Todo list

Noget med Computer Vision

Project Report

Group 18gr842

Aalborg University
Vision, Graphics and Interactive Systems

This report is compiled in LaTeX. Additionally is Mathworks MATLAB, Adobe Illustrator, Lucidcharts.com, Inkscape, and Autodesk Eagle used to draw figures, schematics, and charts.

AALBORG UNIVERSITY
STUDENT REPORT

Vision, Graphics and Interactive Systems
Aalborg University
http://www.aau.dk

Title:
Noget med Computer Vision

Theme:
Computer Vision

Project Period:
Spring Semester 2018

Project Group:
Group 18gr842

Participants:
Marike Koch van den Broek
Shagen Djanian
Niclas Hjorth Stjernholm

Supervisor:
Kamal Nasrollahi

Number of Pages: 38

Date of Completion:
May 30, 2018

Abstract:

□

Preface

Aalborg University, May 18, 2018

Marike Koch van den Broek
<mvanam13@student.aau.dk>

Shagen Djanian
<sdjani14@student.aau.dk>

Niclas Hjorth Stjernholm
<nstjer14@student.aau.dk>

Contents

Glossary

CDF Cumulative Distribution Function. 26
CNN Convolutional Neural Network. 5, 6, 7, 13, 14, 30, 31

DBN Deep Belief Network. 13
DeepID Deep hidden identity features. 5, 6
DeepID2 Deep IDentification-verification features. 6, 7

FC Fully Connected. 31
FCN Fully Convolutional Network. 14, 15

HMM Hidden Markov Model. 5

KNN K-Nearest-Neighbours. 13, 16, 28

LDA Linear Discriminant Analysis. 5, 13, 28, 29
LFW Labeled Faces in the Wild. 6, 7

MCS Multiple Classifier Systems. 16

NIR Near Infra-Red. 8, 11, 12, 15, 21

PCA Principal Component Analysis. 5

ReLU Rectified Linear Unit. 30

SPDNN Semi Parallel Deep Neural Networks. 15
SVM support vector machines. 5, 13, 16, 28

VL Visible Light. 8, 11, 12, 15, 21

Chapter 1

Introduction

The problem of protection of physical elements or information by limiting the access to only the people is well known. The central challenge to the problem of security is the challenge of verifying the identity of a person trying to acquire access. The emergence of biometric techniques has induced an increasing interest in biometric-based security rather than knowledge-based or token-based security. This is mainly due to the fact that the more traditional methods for security systems are easier breached or spoofed [Ross and Jain, 2003]. Through the last decades researchers have investigated identity verification based on different biometric modalities. In the last decade investigations have been conducted in combining several biometric modalities in one system with the purpose of creating a system that performs better than the ones only utilising a single one. Results shows that combining modalities performs better than any of the modalities seperately [Chen and Te Chu, 2005]. However, increased accuracy is not the only benefit of utilising multiple biometric traits. More modalities increases the universality of the system and decreases the influence of noisy measurements [Ross and Jain, 2003].

One of the fields where biometric-base security is increasingly applied is security for handheld devices such as smart phones. Although technology is advancing and mobile devices are equipped with still more advanced components, the computing power and the quality of the data from sensors, are both constrains of mobile devices. These limiting factors makes it more challenging to make successful biometric-based identity verification on mobile devices rather than in other applications.Though the use of machine learning for image processing is well known, it has only scarcely been applied on data obtained by cameras on mobile devices. **?** presents different machine learning methods applied on iris images obtained by smartphone. Bazrafkan et al. [2017] applies deep learning for segmentation purposes on a database containing images acquired using a smart phones among others.

add source?

The work described in this report strives to make a system for identity verification based on multiple biometric traits for use on mobile devices. The biometric traits used are the iris and face. More specifically, this choice is made due to the arguments found in literature that iris is very distinctive, while face is non-invasive [**?**].

describe more challenges (found in literature) before talking plan for solution

argue some more

Chapter 2

Research

Hat

2.1 Face Recognition

The first computer based face recognition was made in 1973. This was based on a feature approach, meaning the program identifies basic face features such as mouth, eye, and nose placement. From here three different types of approaches were made, namely a holistic, feature extraction and a hybrid approach.

The holistic approach encodes the entirety of a face and then identifies using template-matching, the feature extraction approach extracts a defined amount of features from the face, whereas the hybrid method uses both template-matching and feature extraction [Wechsler, 2007]. In 1990, Principal Component Analysis (PCA) was introduced for holistic face recognition. The PCA approach makes use of eigenfaces, each eigenface represents a component a face is encoded. But as Wechsler [2007] claims, Linear Discriminant Analysis (LDA) is a more effective suitable approach for face identification and authentication. Another holistic approach is using support vector machines (SVM) for face recognition [Wechsler, 2007].

The feature approach gave way for what is now known as recognition-by-parts, which uses the features and a global structure to link these features. A structure for linking 2D features is the Hidden Markov Model (HMM). PCA is also used in this approach, but is used to model shape or texture of the face.

Niclas: We need an "introduction" to the subsection

2.1.1 DeepID

Deep hidden identity features (DeepID) is a Convolutional Neural Network (CNN) which aims to use feature extraction for face identification and verification. It detects five facial landmarks; the two eye centres, the nose tip, and the two mouth corners. The network is made of four convolutional layers with max-pooling, which are used to extract features hierarchically. These are followed by the fully-connected DeepID layer and a softmax output layer to indicate identity classes. The feature extraction and recognition is done in two steps, where the first feature extraction is learned with the target of face identification [Sun et al., 2014a].

In the CNNs the neuron number of the last hidden layer in the network is much smaller than that of the output layer. This is done, to better classify faces [Sun et al., 2014a]. The network extracts low-level features in the bottom layers, where feature numbers decreases for each layer. In opposition, the high-level features are formed in the top layers.

The network is tested using the Labeled Faces in the Wild (LFW) database. This database is images of faces from different angles and scenarios consisting of 13.233 images [Huang et al., 2007]. It achieves 97.45% accuracy on this dataset, requiring weakly aligned faces [Sun et al., 2014a].

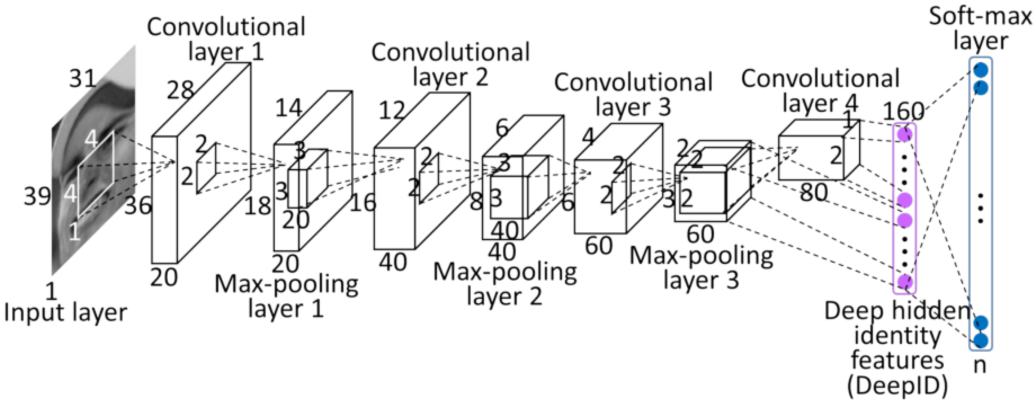


Figure 2.1: Structure of the CNN used in DeepID [Sun et al., 2014a]

2.1.2 DeepID2

Deep IDentification-verification features (DeepID2) is an expansion upon DeepID and is a deep Convolutional Neural Network used for face identification and verification. This is done by using feature extraction.

Just like DeepID, DeepID2 uses four convolutional layers but only the first three uses max-pooling. It uses 400 face patches instead of 60 [Sun et al., 2014a,b] and detects 21 landmarks of the face.

The DeepID2 layer is after the four convolutional layers. This layer is learned under two supervisory signals. The first is identification classifying the images into identities. The second is face verification which manipulates the DeepID2 data to be similar to a matching identity should this be the same.

DeepID2 also uses the LFW database and achieves a 99.15% accuracy.

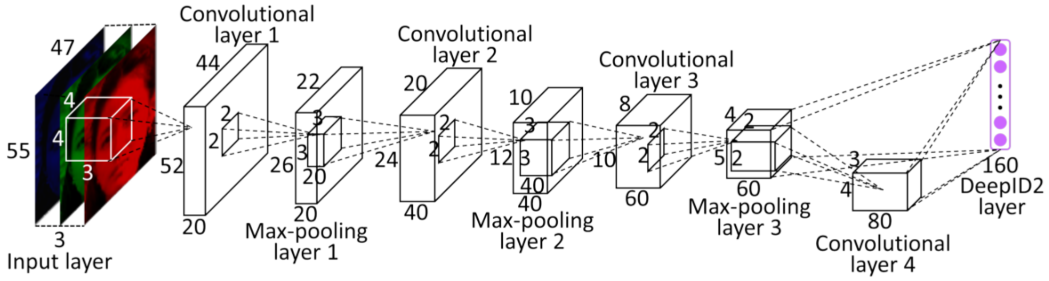


Figure 2.2: Structure of the CNN used in DeepID2 [Sun et al., 2014b]

2.1.3 DeepID3

DeepID3 is a further expansion of both DeepID and DeepID2 but is also drawing on some from elements from VGG net and GoogleNet [?]. The qualities from these two networks is the use of stacked convolution and inception layers. DeepID3 is in general a deeper network than DeepID2 and its expansion DeepID2+.

DeepID3 resembles DeepID2 in the use of adding supervisory signals to early layers.

? proposes two different networks with DeepID3. One is using eight convolutional layers with max pooling after every other convolutional layer. The second network has four convolutional layers with max pooling after every other, following are five inception layers.

DeepID3 is also tested on the LFW dataset with an accuracy of 99.52% which is an increase in accuracy compared to DeepID2, but as stated in **?** it is not an improvement of DeepID2+.

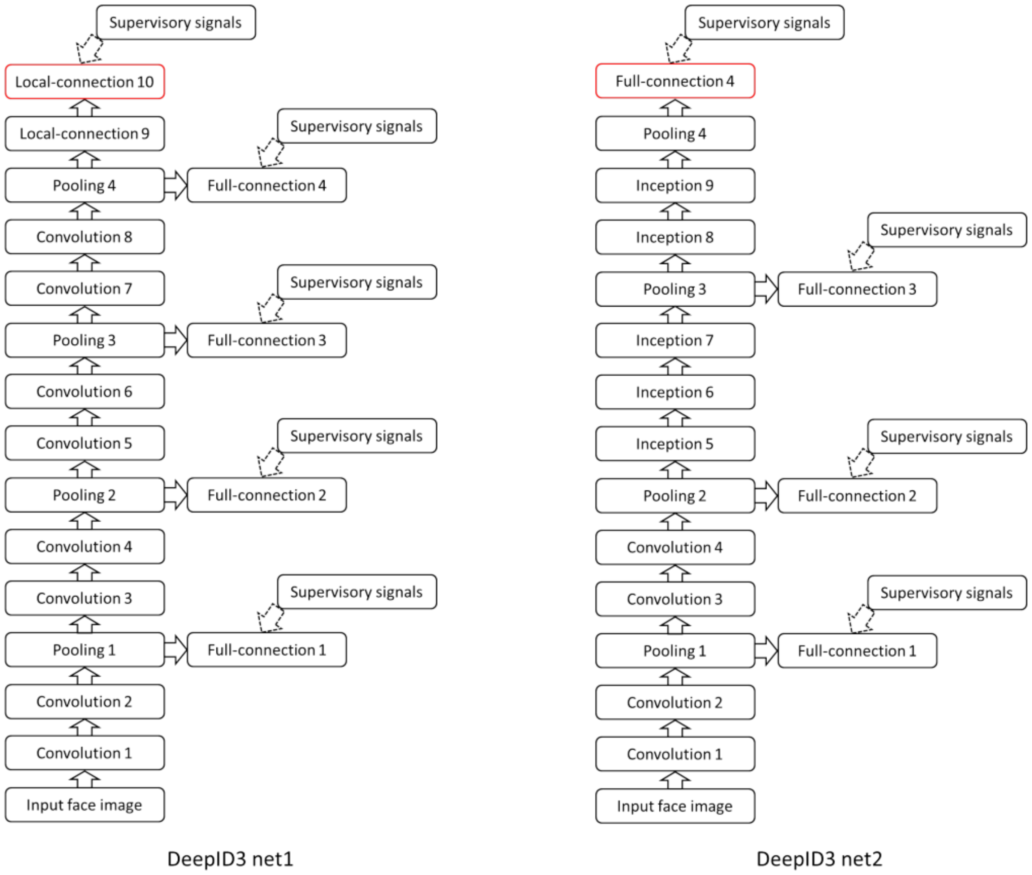


Figure 2.3: Structure of the CNN used in DeepID3 [**?**]

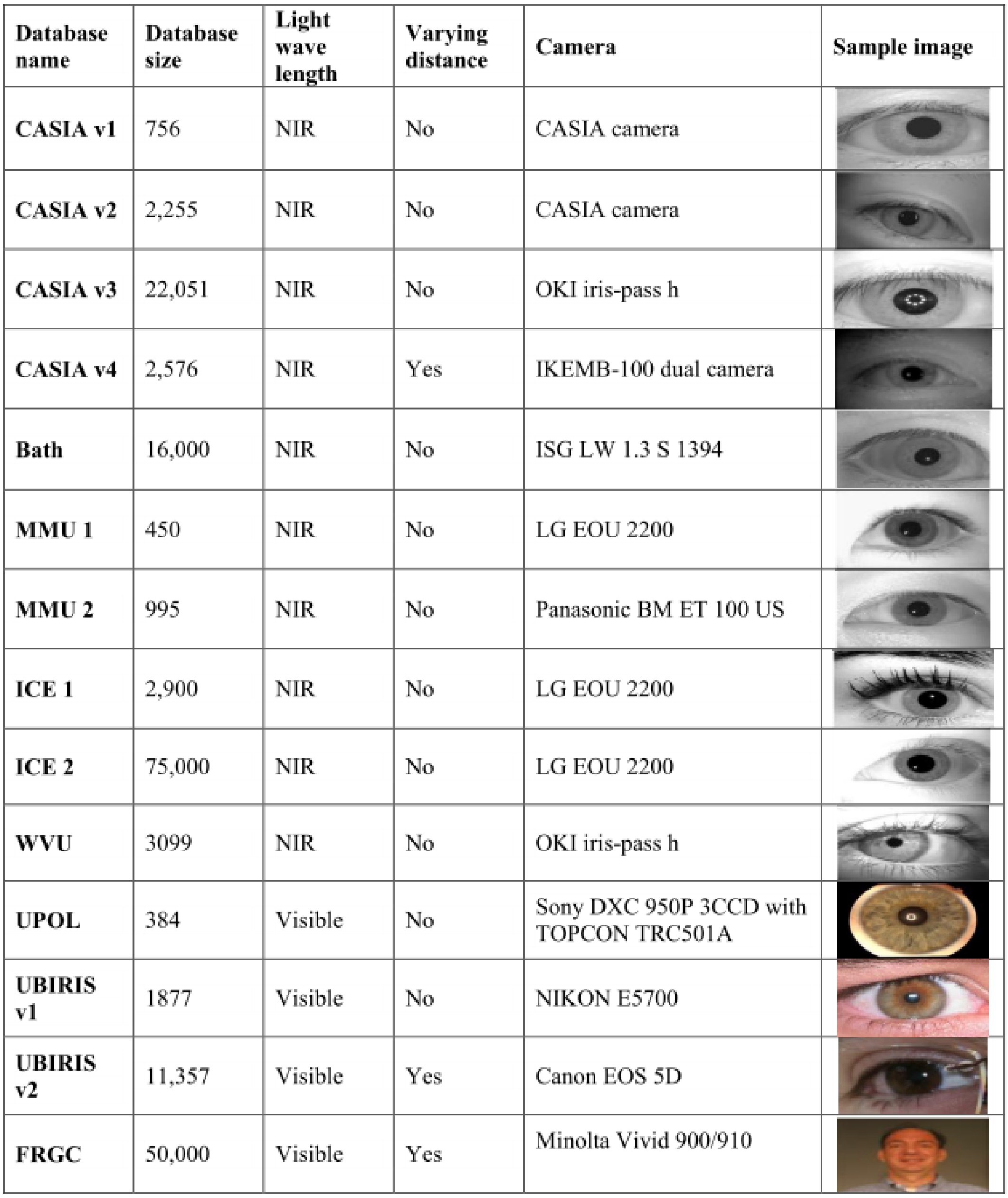
2.2 Iris Recognition

Modern iris recognition was first introduced in an article by Daugman [1993] discussing the security of using iris for recognition. Here an outline of how to do recognition was laid out and even though the field has been extensively since, the general methodology is more or less the same as Daugman proposed. A modern system is typically composed of image acquisition, segmentation and normalisation, feature extraction and matching.

extensively what?

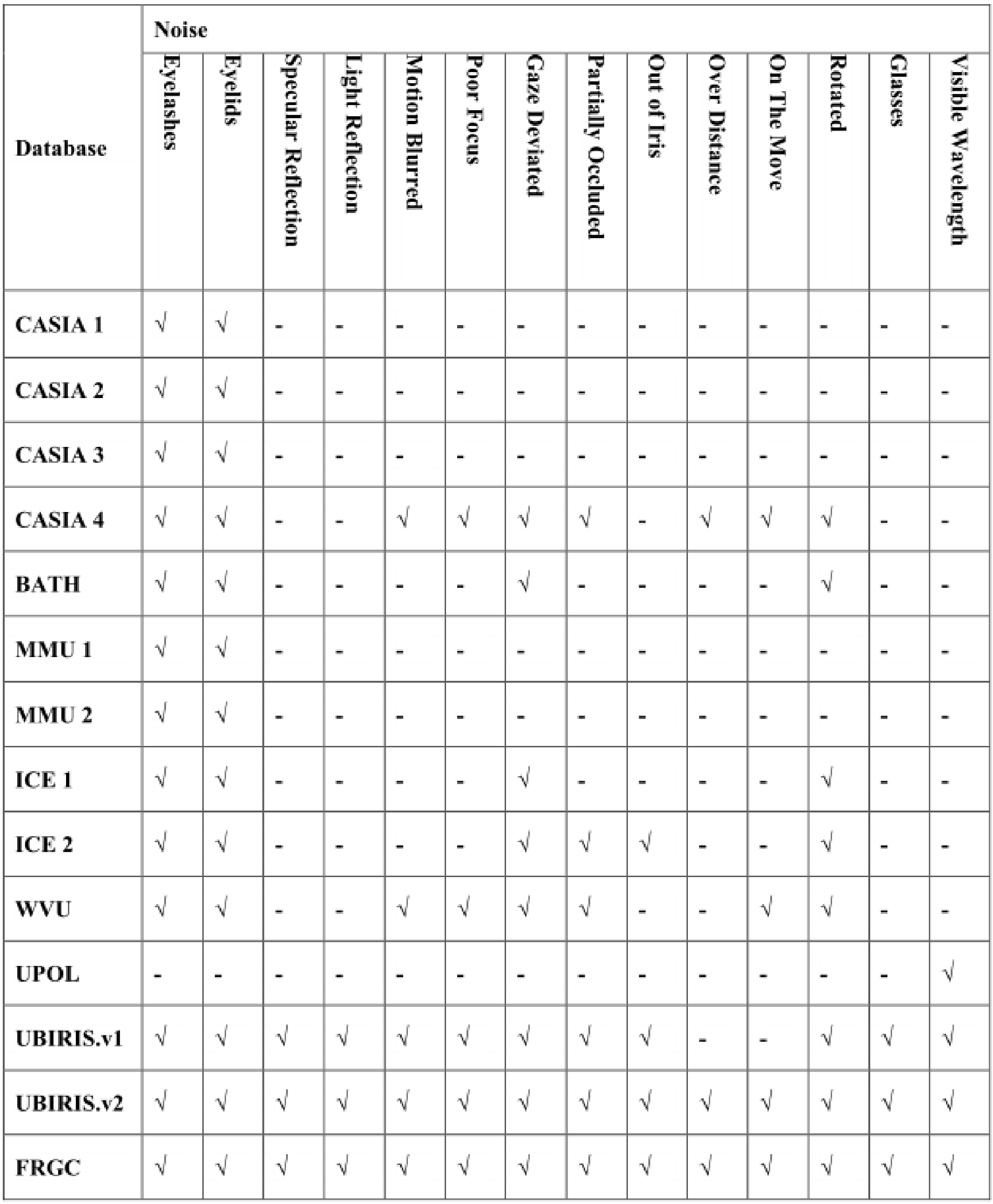
2.2.1 Images

The images acquired are often taken in the Near Infra-Red (NIR) spectrum, which is ranging between 700 nm and 900 nm in wavelength. In this band, the melanin in the iris, which is the substance that gives the iris its colour e.g. brown or blue, is typically less prominent making the unique structure in the iris very distinct. To acquire usable NIR images, the user has to be in the millimetre range of the NIR camera. In the visible light, the melanin is much more prominent and thus makes is harder to detect the structure. The band of visible light has many names in the literature, namely Visible Spectrum (VIS), Visible Wavelength (VW), and Visible Light (VL). VL will be used in this report. While NIR is beneficial in some cases, other useful features can be observed in the VL that cannot be seen with NIR. These can include the moles, freckles and conjunctival vasculature, which can help in making more accurate recognition systems. To make iris recognitions systems comparable with each other some publicly available databases are often used. **?** give an outline of the some of the free databases that are used. A comprehensive table summarising the visual properties, statistics and type of subject used in various database are tabulated in Figure 2.4 and Figure 2.5. Most of the databases available are NIR images with CASIA being one of the most used database. For VL images, UBIRIS in the most commonly used. These contain more "real world" data as the images contain more noise in the form of eyelids obstruction, eyelash obstruction, glare, motion blur, out-of-focus or poor focused iris, partial iris and specular reflection [Rattani and Derakhshani, 2017]. The database has also been used in Noisy Iris Challenge Evaluation (NICE) I. The increasing usage of mobile devices also proposes an opportunity to integrate iris recognition as a biometric for verifying the identity of the user. For this purpose a competition, Mobile Iris CHallenge Evaluation (MICHE), is made to compare the state of the art mobile iris. They provide a the MICHE database that can be used, which they claim is a better database than UBRIS for mobile systems. The database contains noisy iris images taken with a Galaxy Samsung IV, iPhone 5, and Galaxy Tablet II. The noise includes noise from both artificial and natural light sources during acquisition, motion blur, occlusion due to eyelids, glasses, eyelashes, hair, or shadows, which can naturally occur when a user is trying to unlock a phone using their iris.



Database name	Database size	Light wave length	Varying distance	Camera	Sample image
CASIA v1	756	NIR	No	CASIA camera	
CASIA v2	2,255	NIR	No	CASIA camera	
CASIA v3	22,051	NIR	No	OKI iris-pass h	
CASIA v4	2,576	NIR	Yes	IKEMB-100 dual camera	
Bath	16,000	NIR	No	ISG LW 1.3 S 1394	
MMU 1	450	NIR	No	LG EOU 2200	
MMU 2	995	NIR	No	Panasonic BM ET 100 US	
ICE 1	2,900	NIR	No	LG EOU 2200	
ICE 2	75,000	NIR	No	LG EOU 2200	
WVU	3099	NIR	No	OKI iris-pass h	
UPOL	384	Visible	No	Sony DXC 950P 3CCD with TOPCON TRC501A	
UBIRIS v1	1877	Visible	No	NIKON E5700	
UBIRIS v2	11,357	Visible	Yes	Canon EOS 5D	
FRGC	50,000	Visible	Yes	Minolta Vivid 900/910	

Figure 2.4: A table depicting the contents of free iris image databases [?].



Database	Noise													
	Eyelashes	Eyelids	Specular Reflection	Light Reflection	Motion Blurred	Poor Focus	Gaze Deviated	Partially Occluded	Out of Iris	Over Distance	On The Move	Rotated	Glasses	Visible Wavelength
CASIA 1	√	√	-	-	-	-	-	-	-	-	-	-	-	-
CASIA 2	√	√	-	-	-	-	-	-	-	-	-	-	-	-
CASIA 3	√	√	-	-	-	-	-	-	-	-	-	-	-	-
CASIA 4	√	√	-	-	√	√	√	√	-	√	√	√	-	-
BATH	√	√	-	-	-	-	√	-	-	-	-	√	-	-
MMU 1	√	√	-	-	-	-	-	-	-	-	-	-	-	-
MMU 2	√	√	-	-	-	-	-	-	-	-	-	-	-	-
ICE 1	√	√	-	-	-	-	√	-	-	-	-	√	-	-
ICE 2	√	√	-	-	-	-	√	√	√	-	-	√	-	-
WVU	√	√	-	-	√	√	√	√	-	-	√	√	-	-
UPOL	-	-	-	-	-	-	-	-	-	-	-	-	-	√
UBIRIS.v1	√	√	√	√	√	√	√	√	√	-	-	√	√	√
UBIRIS.v2	√	√	√	√	√	√	√	√	√	√	√	√	√	√
FRGC	√	√	√	√	√	√	√	√	√	√	√	√	√	√

Figure 2.5: A table depicting the noise present in the databases [?].

2.2.2 Segmentation

Segmentation of an iris in iris recognition systems tries to detect the iris and find the pupillary boundary and the limbus boundary as well the eyelids and eyelashes that can cause noise on the image. The boundaries along with other parts of the iris can be seen in Figure 2.6. The approach used for segmentation depends among other things on the wavelength of the image; NIR or VL. They both have some common challenges to them. Often the eyelids can cover a small part of the iris, causing the limbus boundary of the iris to not be circular or elliptical. Eyelashes can also cause a similar disturbance as they also can cover parts of the iris. Poor lighting can also make it extremely difficult to detect the boundaries. Specular reflections in the iris can also cause difficulties as they can lie on the iris boundary or close to. Most systems also require a great deal of user cooperation as an off angle iris, motion blur, or glasses or contact lenses can make it even more difficult to detect the boundaries. This can especially be the case for iris recognition in a phone as it cannot be expected of the user knows how to acquire a good iris image.

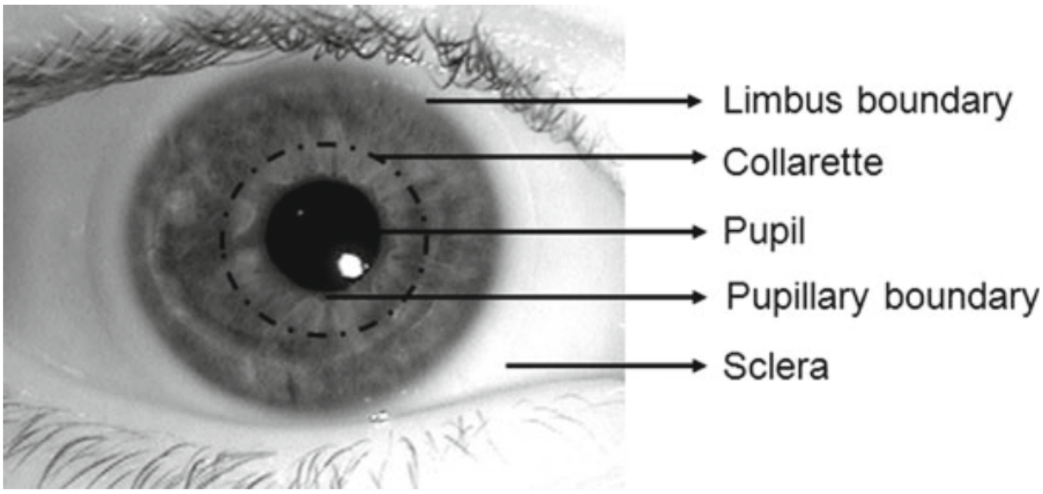


Figure 2.6: A close up NIR image of an iris depicting the different parts of the iris along with their names [Connaughton et al., 2016]

Two commonly observed approaches for segmentation in NIR band images are Daugman's approach [Daugman, 1993] , [Saha et al., 2017], [Rattani and Derakhshani, 2017], [Khan et al., 2017], and Hough Transform [Luhadiya and Khedkar, 2017], [Uka et al., 2017]. Daugman's approach consists of using a Gaussian filter on the image to attenuate the effect of noise and eliminate undesired weak edges like the boundaries within the iris while keeping the strong edges like iris boundaries and eyelid boundaries. An integro-differential operator is then used as a circular edge detector. It is then used iteratively to find the pupillary boundary and the limbus boundary. Hough Transform on the other hand is a histogram based model fitting approach. An edge map of the input map is generated using a gradient-based edge detector. Then a voting procedure is applied on the thresholded edge map to determine the parameters for a contour that best fits a circle. This operation gives an approximate edge map of the iris boundary. Lastly, the segmented iris is often normalised using Daugman's rubber sheet model that maps every point in the segmented region from cartesian to polar coordinates. An open source MATLAB implementation based on updated version of the Daugman approach is a commonly used tool. There exist other methods for different circumstances, but these are two most commonly used. They can also be used on VL images if the image is converted from RGB to grey scale images [Connaughton et al., 2016] .

Remember to add a proper citation

How to properly cite a book where each chapter is written by different people?

2.2.3 Feature extraction and classification

There are multiple ways that the features can be extracted from the segmented iris. The most commonly used in the literature is a 2D Gabor filter which is a linear filter used for edge detection [Daugman, 1993]. The Hamming distance is then used as way to classify the iris. The Hamming distance is a measurement of how many bit flips a piece of data need to have to match another piece of data. The bits of the extracted features are then measured against the whole database and the pair with the lowest score is a match. This is called "1-to-N search". As the database gets larger and larger the computation time also grows as it will have to search through the whole database. That's why Kuehlkamp and Bowyer [2016] have suggested using a "1-to-first search" instead to improve the speed of the search. Here a threshold is chosen and as soon as a match has been found below the threshold it will stop the search. Other approaches to the categorisation have been proposed using machine learning. Khan et al. [2017] proposed using SVM, K-Nearest-Neighbours (KNN) and LDA with respective test accuracies of 97%, 95.1%, 94.28%. In comparisons the commercial systems ranged from 94.57% to 99.67% with VeriEye having the lowest and IriCore having the highest accuracy.

2.2.4 Deep Learning

According to **?** research in neural networks within iris recognition is still very new and not much has been done. Some of the work that has been done have used CNN and a Deep Belief Network (DBN). Al-Waisy et al. [2017] used a common CNN to extract features and classify a segmented and normalised iris image. The segmentation was done using Circular Hough Transform (CHT) normalisation was done using Daugman's rubber sheet method. They named the network IrisConvNet and the architecture was inspired by the Spoofnet as it can be seen in Figure 2.7. This was the general architecture and they tried different numbers of maps and layers to find the best architecture. In general a 3x3 input kernal was used on a 64x64 or 128x128 pixel input image to create feature maps followed by 2x2 max pooling, 5x5 convolution, 2x2 max pooling 5x5 convolution, 2x2 max pooling, 5x5 convolution and finally a two fully connected layers to to a softmax regression classifier layer. A ReLU activation function is applied on the top of the convolutional and fully connected layers because it results in several times faster training without sacrificing accuracy. The AdaGrad algorithm was used for training the network. Three databases were used; SDUMLA-HMT, CASIA-Iris-V3 and IIT Dehli (IITD). IrisConvNet scored an accuracy of 99.82% at categorising CASIA-Iris-V3 in 0.65 s. architecture

maybe add a source or explanation if needed?

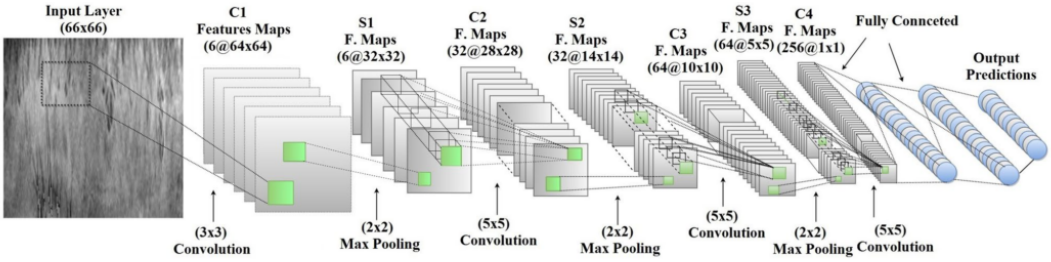


Figure 2.7: Example of a CNN architecture from Al-Waisy et al. [2017].

? claim that the problem with traditional networks are that they are database specific and not very generalisable. They created a Fully Convolutional Network (FCN) using a loss function they created specifically for iris networks called Extended Triplet Loss (ETL). They claim this network is more generalisable than the previous networks and it shows superior results compared with IrisCodewhich is Daugman's segmentation and normalisation approach. A FCN differs from a CNN in that there are no fully connected layers, only convolutional layers, max pooling etc. They proposed an architecture called UniNet that can be seen in Figure 2.8. It consists of two FCNs; FeatNet and MaskNet. The network takes an iris that has been segmented and normalised using a recent approach [**?**]. The segmented iris has a resolution of 64x512 pixels. The image is then fed through FeatNet and MaskNet. FeatNet extracts the iris features while MaskNet masks the non-iris part of the image. e.g an eyelid that occludes. Three of these UniNet networks are then trained in parallel in Triplet-based network architecture that uses the ETL loss function. They used four databases to train and the networks; ND-IRIS-0405, CASIA V4, IITD and WVU Non-ideal Iris Database. It was then compared with a CNN network based on VGG-16 that used softmax, CNN with triplet, FeatNet only and DeepIrisNet which is a CNN that is proposed directly for iris recognition. The results can be seen in Figure 2.9 where UniNet outperforms the other nets and FeatNet is the by far worst performing network, which suggests that MaskNet is needed for the network to perform well.

IrisCode? what is this?

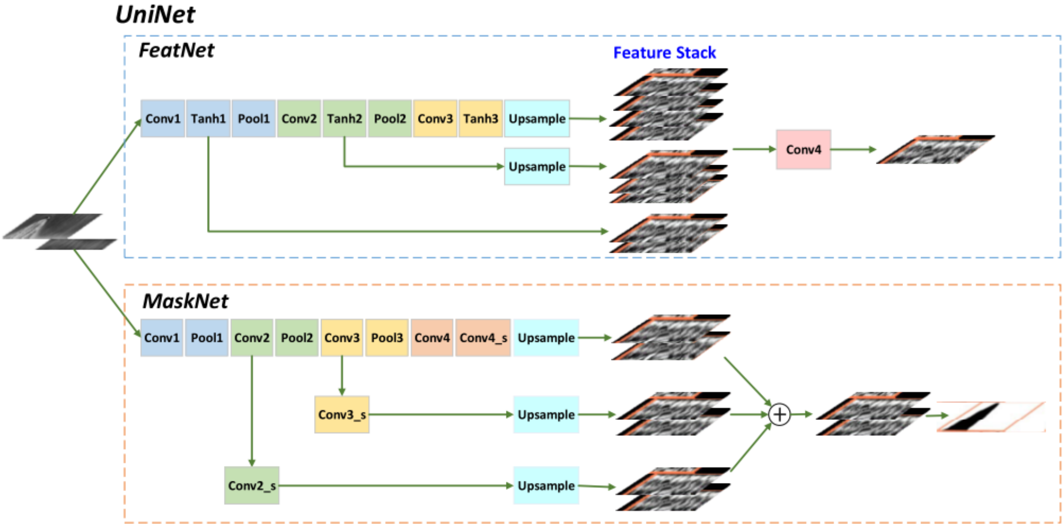


Figure 2.8: Example of a CNN architecture from Zhao and Kumar [2017].

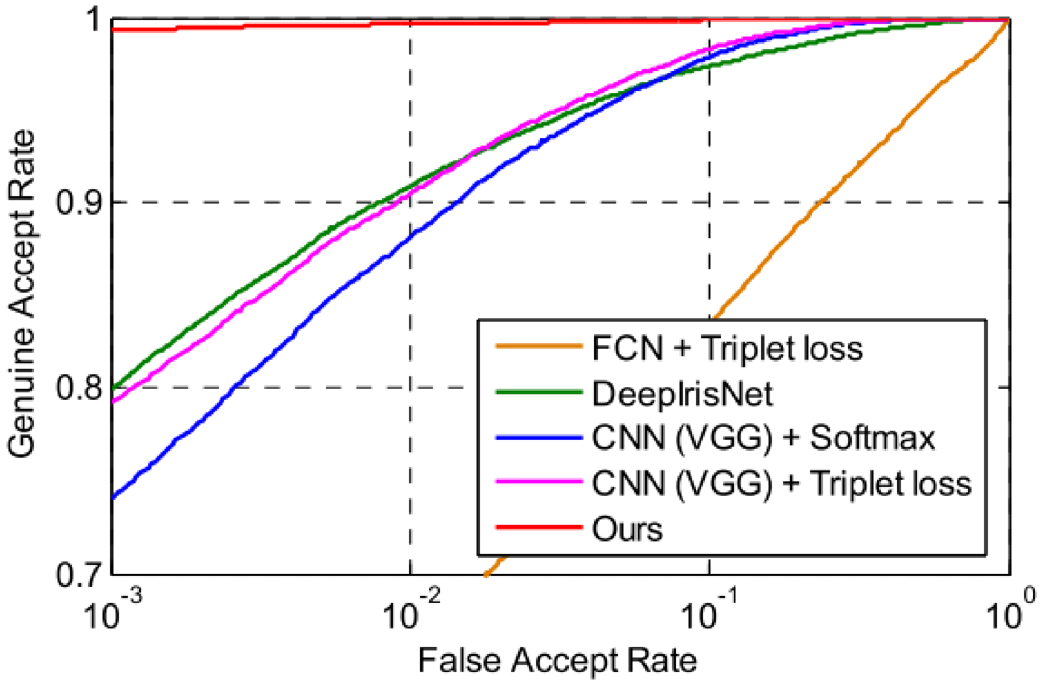


Figure 2.9: Results from the different networks tested on the ND-IRIS-0405 database Zhao and Kumar [2017]. "Ours" is UniNet and FCN + Tripletloss is FeatNet

Another promising approach by Bazrafkan et al. [2017] targets iris segmentation in low-quality consumer images obtained from smartphones is using Semi Parallel

Deep Neural Networks (SPDNN) for generating iris maps from low quality iris images. In SPDNN several deep networks are merged into a single model. This way it is possible to include different networks designs and combine their strengths. They combined four FCNs of different architectures that used a variaty of kernel sizes and layers. An example of one of the FCNs has 12 convolutional layers in this order; 3x3,5x5,7x7,9x9,11x11,13x13,15x15,13x13,11x11,9x9,7x7,5x5 and an output layer that is 3x3.It was trained by using the databases Bath800, CASIA Thousand, UBRIS and MobBio. They contained both NIR and VLimages. Extensive noise was also added to the training images in the form of blur and lower contrast to generate degraded versions of the high quality images and utilise the databases better by creating more samples. Using the UBRIS database as a comparison with other state of the art segmentation techniques it achieved the highest accuracy with 99.30%. The technique with the lowest accuracy, of the ones it was compared to, had an accuracy of 98.10%.

consider whether it makes sense to call it VL images rather than RGB images

2.3 Information Fusion

A system that strives to utilise information from two or more different biometric traits in order to obtain one combined result is a called a multi-modal system which is a kind of multi-biometric system. Besides the multi-modal approach there are several other approaches, which results in information from two or more sources, therefore, methods for fusion of information from different sources into one system for classification purposes is a widely investigated area. [Connaughton et al., 2016]

Fusion of information can happen on different levels. It can happen on one of five levels, each with a higher level of preprocessing: signal level, feature level, score level, rank level, or decision level. The level on which the fusion should be done depends on the kinds of multi biometric data that has to be fused, and the purpose of the fusing. In general score-level and feature-level are the most popular fusing techniques [Connaughton et al., 2016]. Fusing at the lowest level, signal level, might be done in order to merge data from different sources to construct a more detailed or larger dataset or signal. At the feature level the fusing might happen through merging of extracted features from different sources into one feature vector [Ross and Jain, 2003]. At the score level the fusion can be done in order to determining the best sample to use for the processing based on which has the highest score and thus is the best match to the gallery samples. Rank level can be somewhat similar to the scores but depend on match rankings. At the decision level it can be making the decision based multiple classifiers, e.g. one for each modality[Fierrez et al., 2018].

In the literature a large variety of methods and algorithms have been utilised for fusing information on different levels. Some algorithms are fairly simple and basically makes decisions about which sample to use onwards for classification based on matching scores between samples and the gallery. Other methods are more advanced and well known for use in applications such as classification. These are methods such as SVM, KNN, decision trees, and bayesian methods [Ross and Jain, 2003].

The highest rank method and Borda count are both well known methods for combining information based on ranking. The highest rank method ranks possible classes based on the highest rank assigned to the class by a classifier across all classifiers.[Ho et al., 1994] The Borda count is a kind of majority voting which can be used in combination with Multiple Classifier Systems (MCS) [Connaughton et al., 2016; Ho et al., 1994]. An important aspect to consider is that whenever different kinds of information are fused together, the information should be represented in the same data space, otherwise unwanted weighing of certain information above other might occur.

In this report, the focus is to implement a multimodal system utilising the biometric traits face and iris. In literature different approaches for the fusion of these traits have been presented. **?** as one of the lates researches within the area presented a work utilising deep learning for feature extraction and matching of the biometric traits and tests different methods for score and rank level fusion of the modalities. The tested methods on score level include eg. sum, weighted sum, max etc while the rank level fusion uses the mentioned Borda count, Highest rank, or Logistic regression. Chen and Te Chu [2005] did processing of iris and face separately and only combined the two modalities at the decision-level by adding inferred probabilistic values in a linear combination.

2.4 Multi-Modal Databases

Even though recognition based on biometric traits is widely investigated, and research shows that multimodal systems perform better than the uni-modal systems based on the same data, the research in this area is limited and incomplete [Chen and Te Chu, 2005; Connaughton et al., 2016]. Because of the limited availability of multimodal datasets, such datasets are often synthetically constructed based on randomly combined data, eg. iris and face datasets [Chen and Te Chu, 2005]. Only a limited amount of studies utilise a multimodal dataset obtained from the same test subjects or maybe even with one sensor. However, a few multimodal datasets have been encountered in literature. The multimodal datasets varies in which modalities they include. The modalities can be different images of face, recordings of gait, hand geometry, handwriting, signature, fingerprint, finger vein, iris, speech etc. [Yin et al., 2011; Dessimoz et al., 2007; Ross and Jain, 2003; Ortega-Garcia et al., 2010]. Examples of multimodal datasets containing both iris and face are the database IV^2 [Petrovska-Delacrétaz et al., 2008], consisting of data obtained from 300 subjects, the MBioID based on 120 subjects [Dessimoz et al., 2007], The BiosecureID based on 400 subjects, The BioSec database based on 250 subjects, the BMDB DS2 based on 667 subjects [Ortega-Garcia et al., 2010], the SDUMLA-HTM database based on 106 subjects, the MobBio containing data from 105 subjects acquired through a mobile device [Sequeira et al., 2014],and the datasets provided for the The Multiple Biometric Grand Challenge (MBGC) [Connaughton et al., 2016]. The latter is available in two versions and can be obtained on request. Furthermore, it serves as a common test set in order to compare performance. However, some multimodal datasets have been created by researchers during their work, which are not named nor generally available [Connaughton et al., 2016].

Database Name	**Responsible**	**Number of Subjects**	**Modalities**
IV^2	Petrovska-Delacrétaz et al. [2008]	300	VL Face 3D face laser scan 3D face stereoscopic IR Iris
MBioID	Dessimoz et al. [2007]	120	VL Face 3D face Iris Fingerprint Voice Online signature
BiosecureID	**?**	400	Face Iris Fingerprint Voice Signature Hand Handwriting Keystroke
BioSec	**?**	250	Face Iris Fingerprint Voice
BMDB DS2	Ortega-Garcia et al. [2010]	667	VL Face 3D face laser scan IR Iris Fingerprint Voice Signature Hand
SDUMLA-HTM	Yin et al. [2011]	106	VL Face NIR Iris Fingerprint Finger Vein Gait
MobBio	Sequeira et al. [2014]	105	VL Face VL Iris Voice
MBGC v1/v2	NIST	-	Face Iris
	Connaughton et al. [2016]	363	Face Iris

Table 2.1: Multi modal biometric databases containing both iris and face information.

Discuss this table

dkjfnvkd mbmb
Sok

Chapter 3

Implementation

3.1 Basic Methods

In order to get some basic understanding of the methods commonly used for iris classification the work presented in an article is implemented. The work implemented is the work of **?** described in [**?**].In the work VL images of the iris obtained by smartphones are processed and used to train different classifiers. The processing consists of a sequence of steps. The steps included are

make sure references read right

- Iris Location
- Eyelid Suppression
- Iris Normalisation
- Noise Removal
- Histogram Equalisation
- Feature Extraction
- Training and Classification

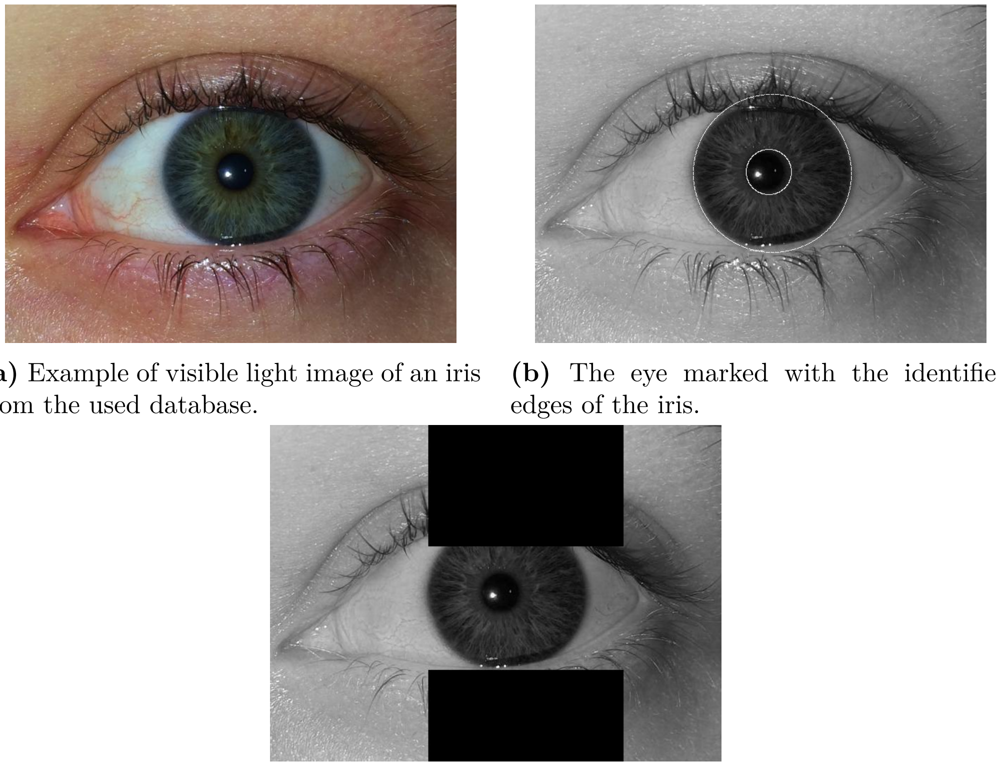
Figure 3.1a shows an example of an image from the database. Before the first step some simple preprocessing was done. The red channel is preserved and the other two other colour channels are neglected. This is done because this simulates the use of NIR light. The result is a greyscale image based on the red channel. In the following sections the implementation of each of the steps will be elaborated.

3.1.1 Iris Location

The first step of processing is locating the iris. For this purpose Daugman's Integro-differential operator was used. The operator identifies the circular contour, which has the greatest change in intensity, by varying the three parameters defining the circle $(r, x_0, y_0$.

$$\text{Threshold} = \text{Low Value} + \text{Fraction} \cdot (\text{High Value} - \text{Low Value}) \tag{3.1}$$

Figure 3.1b shows an example of the identified edges of the iris.



(a) Example of visible light image of an iris from the used database.

(b) The eye marked with the identified edges of the iris.

(c) The eye with the eyelid suppression applied.

Figure 3.1: The original image before and after different steps of processing.

3.1.2 Eyelid Suppression

The annulus identified by the iris location as the area of the iris might not only contain the iris. The subject might in some cases not open the eyelids fully or the angle of the camera will be such that parts of the iris will be covered by the eyelids. Therefore an algorithm is used which eliminates the eyelids. The algorithm is detecting applied on the image cutout of the original image which has the boundary box of the identified iris as edges. The small image is divided into an upper and a lower part which are searched for the upper and the lower eyelid respectively. The search is done by using the

3.1.3 Iris Normalisation

For the normalisation Daugman's rubber sheet model is used. The purpose is to get a rectangular image of the iris, which corresponds to taking the annulus covered by the iris region, cutting it open and unfolding or stretching it to a rectangular shape.

The model does this by mapping the iris from the original image to a rectangular image. The mapping is done based on polar coordinates radius and angle, (r, ϑ),

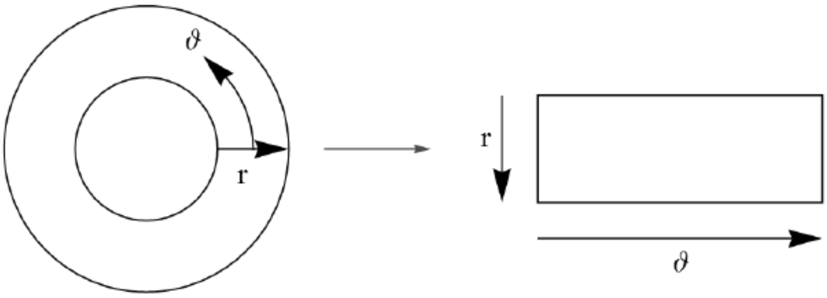


Figure 3.2: Daugman's Rubber Sheet Model. [?]

where $r \in [0,1]$ and $\vartheta \in [0, 2\pi]$. In the rectangular image angle ϑ is on the x-axis and the radius r is on the y-axis of the image. Figure 3.2 shows the mapping of the method. This was implemented using a function in the library crated my Libor Masek [?]. The resolution of the resulting image is dependent on arguments. The image of the normalised iris is shown in Figure 3.3.

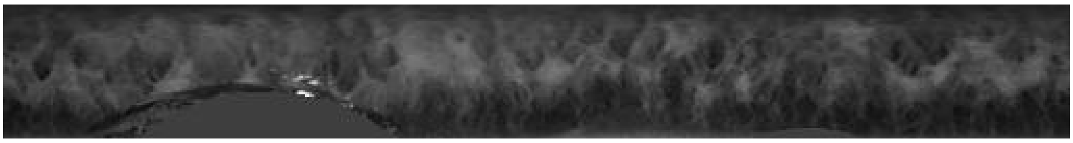


Figure 3.3: The normalised image of the iris before applying functions.

3.1.4 Noise Removal

Though the article uses several well known methods that are commonly used in processing of images of irises, there descriptions of the methods are quite inadequate. After obtaining the normalised iris the next step applied is noise removal. The purpose of the noise remover function is to remove noise occurring in the form of eyelashes covering parts of the iris. Usually the pixels showing the lashes will be among the darkest pixels. Since every image of the iris is different and how dark the iris is also varies, a threshold has to be identified adaptively. The article does not describe in depth how this is implemented, it simply states that some histogram analysis is done in order to obtain the lowest pixel values. Figure 3.4 shows the histogram of the normalised image before any noise removal. Because the information about the exact approach used in the article is inadequate, an adaptive algorithm was created. The algorithm implemented identifies a threshold value based on the histogram. This is done by first identifying the highest and lowest bin-value, which has a frequency of more than a specified "recognition value", which was set to 10 in this project. The recognition value is introduced to make sure outliers are not defining for the threshold. Afterwards the threshold is calculated by the formula in Equation 3.1, where "Fraction" is a parameter set manually defining how large a part

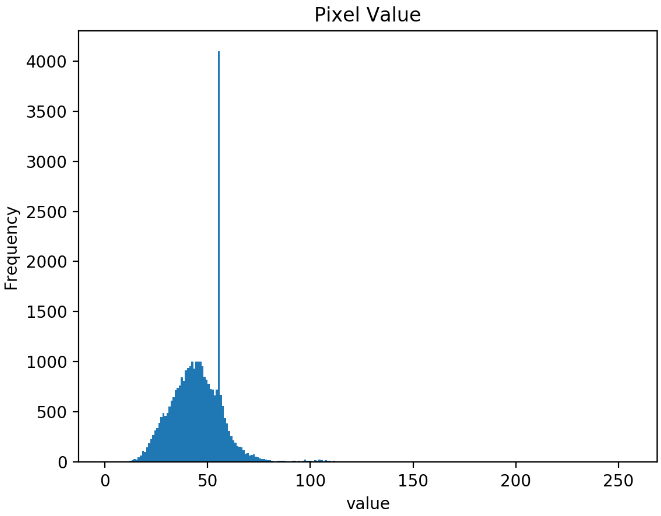


Figure 3.4: The histogram before applying the functions.

of the identified pixel value range has to me thresholded. During the processing in this project "Fraction" was set to be equal 0.1.

$$\text{Threshold} = \text{Low Value} + \text{Fraction} \cdot (\text{High Value} - \text{Low Value}) \tag{3.2}$$

After the threshold has been found it is applied to all pixels in the image. The pixels lower than the threshold are eliminated and have to be reconstructed from neighbouring pixels. Also here the article provides very limited information about the algorithm applied. Therefore an algorithm was implemented, which restores pixels from non-occluded neighbour regions. A part of the algorithm identifies the pixels with values lower than the threshold and saves the pixel coordinates of the pixels. The saved pixels are reconstructed iteratively from neighbouring pixels following the 4-connectivity principle. The pixels are reconstructed when there are at least 2 neighbour pixels they can be reconstructed from. They are only reconstructed from pixels above the threshold, this can be pixels that initially were above the threshold, or it can be pixels that have already been reconstructed. The pixels are reconstructed by assigning the average of the neighbours with values above the threshold as the new pixel value. Once all thresholded pixels have been reconstructed the reconstructed image is returned.

Because the eliminated pixels are reconstructed only from pixels with a value higher than the threshold there are certain traits that can be expected in the histogram of the reconstructed image. One of the traits is that there is a flatline from 0 to the threshold value. A second trait is that a peak close to the threshold is likely to occur because the eliminated values are reconstructed from neighbours which are likely to be close to the value of the eliminated pixels. However, the plotted histogram after reconstruction is as shown in Figure 3.5. As can be seen there is a small

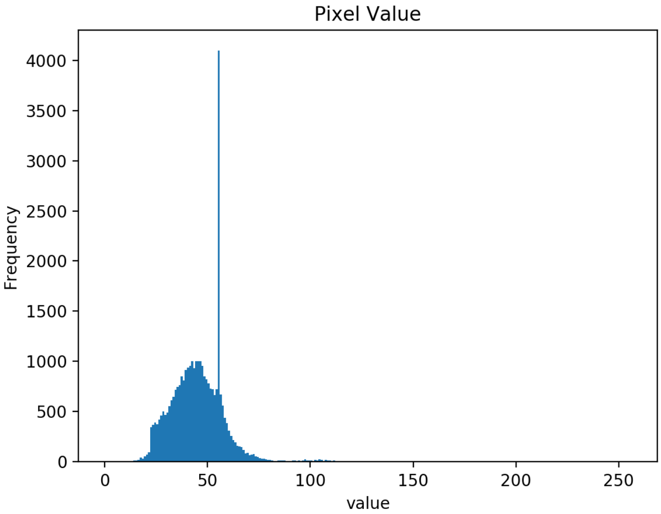


Figure 3.5: The histogram of the image after reconstruction of pixels.

peak as expected, however, there seem to be a "spill over" across the threshold to the lower values. By closer examination of the code it was discovered that this was caused due to a programming mistake. The mistake was that the values used for reconstruction were obtained from the original image and not from the reconstructed image. Once this mistake was corrected the histogram was as expected as shown in Figure 3.6. In relation to the reconstruction of pixels it should be noted that it could have been done with 8-connectivity, and the iterative process could be split up to more steps, such that pixels are always constructed from as many neighbours as possible. This may give a better reconstruction, however, this has not been investigated. Furthermore, small tests showed that the adaptive threshold is difficult to define in an optimal way. If the threshold is too low the eyelash might be reconstructed from edge pixels of the eyelash creating just a lighter eyelash, which is still darker than the iris in the background. If the threshold is higher, it might eliminate the eyelashes, but also be destructive to darker parts of the iris. Maybe this could also be solved if the reconstruction happened based on the neighbourhood and not just the most adjacent neighbour pixels.

During implementation the histograms were frequently inspected in order to ensure the results were as expected. Figure 3.7 show a histogram with some eliminated pixel values. It turns out that the function used for generating histograms in python by default takes the range of the pixel values and splits that into as many bins as specified. As a result some of the bins cover a range of values that is entirely between two integer values and thus do not count any instances of pixel values as they are always integers between 0 and 225. This was solved by passing a specific range (0:256) as an argument and then the histogram was as described in Figure 3.6.

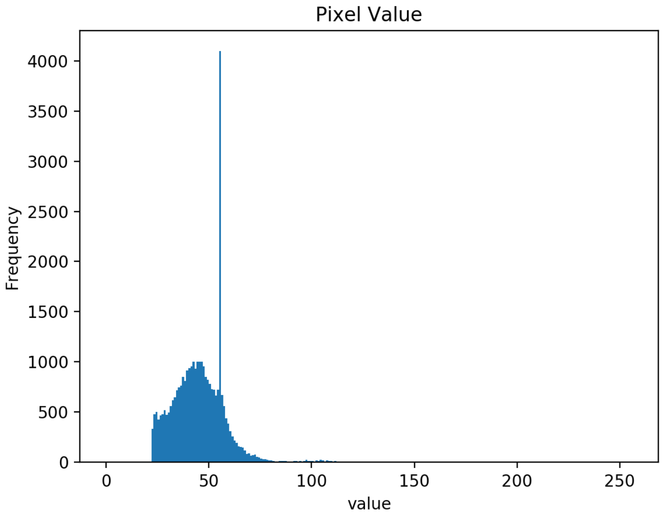


Figure 3.6: The histogram of the image after applying the corrected noise remover function.

3.1.5 Histogram Equalisation

The histogram equalisation is applied to increase contrast. This is necessary to enhance the structures in the iris. This was implemented manually. Just as in the noise removal the lowest and highest bin values with more instances than a specified value are found. Based on the found values the histogram is stretched using the formula in Equation 3.2.

$$\text{New Pixel Value} = (\text{Pixel Value} - \text{Low Value}) \cdot \frac{255}{\text{High Value} - \text{Low Value}} \tag{3.3}$$

The pixels, which have values above 255 or below 0 after the formula is applied are set to 255 or 0 respectively. The resulting image can be seen in Figure 3.8, while Figure 3.9 shows the histogram after applying the formula to each pixel. However, it was discovered that the implemented method was actually histogram stretching which was mistakenly taken as the same as histogram equalisation. Though the two methods have somewhat the same effects histogram equalisation is better at ensuring a uniform spreading of the pixels across the histogram. This is done by identifying a transform of the bin values which causes the Cumulative Distribution Function (CDF) of the histogram to be as linear as possible, and apply it to the grey levels or bin values. The result of the histogram equalisation is showed in Figure 3.10 and Figure 3.11. When comparing the two images of the iris after applying the two different contrast adjustment methods it can be concluded that the histogram equalisation does indeed result in better contrast and a more enhanced and out spoken appearance of the structures in the iris than the histogram stretching does. Testing also showed that using the equalisation rather than the stretching increased the accuracy, however, this will be further addressed in section 3.1.7.

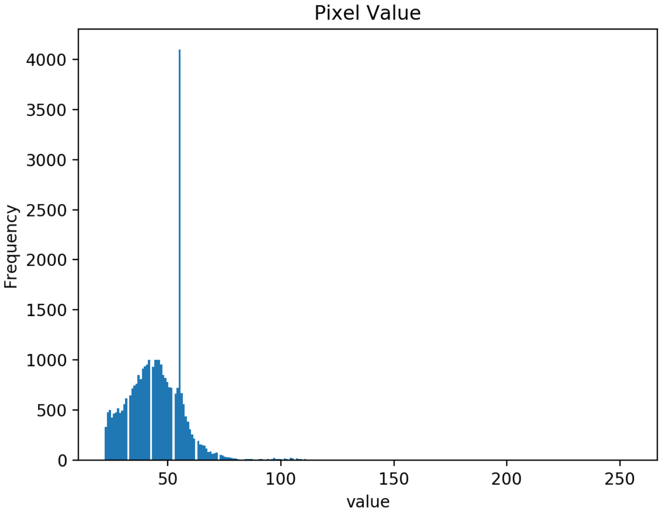


Figure 3.7: The histogram of the reconstructed image with eliminated values.

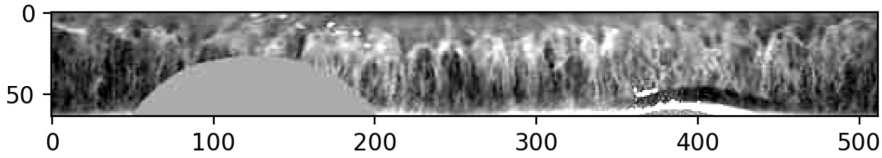


Figure 3.8: The image of the iris after applying the initial "equalise histogram" function.

3.1.6 Feature extraction

As it is difficult to extract all the complex structures in the iris the work presented uses the iris image as input. However, because the image in full will result in to large amounts of data to handle during feature extraction is done. Feature vectors are extracted using Haar wavelets in a wavelet decomposition.

Haar wavelets are an fast and simple way to obtain wavelets. In Haar wavelets the decomposition is done by calculating the average of neighbouring pixels as well as half the difference between neighbouring pixels. Due to the way the method works the image is lowpass or hi-pass filtered two times in each of the possible combinations. The image of the greatest interest in this case is the one that has been lowpass filtered in both horizontal and vertical direction, this is referred to at LL. This image will be a compressed version of the original image, showing an approximation or the tendency of the original image. The other images created in one stage are HH, HL, LH all of which contain detail coefficients. HH is representing the diagonal detail, while HL presents horizontal detail and LH presents vertical detail. The LL will naturally appear in the upper left corner of the output matrix.

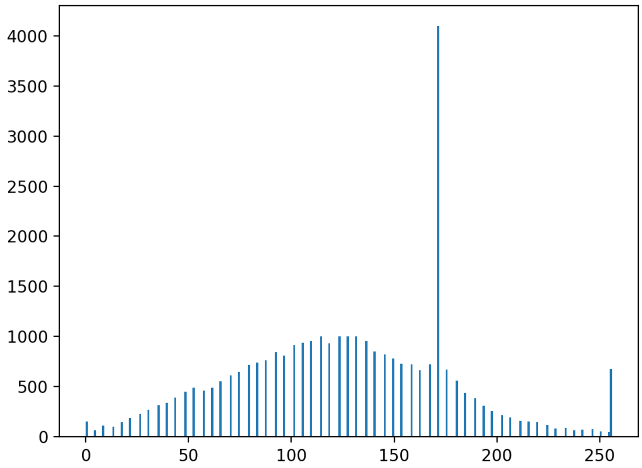


Figure 3.9: The histogram of the image after applying the initial "equalise histogram" function.

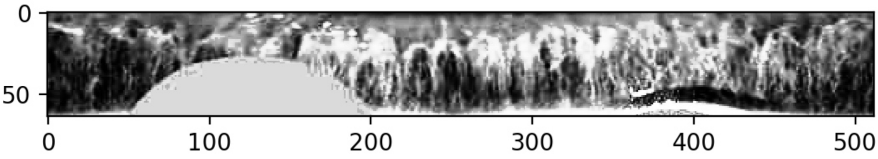


Figure 3.10: The image of the iris after applying the real equalisation of the histogram.

In order to compress the image sufficiently three stages or passes of Haar wavelet decomposition was carried out. The result is showed Figure 3.12. The resulting matrix of LL3 coefficients is then put into a feature vector, which is used for training the classifiers described in section 3.1.7.

3.1.7 Training and Classification

For classification different classifiers were tested. The best of the investigated classifiers are the KNN, LDA, as well as a SVM with linear or quadratic kernel. In the following sections the theory behind the different classifiers will briefly be described

K-Nearest Neighbour

KNN is one of the most simple supervised classification methods. It is a non-parametric method. This means that the train data is used "raw" and there are no parameters that are deducted from the data in order to form a model.

KNN utilises Bayes theorem in order to determine the probability of a sample belonging to a certain class. The probability of the sample belonging to different

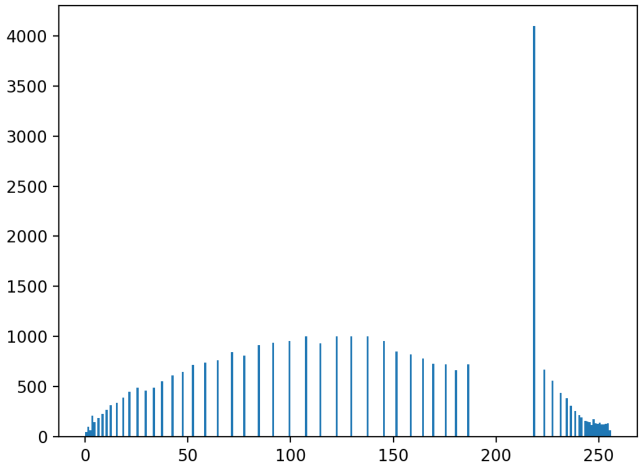


Figure 3.11: The histogram of the image after applying the real equalisation of the histogram.

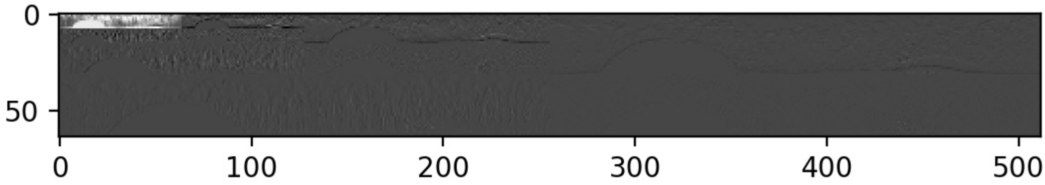


Figure 3.12: The result of the three level haar wavelet decomposition. Note that the small light image in the upper left corner is the LL3, which is used onwards.

classes is found by looking at the density of datapoint belonging to the different classes among the k nearest neighbours. The probability of a class given the sample can be estimated by the formula in Equation 3.3.

$$p(C_k|\mathrm{x}| = \frac{K_k}{K} \tag{3.4}$$

Where C_k is a given class, x is the sample, K is the number of neighbours considered, and K_k is the amount of data points among the K neighbours belonging to the class C_k.

Finally, a class is assigned to the sample based on which class has the highest conditional probability.

Linear Discriminant Analysis

LDA is a parametric method which is trained by supervised learning. LDA is a classifier in which linear decision boundaries are learned based on training data. It is part of a more general group of linear discriminant functions. The decision boundary is found such that interclass variance is maximised while intra class variance is minimised. There are a range of possibilities for identifying the equation for the linear decision border eg. using SVD or EVD. The main principle that separates LDA from Principal component analysis (PCA) is the

Support Vector Machines

Cross Validation

Just as in the article

3.2 What is a Convolutional Nerual Network

This section describes what a CNN is and what structure was used for creating the iris recognition CNN. The section is based mainly on **?** and **?**.

3.2.1 Convolution

A CNN is type of Nerual Network that is especially good for image recognition. Instead of the input being fully connected it operates by using a kernel instead. A kernel can be seen as a small window which moves through the input image. It performs an operation known as a convolution where the coefficients, also known as weights, of the kernel are multiplied with the pixel it is covering and summing the value. A 1D example of this is shown in Figure 3.13 where a kernel of size 3 with weights $\begin{bmatrix}1 & 0 & -1\end{bmatrix}$ are convolved through bottom rows which produces the top rows. Two other concepts are also shown in the example; stride and zero padding. Stride is the amount the kernel moves each time. The left example has a stride of 1 and the right example has a stride of 3. Zero padding is when pixels with value 0 are added at the edge of the image. This is done to ensure that the convolution can actually be made on all pixel because otherwise only one convolution would be possible in the example with a stride of 3. It also has the benefit of keeping the dimensions of the input image and output image the same when the stride is 1, as convolution otherwise shrinks the image.

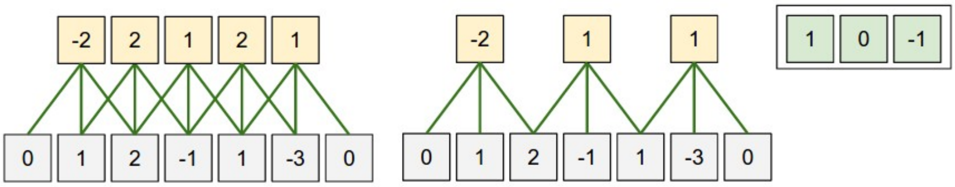


Figure 3.13: Example of a kernel in a CNN [**?**]

The output of the convolution creates a new image. This image becomes the input of an activation function. An activation can be though of as how much should the neuron fire, e.g how active is it. A common activation function for CNN is the Rectified Linear Unit (ReLU), Equation 3.4, that gives an output of x if the x is positive and 0 otherwise. This activation function has proved to be consistently better and faster for deep neural networks than the previously popular sigmoid activation function.

$$f(x) = max(0, x) \tag{3.5}$$

3.2.2 Activation function/feature maps

The output of the convolution is the hidden layer and is called a feature map. An example is shown in Figure 3.14 where a kernel of $5x5$ is used on a input image of $28x28$ to produce a feature map of $24x24$. The dimensions can be calculated by using Equation 3.5 where W_2 is the width of the feature map, W_1 is the width of the input image, F is the size of the kernel and P is the amount of padding.

$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1 \tag{3.6}$$

The weights of the kernel is what is being learned by the network. The kernel has what is known as shared weights and biases, as its the same weights that convolve through the whole image. This is shown in Equation 3.6 which is how a single neuron of the feature map in example Figure 3.14 is computed. σ is an activation function, b is the shared bias, w are the weights of the kernel with l, m being their position and a is the input pixel at the j, kth.

$$\sigma(b + \sum_{\mathrm{l=0}}^{4} \sum_{\mathrm{m=0}}^{4} w_{\mathrm{l,m}} * a_{\mathrm{j+l,k+m}}) \tag{3.7}$$

In other words the kernel learns to detect a feature that can be found in the image, e.g. straight lines. As a single feature is not enough for recognition, multiple kernels are used to make multiple feature maps. These new feature maps become the input of another convolution layer that can learn higher level feature, e.g. combining the vertical and horizontal feature to create a kernel that detects edges. Each convolution adds a new level of abstraction.

3.2.3 Pooling

Pooling is an operation aften added after a convolutional layer. Like a convolution a small windows moves through the image, except it does not convolve but simply looks at the pixels in its current location and outputs a single pixel depending of the type of pooling used. Maxpooling is a common type of pooling used in CNNs. An example of maxpooling with a windows of size $2x2$ and stride 2 is shown in Figure 3.15. This has the benefit of downscaling the spatial size of the image for faster computation and reducing overfitting.

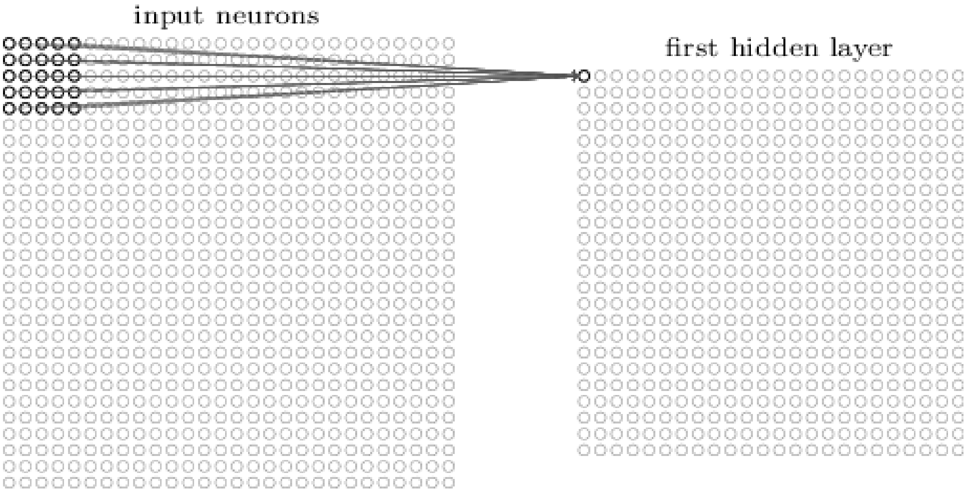


Figure 3.14: Example of a kernel in a CNN [?]

3.2.4 Classification

To use the features extracted from the convolutional layer one or more Fully Connected (FC) layers are added at the end of a network. In a glsfc layer all the neurons from the previous layer are connected to all of the neurons in current layer. This relationship is described in Equation 3.7. The output of a single neuron is the sum of the all the previous weights w_i and their bias b put through some activation function f.

$$f\left(\sum_i w_i x_i + b\right) \tag{3.8}$$

To classify N number of classes the last FC layer usually has the same amount of neurons as classes. A common activation function for multi class categorisation is the softmax function. It is a normalised exponential function that calculates the probability of each class over all possible classes. The outputs is in a range of $0 - 1$ and the sum of all class probabilities is 1. The function is described in **??** where Z is a vector of weights from the previous layer, j is the output class and K is the total amount of classes.

$$\sigma(Z)_j = \frac{e^{Z_j}}{\sum_{\mathrm{k=1}}^{K} e^{Z_k}} \tag{3.9}$$

A common loss function that is used to optimise the softmax function is a cross-entropy loss function. Cross-entropy is defined as **??** where q is the is the estimated distribution and p is the actual distribution. In other words it's a measurement of how far the estimated classes are from the true classes.

$$H(p,q) = -\sum_x p(x) log(q(x)) \tag{3.10}$$

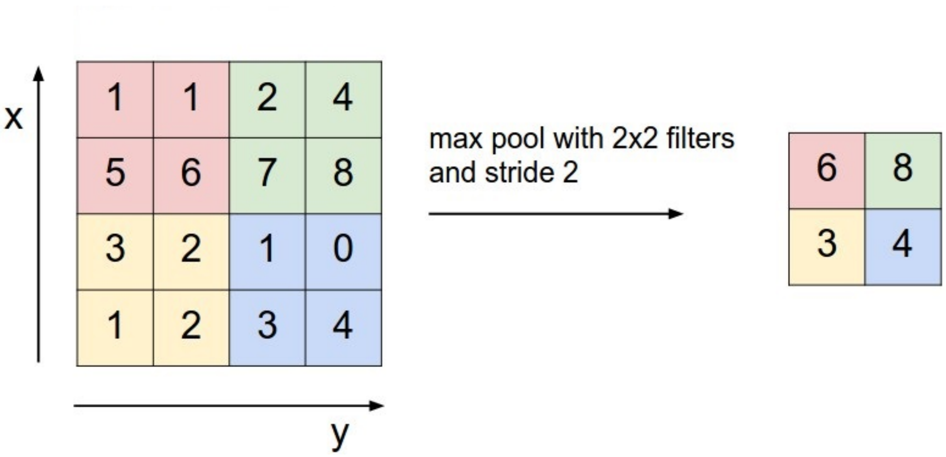


Figure 3.15: Example of maxpooling with size $2x2$ and stride 2 [**?**]

The estimated classes are softmax function, in other words $q = \sigma(Z)_j$. The loss function that is optimised is then **??**.

$$J(\theta) = -\frac{1}{m}\left[\sum_{\mathrm{i=1}}^{m}\sum_{\mathrm{j=1}}^{K} 1\{y_i = j\} log\left(\frac{e^{\theta_j}}{\sum_{\mathrm{k=1}}^{K} e^{\theta_k}}\right)\right] + \frac{\lambda}{2}\sum_{\mathrm{i=1}}^{K}\sum_{\mathrm{j=0}}^{n}\theta_{\mathrm{ij}}^2 \tag{3.11}$$

θ are the weights that are randomly set and are learned through back propagation. K is the number of classes and n is the number of labelled training samples. y_i is the label and 1· is a function that is 1 when the input is true and 0 when it is not. The second term is weight decay regularization term that reduces the magnitudes of weights and prevents it from overfitting.

3.2.5 IrisConv structure

What and why. What is their structure? How did we implement it(python/keras)

3.2.6 The structure we ended up using.

Bibliography

Al-Waisy, A. S., Qahwaji, R., Ipson, S., Al-Fahdawi, S., and Nagem, T. A., 2017. A multi-biometric iris recognition system based on a deep learning approach. *Pattern Analysis and Applications*, (0123456789), pp. 1–20. ISSN 14337541. doi: 10.1007/s10044-017-0656-1. Available at: <https://doi.org/10.1007/s10044-017-0656-1>.

Bazrafkan, S., Thavalengal, S., and Corcoran, P., 2017. An End to End Deep Neural Network for Iris Segmentation in Unconstraint Scenarios. dec. Available at: <http://arxiv.org/abs/1712.02877>.

Chen, C.-H. and Te Chu, C., 2005. Fusion of Face and Iris Features for Multimodal Biometrics. pp. 571–580. ISSN 03029743. doi: 10.1007/11608288_76. Available at: <http://link.springer.com/10.1007/11608288{_}76>.

Connaughton, R., Bowyer, K. W., and Flynn, P. J. *Chapter 17 Fusion of Face and Iris Biometrics*. 2016. ISBN 978-1-4471-6782-2. doi: 10.1007/978-1-4471-6784-6. Available at: <http://link.springer.com/10.1007/978-1-4471-6784-6>.

Daugman, J., 1993. High confidence visual recognition of persons by a test of statistical independence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15 (11), pp. 1148–1161. ISSN 01628828. doi: 10.1109/34.244676. Available at: <http://ieeexplore.ieee.org/document/244676/>.

Dessimoz, D., Richiardi, J., Champod, C., and Drygajlo, A., 2007. Multimodal biometrics for identity documents ({A figure is presented}). *Forensic Science International*, 167(2-3), pp. 154–159. ISSN 03790738. doi: 10.1016/j.forsciint.2006.06.037.

Fierrez, J., Morales, A., Vera-Rodriguez, R., and Camacho, D., 2018. Multiple classifiers in biometrics. part 1: Fundamentals and review. *Information Fusion*, 44 (November 2017), pp. 57–64. ISSN 15662535. doi: 10.1016/j.inffus.2017.12.003. Available at: <https://doi.org/10.1016/j.inffus.2017.12.003>.

Ho, T. K., Hull, J. J., and Srihari, S. N., 1994. Decision Combination in Multiple Classifier Systems. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(1), pp. 66–75. ISSN 0162-8828. doi: http://dx.doi.org/10.1109/34.273716.

Huang, G. B., Ramesh, M., Berg, T., and Learned-Miller, E., 2007. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. *University of Massachusetts Amherst Technical Report*, 1, pp. 07–49. ISSN 1996756X. doi: 10.1.1.122.8268. Available at: <http://vis-www.cs.umass.edu/lfw/lfw.pdf>.

Khan, F. F., Akif, A., and Haque, M. A., 2017. Iris Recognition using Machine Learning from Smartphone Captured Images in Visible Light. pp. 26–28.

Kuehlkamp, A. and Bowyer, K. W., 2016. An analysis of 1-to-first matching in iris recognition. *2016 IEEE Winter Conference on Applications of Computer Vision, WACV 2016*. doi: 10.1109/WACV.2016.7477687.

Luhadiya, R. and Khedkar, A., 2017. Iris detection for person identification using multiclass SVM. *2016 IEEE International Conference on Advances in Electronics, Communication and Computer Technology, ICAECCT 2016*, pp. 387–392. doi: 10.1109/ICAECCT.2016.7942619.

Ortega-Garcia, J., Fierrez, J., Alonso-Fernandez, F., Galbally, J., Freire, M. R., Gonzalez-Rodriguez, J., Garcia-Mateo, C., Alba-Castro, J. L., Gonzalez-Agulla, E., Otero-Muras, E., Garcia-Salicetti, S., Allano, L., Ly-Van, B., Dorizzi, B., Kittler, J., Bourlai, T., Poh, N., Deravi, F., Ng, M. N., Fairhurst, M., Hennebert, J., Humm, A., Tistarelli, M., Brodo, L., Richiardi, J., Drygajlo, A., Ganster, H., Sukno, F. M., Pavani, S. K., Frangi, A., Akarun, L., and Savran, A., 2010. The multiscenario multienvironment biosecure multimodal database (BMDB). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6), pp. 1097–1111. ISSN 01628828. doi: 10.1109/TPAMI.2009.76.

Petrovska-Delacrétaz, D., Lelandais, S., Colineau, J., Chen, L., Dorizzi, B., Ardabilian, M., Krichen, E., Mellakh, M. A., Chaari, A., Guerfi, S., D'Hose, J., and Amor, B. B., 2008. The IV2 multimodal biometric database (including Iris, 2D, 3D, stereoscopic, and talking face data), and the IV2-2007 evaluation campaign. *BTAS 2008 - IEEE 2nd International Conference on Biometrics: Theory, Applications and Systems*, 00, pp. 3–9. doi: 10.1109/BTAS.2008.4699323.

Rattani, A. and Derakhshani, R., 2017. Ocular biometrics in the visible spectrum: A survey. *Image and Vision Computing*, 59, pp. 1–16. ISSN 02628856. doi: 10.1016/j.imavis.2016.11.019. Available at: <http://dx.doi.org/10.1016/j.imavis.2016.11.019>.

Rifaee, M., Abdallah, M., and Okosh, B., 2017. A Short Survey for Iris Images Databases. *international journal of multimedia & its applications*, (April). Available at: <https://www.researchgate.net/publication/316093004{_}A{_}Short{_}Survey{_}for{_}Iris{_}Images{_}Databases>.

Ross, A. and Jain, A., 2003. Information Fusion in Biometrics. 24(13), pp. 2115–2125.

Saha, R., Kundu, M., Dutta, M., Majumder, R., Mukherjee, D., Pramanik, S., Thakur, U. N., and Mukherjee, C., 2017. A Brief Study on Evolution of Iris Recognition System. *Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2017 8th IEEE Annual*, pp. 685–688. Available at: <http://ieeexplore.ieee.org.ezproxy.psu.edu.sa/stamp/stamp.jsp?arnumber=8117234>.

Sequeira, A. F., Monteiro, J. C., Rebelo, A., and Oliveira, H. P., 2014. MobBIO: A Multimodal Database Captured with a Portable Handheld Device. *9th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, (c), pp. 1–14.

Sun, Y., Wang, X., and Tang, X. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1891–1898, 2014a. ISBN 9781479951178. doi: 10.1109/CVPR.2014.244. Available at: <http://mmlab.ie.cuhk.edu.hk/pdf/YiSun{_}CVPR14.pdf>.

Sun, Y., Wang, X., and Tang, X. Deep Learning Face Representation by Joint Identification-Verification. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1891–1898, 2014b. ISBN 9781479951178. doi: 10.1109/CVPR.2014.244. Available at: <https://arxiv.org/pdf/1406.4773.pdfhttp://arxiv.org/abs/1406.4773>.

Uka, A., Roçi, A., and Koç, O., 2017. Improved Segmentation Algorithm and Further Optimization for Iris Recognition. *IEEE EUROCON 2017 -17th International Conference on Smart Technologies*, (July), pp. 6–8.

Wechsler, H. *Reliable face recognition methods: System design, implementation and evaluation.* Springer US, Boston, MA, 2007. ISBN 038722372X. doi: 10.1007/978-0-387-38464-1. Available at: <http://link.springer.com/10.1007/978-0-387-38464-1https://link-springer-com.zorac.aub.aau.dk/book/10.1007{%}2F978-0-387-38464-1{#}about>.

Yin, Y., Liu, L., and Sun, X. SDUMLA-HMT: A multimodal biometric database. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7098 LNCS, pp. 260–268, 2011. ISBN 9783642254482. doi: 10.1007/978-3-642-25449-9_33.

Zhao, Z. and Kumar, A., 2017. Towards More Accurate Iris Recognition Using Deeply Learned Spatially Corresponding Features. *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 3829–3838. doi: 10.1109/ICCV.2017.411. Available at: <http://ieeexplore.ieee.org/document/8237673/>.