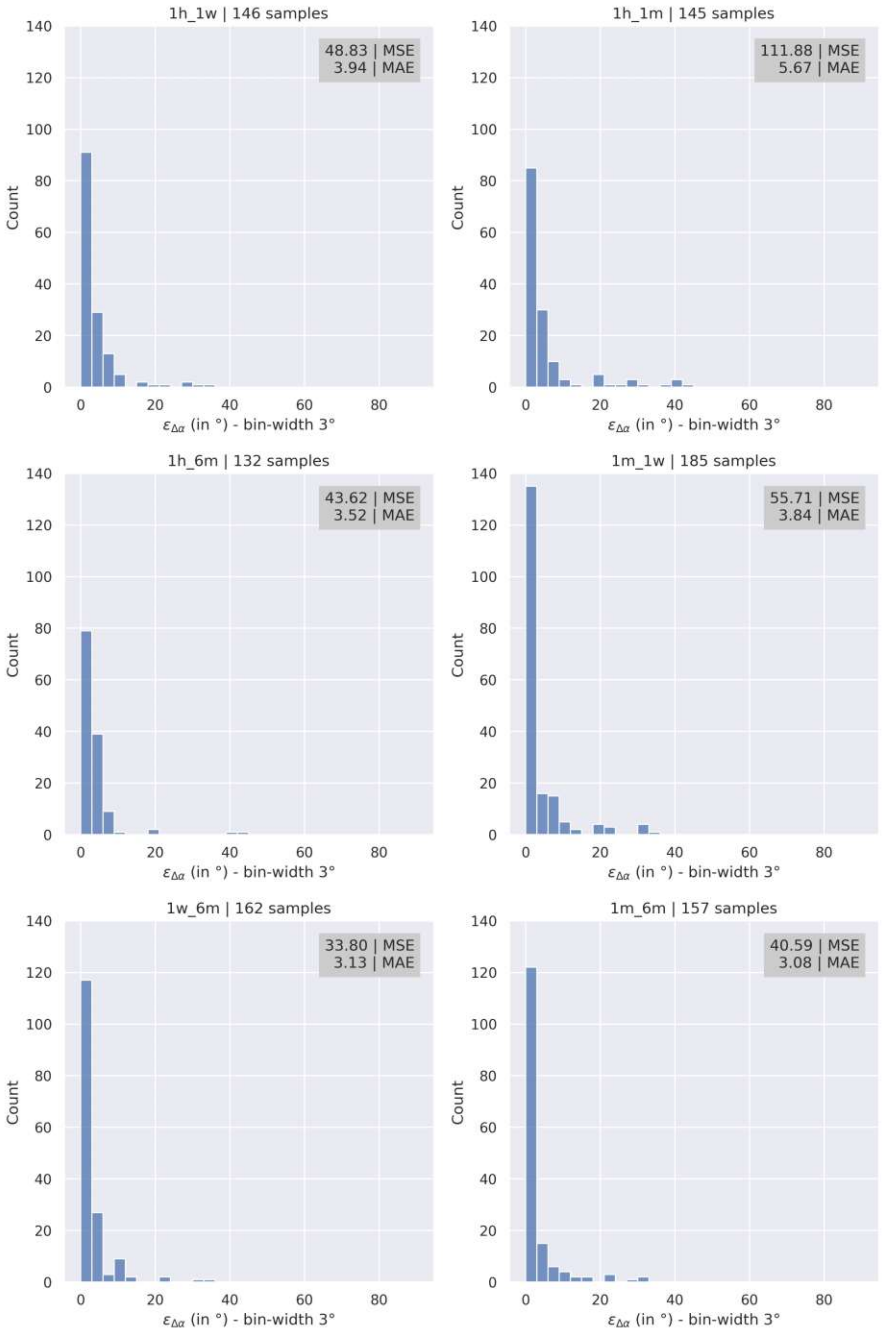

Figure 6.9: $\epsilon_{\Delta\alpha}$ distribution for experiment D - bin-width 3°

	1h_1w	1h_1m	1h_6m	1w_1m	1w_6m	1m_6m
015	-	3.20°	2.22°	-	2.48°	2.71°
017	1.59°	2.17°	3.11°	1.60°	4.27°	3.40°
026	2.68°	3.11°	4.27°	0.47°	1.47°	1.00°
027	1.27°	2.69°	1.88°	1.50°	0.54°	0.89°
028	-	-	-	0.11°	0.35°	0.79°
034	2.28°	0.32°	1.08°	0.87°	0.33°	0.76°
037	6.67°	7.41°	-	6.32°	-	-
054	1.46°	1.20°	0.21°	0.45°	1.43°	1.02°
063	1.37°	4.93°	-	3.55°	-	-
071	2.06°	0.41°	0.65°	1.72°	2.72°	0.94°
073	3.69°	0.92°	0.73°	2.73°	-	-
077	8.51°	8.99°	-	0.60°	-	-
080	2.77°	22.17°	22.44°	21.76°	22.10°	0.77°
081	3.59°	1.98°	-	2.38°	-	-
089	1.52°	?	-	?	-	?
102	0.61°	6.17°	0.80°	0.41°	0.14°	0.33°
109	3.22°	0.67°	0.92°	0.69°	2.11°	1.17°
111	-	0.98°	-	-	-	-
120	0.20°	0.49°	2.23°	1.84°	1.37°	0.11°
121	0.90°	0.53°	0.25°	1.20°	1.00°	0.11°
122	1.64°	0.95°	0.54°	2.94°	1.01°	1.58°
123	-	-	-	0.32°	1.33°	1.22°
124	10.81°	8.73°	-	1.17°	-	-

$\epsilon_{\Delta\alpha} < 3°$	$3° < \epsilon_{\Delta\alpha} < 6°$	$6° < \epsilon_{\Delta\alpha}$	90% Margin of error
$\epsilon_{\Delta\alpha}$	$\epsilon_{\Delta\alpha}$	$\epsilon_{\Delta\alpha}$	±0
			±3°
			±6°
			±12°

Figure 6.10: The range of $\epsilon_{\Delta\alpha}$ for different timepoint pairs for experiment D is indicated by the colors green, yellow and red in increasing order. Red cells mark consistent results that produce a large $\epsilon_{\Delta\alpha}$. White cells can be interpreted as inconsistent $\Delta\alpha$ results or no results for this combination. "-" means no measurement was possible, "?" means no value available due to lack of label data.

	1h_1w	1h_1m	1h_6m	1w_1m	1w_6m	1m_6m
015		?				
017			IR		IR	IR
026		MC	MC			
027						
028						
034						
037	LR (1w)	IR		LR (1w,1m)		
054						
063		MC		?		
071						
073	MC					
077	LR (1w)	LR (1m)				
080		LR (1h)	LR (1h)	LR (1w)	LR (1w)	
081	IR					
089						
102		LR (1h)				
109	LR (1h)					
111						
120						
121						
122						
123						
124	LR (1w,1h)	LR (1h)				

Shorthand	Description	Color
LR	Error caused mainly by γ measurement (timepoint(s) in brackets)	
IR	Error caused mainly by $\Delta\beta$ calculation	
MC	Multiple causes, combination of $\Delta\beta$ and γ error	
?	Error cause not discernible	

Figure 6.11: Error causes for several timepoint pairs for experiment D. See Figure 6.10 for corresponding $\epsilon_{\Delta\alpha}$ errors.

6.4.1 Noticeable images

The goal of this section is to determine, why more errors are accumulated for some patients. For this reason, the patients that consistently show inaccurate or inconsistent results in Table 6.10 are analyzed and potential limitations of the algorithm are derived. Sample images for each patient can be seen in Figure 6.12.

- **Patient 037** has partially obfuscated lens markers, which worsens prediction accuracy. Additionally, the patient has a multitude of small blood vessels, which could affect the ability of the registration algorithm to properly align the images. Larger blood vessels that are more constant in size, as are present in other patients, seem to produce better results.
- **Patient 077** has partially obfuscated markers. This reduces prediction accuracy, and some algorithms cannot process missing markers properly or are dependent on multiple markers being visible.
- **Patient 080** has badly visible markers at one side across multiple images.
- **Patient 089** shows a irregularly shaped pupil, more oval than circular. This breaks the assumption that pupils are always circular in shape, and these pictures are not deemed processable. In addition, it shows poor focus and obfuscated markings, which decrease the quality of the result.
- **Patient 111** has suboptimal results, because markers are partially obfuscated and the number of images is on the lower side.
- **Patient 124** shows a multitude of small blood vessels, which could affect the ability of the registration algorithm to properly align the images.

These facts lead to several limitations that apply to the all algorithms similarly:

- Missing markers severely impact the performance of the algorithms. In addition, there is a physical limit to the potential accuracy achievable with the given image quality and lens type (see Figure 6.13). With all markers perfectly visible, two lines could be drawn that intersect all marker points, but their angles differs by approximately 1.2°. When completely removing one side of the markers, this difference rises to 17.2°, making accurate predictions impossible.
- When the small blood vessels of the conjunctiva are visible due to irritation, image rotation cannot be performed as accurately anymore.
- Irregularly shaped pupils can be a problem for detecting the image mask properly.

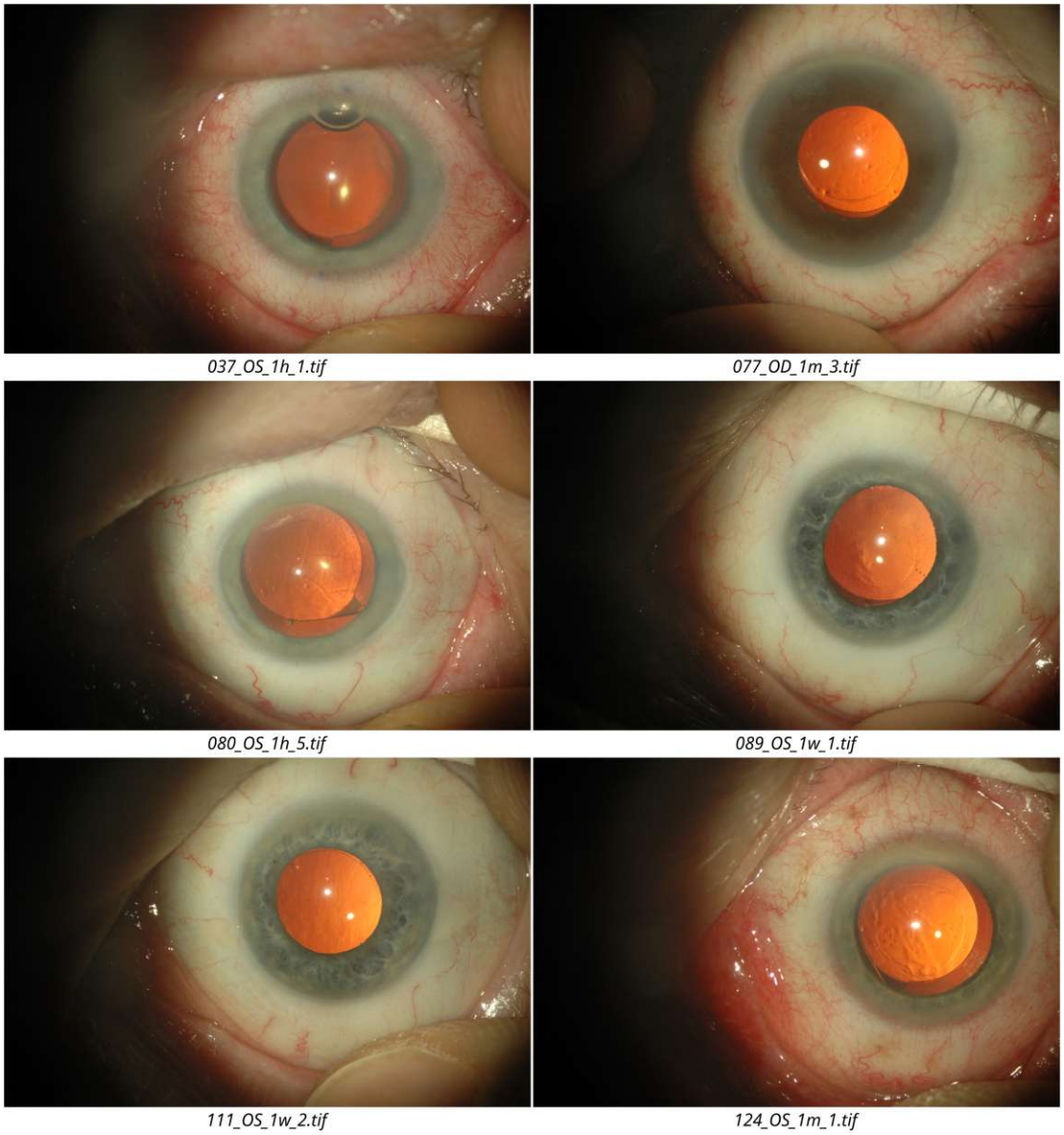

Figure 6.12: Samples of badly processable images

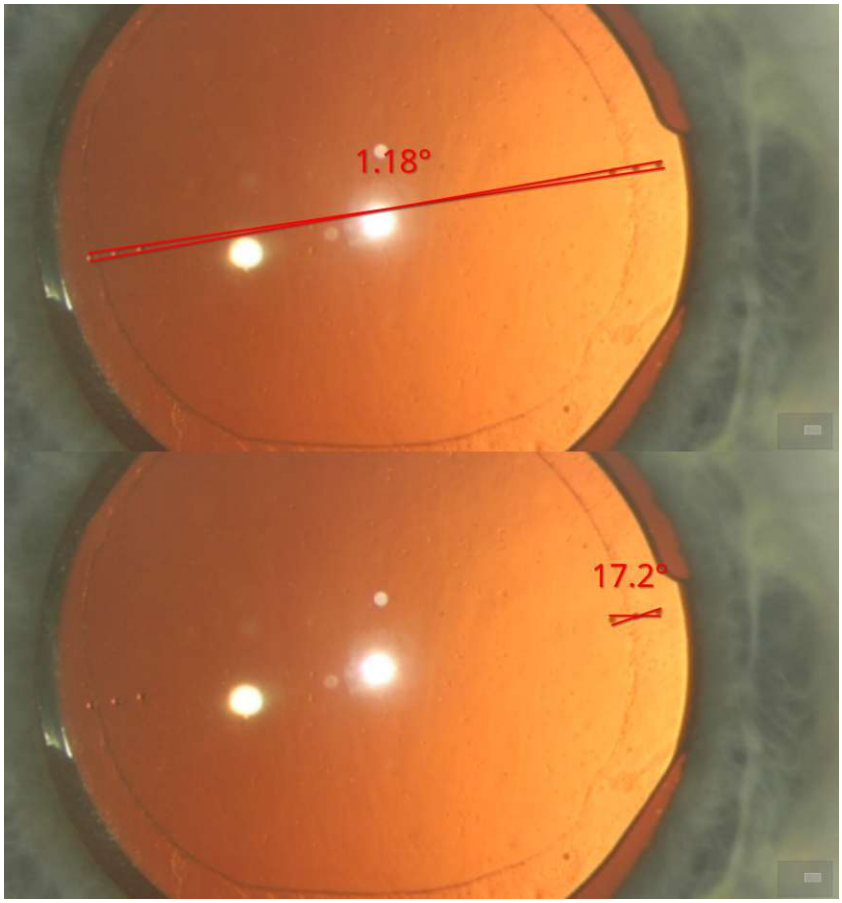

Figure 6.13: Sketch visualizing the maximum angular error of a line drawn that is fully inside the markers. Top shows maximum error using six markers, bottom maximum error using three markers on one side.

6.5 Performance considerations

The performance discussed in this chapter is based on the CNN model test-set. Comparing Tables 6.10 and 6.11 it can be observed that algorithm A and B have roughly the same speed. The main part of time spent in the calculations is always the alignment of the images (algorithm C), which is derived from the fact that all combinations of images are compared, drastically increasing the number of iterations. This can be concluded from Table 6.12. Performance of algorithm D can be derived by combining the runtimes of algorithms B and C.

Step	Mean	standard-deviation	Min	Max
Preprocessing (Step 1-2)	1.43s	0.86s	0.52s	3.57s
Process time (Step 3-5a)	0.67s	0.27s	0.23s	1.17s

Table 6.10: Processing time per patient for experiment A

Step	Mean	standard-deviation	Min	Max
Preprocessing (Step 1-2)	1.52s	0.92s	0.55s	3.6s
Process time (Step 3)	0.93s	0.35s	0.31s	1.5s

Table 6.11: Processing time per patient for algorithm B

Step	Mean	standard-deviation	Min	Max
Image alignment time	60.73s	27.8s	12.08s	121.82s

Table 6.12: Processing time per patient for algorithm C

A benefit of algorithm B that is not utilized in the comparison at hand is the fact that the process time (step 3 of algorithm B) rises slower when more images are added, due to parallelism capabilities of CNN models. This is in contrast to algorithm A, where the runtime scales linearly with each image added. However, it has little impact on algorithm D, which combines algorithm B and C, because the runtime is dominated by the image alignment steps.

The performance of the CNN training used in algorithm B and D depends mainly on the number of iterations performed, but training generally finishes in 8 to 12 hours with the given hardware. In theory this could be sped up significantly with newer and faster hardware, thus only a rough estimate is given for context.

CHAPTER 7

Conclusion

The concluding chapter summarizes the key insights derived from the experiment data and analysis. By reviewing the primary objectives, this chapter aims to answer the research question and address potential limitations and areas for future research. The main parts in this section are composed of comparing the results of algorithms A and B, evaluating the DL dominated algorithm D on predicting lens angle differences and recommending when the algorithms can be used in a practical setting.

7.1 Comparison of computer vision based and DL dominated algorithms for predicting γ

In this section, algorithms A and B are compared to each other. To enable comparison, the accuracy of the γ measurements is used. Focus is placed on the best performing experiments with respect to accuracy (lowest MAE and highest ratio of processable pictures with an error $\epsilon_\gamma < 3°$). Other experiments are mentioned if they have a significant benefit, like e.g. computational speed. In addition, all versions of algorithm B are compared and discussed.

Timepoint OP is excluded from the evaluation since the number of samples processable is much lower and the data is not comparable. Almost all DL dominated algorithms - as discussed in algorithm B and shown in Table 6.3 - outperform the computer vision algorithm A. Therefore, if using the best performing version (3) of algorithm B, it can be said that the DL dominated algorithm has considerably better performance; it has an almost 10 times smaller MAE of approximately 3 and the histograms for experiment B lack the uniformly distributed error component, meaning that most errors are closely grouped around zero. In addition, the histograms demonstrate that the fraction of samples below a ϵ_γ threshold of 3° is around 80% for algorithm B and only around 40% for algorithm A.

Comparing the different versions of algorithm B yields insight, which part of the input is more important for good performance: Version (1) shows, that a 3-channel input of the image without additional feature information performs in the same magnitude as the visual computing based algorithm A. If only the additional 4th channel is used (version (5)), the MAE diminishes, implying that the algorithm extracts key information from the encoded image features. Combining the additional 4th channel with one randomly chosen color channel lowers the MAE slightly (version (4)), but the effect is larger when supplying the CNN with all three color channels and the unaltered additional 4th channel (version (2)). When noise points are removed from the 4th channel, the accuracy further increases leading to version (3) having the lowest overall MAE.

This leads to the summarizing conclusions that:

- Algorithm A is not appropriate for γ prediction due to its low prediction accuracy (approx. 40% of ϵ_γ are less than 3°)
- Algorithm B is better than algorithm A in terms of prediction accuracy
- Version (3) of algorithm B has the highest prediction accuracy, which can be characterized as
  - 78% - 81% of ϵ_γ less than 3° depending on the timepoint
  - Most of the error (85% - 94%) falls below 4°, depending on the timepoint (the best value is 94% for 1m)

Based on the above, algorithm B should be used for computing the marker angle γ in algorithm D.

7.2 Assessing algorithm C for image rotation calculation

Before being able to assess the lens angle differences $\Delta\alpha$ of algorithm D, the image rotation quality $\Delta\beta$ of algorithm C must be investigated. This enables the analysis of the individual components composing the angle difference error $\Delta\alpha$.

Again, timepoint OP has to be excluded since the number of samples processable is much lower and the data is not comparable. In experiment C it can be seen, that $\epsilon_{\Delta\beta}$ for the majority of timepoint pairs have a MAE between 0.65 and 1.7. The fraction of images that have a $\epsilon_{\Delta\beta} < 2°$ is at least 95%, except timepoint pairs where one of the images is taken at timepoint "1h". The percentage drops by approximately 5% to 10% there. This indicates that the error introduced by image rotation is very small for comparisons "1w_1m", "1w_6m" and "1m_6m" and a bit larger for all timepoints containing "1h". This leads to the conclusion that the achieved prediction accuracy with this algorithm is approximately 2°.

7.3 Evaluating of the DL dominated algorithm D to calculate lens angle differences

This section aims to evaluate the prediction accuracy of a DL dominated algorithm D for measuring the angle difference $\Delta\alpha$.

Due to the fact that timepoint OP has a much higher MAE in algorithm B and C, it is excluded from the analysis of algorithm D. The experiment shows, that algorithm D has consistent results when comparing inter-timepoint combinations. It should be noted that images taken directly one hour after operation (timepoint "1h") cannot be processed as accurately as images taken at other timepoints. For either case, a large amount of samples (75% to 84%, depending on the timepoint pairs) have an error that is smaller than 6°. It can be seen from the distribution of the medical labels of the lens angle differences (see Table 5.4 in Section 5.5.3) that the highest lens angle difference arises between 1h and 6m and 1h and 1m. Therefore, these are most appropriate for detecting the rotation of the lens. For these timepoint pairs, the proportion of cases having medical labels outside -2 to 2 degrees and detected angles outside (-2 - 2), (-5 - 5) and (-8 - 8) degrees, respectively, are shown in table 7.1.

	$\lvert\phi_{\Delta\alpha}\rvert > 2°$	$\lvert\Delta\alpha\rvert > 2°$	$\lvert\Delta\alpha\rvert > 5°$	$\lvert\Delta\alpha\rvert > 8°$
1h_1m	10%	55%	20%	10%
1h_6m	11.1%	44.4%	5.6%	0%

Table 7.1: Proportion of pathological cases $\lvert\phi_{\Delta\alpha}\rvert > 2°$ and detected angles $\lvert\Delta\alpha\rvert > 2$, 5 and 8 degrees respectively

It is evident from tables 6.8, 6.9 and 7.1 that the lens angle difference error is smaller for 1h_6m than for 1h_1m. Therefore, we continue the evaluation for the case 1h_6m. The following tables show the number of images with true positive, false positive, true negative and false negative for the cases $\Delta\alpha$ higher than 2, 5 and 8 degrees, respectively.

From these three evaluated cases, the case with $\lvert\Delta\alpha\rvert < 5°$ shows the best performance. It has 0 false positives and only one false negative.

1h_6m	$\lvert\Delta\alpha\rvert < 2°$	$\lvert\Delta\alpha\rvert > 2°$
$\lvert\phi_{\Delta\alpha}\rvert < 2°$	9	7
$\lvert\phi_{\Delta\alpha}\rvert > 2°$	1	1

Table 7.2: Number of cases in the case $\lvert\Delta\alpha\rvert < 2°$

1h_6m	$\lvert\Delta\alpha\rvert < 5°$	$\lvert\Delta\alpha\rvert > 5°$
$\lvert\phi_{\Delta\alpha}\rvert < 2°$	16	0
$\lvert\phi_{\Delta\alpha}\rvert > 2°$	1	1

Table 7.3: Number of cases in the case $\lvert\Delta\alpha\rvert < 5°$

1h_6m	$\lvert\Delta\alpha\rvert < 8°$	$\lvert\Delta\alpha\rvert > 8°$
$\lvert\phi_{\Delta\alpha}\rvert < 2°$	16	0
$\lvert\phi_{\Delta\alpha}\rvert > 2°$	2	0

Table 7.4: Number of cases in the case $|\Delta\alpha| < 8°$

7.4 Practical usability of the algorithm in the medical field

The analysis of algorithm D shows, that in general the algorithm can be used to generate consistent results when measuring the rotational change of a IOL for a patient. Some limitations have to be given: The images cannot be compared to images taken during surgery (timepoint OP) due to consistently worse performance.

For practical usage the timepoint pair 1h_6m is recommended since

- the lens angle difference between OP and 1h is negligible
- 1h_6m has one of the highest lens angle differences, giving the highest chance to detect it
- 1h_6m has a smaller lens angle difference error than the case 1h_1m

The prediction accuracy of algorithm D is approx. 6 degrees, the lens rotation can be considered pathological for this thesis when the lens angle difference is higher than 2 degrees. The best performance for practical usage can be achieved by using algorithm D in a classificator setting above and under the threshold 5°. Using 5 degrees instead of 2 degrees can be explained by the MAE being in the magnitude of 3 degrees (see Table 6.8). As shown in Table 7.3 the achieved classification performance is characterized as:

- Accuracy of 94.12%
- Precision of 100%
- Recall of 50%

Therefore, algorithm D can detect 50% of the pathological cases for the considered image set. However, this is only a preliminary result due to the low number of pathological cases and therefore further investigations are necessary. This is the most important practically usable result of this thesis.

7.5 Discussion of the research questions

This section focuses on explicitly answering Research Questions 1-3. Based on the previous sections, it should provide a differentiated view on the question and give an answer outlining possible drawbacks, limitations, and possible further steps if the conclusion is not definitive.

7.5.1 RQ1: Can the predictive accuracy of a fully automated computer vision based method for assessment of the toric IOL rotation without deep learning achieve a level of quality equivalent to those of a manual data analysis method as performed by medical professionals?

Experiments based on algorithm A have shown that the CV algorithm produces measurements with a large error for more than half of the images. Therefore, the algorithms can neither be used to generate labels to train CNNs, nor directly in an algorithm that calculates the lens angle difference.

The main limitations are the high sensitivity to noise and certain preconditions that have to be met. If only some of the six markers are visible in an image, or the picture was not taken under ideal circumstances, algorithm A (the CV algorithm) fails to produce proper results. This cannot be fixed easily, since the whole algorithm is based on the assumption, that all six markers are clearly visible. Therefore, a CV algorithm has limited potential for the given lens type.

7.5.2 RQ2: Which deep learning (DL) architecture performs best for automatic assessment of toric IOL rotation?

As seen in experiment B, the DL architecture ResNet18 can perform best with utilizing explicit feature extracting, i.e. utilizing all three channels of any given image with additional 4th channel information about the possible location of the lens markers. The main caveat is that the comparisons were conducted only on a very limited range of variants of a single standard model (ResNet18).

7.5.3 RQ3: Can a deep learning dominated algorithm outperform a traditional computer vision algorithm in terms of predictive accuracy for the assessment of the toric IOL rotation?

A deep learning dominated algorithm B outperforms a CV algorithm A, as seen in experiments with algorithms B and A. Additionally, the experiment with algorithm D shows that involving a DL based algorithm B produces results that can be used to estimate the lens angle difference with approximately 6 degrees prediction accuracy.

7.6 Limitations and further research

The DL dominated approach has potential that has not been leveraged yet. Since only ResNet18, a small out-of-the-box CNN, has been explored in this study, it seems reasonable to assume that a different type of CNN model could improve results. As specific next steps, a larger model can be trained when having more test data and faster hardware; more specialized or custom CNN models could potentially increase the quality by utilizing a tailored structure and layer parameters adjusted to the problem at hand.

For this, hyper-parameter tuning is a viable strategy that has not been applied yet. In addition, approaches that have image pairs as input should be explored, because they might improve accuracy.

Another avenue of improvement could be the use of an improved or different image-rotation calculation (image registration) algorithm. The current approach has a low error for processable samples, but cannot produce results for all images. A method that can deal better with slight differences in images could yield more results with a high quality, which would be beneficial for improving algorithms C and D.

Images made with other camera equipment and lighting settings worsen the performance of the algorithms, given the used fixed set of parameters. In the analyzed dataset, this applies to all images made in OP - compared to other images, they have a lower prediction accuracy. A possible approach to improve results for images at timepoint OP is to gather more images that can be used in training a CNN model, more data can be extracted from the video of the operation. In the experiments performed in this thesis, five frames have been extracted at most - depending on the amount of artifacts and length of the video, many more are available. The additional frames would provide more data for CNN training, which can improve the ability of the network to generate results of higher accuracy for intra-operational videos. The prediction accuracy for this timepoint might be limited by the quality of the videos - the motion artifacts and compression algorithm used could prevent better results, a different method of capturing the source videos might be needed.

A different topic that has not been explored yet is the use of adjusted algorithms for lenses of different manufacturers. Especially if more distinct markers are present, the proposed methods could yield reliable results and might come even closer to human level of precision. The adjustments that would have to be done are expected to be rather small, especially when adapting neural networks; therefore, research in this direction seems promising.

7.7 Final remarks

In conclusion, the general approach for finding and comparing the rotation of the toric IOL has been found and some fundamental steps were taken in the right direction. The prediction accuracy of the algorithm can be still significantly improved and there are promising ways for doing that. Such enhancements would decrease the error to a minimum level, so that the algorithms devised can be used for any patient. To accomplish this, more research should focus on improving CNN and optimizing out-of-the-box CV methods.

This work serves as a solid foundation for future research, providing domain-specific entry points into computer vision and deep learning. It introduces various approaches that enable fully automated development of deep learning algorithms for finding the orientation of toric IOLs. The utilized methods are open to improvements and not limited to their

current state. Further development and refinements will lead to software that reduces workload for medical personnel, especially in research where a large number of patients have to be processed. This may lead to faster testing and more thorough and precise analysis of available IOL products, which is beneficial for patients and manufacturers alike.

Ethical considerations

Due to the dataset being identical to the reference [SSS+21], the same ethical considerations apply:

> „Patients referred for monolateral or bilateral cataract surgery were included in the study. The prospective clinical trial included 130 eyes of 68 patients. In all eyes, a nontoric hydrophobic acrylic single-piece IOL was implanted. Patients provided written informed consent prior to the study. All study procedures adhered to the tenets of the Declaration of Helsinki. The local ethics committee of the Medical University of Vienna approved the study (EK 1433/2018) and the study was registered to a public clinical trial registry (NCT03803852).“[SSS+21]

The data was only provided pseudonymized and is treated confidentially. Only randomly chosen images of the eyes are used as reference figures in this study, and the full dataset is not released along with the study.

List of Figures

List of Tables

Bibliography

[AASIK23] Rasha Saad Aldoury, Nadia M.G. Al-Saidi, Rabha W. Ibrahim, and Hasan Kahtan. A new x-ray images enhancement method using a class of fractional differential equation. *MethodsX*, 11:102264, 2023.

[Abb17] Qaisar Abbas. Glaucoma-deep: detection of glaucoma eye disease on retinal fundus images using deep learning. *International Journal of Advanced Computer Science and Applications*, 8(6), 2017.

[ABD12] Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J. Davison. Kaze features. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, pages 214–227, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[Alc24] TX Alcon, Fort Worth. Clareon toric iol, 2024.

[Aph15a] Wikimedia Commons Aphex34. Max pooling, 2015.

[Aph15b] Wikimedia Commons Aphex34. Typical cnn structure, 2015.

[AS11] Pablo F Alcantarilla and T Solutions. Fast explicit diffusion for accelerated features in nonlinear scale spaces. *IEEE Trans. Patt. Anal. Mach. Intell*, 34(7):1281–1298, 2011.

[Bal81] Dana H Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2):111–122, 1981.

[BETV08] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008. Similarity Matching in Computer Vision and Multimedia.

[BGK15] Dibya Jyoti Bora, Anil Kumar Gupta, and Fayaz Ahmad Khan. Comparing the performance of l*a*b* and HSV color spaces with respect to color image segmentation. *CoRR*, abs/1506.01472, 2015.

[BGV14] Inessa Bekerman, Paul Gottlieb, and Michael Vaiman. Variations in eyeball diameters of the healthy adults. *Journal of Ophthalmology*, 2014:1–5, 2014.

[Can86] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8:679 – 698, 12 1986.

[CEL+23] Lorenzo Ciampiconi, Adam Elwood, Marco Leonardi, Ashraf Mohamed, and Alessandro Rozza. A survey and taxonomy of loss functions in machine learning, 2023.

[Com15] Wikimedia Commons. Diagram of four purkinje images, 2015.

[CSS+20] Gulpreet Kaur Chadha, Aakarsh Srivastava, Abhilasha Singh, Ritu Gupta, and Deepanshi Singla. An automated method for counting red blood cells using image processing. *Procedia Computer Science*, 167:769–778, 2020. International Conference on Computational Intelligence and Data Science.

[CTH22] Yao-Mei Chen, Jinn-Tsong Tsai, and Wen-Hsien Ho. Automatic identifying and counting blood cells in smear images by using single shot detector and taguchi method. *BMC Bioinformatics*, 22(S5), December 2022.

[DHB98] G. De Haan and E.B. Bellers. Deinterlacing-an overview. *Proceedings of the IEEE*, 86(9):1839–1857, 1998.

[dJdGN+14] Marcus C. de Jong, Pim de Graaf, Daniel P. Noij, Sophia Göricke, Philippe Maeder, Paolo Galluzzi, Hervé J. Brisse, Annette C. Moll, and Jonas A. Castelijns. Diagnostic performance of magnetic resonance imaging and computed tomography for advanced retinoblastoma. *Ophthalmology*, 121(5):1109–1118, May 2014.

[DM11] Steven B Damelin and Willard Miller. *Cambridge texts in applied mathematics: The mathematics of signal processing series number 48*. Cambridge University Press, Cambridge, England, December 2011.

[DZQY17] Yanyan Dong, Qinyan Zhang, Zhiqiang Qiao, and Ji-Jiang Yang. Classification of cataract fundus image based on deep learning. In *2017 IEEE International Conference on Imaging Systems and Techniques (IST)*. IEEE, October 2017.

[Eur23] Eurostat. Surgical procedures, 2023.

[FB81] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[FR20] Tiago B Ferreira and Filomena Ribeiro. How can we improve toric intraocular lens calculation methods? current insights. *Clinical Ophthalmology*, Volume 14:1899–1908, July 2020.

[Gar02] C. J Garvey. Computed tomography in clinical practice. *BMJ*, 324(7345):1077–1080, May 2002.

[GB19] Savy Gulati and Rosepreet Kaur Bhogal. Detection of malignant melanoma using deep learning. In Mayank Singh, P.K. Gupta, Vipin Tyagi, Jan Flusser, Tuncer Ören, and Rekha Kashyap, editors, *Advances in Computing and Data Sciences*, pages 312–325, Singapore, 2019. Springer Singapore.

[GE08] Emmanouil Psarakis Georgios Evangelidis. Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1858–1865, 2008.

[GK16] M.-M. Gellrich and C. Kandzia. Purkinje-bilder in der spaltlampenvideographie: Videobeitrag. *Der Ophthalmologe*, 113(9):789–793, August 2016.

[GL17] Rishab Gargeya and Theodore Leng. Automated identification of diabetic retinopathy using deep learning. *Ophthalmology*, 124(7):962–969, 2017.

[Grz21] Andrzej Grzybowski. *Artificial Intelligence in Ophthalmology*. Springer International Publishing, 2021.

[GT13] Dilraj S. Grewal and Angelo P. Tanna. Diagnosis of glaucoma and detection of glaucoma progression using spectral domain optical coherence tomography. *Current Opinion in Ophthalmology*, 24(2):150–161, March 2013.

[GWK+18] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, and Tsuhan Chen. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377, 2018.

[HHHJ90] D Henderson, Richard E Howard, W Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*. Neural Information Processing Systems, 1990.

[Hou60] Paul V C Hough. Patent us3069654a, method and means for recognizing complex patterns, 1960.

[HS+88] Chris Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*. Citeseer, 1988.

[HSL+91] David Huang, Eric A. Swanson, Charles P. Lin, Joel S. Schuman, William G. Stinson, Warren Chang, Michael R. Hee, Thomas Flotte, Kenton Gregory, Carmen A. Puliafito, and James G. Fujimoto. Optical coherence tomography. *Science*, 254(5035):1178–1181, November 1991.

[HZRS15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[IPS20] Vilas B. Inchur, L. S. Praveen, and Preetham Shankpal. Implementation of blood cell counting algorithm using digital image processing techniques. In *2020 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*, pages 21–26, 2020.

[IS15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

[ISD12] Aparna Irodi, Arathi Simha, and Sarada David. Magnetic resonance imaging for the ophthalmologist: A primer. *Indian Journal of Ophthalmology*, 60(4):301, 2012.

[Kal00] Willi Kalender. *Computertomographie*. Wiley-VCH Verlag, Weinheim, Germany, August 2000.

[KB14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[KCA+22] Tiarnan D.L. Keenan, Qingyu Chen, Elvira Agrón, Yih-Chung Tham, Jocelyn Hui Lin Goh, Xiaofeng Lei, Yi Pin Ng, Yong Liu, Xinxing Xu, Ching-Yu Cheng, Mukharram M. Bikbov, Jost B. Jonas, Sanjeeb Bhandari, Geoffrey K. Broadhead, Marcus H. Colyer, Jonathan Corsini, Chantal Cousineau-Krieger, William Gensheimer, David Grasic, Tania Lamba, M. Teresa Magone, Michele Maiberger, Arnold Oshinsky, Boonkit Purt, Soo Y. Shin, Alisa T. Thavikulwat, Zhiyong Lu, Emily Y. Chew, Priscilla Ajilore, Alex Akman, Nadim S. Azar, William S. Azar, Bryan Chan, Victor Cox, Amisha D. Dave, Rachna Dhanjal, Mary Donovan, Maureen Farrell, Francisca Finkel, Timothy Goblirsch, Wesley Ha, Christine Hill, Aman Kumar, Kristen Kent, Arielle Lee, Pujan Patel, David Peprah, Emma Piliponis, Evan Selzer, Benjamin Swaby, Stephen Tenney, and Alexander Zeleny. Deeplensnet: Deep learning automated diagnosis and quantitative classification of cataract type and severity. *Ophthalmology*, 129(5):571–584, May 2022.

[Key81] R. Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6):1153–1160, 1981.

[KFSN14] Sanjeev Kasthurirangan, Lucas Feuchter, Pamela Smith, and Donald Nixon. Software-based evaluation of toric iol orientation in a multicenter clinical study. *Journal of Refractive Surgery*, 30(12):820–826, December 2014.

[Kol95] Helga Kolb. *Gross Anatomy of the Eye*. University of Utah Health Sciences Center, Salt Lake City (UT), 1995.

[KPL+12] Pearse A. Keane, Praveen J. Patel, Sandra Liakopoulos, Florian M. Heussen, Srinivas R. Sadda, and Adnan Tufail. Evaluation of age-related macular

degeneration with optical coherence tomography. *Survey of Ophthalmology*, 57(5):389–414, September 2012.

[LBBH98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[LBK+18] Sang-gil Lee, Jae Seok Bae, Hyunjae Kim, Jung Hoon Kim, and Sungroh Yoon. Liver lesion detection from weakly-labeled multi-phase CT volumes with a grouped single shot multibox detector. *CoRR*, abs/1807.00436, 2018.

[LBL17] Cecilia S Lee, Doug M Baughman, and Aaron Y Lee. Deep learning is effective for classifying normal versus age-related macular degeneration oct images. *Ophthalmology Retina*, 1(4):322–327, 2017.

[Lei07] Timothy G. Leighton. What is ultrasound? *Progress in Biophysics and Molecular Biology*, 93(1):3–83, 2007. Effects of ultrasound and infrasound relevant to human health.

[LGB+01] Marc Lalonde, Langis Gagnon, Marie-Carole Boucher, et al. Automatic visual quality assessment in optical fundus images. In *Proceedings of vision interface*, volume 32, pages 259–264. Ottawa, 2001.

[LGVSC17] James C Liu, Wesley Green, Gregory P Van Stavern, and Susan M Culican. Assessing the utility of 2.5% phenylephrine for diagnostic pupillary dilation. *Canadian Journal of Ophthalmology*, 52(4):349–354, 2017.

[LLC+20] Wonje Lee, Seungho Lee, Semin Chong, Kyungmin Lee, Jongha Lee, Jae Chol Choi, and Changwon Lim. Radiation dose reduction and improvement of image quality in digital chest radiography by new spatial noise reduction algorithm. *PLOS ONE*, 15(2):e0228609, February 2020.

[LMY24] Xuanqiao Lin, Dongmei Ma, and Jin Yang. Insights into the rotational stability of toric intraocular lens implantation: diagnostic approaches, influencing factors and intervention strategies. *Frontiers in Medicine*, 11, 2024.

[Low04a] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[Low04b] David G Lowe. Patent us6711293b1, method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image, 2004.

[LSC+23] Achim Langenbucher, Nóra Szentmáry, Alan Cayless, Jascha Wendelstein, and Peter Hoffmann. Prediction of iol decentration, tilt and axial position using anterior segment oct data. *Graefe's Archive for Clinical and Experimental Ophthalmology*, 262(3):835–846, September 2023.

[LSW+23] Xiangjie Leng, Ruijie Shi, Yanxia Wu, Shiyin Zhu, Xingcan Cai, Xuejing Lu, and Ruobing Liu. Deep learning for detection of age-related macular degeneration: A systematic review and meta-analysis of diagnostic test accuracy studies. *PLOS ONE*, 18(4):e0284060, April 2023.

[LWK+17] Yu-Chi Liu, Mark Wilkins, Terry Kim, Boris Malyugin, and Jodhbir S Mehta. Cataracts. *The Lancet*, 390(10094):600–612, 2017.

[Mal08] Raman Malhotra. 5 - techniques in cataract surgery. In Raman Malhotra, editor, *Cataract*, pages 89–105. Butterworth-Heinemann, Edinburgh, 2008.

[ML09] Marius Muja and David Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP 2009 - Proceedings of the 4th International Conference on Computer Vision Theory and Applications*, volume 1, pages 331–340, 01 2009.

[MSK11] Achyut N. Mukherjee, Chunwah So, and Vinod Kumar. Clinical and experimental validation of a slit lamp modification to measure toric lens position. *Contact Lens and Anterior Eye*, 34(3):111–113, 2011.

[OOS+21] Ryoko Osawa, Tetsuro Oshika, Masahiko Sano, Takuma Yuguchi, and Tadayoshi Kaiya. Rotational stability of modified toric intraocular lens. *PLOS ONE*, 16(3):1–10, 03 2021.

[QTJ+15] Elizabeth A Quigley, Barbara A Tokay, Sarah T Jewell, Michael A Marchetti, and Allan C Halpern. Technology and technique standards for camera-acquired digital dermatologic images: a systematic review. *JAMA dermatology*, 151(8):883–890, 2015.

[Rad17] Johann Radon. Über die bestimmung von funktionen längs gewisser mannigfaltigkeiten. *Berichte über die Verhandlungen der Königlich-Sächsischen Gesellschaft der Wissenschaften zu Leipzig. Mathematisch-Physische Klasse.*, 69:262–277, 1917.

[RKK+20] Tawsifur Rahman, Amith Khandakar, Muhammad Abdul Kadir, Khandaker Rejaul Islam, Khandakar F. Islam, Rashid Mazhar, Tahir Hamid, Mohammad Tariqul Islam, Saad Kashem, Zaid Bin Mahbub, Mohamed Arselene Ayari, and Muhammad E. H. Chowdhury. Reliable tuberculosis detection using chest x-ray with deep learning, segmentation and visualization. *IEEE Access*, 8:191586–191601, 2020.

[Roj96] Raul Rojas. *Neural Networks*. Springer, Berlin, Germany, July 1996.

[RRKB11] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571, 2011.

[SCG24] Wiktor Stopyra, David L. Cooke, and Andrzej Grzybowski. A review of intraocular lens power calculation formulas based on artificial intelligence. *Journal of Clinical Medicine*, 13(2):498, January 2024.

[Sch30] Fritz Schroeter. Patent de574085c, verfahren zur abtastung von fernsehbildern, 1930.

[SHK+14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[SKBK22] Abhinav Singh, Gaurav Kapoor, Vinod K. Baranwal, and Nidhi Kalra. Rotational stability of toric intraocular lenses. *Medical Journal Armed Forces India*, 78(1):68–73, January 2022.

[SMS94] Kimiya Shimizu, Akiko Misawa, and Yoko Suzuki. Toric intraocular lenses: correcting astigmatism while controlling axis shift. *Journal of Cataract & Refractive Surgery*, 20(5):523–526, 1994.

[Sou09] Simon Southern. Ultrasound of the eye. *Australasian Journal of Ultrasound in Medicine*, 12(1):32–37, February 2009.

[SSF13] Harold A Stein, Raymond M Stein, and Melvin I Freeman. Chapter 10 - refractive errors and how to correct them. In *The Ophthalmic Assistant (Ninth Edition)*, pages 173–197. W.B. Saunders, London, ninth edition edition, 2013.

[SSM+20] Daniel Schartmüller, Luca Schwarzenbacher, Elias Laurin Meyer, Sabine Schriefl, Christina Leydolt, and Rupert Menapace. Comparison of long-term rotational stability of three commonly implanted intraocular lenses. *American Journal of Ophthalmology*, 220:72–81, December 2020.

[SSS+21] Daniel Schartmüller, Luca Schwarzenbacher, Sabine Schriefl, Veronika Röggla, Irene Steiner, Claudette Abela-Formanek, Christina Leydolt, and Rupert Menapace. Rotational stability of intraocular lenses: A standardized method for more accurate measurements in future studies. *American Journal of Ophthalmology*, 231:200–207, November 2021.

[TPT+20] Phuong Truong, Alex Phan, Buu Truong, Benjamin Suen, Gerrit Melles, and Frank Talke. A smartphone attachment for remote ophthalmic slit lamp examinations. *Microsystem Technologies*, 26(11):3403–3407, May 2020.

[VCB+23] Maria Vakalopoulou, Stergios Christodoulidis, Ninon Burgos, Olivier Colliot, and Vincent Lepetit. *Deep Learning: Basics and Convolutional Neural Networks (CNNs)*, page 77–115. Springer US, 2023.

[WB10] James S. Wolffsohn and Phillip J. Buckhurst. Objective analysis of toric intraocular lens rotation and centration. *Journal of Cataract & Refractive Surgery*, 36(5):778–782, 2010.

[Wen21] Jascha Wendelstein. Rotational stability, tilt and decentration of a new iol with a 7.0 mm optic. *Current Eye Research*, 46(11):1673–1680, 2021. PMID: 34121551.

[YZL21] Song Yang, Jun Zhou, and Dengwen Li. Functions and diseases of the retinal pigment epithelium. *Frontiers in Pharmacology*, 12, 2021.

[ZLC+16] Xiaojian Zhong, Erping Long, Wan Chen, Wu Xiang, Zhaochuan Liu, Hui Chen, Jingjing Chen, Zhuoling Lin, Haotian Lin, and Weirong Chen. Comparisons of the in-the-bag stabilities of single-piece and three-piece intraocular lenses for age-related cataract patients: a randomized controlled trial. *BMC Ophthalmology*, 16(1), July 2016.

[ZNX+19] Hongyan Zhang, Kai Niu, Yanmin Xiong, Weihua Yang, ZhiQiang He, and Hongxin Song. Automatic cataract grading methods based on deep learning. *Computer Methods and Programs in Biomedicine*, 182:104978, December 2019.

[ZWB23] Shuhe Zhang, Carroll A.B. Webers, and Tos T.J.M. Berendschot. Luminosity rectified blind richardson-lucy deconvolution for single retinal image restoration. *Computer Methods and Programs in Biomedicine*, 229:107297, 2023.