



h_da

HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

fbi

FACHBEREICH INFORMATIK

Hochschule Darmstadt, University of Applied Science
- FACHBEREICH INFORMATIK -

Automatic detection of glasses in ocular images

Detection of glasses using 3 different approaches

Abschlussarbeit zur Erlangung des akademischen Grades
Bachelor of Science (B.Sc.)

vorgelegt von

Florian Struck

Referent:

Prof. Dr. Christoph Busch

Korreferent:

Dr. Christian Rathgeb

Florian Struck: *Automatic detection of glasses in ocular images*, Detection of glasses using 3 different approaches, Bachelor of Science (B.Sc.), ©July 7, 2017

Supervisors:

Prof. Dr. Christoph Busch

Dr. Christian Rathgeb

Location:

Darmstadt

Time frame:

24.04.2017 - 07.07.2017

Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe.

Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht.

Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen.

Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Darmstadt, 03. Juli 2017

Abstract

In these days there is a large number of deployments for biometric systems. One of the biggest deployments is the India Aadhaar Identification program with more than 1 billion registred citizens. In addition there are many border control programs that use biometric systems for identification. Hence one can suggest that biometric systems become an integral part of the digital world. Iris is one of the most frequently used biometric characteristic in operational systems because the recoding is convenient and these systems have a high reliability. Due to the increasing size of the operational deployments of biometric systems, the requirements in terms of, among others, biometric performance, efficiency and reliability increase. Especially the large number of glasses wearers is a challenge for iris recognition systems.

There exist some studies which showed that glasses can deteriorate the biometric performance of iris recognition systems, but none of these showed the causes of this deterioration in detail. We analyzed the influence of glasses on the performance of an iris recognition system with different experiments and we discovered, that the biometric performance loss of iris recognition systems correlates strongly with the subjects wearing glasses. A possible solution to solve this problem is to automatically detect glasses and handle such attempts separately. We propose 3 approaches to automatic detection of glasses and perform a comparative assessment of their accuracy. The proposed approaches are based on: an explicit algorithmic approach with edge and reflection detection, a deep learning approach and an approach using filters of binarized statistical image features.

The benchmark that we used was carried out on the CASIA-IrisV4-Thousand database, which contains 20000 near-infrared eye images; 5336 with and 14664 without glasses. The explicit algorithm achieved a classification accuracy of 97.18 %, the statistical approach achieved an accuracy of 98.08 % and the deep learning approach achieved an classification accuracy of 98.97 %. When using a fusion of all three approaches we were able to classify 99.06 % of images with glasses and 99.71 % of images without glasses. Thereby we were able to classify 99.54 % of all images on the CASIA-Thousand database in glasses and non-glasses images. With rejecting the detected glasses on the CASIA-Thousand database, we increased the iris recognition performance from 9.19 % to 6.61 % EER.

Zusammenfassung

Heutzutage gibt es eine große Anzahl an Anwendungen für biometrische Systeme. Eine der größten Anwendungen für biometrische Systeme ist das India Aadhaar Identification Program mit mehr als einer Milliarde registrierter Einwohner. Zusätzlich verwenden aber auch schon viele Grenzschutz-Programme biometrische Systeme für die Identifizierung. Daher kann man behaupten, dass biometrische Systeme mittlerweile ein integraler Bestandteil unserer digitalen Welt sind. Eines der am häufigsten genutzten biometrischen Merkmale ist die Iris-Textur, da diese komfortable zu erfassen ist und eine hohe Zuverlässigkeit aufweist. Durch die steigende Größe von biometrischen Anwendungen steigen auch die Anforderungen an die biometrische Erkennungsleistung, die Effizienz und Zuverlässigkeit. Insbesondere die große Anzahl an Brillenträger ist eine neue Herausforderung für heutige Iris- Erkennungssysteme.

Es existieren zwar einige Forschungen, die zeigen, dass Brillen die Erkennungsleistung eines biometrischen Systems verschlechtern können, jedoch wurden die Gründe für diesen Zusammenhang nicht näher untersucht. Wir untersuchten den Einfluss von Brillen anhand verschiedener Experimente und stellten fest, dass es einen starken Zusammenhang zwischen der Verschlechterung der Erkennungsleistung und der Verwendung von Brillenbilder gibt. Eine Möglichkeit zur Lösung dieses Problems ist, Brillen automatisch zu erkennen, um diese einer separaten Behandlung unterziehen zu können. Wir testeten drei unterschiedliche Herangehensweisen für die Erkennung von Brillen und verglichen diese anhand ihrer Genauigkeit. Zum einen wurde ein expliziter Ansatz mit Kanten- und Reflexionserkennung verwendet, zum anderen aber auch ein Deep-Learning Ansatz und eine statistische Herangehensweise mit Verwendung des BSIF (Binarized Statistical Image Features) -Filters.

Für den Benchmark verwendeten wir die CASIA-IrisV4-Thousand Datenbank mit 20000 Bilder (5336 Brillenbilder und 14664 Nicht-Brillenbilder) im Nah-infrarot-Bereich. Der explizite Algorithmus erzielte eine Genauigkeit von 97,18 %, der statistische Ansatz konnte 98,08 % der Bilder richtig zuordnen und der Deep-Learning Ansatz hatte eine Genauigkeit von 98,97 %. Durch Fusion aller drei Konzepte waren wir in der Lage, 99,06 % aller Brillenbilder und 99,71 % aller Nicht-Brillenbilder richtig zuzuordnen. Auf der CASIA-Thousand Datenbank konnten wir dadurch 99,54 % aller Bilder erfolgreich klassifizieren. Durch zurückweisen der erkannten Brillenbilder konnten wir damit die Iris-Erkennungsleistung auf der CASIA-Datenbank von 9.19 % auf 6.61 % EER verbessern.

Acknowledgments

First of all, I would like to thank my supervisors Christian Rathgeb and Pawel Drozdowski for all the suggestions and assistance during the practice phase and the thesis. Especially, I would like to thank Pawel for the many hours, he corrected my thesis and assisted with \LaTeX knowledge.

Further thanks go to my family Susanne, Rolf and Nadine for the support and motivation during my study. Without you, I would not be able to do this study and write this thesis.

Contents

List of Figures	ix
List of Tables	xi
1 Introduction	1
1.1 Background	2
1.1.1 Biometric system	2
1.1.2 Topology	2
1.1.3 Performance estimation	4
1.1.4 Iris signal processing	5
1.2 Motivation	6
1.2.1 Experiments	7
1.3 Research questions	11
2 Fundamentals	12
2.1 Machine Learning	12
2.1.1 Support Vector Machine	12
2.1.2 Deep Learning	13
2.2 Binarized statistical image features	15
3 Glasses detection	16
3.1 Algorithmic feature extraction	16
3.1.1 Reflection	16
3.1.2 Edge detection	18
3.1.3 SVM	23
3.2 Deep Learning	23
3.2.1 Structure	24
3.2.2 Training	25
3.3 Binarized statistical image features	26
3.3.1 Functionality	26
3.3.2 SVM	27

4	Evaluation	28
4.1	Procedure	28
4.2	Results	30
4.3	Fusion	30
5	Discussion	33
5.1	Error classes	33
5.1.1	Explicit algorithm	33
5.1.2	Deep learning	36
5.1.3	Statistical approach with BSIF	37
5.2	Strengths and weaknesses	39
5.2.1	Classification accuracy	39
5.2.2	Adaptability	40
5.2.3	Transparency	41
5.2.4	Throughput	43
5.2.5	Expandability	44
5.2.6	Summary	44
6	Conclusion	45
	Bibliography	I

List of Figures

1.1	General topology of a biometric system [Joi06]	2
1.2	Dependence between the False Match Rate, the False Non-Match Rate and the used threshold	4
1.3	Iris preprocessing steps for feature extraction	5
1.4	Feature extraction from the normalized iris texture to a binary string	5
1.5	Sample images from the CASIA-Thousand database and the MobBIO database	7
1.6	Found error classes of failed segmentations	9
1.7	Quality distribution of all successful segmented images	10
2.1	Example of a 2D SVM Classification, adapted from [Sch16]	13
2.2	General structure of a deep neuronal network, adapted from [ima]	13
2.3	General structure of a deep convolution network, adapted from [con]	14
2.4	Learned BSIF filters [KR12]	15
3.1	Comparison between natural reflections and artificial reflections caused by glasses	17
3.2	Visualization of the splitting process	17
3.3	Visualization of the relative brightness deviation	18
3.4	Histogram of the reflection score distribution	18
3.5	Visualization of the edge highlighting using a horizontal edge operator	19
3.6	Applying the calculated threshold for binarization of the grey-value image	20
3.7	Dilatation process example	21
3.9	Visualization of the assignment of the pixels to edges	22
3.10	Feature score distribution of the explicit algorithm on the CASIA database	23
3.11	Structure of our deep convolutional network for glasses detection	24
3.12	Applying the BSIF filter on the input image	26
3.13	Applying the BSIF filter on the input image with glasses	26
4.1	Splitting process for cross validation on the CASIA database	29
5.1	Wrongly classified image without glasses due to bad lightning conditions	34
5.2	Wrongly classified image without glasses due to bad alignment of the eye	34
5.3	Wrongly classified image without glasses due to a strong eyebrows	35

5.4	Wrongly classified image of a glasses due to a strong curvature of the frame edges	35
5.5	Wrongly classified image of a glasses due to transparent glasses frames . . .	36
5.6	Error classes of the deep learning approach	36
5.7	Wrongly classified image without classes due to a strong upper eye edge . .	38
5.8	Wrongly classified image with glasses due to a thin glasses frame	38
5.9	Wrongly classified image with classes due to a transparent glasses frame . .	39
5.10	Feature score distribution of the explicit algorithm on the CASIA database	42
5.11	Part of the average histogram from the statistical approach with the BSIF filter on the CASIA database	42
5.12	Learned convolution filter in our glasses identification network	43

List of Tables

1.1	Iris recognition performance on our base experiment	8
1.2	Statistics of the OSIRIS segmentation performance	8
1.3	Iris recognition performance on successfully segmented images	9
1.4	Iris recognition performance on successful segmented images and usage of masks and a threshold of 0.60	10
3.1	Example list of measured edges	22
4.1	The characteristics of the CASIA database	28
4.2	Results of the evaluation on the CASIA database	30
4.3	Results of the AND fusion on the CASIA database	30
4.4	Results of the OR fusion on the CASIA database	31
5.1	Summary of the strengths and weaknesses of the proposed approaches	44

Acronyms

FMR	False match rate
FNMR	False non-match rate
EER	Equal Error Rate
NIR	near-infrared
SVM	Support Vector Machine
DNN	Deep Neuronal Network
CNN	Convolution Neural Network
BSIF	binarized statistical image features

Chapter 1

Introduction

Due to growing networking and digitalisation, reliable, efficient and user-friendly authentication of users becomes more and more important. Besides traditionally authentication methods that are based on either knowledge (e.g. passwords) or on possession (e.g. tokens, access cards), biometrics has arisen as an increasingly viable alternative. Biometric systems make it possible to identify users using biological and behavioural features of an individual [Joi12]. In contrast to knowledge or physical tokens biometric features cannot be forgotten, lost or passed on to another person. This makes it a comfortable and secure way to authenticate users.

One of the most popular biometric characteristics is the iris texture. One reason for this is that the iris can be recorded comfortable because the eye is an externally visible organ. Further the iris has a high collectability and permanence [JBP06] so that it can be used for a fast and reliable biometric system. An important factor for the biometric performance of a biometric recognition system is the quality of the captured sample. The better the sample quality the better the biometric recognition performance. For iris recognition it is important that the eye is opened widely and the iris texture is clearly visible and unoccluded. Besides natural noise factors such as eyelids or eyelashes, which cannot be prevented, there are environmental factors that can further deteriorate the quality of a sample [Joi15]. Since many of today's biometric systems are unsupervised, there is no human who controls the capturing process, these factors have to check automatically by the biometric system. One of these factors is the wearing of glasses during the capturing process. In contrast to many other environmental factors (e.g. lighting, reflections) that can already be detected automatically, the detection of glasses is more complex and needs greater attention. Therefore we tried to detect glasses on eye images automatically, so that an iris recognition system would be able to handle them separately to achieve a better biometric performance. For this purpose we tried 3 different methods for glasses detection. The first approach was an explicit algorithm that calculates 2 metrics that describe the edges and reflections on an input image and uses a Support Vector Machine

(SVM) to classify the images with the calculated metrics. The second approach that we have tested uses a deep convolution network for direct classification. As third method, we used a statistical approach that creates a histogram of an input image using the BSIF filter and uses a SVM for classification. After we tested the approaches on their own we merged the decisions with different rules to get better results.

1.1 Background

In order to understand the experiments that we have done to investigate the influence of glasses on iris recognition systems, we give a short introduction about biometric systems and techniques to estimate their performance.

1.1.1 Biometric system

A biometric system is used to identify or verify an individual based on its biometric features. It uses the distinguishable biological or behavioural characteristics to recognize a registered user of a system. For this it saves the biometric characteristics of an individual in a database so that it is able to compare the characteristics with other samples.

1.1.2 Topology

The general topology of a biometric system can be seen in figure 1.1.

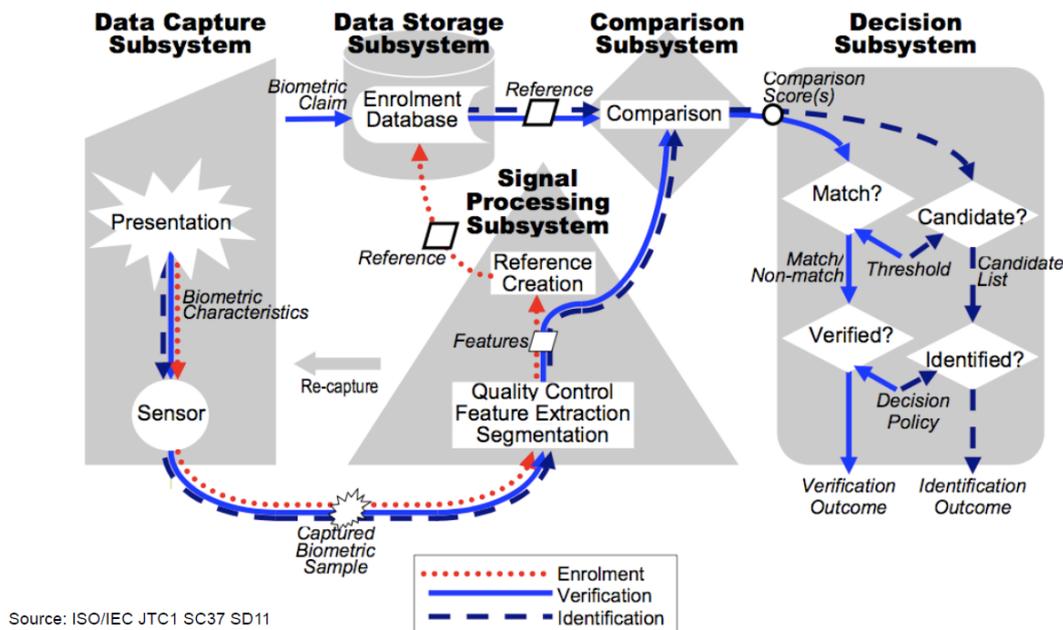


Figure 1.1: General topology of a biometric system [Joi06]

In the following part, the subsystems of a general biometric system are explained. The used biometric terms are defined in ISO/IEC 2382-37 [Joi12].

Data Capture Subsystem This subsystem has the tasks to capture the biometric features from a user into a biometric sample. This is done by sensors which record the needed characteristics. In the case of iris recognition these sensors are generally cameras that take pictures of the eyes.

Signal Processing Subsystem This subsystem processes the captured sample of the [Data Capture Subsystem](#). For this it runs multiple steps such as quality control, feature extraction and segmentation to create a reference. While the enrolment process the reference is given directly to the [Data Storage System](#), where it will be saved in a database. Otherwise the reference goes to the [Comparison Subsystem](#).

Data Storage Subsystem While running an enrolment process, this subsystem saves the enrolled reference in the database, so that it can be loaded later for verification or identification. For verification, it loads the reference of the user based on an identity claim and sends it to the [Comparison Subsystem](#). When running an identification process, it loads all references from the database and sends them to the [Comparison Subsystem](#).

Comparison Subsystem In this subsystem, the captured sample from the [Signal Processing Subsystem](#) and a saved reference from the [Data Storage System](#) are compared. As a result of the comparison, one or more comparison scores are generated that indicate how similar or dissimilar the two samples were. These scores are sent to the [Decision Subsystem](#).

Decision Subsystem This is the last subsystem and runs in two different modes: verification and identification. In the verification mode, the system decides on the basis of a single comparison score from the [Comparison Subsystem](#) and an associated threshold whether there is a match or not. If there is a match, the person whose characteristic was captured in the [Data Capture Subsystem](#) is verified successfully, otherwise the verification fails. In the identification mode, this subsystem receives a list of comparison scores from the [Comparison Subsystem](#). With the aid of the best score or an additional decision logic, the system decides with thresholds whether there is a candidate or not. If there is a candidate, the associated identity according to the database will be outputted, else the identification fails.

1.1.3 Performance estimation

For comparing and benchmarking the performance of a biometric system, there are different metrics which are standardized in ISO/IEC 19795-2 [Joi07]. Two of these are the False Match Rate (FMR) and the False Non-Match Rate (FNMR).

False Match Rate (FMR) This metric indicates the proportion of impostor attempts that were accepted falsely. It depends on the used decision threshold. For dissimilarity scores, the higher the threshold the higher is the False Match Rate.

False Non-Match Rate (FNMR) This metric indicates the proportion of genuine attempts that were rejected falsely. Like the False Match Rate it depends on the used decision threshold. For dissimilarity scores, the higher the decision threshold, the lower the False Match Rate.

Equal Error Rate (EER) This metric indicates the point where the False Match Rate and the False Non-Match Rate are equal. It means that there is the same proportion of impostor attempts that were accepted falsely, as genuine attempts that were rejected falsely. This is a common, one-number metric to evaluate the performance of a biometric system.

The dependence between the FMR, FNMR and the used threshold can be seen in figure 1.2.

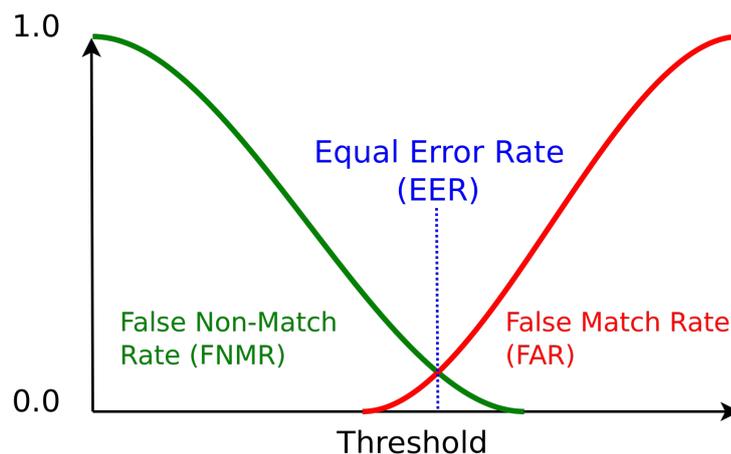


Figure 1.2: Dependence between the False Match Rate, the False Non-Match Rate and the used threshold

1.1.4 Iris signal processing

The signal processing in a iris recognition system can be distinguished into segmentation/normalization and feature extraction. The described processes are used in essentially all operational iris deployments.

Segmentation/Normalization

In this step, an algorithm locates the outline of the iris (visualized in figure 1.3b) and normalizes it into a rectangular area (figure 1.3c) of a constant size according to Daugman's rubber sheet model [Dau04]. In addition it can generate a noise mask (figure 1.3d) of parts that do not belong to the iris texture such as eyelids or eyelashes. This prevents that the feature extraction algorithm processes areas that do not contain iris feature, and thereby deteriorate the recognition performance. This preprocessing ensures that the feature extraction algorithm has an input of a constant size and can extract the features independent of the iris position and pupil dilation.

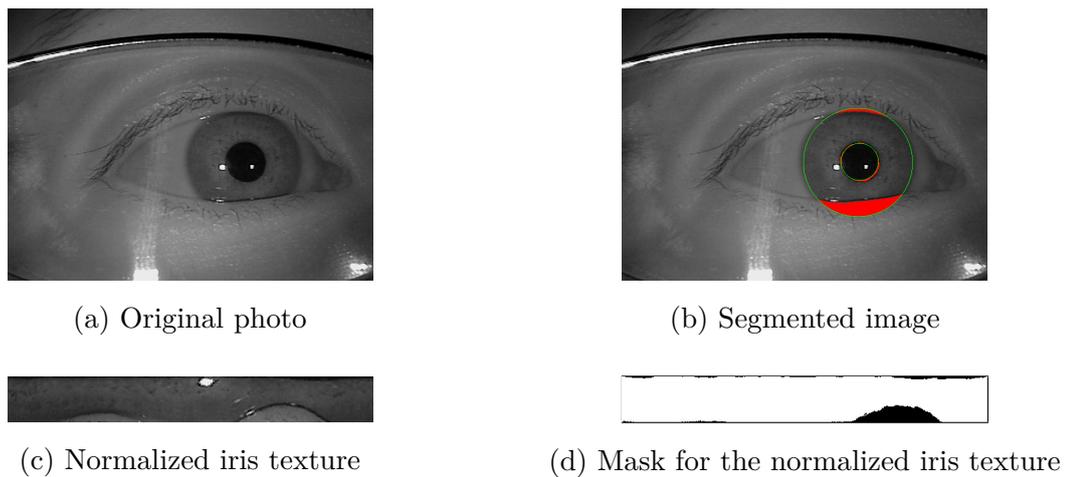


Figure 1.3: Iris preprocessing steps for feature extraction

Feature extraction

After creating a normalized texture with an optional noise mask, a feature extraction algorithm transforms it into a binary template of a constant size (called iris code).



Figure 1.4: Feature extraction from the normalized iris texture to a binary string

As J. Daugman [Dau04] already introduced, the comparison of two binary iris templates

can be done by calculating the Hamming distance. The formula for the calculation of the dissimilarity scores of two binary iris templates can be expressed as

$$hd = \frac{\|(codeA \oplus codeB) \cap maskA \cap maskB\|}{\|maskA \cap maskB\|} \quad (1.1)$$

where codeA is the bit vector of the first template, codeB is the bit vector of the second template, maskA is the bit vector of the mask of the first template and maskB is the bit vector for the mask of the second template.

1.2 Motivation

With increasing number of use cases for biometric systems, the systems have to become more robust against different environmental influences which can deteriorate the quality of the captured samples. While earlier most biometric systems were permanently installed with consistently good lighting conditions, more and more of today’s biometric systems are based on mobile hardware, such as mobile iris sensors or smartphones , operating under unconstrained and unsupervised environments. In addition, while earlier only instructed persons used these relatively small systems, today’s large-scale biometric systems are used by the general public, who do not always know how to use such systems correctly. A wrong usage of the sensors during the capturing process can result in a bad sample quality and as H. Proença and L. Alexandre [PA05] have already showed, leads to a worse segmentation performance. That is why automatic sample quality control must be performed by a biometric system, in order to ensure correct operation.

There are many quality metrics defined in the ISO/IEC 29794-6 [Joi15] to measure the quality of iris samples. The task for a biometric system is to calculate those quality metrics for a captured sample automatically to decide if the quality is sufficient. Such an automatic quality estimation was done, for example, by Kalka *et al.* [KZSC10] who developed an automatic method to estimate the quality of iris samples.

Besides the general sample quality, there is an additional factor. Many people have to wear glasses. For example, in Germany, almost two thirds (63.5 %) of the population which age is over 15 are glasses wearers [fDA15]. The number of glasses wearers between 20- and 29 is increasing noticeably. As Lim *et al.* [LLBK01] has already investigated, the usage of samples with glasses deteriorate the preprocessing performance and therefore the overall performance of an iris recognition system. However, the causes for this deterioration were analyzed rarely by the authors so that we investigated the underlying causes of this quality deterioration in greater detail.

1.2.1 Experiments

To investigate the influence of glasses under different basis conditions, we used both near-infrared (NIR) and visible wavelength (VW) images of the CASIAv4 thousand database [Chi], thereby yielding more general and comprehensive results. We manually categorized the images in two classes: images with glasses and images without glasses, thus we were able to evaluate the recognition performance of these classes separately. Then we segmented all images using the OSIRIS [ODGS16] tool, extracted their features using the LogGabor implementation of the USITv2-Toolkit [RUWH16] and saved the references sorted by the subject id. After that we were able to calculate the genuine and impostor scores for both classes by applying the commonly used, Hamming distance based, comparator. At the end of each experiment, we calculated the Equal Error Rate (EER) on the basis of the calculated scores, so that we were able to assess the recognition performance.

In figure 1.5 you can see sample images from the databases that we used in our experiments. Image 1.5a-d is from the MoBIO database, image 1.5e-h is from the CASIA-Thousand database.

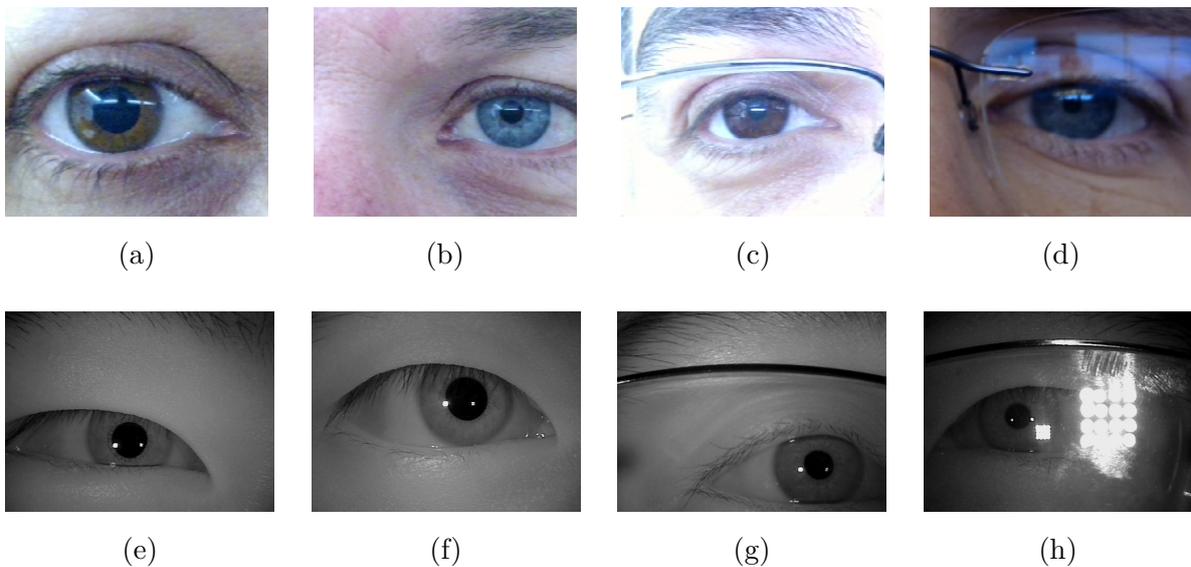


Figure 1.5: Sample images from the CASIA-Thousand database and the MoBIO database

Baseline

At the beginning of our research, we started with a baseline evaluation of the categorized databases. The results are shown in table 1.1. As you can see, using samples with glasses leads to a strong deterioration of the iris recognition performance on both databases. Further analysis showed that there were many high genuine comparison scores on samples of glasses, which causes a higher EER.

Database	Glasses			No glasses		
	Genuine comp.	Impostor comp.	EER	Genuine comp.	Impostor comp.	EER
CASIA	10338	999000	12.16 %	52278	999000	6.86 %
MobBIO	658	10712	40.04 %	4020	10712	35.26 %

Table 1.1: Iris recognition performance on our base experiment

Segmentation evaluation

Next, we evaluated the segmentation performance by visually inspecting the segmented images for correctness. In table 1.2 you can see the statistics of the OSIRIS segmentation performance on the used databases.

Database	Glasses			No glasses		
	Successful segm.	Failed segm.	Quota	Successful segm.	Failed segm.	Quota
CASIA	5105	231	95.67 %	14031	633	95.68 %
MobBIO	127	136	48.29 %	842	535	61.15 %

Table 1.2: Statistics of the OSIRIS segmentation performance

While on the MobBIO database the quota of the successful segmentations on images with glasses was much worse than on images without glasses, on the CASIA-Thousand database the proportions were almost equal. This shows that on the CASIA-Thousand database glasses have significantly less influence on the segmentation performance than they have on the MobBIO database. One reason for the small difference could be that glasses have less influence on the photo quality when using near-infrared light instead of visible light.

We analyzed the wrongly segmented samples and found 4 error glasses, which are shown in figure 1.6.

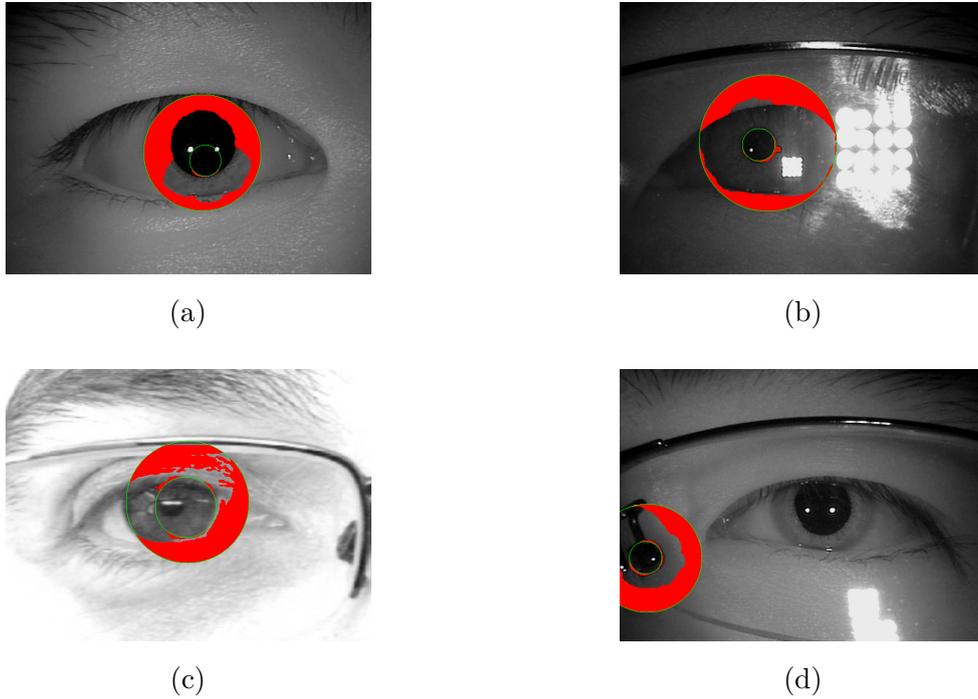


Figure 1.6: Found error classes of failed segmentations

As you can see 3 of 4 error classes were caused by glasses, whereas the most common error class (figure 1.6a) was independent of glasses. After sorting out all failed segmentations, we repeated the evaluation. The results can be seen in table 1.3.

Database	Glasses			No glasses		
	Genuine comp.	Impostor comp.	EER	Genuine comp.	Impostor comp.	EER
CASIA	9729	998001	10.67 %	48744	998001	3.79 %
MobBIO	263	9037	27.92 %	2264	9037	28.91 %

Table 1.3: Iris recognition performance on successfully segmented images

The removal of false segmentation has led to a significantly better performance on both classes. However, on samples without glasses the performance is still better than on samples with glasses. We think that the EER of 27.92 % on the MobBIO database is a measuring error because, this is the only exception where the performance on samples with glasses is better than the performance on samples without glasses. All other calculations show consistent results. One possible reason for this measuring error could be that there were only few genuine comparisons (263).

Quality evaluation

In the last experiment we analyzed the quality of the samples using the *Usable iris area*, which is a general metric for quality estimation of iris images and is standardized in ISO 29794-6 [Joi15]. At first, we plotted the quality distribution of both databases, which can be seen in figure 1.7.

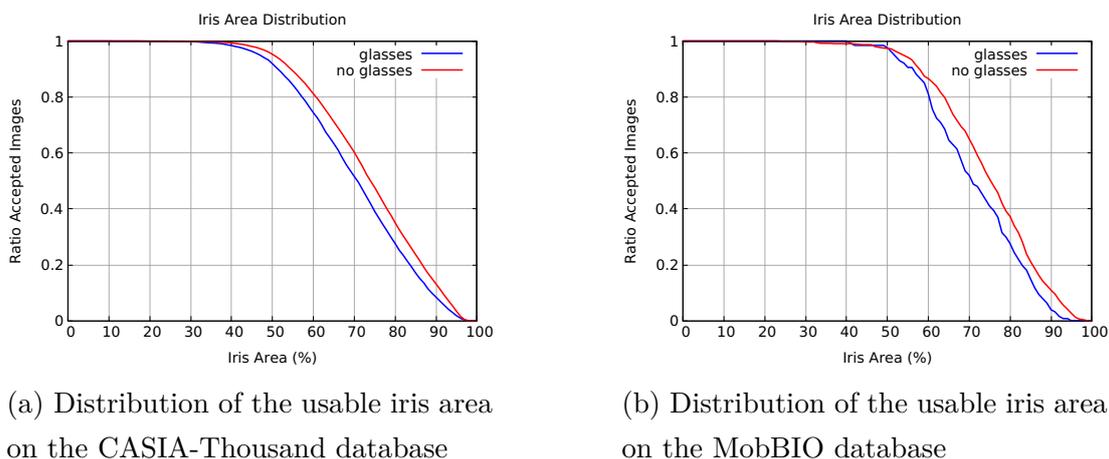


Figure 1.7: Quality distribution of all successful segmented images

The diagram shows that samples with glasses have a consistently smaller visible iris area than samples without glasses. After getting an impression of the quality distribution, we decided to exclude some of the images with small visible iris area. For our evaluation we used the threshold 0.60. This threshold rejects the poorest images and nevertheless accepts the majority of images with a better quality. The results are shown in table 1.4.

Database	Glasses			No glasses		
	Genuine comp.	Imposter comp.	EER	Genuine comp.	Imposter comp.	EER
CASIA	6366	935145	7.83 %	36573	935145	2.50 %
MobBIO	194	8572	26.06 %	1803	8572	25.11 %

Table 1.4: Iris recognition performance on successful segmented images and usage of masks and a threshold of 0.60

Although we sorted out all images with a small usable iris area, the performance difference between samples of glasses and samples without glasses kept significantly. This means that even in systems with high quality requirements, there can be a strong deterioration of the iris recognition performance when accepting samples with glasses.

Conclusion

During the experiments, we found multiple causes for the deterioration of the iris recognition performance when using samples with glasses. In our base experiment, we showed that even if the overall iris recognition performance is relatively bad, glasses have a great negative impact on the performance. After analyzing the segmentation performance, we found out that the most types of unsuccessful segmentations occurred only on images of glasses even, though the most common type of segmentation error occurred independently of glasses. It turned out that the influence of glasses on the segmentation performance depends on the used wavelength. While on the NIR photos of the MobBIO database there was a significant difference, on the VW photos of the CASIA database the error rates were almost equal. The calculation and visualization of the usable iris area of the samples showed that images without glasses had a consistently greater area. In addition, we demonstrated that even higher quality control thresholds do not change the fact that images with glasses have a worse performance on iris recognition systems than images without glasses. Overall, we can say that glasses consistently deteriorate the performance of iris recognition systems and should be handled separately. In order to do so, there has to be an automatic method of glasses detection on input samples.

1.3 Research questions

In this thesis, we develop different approaches for glasses detection and benchmark them against each other. The goal of this research was to answer following questions:

1. Which approaches are suitable for detecting glasses? How do they work?
2. Where are their strengths and weaknesses? How can weaknesses be avoided and strengths be used better?
3. How well do they complete each other? Is it possible to offset weaknesses of one approach by adding a second approach? How well does the fusion of all different approaches perform?

Chapter 2

Fundamentals

In the following chapter we describe fundamental concepts that we used in the experiments for the thesis. In addition we mention machine learning approaches and the BSIF filter.

2.1 Machine Learning

Machine Learning makes it possible to recognize relationships in datasets automatically. Therefore, it is a popular technique to handle complex data. Machine Learning can be divided into several approaches, which differ in the way their algorithm handle the data. In this thesis, we only used supervised learning which is commonly used for automatic classification. The usage of machine learning can be distinguished into a training phase and a prediction phase. In the training phase, the algorithm reads input data and its associated output data and learns the relationships between these datasets. By definition of supervised learning, the input data of the training set has to be labeled [sup]. This means that a groundtruth set for the training data has to be created which contains the expected associations. In the prediction phase, the algorithm reads only input data which is unknown to the algorithm and process it based on the previously learned relationships.

2.1.1 Support Vector Machine

A Support Vector Machine (SVM) is a supervised machine learning approach to classify input vectors into two categories. The basic operating principle of classification can be seen in figure 2.1.

During training the SVM reads multiple vectors that are represented as points in a hyperspace, whereby each point pertains to specific category. It tries to create a hyperplane between the points of the different categories, so that the groups of points are separated clearly. For each training iteration the SVM tries to enhance the distance between the hyperplane and the points to optimize the classification performance [Chr04]. This has the advantage that even if there are new points that are lying closer to the points of the

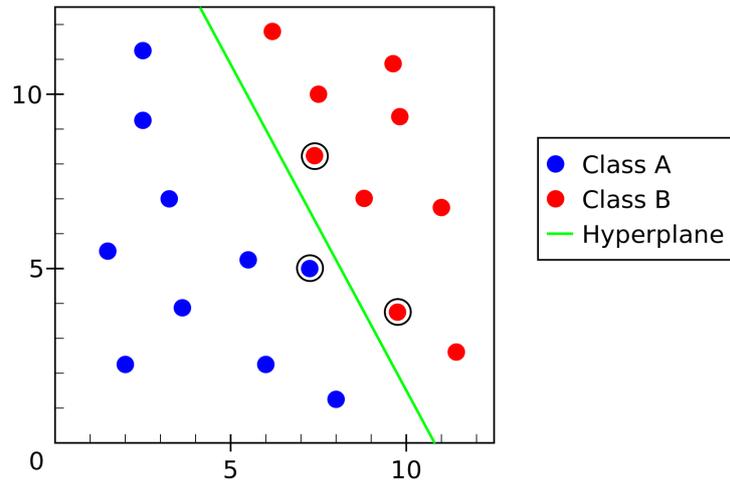


Figure 2.1: Example of a 2D SVM Classification, adapted from [Sch16]

other category the separation still works. For the optimization process only the so-called support vectors, which are points that are lying closely to the other category (outlined points in figure 2.1), matter.

In the prediction phase the SVM gets an input vector and decides on the basis of the position of the associated point in the hyperspace the category to which it belongs. These relatively simple calculations allow it to categorize complex data in a short time.

2.1.2 Deep Learning

Deep Learning describes combination of multiple neuronal layers that are connected to each other. Each of those learns particular connections of the previous layer and passes them to the next layer. This allows to learn very complex relationships for doing costly processing steps. The basic structure of a deep neuronal net (DNN) can be seen in figure 2.2.

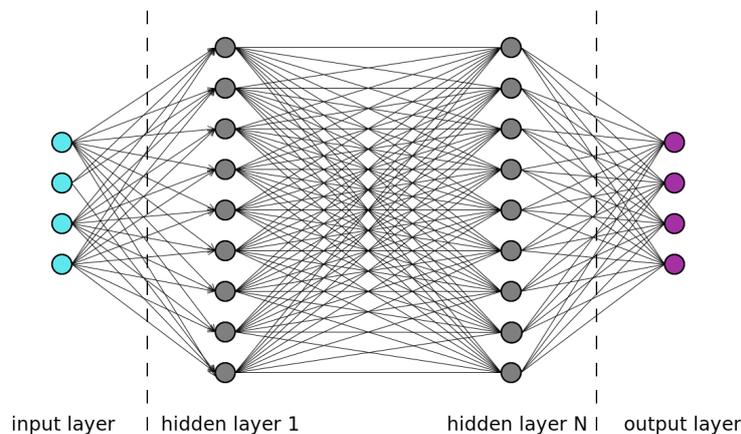


Figure 2.2: General structure of a deep neuronal network, adapted from [ima]

It contains an input layer, multiple hidden layers and an output layer. Each layer contains several neurons that are derived on the biological neurons of the human brain. For communication the neurons are connected to each other whereby each connection has a weight. Like the biological neurons, the artificial neurons have an activation threshold for the sum of the incoming signals. Once this threshold is exceeded the triggered neuron sends a signal to the following neurons.

During the training phase, the source data is given to the input neurons that sends the calculated signals to the next layer. At the end of the neuronal net the output is compared to the expected output of the training data. By using backpropagation [bac], the weights are adjusted so that the output of the net approximates the expected output.

In prediction mode, the input data is directly transformed into the output data by using the learned weights.

Deep Convolutional Network

A convolutional neuronal network [con] is a subset of a neuronal net. In contrast to the neuronal network that uses neurons, a convolutional network contains several filters per layer. Convolutional networks that have many layers are also called deep convolutional networks. This type of neuronal network is particularly very suitable for image processing, because it is able to recognize patterns in images efficiently. The common structure of a deep convolutional network for image processing can be seen in figure 2.3.

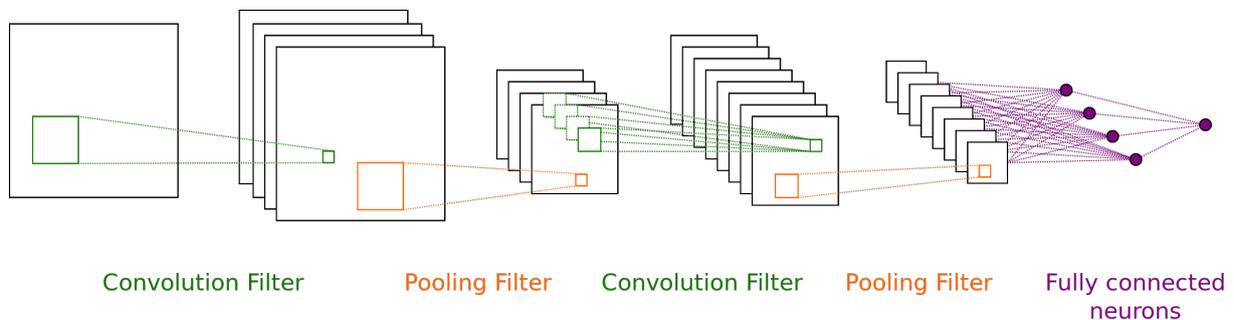


Figure 2.3: General structure of a deep convolution network, adapted from [con]

At the beginning of such a network, there are multiple combinations of convolution and pooling layers. These layers extract the features from the input image. While convolution filters learn known patterns from the input matrix, pooling layers scale down the size of the input, which enhance the calculation performance. At the end of the feature extraction part there is a set of fully connected neurons, which are responsible for the assignment of the extracted features into classes. The more difficult the processing task of the input image the more extensive has to be the network structure.

In summary it can be said that deep learning approaches are very powerful to solve classification problems which is why we used it for our experiments. However, the mentioned parts are only a small extract of the broad and complex subject area of machine learning.

2.2 Binarized statistical image features

In order to recognize image patterns and process images, local image descriptors in form of filters are commonly used. They are able to map local brightness curves into scalar values, which represent the local structure of an image part. A filter consists of a matrix of a fixed size, which contains weights for mapping the pixel values to the final filter response. Commonly, the filter is applied on all pixels of an image to create a detailed overview of the brightness features of an image. In this section, we introduce the Binarized Statistical Image Features (BSIF) filters.

The BSIF filter was first presented by J. Kannala and E. Rahtu in [KR12]. In contrast to the most filters, which use fixed weights, the weights of the BSIF filters are determined by training on natural images. Figure 2.4 shows an example of pre-learned 9x9 pixels BSIF filters.

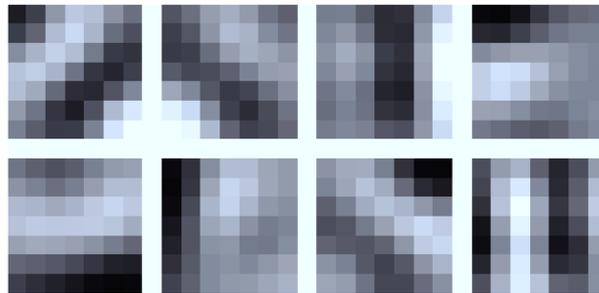


Figure 2.4: Learned BSIF filters [KR12]

The BSIF filter consists of several independent subfilters. Each subfilter has its own weights and therefore produces its own response. The response is binarized by its sign. As final filter response, the output bits of the different subfilters are appended and the results is interpreted as a decimal number. Therefore, the final filter response depends on the training data of the subfilters, their size and the number of filters that were used for the final response. Commonly, the filter response is transformed into a histogram of the length $l = 2^n$ where n is the number of subfilters which were used. For the experiments in this thesis, we used the pre-learned filters and matlab implementation from J. Kannala [bsi].

Chapter 3

Glasses detection

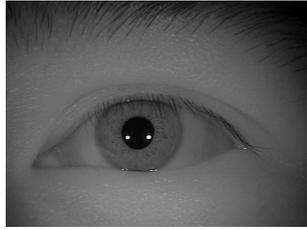
In this chapter, we present 3 different automatic approaches for glasses detection in ocular images. All approaches uses completely different strategies to select and extract features from the images and use them for classification.

3.1 Algorithmic feature extraction

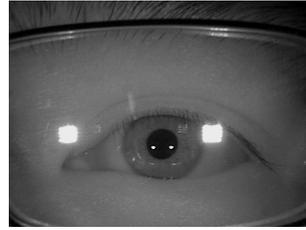
This approach is based on an explicit algorithm, that analyzed fixed features on images with glasses to generate metrics which can be used for classification. The procedure of this algorithm is inspired by how a human might perceive glasses. We analyzed multiple images to find features that are suitable for detecting glasses. These features had to be extracted easily by an machine algorithm and had to be robust against environmental factors such as fluctuations in illumination. In our implementation we used the OpenCV [Ope17] library for the image processing and uses auxiliary functions of the Boost [Boo17] library.

3.1.1 Reflection

The first feature that we have considered as suitable was the strong reflection on many glasses. As you can see in figure 3.2a, the natural reflections on images without glasses are much smaller than the reflections in figure 3.2b, which shows an image with glasses. This makes it easy to distinguish between natural reflections and artificial reflections which are caused by glasses.



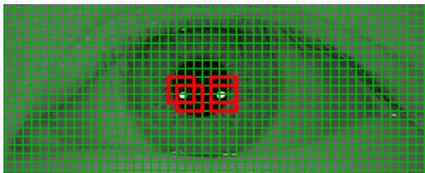
(a) Image without glasses



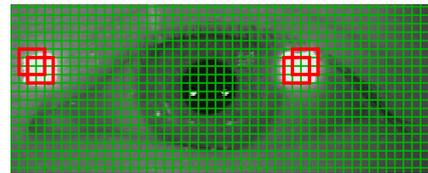
(b) Image with glasses

Figure 3.1: Comparison between natural reflections and artificial reflections caused by glasses

For the reflection detection we split the image into overlapping blocks of a fixed length. These blocks had to be big enough to ignore small natural reflections, but small enough to highlight bigger reflections which are caused by the glasses. On the CASIA database we used a block size of 30x30 pixels. In order that the blocks cover the reflection areas as well as possible we shifted the blocks by one-third of the block size which resulted in 10 pixels. In figure 3.2 you can see an example how the image is divided into blocks where the alignments for the blocks are marked green and the relevant blocks are marked red.



(a) Image without glasses



(b) Image with glasses

Figure 3.2: Visualization of the splitting process

The measuring of the brightness of each block was done by calculating the average grey value of the block and divided it by the average grey value of the complete picture. The usage of relative brightness scores instead of the average pixel value in a block makes it more robust against fluctuations in illumination. For instance, when an image is overexposed, the bright areas must not be identified as reflections. Since the average grey value of such images is relatively high, bright blocks lead to a significant lower reflection score than on images with a better exposure. At the end of this step we get a 2-D map of relative brightness deviation. The visualization of this map can be seen in figure 3.3.

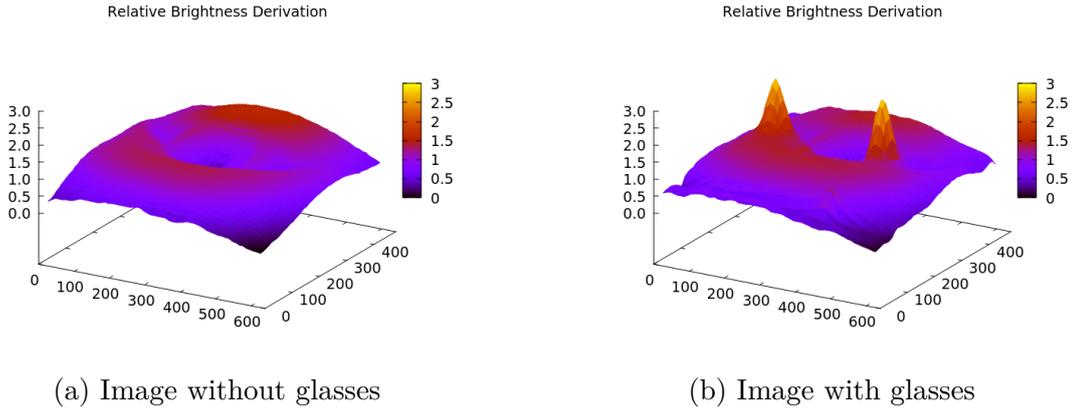


Figure 3.3: Visualization of the relative brightness deviation

The big reflections lead to high relative brightness scores whereas small reflections have very little influence on the scores. For the final score, only the highest brightness score is considered. This final score can be used for classification either with a specific threshold or with a machine learning approach such as SVM. The distribution of the reflection scores on the CASIA database can be seen in figure 3.4. In our first version with a fixed relative brightness threshold, a threshold of 2.0 turned out as suitable.

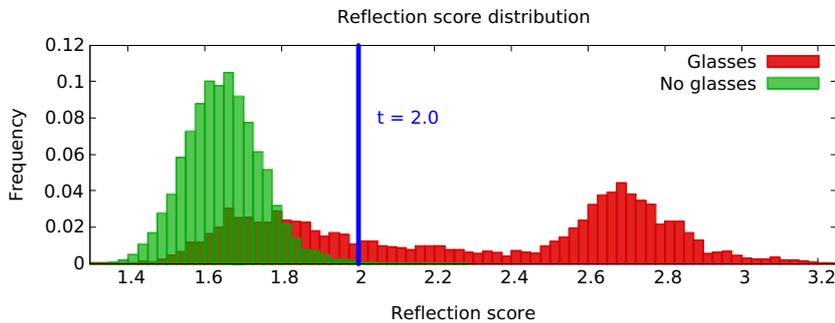


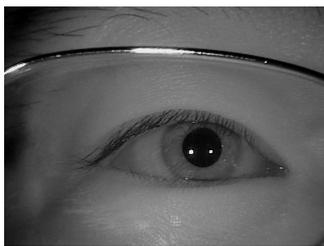
Figure 3.4: Histogram of the reflection score distribution

3.1.2 Edge detection

After we were able to measure strong reflections and used it for glasses classification, we searched for further features because not all glasses cause reflections. The next feature that we considered as useful were the edges of the glasses frame. In contrast to the natural small edges on the eye, these frame edges are more pronounced and longer. However, the measuring of these edges is more complex than the measuring of the reflection. In contrast to the human brain that can detect edges very easily, the edge detection on computer is significantly harder. Therefore, we split the edge measuring algorithm into several processing steps. The first 3 steps enhance the edges, so that the last 3 steps are able to measure them.

Edge enhancement

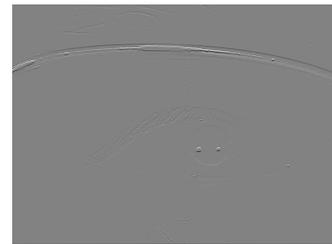
In the first step, we deliberate how to highlight the important frame edges while ignoring small natural edges of the eye. Therefore we analyzed the differences between the 2 classes of edges. While the natural edges are mostly short and have no specific direction, the frame edges are significantly longer and run commonly horizontally as long as the eye is aligned properly. The edge detection is commonly done by specific convolution matrices that measure the brightness gradient of a part of an image. One of the most popular edge detection operator is the Laplacian Operator, which is able to detect edges of both horizontal direction and vertical direction. As mentioned above, we only want to detect horizontal edges which is why we used another more simple edge detection operator, which can be seen in figure 3.5b. After applying the operator on the input image 3.5a, we obtain an image of the detected edges which is shown in figure 3.5c. Since these convolution matrices do only calculate the local brightness gradient, the output image is independent of the average brightness of the input image.



(a) Original image

0	0	0
0	-1	0
0	1	0

(b) Simple horizontal edge detection operator



(c) Image after edge operator¹

Figure 3.5: Visualization of the edge highlighting using a horizontal edge operator

The result of this process step is grayscale image with a average pixel value of 127.

Edge binarization

In the next step, we transformed the grayscale image into a black and white image by applying a relative threshold. Thereby, we ignore blurred edges and brightness transitions from dark to light. This means we accept only sharp brightness transitions from light to dark, mostly the bottom edge of a glasses frame.

The absolute threshold is selected, so that only a specified percentage of the brightest pixels is accepted. In our evaluations the best results were achieved by using the 3% of the brightest pixels. The pseudo code 1 shows, how the transforming of the relative threshold to the absolute threshold is done.

¹For visual inspection the contrast was enhanced

Input: The source image *src_image*

Input: The relative threshold *rel_threshold*

Output: The absolute threshold *abs_threshold*

accepted_pixel_number \leftarrow *pixel_number(src_image)* * (1 - *rel_threshold*)

histogram \leftarrow *create_histogram(dstImgPath)*

hist_sum \leftarrow 0

for *histogram_index* \leftarrow 0 **to** *length(histogram)* - 1 **do**

hist_sum \leftarrow *hist_sum* + *histogram*[*histogram_index*]

if *hist_sum* > *accepted_pixel_number* **then**

 | **return** *histogram_index*

end

end

Algorithm 1: Pseudo code for calculating the absolute threshold from the relative threshold for an input image

After calculation of the threshold, we applied it on the grayscale image to accept only relevant strong edges. Therefore, we set all pixel values that are less than the calculated threshold to 0 and all pixel values that are higher or equal to the threshold to 255. An example of this processing step can be seen in figure 3.6.

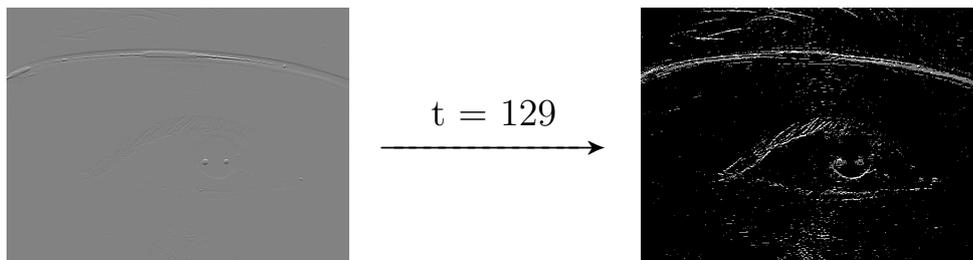


Figure 3.6: Applying the calculated threshold for binarization of the grey-value image

Dilatation

The dilatation process has the task to close the gaps between nearby edges. Due to artifacts caused by illumination or compression many edges are not recorded completely. However, the next processing steps needs long, connected edges to get correct results. In the dilatation process the white areas are expanded, so that small gaps between those are closed. For our implementation, we used the dilate function of the OpenCV [Ope17] library with a rectangular filter of the size of 7 x 7 pixels. These values were determined by trial and error and offered the best results. An example of this processing step can be seen in figure 3.7.

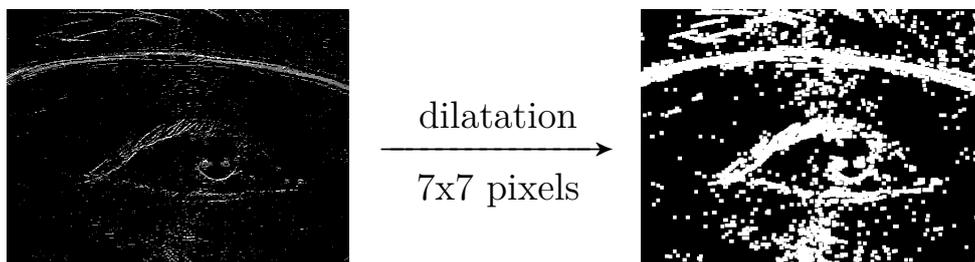
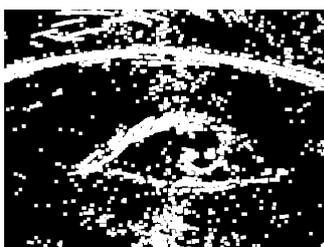


Figure 3.7: Dilatation process example

Outsorting of natural edges

Since the glasses frames are mostly at the margins of the images we analyzed only edges which begin or end at the outer regions of the image. On the CASIA database the best results were achieved by using the 10 percent of the left and right side and 10 percent of the bottom side, which can be seen in figure 3.8b. We ignored edges on the top because strong eyebrows were often wrongly detected as edges due to the high contrast between the eyebrows and the light skin. By applying the edge search mask on the input image we reduce the number of edges significantly. The result can be seen in figure 3.8c.



(a) Image with all edges



(b) Edge search mask



(c) Edges after sorting out

Assignment of pixels to edges

This processing step assigns each pixel of an edge to an edge id, so that the different edges are distinguishable. We used the flood fill algorithm with 8 directions. This means that the algorithm searches both for horizontal/vertical neighbour pixels and for diagonal neighbour pixels. The recursion-based algorithm assigns pixels that do not belong to a registered edge to either the edge of the neighbourhood pixels if these are already registered or, otherwise, a new edge if the neighbourhood pixels do not belong to a registered edge. The visualization of this processing step can be seen in figure 3.9.

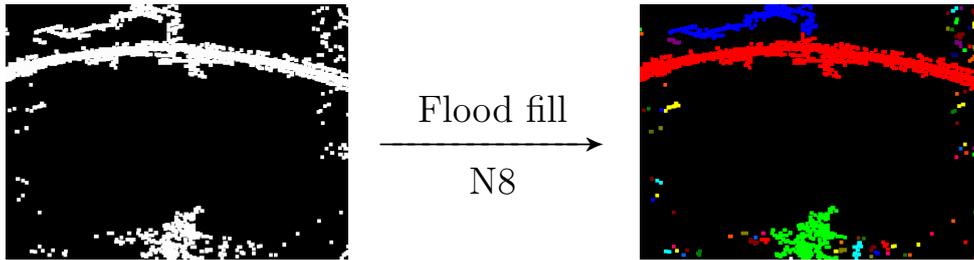


Figure 3.9: Visualization of the assignment of the pixels to edges

Edge measurement

At the end of the processing steps, we were able to measure the edges to classify whether the remained edges are a natural edge of the eye or an artificial edge of the glasses frame. For every remaining edge we measure the most left, the most right, the most top and the most bottom pixel to calculate the dimensions of the edges. This was done by an recursive algorithm, which iterates trough all pixels of an edge and sets the minimums and maximums of the positions of the pixels. We ignored edges whose width was less than a specific threshold to avoid that a small edge with only a few pixels gets an high ratio score. A minimum edge length of 120 pixels has proven to work well. As a result we get a list of edges with their dimensions and their ratio. By processing the example image 3.9, the generated list can be seen in table 3.1.

Edge id	width	height	ratio
0	639	111	5.76
1	279	66	4.23
2	142	100	1.42

Table 3.1: Example list of measured edges

As was already the case in the reflection metric, the highest ratio is used as edge metric, which can be used for classification either with a fixed threshold or an additional system like a SVM. In figure 3.10 you can see the distribution of edge scores on images of the CASIA database in which suitable edges could be detected. It should be mentioned that on 6.7 % of all images with glasses and 56.8 % of all images without glasses, no suitable edges could be detected.

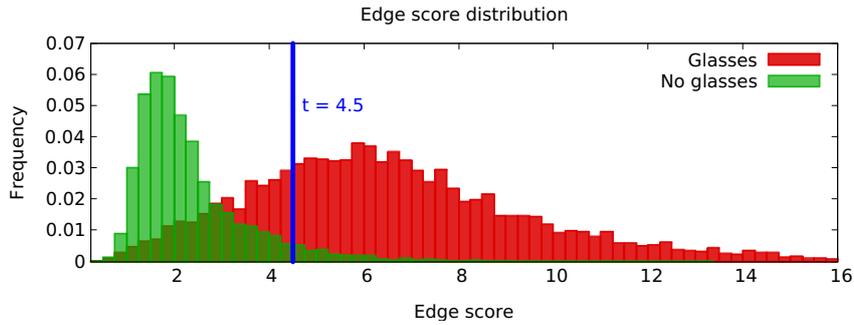


Figure 3.10: Feature score distribution of the explicit algorithm on the CASIA database

In our early versions, we used a fixed threshold of 4.5 for the CASIA database in which case the example image would be classified as glasses successfully.

3.1.3 SVM

While we used in our early version fixed thresholds to classify the 2 feature metrics of our algorithmic approach, we used later a Support Vector Machine (SVM) for classification. In contrast to a static threshold, which has to be determined expensively by many experiments and is relatively inaccurate, a SVM needs only a short time for training and achieve a high classification performance. Therefore, we used the output values of the algorithm as labeled training data for the SVM to create a trained SVM model. Afterwards, the model was used to categorize the tuple of edge detection score and reflection score into the categories glasses and no glasses. For our implementation we used the lightweight libsvm [CL11] library. We experimented with different kernel and cost parameter while training; the best results were achieved by using the RBF (radial basis function) kernel with the cost parameter of 10000.

3.2 Deep Learning

The second approach that we tried was a full machine learning approach. That means that we do not select suitable features by hand as we done it in the explicit algorithmic approach, but instead let the machine decide which features it wants to use and how the features are used. So we built a deep convolutional network using the deep learning framework Caffe [JSD⁺14a] and trained it with labeled images of the CASIA database.

3.2.1 Structure

The structure of our deep convolutional network is shown in figure 3.11. It can be divided into a feature extraction part, in which the algorithm learns features from the input image and a classification part, in which the network classify the extracted features. The deep learning network expects a grayscale input image with the fixed size of 320x240 pixels and outputs 2 metrics, which can be used for the final classification. Therefore, if necessary, the input images have to be scaled before they can be classified by the network. The used layers and their parameters were selected based on previous empirical experiments.

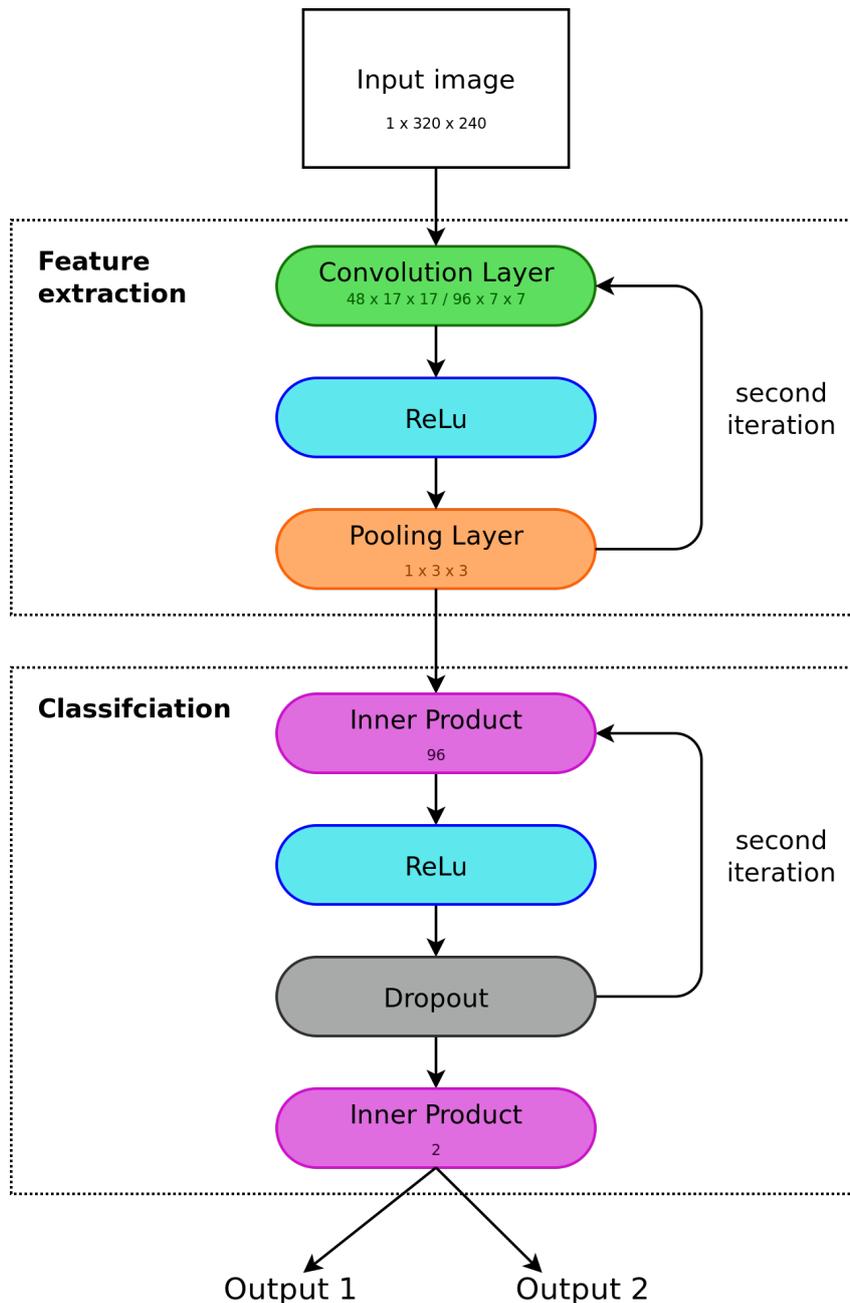


Figure 3.11: Structure of our deep convolutional network for glasses detection

Feature extraction The first part of the convolutional network contains layers for feature learning. It contains a sequence of a convolution layer, a ReLu (Rectified Linear Unit) layer and a pooling layer, which have to be run twice with different parameters. In the first iteration, 48 convolution filters with the size of 17x17 pixels are trained on the given input image. We think that a filter of this size should recognize frame edges well and further analyses showed that this filter size is a good choice. It is followed by a ReLu activation layer, which is a common way to improve the performance of a convolutional net. After the activation function, there is a pooling layer with a filter size of 3x3 pixels. This filter is applied with a stride of 2, which downscales the size of the input by factor of 4, which enhances the training and prediction performance. Furthermore, it prevents overfitting, which means, that the network learns too many variables from the training data, so that it is unable to classify unknown input data from other datasets. This sequence is repeated a second time with the difference, that the convolution layer consists of 96 filters with a size of 7x7 pixels, which is approximate a quarter of the size of the input data. At the end of the first part, the output data consists of 96 blocks of 4x5 pixels.

Classification The classification part reads the output data of the feature extraction part and learns how the features can be used for classification. Therefore, we used - similarly to the first part - a sequence of 3 layers: an Inner Product (IP) Layer, a ReLu layer and a Dropout layer. The IP layer consists of 96 neurons that are fully connected to the neurons of the second IP layer, which contains also 96 neurons. Between each IP layers are a ReLu activation function and a Dropout layers. While the activation function improves the classification performance, the Dropout layer discards small signals between the neurons and therefore prevents the network from overfitting. The second IP layer is fully connected to a small third IP layer which is the last layer of the network.

Since the two output values of the network were relatively easy to use for classification, we decided to use a fixed straight line to separate the output values into two classes.

3.2.2 Training

For training we created LMDB [lmd] container, which contained the labeled training set and trained the network on them. A step size of 20 000 was used, which means that the learning rate is adapted each 20 000 iterations. We run the training process over 100 000 iterations with a batch size of 32 images. The batch size is the number of images that the network is able to process in one iteration. A higher batch size can improve the training performance, as well as the accuracy of the network. At the beginning of the training a learning rate of 0.0001 was used and multiplied by 0.25 every step. These parameters were inspired by commonly used parameters of other networks such as LeNet [LeN].

3.3 Binarized statistical image features

In contrast to the other two approaches this statistical approach does not extract specific feature of the images, but analyzes the frequency of differences in brightness. Therefore, we used the BSIF (binarized statistical image features) [KR12] filter which extracts local brightness features, while ignoring global brightness differences. For classification we created a histogram of the filter output and used it as input for a SVM, which learns the relationships between the frequency distributions of the histogram and the ground truth class.

3.3.1 Functionality

In the first step of our implementation, we run the BSIF filter directly on the input image. For this we used the matlab BSIF implementation [bsi] from Juho Kannala and Esa Rahtu and transformed it into C++ code. An example of the filter process can be seen in figure 3.12.

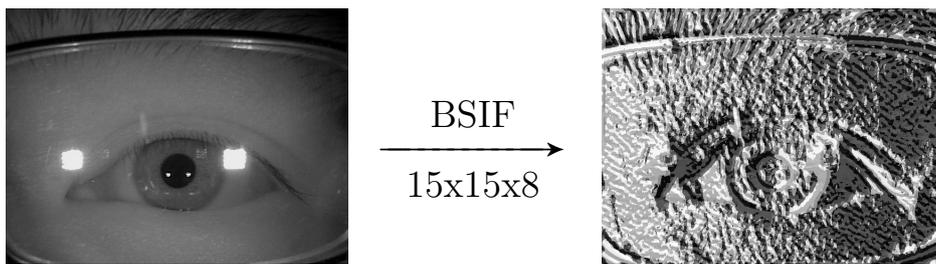


Figure 3.12: Applying the BSIF filter on the input image

The BSIF filter transforms local brightness differences into a gray value in the range of 0-255. After applying the filter, we created a histogram of the image (see figure 3.13) which represents the statistical distribution of the pixel values.

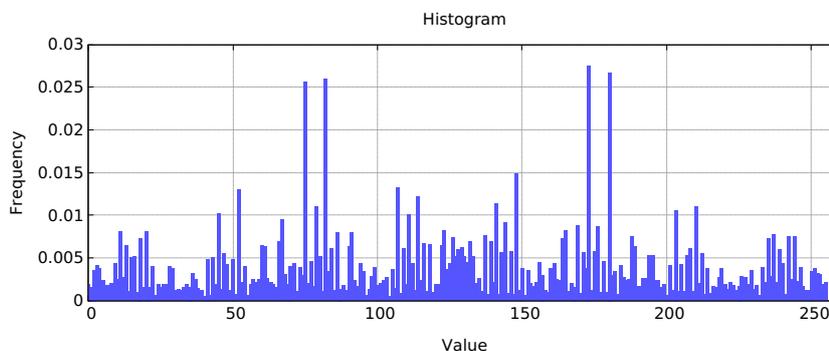


Figure 3.13: Applying the BSIF filter on the input image with glasses

The differences between histograms of images with glasses and histograms of images without glasses are rarely distinguishable by visual inspection, so we had to use a more complex classification approach than merely using a static threshold for specific values.

3.3.2 SVM

As we have a Support Vector Machine (SVM) already used for the explicit algorithmic approach we used it also for classification of the histograms. Therefore the SVM was trained to decide whether a histogram belongs to an image with glasses or to an image without glasses. In contrast to the commonly used RBF (radial basis function) kernel, which achieves good results on low dimensional vectors, for this purpose we used the linear kernel, because it is particularly suitable for high dimensional vectors such as histograms. We used 1000000 as cost parameter and an termination tolerance of 0.001.

After we developed and implemented our approaches, we continued with the evaluation of these to test how well they work.

Chapter 4

Evaluation

After we implemented our approaches, we evaluated their accuracy. We used a fixed procedure for our benchmarking on a large known database, so that we were able to achieve comprehensive results. In addition to testing the approaches individual, we have done a decision fusion to check how the approaches work together.

4.1 Procedure

For getting precise results of our evaluation we choose the CASIAv4-Thousand [Chi] database with 20 000 near-infrared images. The characteristics of this database can be seen in figure 4.1.

Database	CASIA-Thousand
Subjects	1000
Image number	20000
Image number with glasses	5336
Image number without glasses	14664
Quota images with glasses	36.39 %
Light spectrum	near-infrared
Colour model	grey
Photo resolution	640 x 480

Table 4.1: The characteristics of the CASIA database

In order to check the correctness of the classification decisions of the approaches, we had to find out which images contained glasses and which images did not contain glasses. Since the CASIA database does not contain such a groundtruth data for glasses, we had to categorize the images by hand in images with glasses and images without glasses. The accuracy was determined by calculating the ratio of the agreement between the groundtruth decisions and the decisions of the used approach.

In our evaluations we distinguish between the accuracy which was achieved on images without glasses and the accuracy which was achieved on images with glasses. Since the probability that an approach categorizes glasses wrongly and the probability that an approach categorize an images without glasses wrongly is not equal, the overall performance of an approach could depend on the ratio of the existing classes. So if an approach would categorize all images of the CASIA database as image without glasses, the overall accuracy would be 64 % because this is the ratio of images without glasses. To avoid this, we distinguish between the performance on different classes, so that we are independent of the occurrence frequency of each class on the used database.

Since the approaches are trained on the images of the CASIA database, we used cross validation to always evaluate the performance on previously unseen images. Therefore, we splitted the CASIA database into 4 parts - each contains 250 subjects and 5000 images - and built 4 training and evaluation sets. Each training set and its corresponding evaluation set is disjoint, which means that no image of the training set exists in the evaluation set and vice versa. The splitting process is visualized in figure 4.1. We tested each approach on every evaluation set and took the average accuracy as final performance metric.

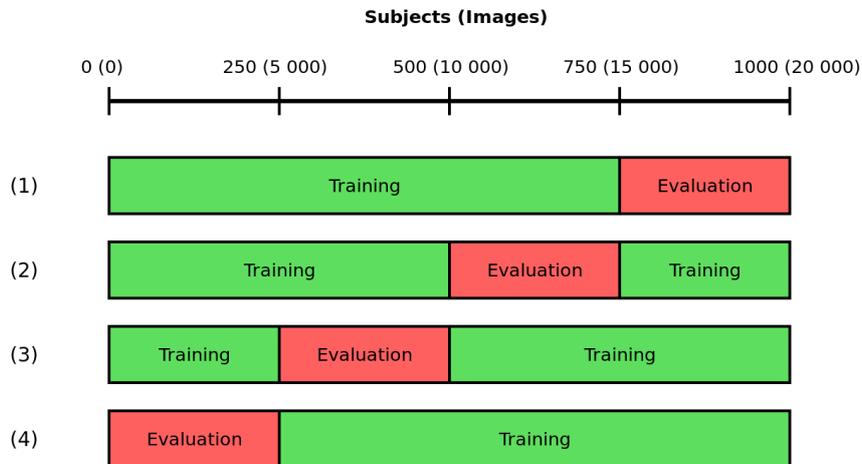


Figure 4.1: Splitting process for cross validation on the CASIA database

4.2 Results

The results of the evaluation on the CASIA database can be seen in table 4.2.

Approach	Accuracy		
	Glasses	No glasses	Overall
Explicit algorithm (fixed threshold)	91.00 %	96.86 %	94.73 %
Explicit algorithm (SVM)	92.37 %	97.18 %	95.43 %
Deep Learning	97.33 %	99.28 %	98.97 %
BSIF with SVM	98.54 %	97.79 %	98.08 %

Table 4.2: Results of the evaluation on the CASIA database

As you can see in table 4.2, the deep learning approach in the form of a deep convolution network has, with an accuracy of 98.97 %, the best overall performance, as well as the best classification performance on images without glasses (99.28 %). On images with glasses, the statistic approach with usage of the BSIF filters has the highest accuracy (98.54 %). The classification performance of the explicit algorithm is, both in the first version with fixed thresholds as well as in the later version with a followed SVM, worse than the other approaches, but nevertheless promising. While the ratio of the accuracy on images with glasses and images without glasses is almost equal when using the statistical approach, the other approaches have a better classification performance on image without glasses than on images with glasses.

4.3 Fusion

We evaluated how the classification accuracy is changing when using an AND, an OR and a majority vote [RNJ06] between the approaches. While the AND and OR was applied on 2 approaches, the majority vote was done on all three approaches. The results of the AND fusion in form of the overall classification performance on the CASIA database can be seen in table 4.3.

AND	Explicit algorithm (SVM)	Deep Learning	BSIF with SVM
Explicit algorithm (SVM)	95.43 %	97.36 %	97.46 %
Deep Learning	97.36 %	98.97 %	99.12 %
BSIF with SVM	99.46 %	99.12 %	98.08 %

Table 4.3: Results of the AND fusion on the CASIA database

The results of the fusion show that the classification performance is improved almost everytime by using an AND link. One exception is the fusion of the deep learning approach with the explicit algorithm, because the AND fusion of this approaches is worse than the single classification performance of the deep neuronal network. The best classification accuracy with the AND link was achieved by fusing the explicit algorithm with the BSIF approach, which results in a accuracy of 99.46 %. It is import to note that the classification accuracy on images with glasses suffered from the AND link, while the classification performance on images without classes was improved. Since on the CASIA database the ratio of images without glasses is higher, the overall performance was improved.

In the next step we did the OR link between the decisions of the approaches. The results can be seen in table 4.4.

OR	Explicit algorithm (SVM)	Deep Learning	BSIF with SVM
Explicit algorithm (SVM)	95.43 %	97.32 %	96.33 %
Deep Learning	97.32 %	98.97 %	96.01 %
BSIF with SVM	96.33 %	96.01 %	98.08 %

Table 4.4: Results of the OR fusion on the CASIA database

In contrast to the first fusion with an AND link the overall performance of the OR fusion often becomes worse than the individual performance. Only the classification performance of fusions with the explicit algorithm became better. As we mentioned above the AND and OR link favours either images of glasses or images without glasses. Since the OR link favours the decision that an image shows glasses, the performance on images with glasses increases while the performance on images without glasses become worse. The small number of images with glasses therefore leads to a worse classification performance when using the OR link.

In the last fusion experiment, we used the majority vote to fuse the classification decisions. In contrast to the simple AND and OR link, the majority vote does not favour one of the classes. Using the majority vote we were able to correctly classify 99.06 % of all images with glasses and 99.97 % of all images without glasses. Therefore, we were able classify 99.54 % of all images of the CASIA database correctly, which is the highest accuracy of all experiments.

The evaluation shows that all three approaches achieved a well classification accuracy. Although the deep learning approach has the best individual classification performance, the best result of the AND fusion was achieved by fusing the explicit algorithmic approach with the statistical approach. In addition, it turns out that some approaches are better in classifying images with glasses than classifying images without glasses and the other way round. In order to understand the classification accuracy better, we run further analysis on the wrongly classified images.

Chapter 5

Discussion

After we evaluated the classification performance of our approaches, we analyzed the results more comprehensively. The purpose is to understand the results of the approaches better, so that we are able to optimize them or change the environment, so that the algorithms work better.

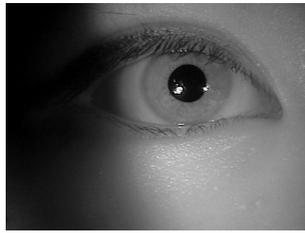
5.1 Error classes

First of all, we analyzed the images that were wrongly classified. We tried to find repeating patterns in images that lead to false classification and combined images with similar patterns into error classes. This allows us to find weaknesses of the different approaches, which helps us to conduct a more precise evaluation of the approaches.

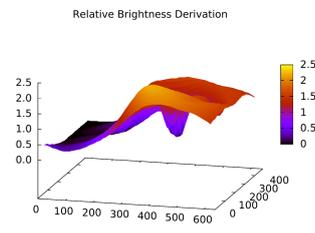
5.1.1 Explicit algorithm

For our explicit algorithmic approach, we were able to make precise investigations because every step is clearly separated from the other and can be visualized easily. We analyzed for each error class how the algorithm handles these images and at which point the algorithm fails to process them correctly. Afterwards, we considered solutions how we can solve the error in the failing process step, so that all images of these error class will be processed correctly.

The first error class, which can be seen in figure 5.1, is caused due to bad lightning conditions.



(a) Original image



(b) Failed processing step

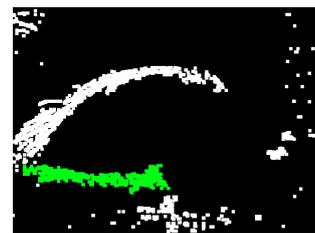
Figure 5.1: Wrongly classified image without glasses due to bad lightning conditions

This leads to dark parts of the image and therefore drops the average brightness level. As you can see in figure 5.1b, the differences between the brighter parts on the right side and the relatively low average brightness level is relatively high. In this example image without strong reflections, a maximum reflection score of 2.20 was calculated, which is higher than the used threshold of 2.0 and therefore, the image was wrongly classified as image of a glasses. As improvement of the reflection measuring, the algorithm could measure only local brightness differences instead of comparing the brightest parts with the global average brightness. This would make the algorithm more resistant against strongly changing brightness levels.

The next error class (figure 5.2) is caused due to the strong zoom on the eye, by which natural edges of the eye are at the margin of the image.



(a) Original image

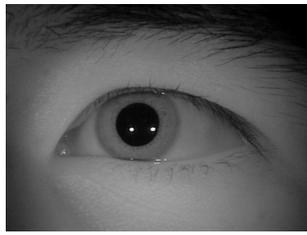


(b) Failed processing step

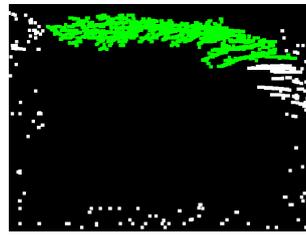
Figure 5.2: Wrongly classified image without glasses due to bad alignment of the eye

The algorithm expects that only frame edges are located at the margin of an image. If edges of the eye are located at the margins the algorithm detects them falsely as frame edge. In the example image the bottom left edge of the eye, marked as green area in figure 5.2b, was detected as frame edge. It can be prevented by adjusting the edge search mask, so that edges of the eye will be filtered stronger.

In the following error class, which can be seen in figure 5.3, dark eyebrows are detected as frame edge.



(a) Original image

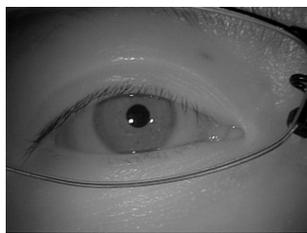


(b) Failed processing step

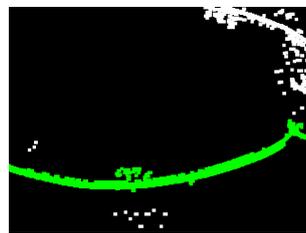
Figure 5.3: Wrongly classified image without glasses due to a strong eyebrows

Despite the adapted edge search mask, the high contrast between the dark eyebrows and the light skin leads to a wrong edge detection, which can be seen in figure 5.3b. It can be prevented, similarly as in the error class mentioned above, by adjusting the edge search mask, so the the eyebrows will be ignored better.

The next error class (figure 5.4) is caused due to a strong curvature of the frame edges which results in lower edge scores.



(a) Original image



(b) Failed processing step

Figure 5.4: Wrongly classified image of a glasses due to a strong curvature of the frame edges

The algorithm measures the width and height of each detected edge and takes the ratio as edge score. If an edge has a strong curvature, its height is relatively big which results in lower edge scores. Even though the example frame edge, which is marked green in figure 5.4b, is very long and relatively thin, its edge score is 4.15 which is lower than the used threshold of 4.5. This problem can be avoided by using a metric, which include both the ratio of the dimension, as well as the area of an edge. While many natural edges and artifacts have a high area/width ratio, the ratio between the area of an frame edge and its width is significantly lower. This can be used to optimize the edge detection performance.

The last error class (figure 5.5) occurs on transparent glasses frames.



Figure 5.5: Wrongly classified image of a glasses due to transparent glasses frames

Due to the small contrast between the transparent glasses frame and the skin, the edge cannot be highlighted enough, so that it can be measured correctly (see figure 5.5b). This problem is more complex than the other mentioned above and many processing steps are involved. For one thing a better edge detection operator can improve the edge contrast, so that the edge can be measured. On the downside better edge highlighting steps such as a different edge binarization threshold or other dilatation parameters can avoid this error.

5.1.2 Deep learning

Since the processing steps performed by deep learning systems are hard to understand, we were unable to find a specific point(s) of failure. We analyzed the correlations between the images to find out the causes for the false classifications. The found error classes can be seen in figure 5.6.

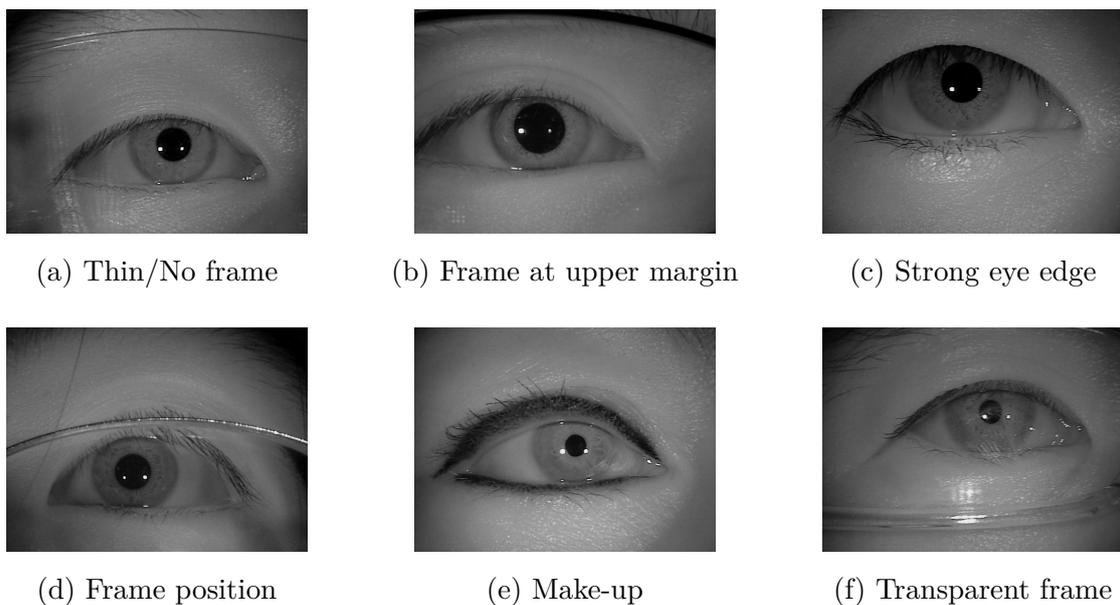


Figure 5.6: Error classes of the deep learning approach

Thin/No frame In these images, the frame has either a very thin frame or even no frame and has therefore only a low edge contrast. This makes it hard to identify these glasses. By enhancing the number of images with this case in the training set, the network should be better in identifying glasses with only a small or no frame.

Frame at upper margin This classification error occurs on images in which the glasses frame is at the upper margin, so that the top of the frame is cut. We assume that there were too few images of this case in the training set, so that the deep learning network did not learn how to handle this frame edges correctly.

Strong eye edge Like it was the case in the deep learning approach, a strong eye edge is very similar to a frame edge and is therefore often wrongly identified as frame edge of glasses. One possibility to prevent this, is to adapt the light conditions, so that the contrast of this edge is lower and it will be not detected as strong edge of a frame.

Frame position This error class exists only for the deep learning approach and occurs on images in which the frame edge is not located at the upper/lower part of the image, but in the middle part where the network does not expect frame edges. As it was mentioned in the first error class, these infrequent cases can be avoided by adapting the training data, so that the deep convolutional network learns that glasses frames exist also in the middle part of an image.

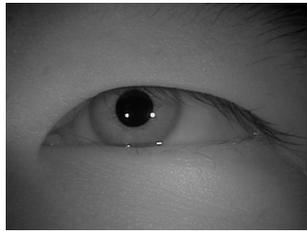
Make-up On images in which the person wears make-up, the convolution network mistakes the dark eye margins as frame. This error class can be avoided by enhancing the number of such images in the training data, so that the network learns the differences between the edges caused by make-up and the edges caused by glasses frames.

Transparent frame As it was the case for the explicit algorithmic approach, the edge detection often fails on transparent frames due to the low contrast between the light frame and skin. This error class can be avoided by adapting the light conditions, so that the contrast between the frame and the skin is enhanced. Another possibility is to create a more complex neuronal network that is able to detect such frames more precisely.

5.1.3 Statistical approach with BSIF

Since the statistic approach has only a few parameters such as filter, filter size and SVM parameters, it is hard to improve it based on the error classes. However, it is important to know on which images the classification fails, so that these images can be avoided. We have found 3 error classes by visual inspection.

The most common error class (see figure 5.7) is caused by a strong upper eye edge. As you can see in figure 5.7b, the upper edge of the eye has a similar contrast as frame edges, so that the SVM classified it wrongly as image with glasses. This problem can be avoided by ignoring edges which are too close to the pupil.



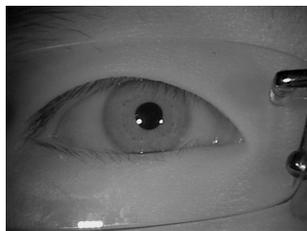
(a) Original image



(b) Image after applying the BSIF filter

Figure 5.7: Wrongly classified image without classes due to a strong upper eye edge

The next error class, which can be seen in figure 5.8 occurs on images that have either no glasses frame or a very thin frame. As you can see in figure 5.8b, the upper edge is weakly pronounced, the bottom frame edge is only pronounced strongly on the right side. The small recognized part of the edge is not sufficient to classify the histogram correctly. The usage of a more sensitive filter could help to reduce the number of images of this error class.



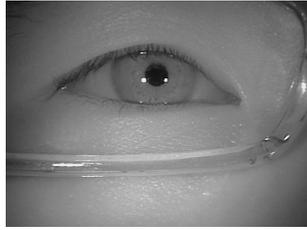
(a) Original image



(b) Image after applying the BSIF filter

Figure 5.8: Wrongly classified image with glasses due to a thin glasses frame

The last error class is caused by transparent frames. Like it was the case with the edge operator of the explicit algorithmic approach, the transparent frame has only a low contrast to the human skin and is therefore hard to recognize by a filter. As it is visualized in figure 5.9, the BSIF filters are unable to detect a clear consistent edge, which is needed to change the histogram, so that the SVM classifies it correctly. One possible solution to solve this problem is to change the lightning position, so that there are no reflections on the frame and the frame becomes darker. This would thus enhances the contrast between the frame and the skin.



(a) Original image



(b) Image after applying the BSIF filter

Figure 5.9: Wrongly classified image with classes due to a transparent glasses frame

5.2 Strengths and weaknesses

For comparing the different approaches, we analyzed their strengths and weaknesses. Since every approach has other strengths or weaknesses, we created 5 valuation categories:

- classification accuracy
- adaptability
- transparency
- throughput
- expandability

For every category we are able to directly compare the approaches to each other.

5.2.1 Classification accuracy

As it was mentioned in the evaluation chapter 4, all three approaches achieve a minimum classification accuracy of 95 % on the CASIA database. This shows that all three approaches are suitable for detecting glasses, whereby each approach has its own weaknesses. While the deep learning approach has the best single classification performance on the CASIA database, the best result of an and/or fusion of 2 approaches (99.46 %) was achieved by fusing the explicit algorithm with the statistical approach using the BSIF filter. It means that these two approaches complement each other better than all other combination of two approaches with the deep convolutional network. The majority-vote fusion improved the classification performance slightly to 99.54 % which shows, that the deep learning approach can only rarely supplement the fusion of the other two approaches.

Further analysis showed that while the explicit algorithm is vulnerable to changing exposures and needs a well aligned eye with straight frame edges to work well, the deep learning approach needs the frame at a specific position and is vulnerable to make-up. The statistical approach with the BSIF filter needs strong frame edges and ignores strong

reflections, which could be used for a more stable classification performance. All three approaches have some problems with glasses that have either only a very thin frame or even no frame, which makes it hard to detect the edges of a glasses. Further transparent frames are also hard to detect for all of the tested approaches because they are relatively bright and have reflections that lead to a low contrast between the skin and the frame, which makes it hard to detect it. In those cases the explicit algorithm has the advantage that it is able to use potential reflections to classify images where no frame edges could be detected correctly. In addition, all approaches detect strong eyelid edges falsely as frame edge.

These analysis showed that the classification performance of each approach strongly depends on the database for which it is used. While on a database with rare reflections the explicit algorithm would fail significantly more often, on a database with a better eye alignment and more reflections on glasses it would have a better performance. As a result, the selection of the right approach/the right fusion is database specific and can reach a different outcome on another database.

5.2.2 Adaptability

An important factor is the adaptability of an algorithm on a specific database. For each database there can be used other hardware for the capturing process with other image resolutions and other distances between the eye and the sensor. Since the algorithms are not robust against every factor, it should be possible to optimize them for specific factors such as image resolution or eye alignment. However, the more parameters exist the more difficult it is to find the right one. Consequently, you cannot achieve a simple, highly robust algorithm, which can be easily adapted to many factors.

The statistical approach with the BSIF filters has the least parameters that can be adapted. While it is relatively robust against factors as different exposures or edge alignments, the downstream SVM has to be retrained for each new database. Although it is relatively easy to create a new SVM model with the usage of provided scripts, it is necessary to have a training set. Furthermore, the size of the used filter has to be adapted to the dimensions of the images. While the size of many filters can be changed freely, the selection of the BSIF filter of the used BSIF implementation is limited.

The deep convolutional network has the highest adaptability due to many parameters that can be set. In contrast to the other two approaches, more background knowledge is needed to find parameters that work properly. For small changes of the database such as other lightning conditions, it is sufficient to retrain the convolution network on a training set of the new database, which does not require much background knowledge about the network structure. When using a different image ratio or image resolution as input, it

could be reasonable to adapt the network parameter or even the network structure to get good results. However, this needs much background knowledge about deep convolutional networks and is time consuming.

In contrast to the statistical approach, which is relatively robust against several factors, the explicit algorithm needs many parameters to work properly. On the one hand, the high number of parameters enhances the adaptability on a database, because the algorithm can be adapted precisely on specific features of a database, on the other hand it makes the algorithm vulnerable against small changes, such as changing lightning conditions. Finding the right parameter for a specific database is time consuming, because they have to be set by trial and error. However, in contrast to the parameter of the deep learning approach, it is not required to have specific specialist background knowledge. Since each processing step is separated, the suitable parameters can be determined by debugging step by step.

5.2.3 Transparency

The next evaluation category is about the transparency of the different approaches. For understanding the decisions and being able to improve the algorithms, it is necessary to know which features the algorithm uses and how the final decision is reached.

While most explicit algorithms can be debugged relatively easy, the machine learning algorithms such as deep learning and SVM are more abstract and therefore harder to understand. The highest transparency has the fully explicit approach, because it extracts defined features and consists of a series of separated processing steps. Each processing step can be visualized and is understandable. If an image was wrongly classified, it is easy to figure out the failed processing step. The statistical approach is harder to understand, because it abstracts from the single features such as edges to one big histogram for classification. Thus, it is hard to say which features are decisive for the final decision.

But even if there is a downstream SVM, whose decisions are hard to debug, you can visualize the feature metric distribution and therefore estimate the decision. In figure 5.10, you can see the distribution of output scores of the explicit algorithmic approach. As you can see, the points of glasses are divided into two even groups: points of images with reflections and points of images without reflections. This means that every second image of the CASIA database can be classified by analyzing the reflections.

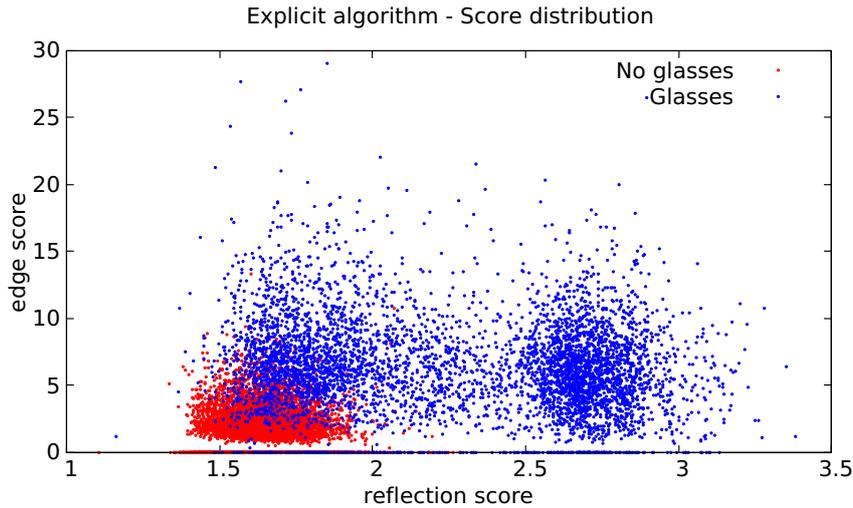


Figure 5.10: Feature score distribution of the explicit algorithm on the CASIA database

In figure 5.11, you can see the average histogram of images with glasses and images without glasses with the associated standard deviation. This diagram shows how the histogram is responding to the usage of glasses and no glasses and make it possible to understand the decision of the SVM. By analyzing the different algorithms, such information can be used to optimize the algorithms for the specific dataset.

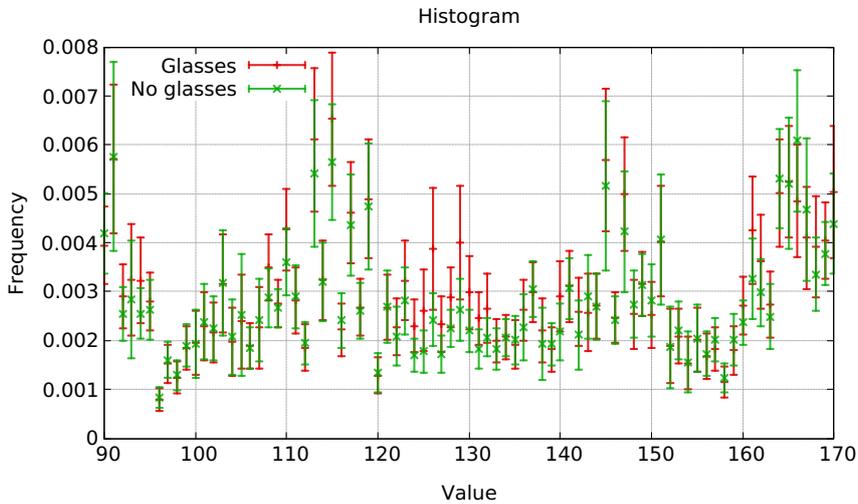


Figure 5.11: Part of the average histogram from the statistical approach with the BSIF filter on the CASIA database

For understanding the decisions of deep learning networks, you can visualize the weights of the learned filters. This allows you to look into the network to understand which features are used for classification. Especially the convolution filter can be visualized well. In figure 5.12, you can see that the deep convolutional network that we have learned on the first training set learns to use horizontal brightness differences as features. You can recognize this by the horizontal distributed weights of the convolution filters. Thus

the deep learning approach uses similar features as the explicit algorithmic approach. In addition to the filter visualisation, you can plot the distribution of the output data to check how sure the network is with its classification decision of an image.



Figure 5.12: Learned convolution filter in our glasses identification network

5.2.4 Throughput

Another factor is the throughput performance. It depends on the complexity of the different processing steps, the number of these steps and how they can be parallelized. Although the deep learning approach has the most processing steps (13 layers), its steps can be parallelized very well, so that it is still the fastest approach. We tested it on a CPU, but it is also able to run on a GPU with a much greater performance. In our tests the explicit approach was slightly slower than the deep learning network. The single steps are mostly simpler than the processing steps of the convolution network, but can be less parallelized. In our implementation we parallelized only the filter operation, such as for the reflection calculation and the edge detection operator, because these were the most expensive processing steps and could be easily parallelized. By optimizing the algorithm for parallelization, the throughput performance can be further enhanced. The slowest approach was the statistical approach by using the BSIF filter. Although this approach has the less processing steps, the filter process is very expensive. Since we used a transformed Matlab implementation which supports only single threaded processing, the performance is slower than it could be. As mentioned above, the filter processes such as applying the BSIF filter can be parallelized very well so that the throughput of this approach can be increased strongly.

5.2.5 Expandability

Besides the adaptability, it should be also possible to expand the algorithms so that they are able to process further features or are able to process specific features better.

The statistical approach can be expanded by adding further filters. As it is already being done by using the BSIF filter, the output of each filter can be summarized to multiple histograms, which can be appended to one another, so that there is one big histogram which can be used for classification. At example, we found out by analyzing the error classes that this approach ignores strong reflections on images, which could be used to detect images with glasses. The reflection features can be used by adding a filter which responses to big bright areas like it was done in the explicit algorithm. This could increase the classification performance significantly.

Expanding the deep convolutional network can be done by adding additional layers so that the network is able to learn more complex relationships of the images. As it was already mentioned for the adaptability, the change of the network need much background knowledge and is often be done by trial and error which need much of time.

Since the explicit algorithm is constructed modularly, there can be added further processing steps for already used features or completely new feature metrics. For example, there can be added a new processing step ahead of the edge measurement that filter eyebrows, so that they are ignored and do not influence the final decision.

5.2.6 Summary

To summarize the strengths and weaknesses of the approaches, we organized them in a scale from - - to ++. A strong weakness was mapped to - -, a small weakness/strength to -/+ and a strong strength to ++. The results can be seen in table 5.1.

Category	Explicit algorithm	Deep learning	Statistical approach (BSIF)
Classification accuracy	-	++	+
Adaptability	+	++	+
Transparency	++	- -	-
Throughput	+	++	-
Expandability	+	++	+

Table 5.1: Summary of the strengths and weaknesses of the proposed approaches

Chapter 6

Conclusion

Previous experiments have shown that glasses cause a significant deterioration of iris recognition performance. Among other things, the usage of glasses leads to more segmentation failures and, on average, smaller visible iris area. This effect can be observed irrespective of the image quality being otherwise fine. These findings motivate development of approaches for automatic detection of glasses.

We developed 3 different approaches for automatic glasses detection. After implementing all approaches and creating groundtruth labels, we evaluated the classification performance on the CASIA database and analysed the wrongly classified images in order to understand the strengths and weaknesses of each of the proposed algorithms. The evaluation shows that the deep learning network has the best single-algorithm classification accuracy (98.97 %) on the CASIA database, whereas the best overall classification performance on the CASIA database (99.54 %) was achieved by using the majority vote of all three approaches. It turns out that the explicit algorithm and the statistical approach with the BSIF filter complement each other better than all other tested combinations. They achieved the best classification accuracy (99.46 %) by using a combination of 2 approaches. The fusion of all three approaches was rarely better than this fusion using the AND operation. It was shown that all approaches have their own classification weaknesses and strengths and that most classification weaknesses of one algorithm can be compensated by the strengths of another approach through decision-level fusion. The greatest difficulty for all developed approaches was the detection of glasses which have either a very thin or transparent frame, which makes it hard for classification by edge detection. While the explicit algorithmic approach supplies the most comprehensible results, the deep learning approach has the best adaptability and the statistical approach has the best classification accuracy on images with glasses. In other words, the three proposed approaches are complementary.

This research can be used to either improve the recognition performance on unsupervised recognition systems or to aid automation of the quality control of the acquisition process. On systems which accepts glasses because they do not have an automatic way of glasses detection and a human controlled capturing process is unacceptable, the recognition performance can be improved by detecting glasses automatically and rejecting these.

As an example application, take an unsupervised border control system at an airport. When a user wants to verify his/her identity, the systems acquires an image of the user's eye and checks by applying our approaches whether the person wears glasses or not. If glasses were detected, the system could abort the verification process and show a message that the person should take off his/her glasses.

Another application is iris recognition on mobile phones such as the iris recognition technology of the Samsung Galaxy S8 [sam] to unlock the phone. There are some reports that using glasses can result in longer runtime of the identification process. We expect that the technology uses multiple frames for the iris recognition until the user is accepted. Using the presented approaches for glasses detection, which work with only a single frame, the user can be asked to take his/her glasses off and retry the identification. This can decrease the runtime of unlocking process and makes it more comfortable.

By applying this procedure on the CASIA database, we were able to improve the iris recognition performance from 9.19 % EER to 6.61 % EER.

For the future work there should be developed techniques to improve the feature extraction on images with glasses so that they need not to be rejected. In addition it is needed to investigate further causes for the performance deterioration of iris recognition systems when using glasses besides the causes that were mentioned in this thesis. Further it should be investigated how well the mentioned approaches perform on greater parts such as complete facial images.

Bibliography

- [bac] *Neural Networks and Deep Learning: Chapter 2 how the backpropagation algorithm works*, <http://neuralnetworksanddeeplearning.com/chap2.html>, Accessed: 2017-05-29.
- [Boo17] Boost, *Boost C++ Libraries*, <http://www.boost.org/>, 2017, Last accessed 2017-03-09.
- [bsi] *Bsif: binarized statistical image features*, <http://www.ee.oulu.fi/~jkannala/bsif/bsif.html>, Accessed: 2017-06-02.
- [Chi] Chinese Academy of Sciences' Institute of Automation (CASIA), *Casia-irisv4*, <http://biometrics.idealtest.org/>, Accessed: 2017-03-09.
- [Chr04] A. Christmann, *Regression depth and support vector machine*, Universität Dortmund, 2004.
- [CL11] C. Chang and C. Lin, *LIBSVM: A library for support vector machines*, ACM Transactions on Intelligent Systems and Technology **2** (2011), 27:1–27:27, Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [con] *deeplearning4j: Convolutional networks*, <https://deeplearning4j.org/convolutionalnets>, Accessed: 2017-05-29.
- [Dau04] J. Daugman, *How iris recognition works*, IEEE Transactions on Circuits and Systems for Video Technology **14** (2004), no. 1, 21–30.
- [fDA15] Instituts für Demoskopie Allensbach, *Brillenstudie 2014*, Tech. report, Instituts für Demoskopie Allensbach, 2015.
- [ima] *Neural Networks and Deep Learning: Chapter 6 deep learning*, <http://neuralnetworksanddeeplearning.com/chap6.html>, Accessed: 2017-05-29.
- [JBP06] A. Jain, R. Bolle, and S. Pankanti, *Biometrics: personal identification in networked society*, vol. 479, Springer Science & Business Media, 2006.
- [Joi06] Joint Technical Committee ISO/IEC JTC 1, Information technology, Subcommittee SC 37, Biometrics, *Iso/iec 19795-1:2006 information technology -*

- biometric performance testing and reporting - part 1: Principles and framework*, ISO ISO/IEC 19 795-1:2006, International Organization for Standardization, 04 2006.
- [Joi07] ———, *Iso/iec 19 795-1:2006 information technology - biometric performance testing and reporting - part 2: Testing methodologies for technology and scenario evaluation*, ISO ISO/IEC 19 795-1:2006, International Organization for Standardization, 02 2007.
- [Joi12] ———, *Iso/iec 2382-37:2017 information technology - vocabulary - part 37: Biometrics*, ISO ISO/IEC 2382-37:2017, International Organization for Standardization, 12 2012.
- [Joi15] ———, *Iso/iec 29794-6:2015 information technology - biometric sample quality - part 6: Iris image data*, ISO ISO/IEC 29794-6:2015, International Organization for Standardization, 2015.
- [JSD⁺14a] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, *Caffe: Convolutional architecture for fast feature embedding*, Proceedings of the 22Nd ACM International Conference on Multimedia (New York, NY, USA), MM '14, ACM, 2014, pp. 675–678.
- [JSD⁺14b] ———, *Caffe: Convolutional architecture for fast feature embedding*, arXiv preprint arXiv:1408.5093 (2014).
- [KR12] J. Kannala and E. Rahtu, *Bsif: Binarized statistical image features*, Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), Nov 2012, pp. 1363–1366.
- [KZSC10] N. D. Kalka, J. Zuo, N. A. Schmid, and B. Cukic, *Estimating and fusing quality factors for iris biometric images*, IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans **40** (2010), no. 3, 509–524.
- [LeN] *Training lenet on mnist with caffe*, <http://caffe.berkeleyvision.org/gathered/examples/mnist.html>, Accessed: 2017-05-30.
- [LLBK01] S. Lim, K. Lee, O. Byeon, and T. Kim, *Efficient iris recognition through improvement of feature vector and classifier*, ETRI journal **23** (2001), no. 2, 61–70.
- [lmd] *Lightning memory-mapped database*, <https://symas.com/lightning-memory-mapped-database/>, Accessed: 2017-05-30.
- [ODGS16] N. Othman, B. Dorizzi, and S. Garcia-Salicetti, *Osiris: An open source iris recognition software*, Elsevier, 10 2016.

- [Ope17] OpenCV, *OpenCV Library*, <http://opencv.org/>, 2017, Last accessed 2017-03-09.
- [PA05] H. Proença and L. Alexandre, *Ubiris: A noisy iris image database*, Image Analysis and Processing–ICIAP 2005 (2005), 970–977.
- [RNJ06] A. A. Ross, K. Nandakumar, and A. Jain, *Handbook of multibiometrics*, International Series on Biometrics, Springer US, 2006.
- [RUWH16] C. Rathgeb, A. Uhl, P. Wild, and H. Hofbauer, *Design decisions for an iris recognition sdk*, Handbook of Iris Recognition (Kevin Bowyer and Mark J. Burge, eds.), Advances in Computer Vision and Pattern Recognition, Springer, second edition ed., 2016.
- [sam] *Iris recognition on galaxy s8*, <http://www.samsung.com/au/iris/>, Accessed: 2017-06-27.
- [Sch16] U. Scherhag, *Presentation Attack Detection for State-of-the-Art Speaker Recognition Systems*, Master’s thesis, Hochschule Darmstadt, 2016, p. 26.
- [SMRO14] A. F. Sequeira, J. C. Monteiro, A. Rebelo, and H. P. Oliveira, *Mobbio: A multimodal database captured with a portable handheld device*, 2014 International Conference on Computer Vision Theory and Applications (VISAPP), vol. 3, 01 2014, pp. 133–139.
- [sup] *deeplearning4j: Introduction to deep neuronal networks*, <https://deeplearning4j.org/neuralnet-overview>, Accessed: 2017-05-29.