Gender identification from video sequences

Master's Thesis in Engineering Mathematics

JOHAN WIEBE

Department of Mathematical Sciences
Mathematical Statistics
Chalmers University of Technology
Gothenburg, Sweden 2014

Abstract

Automated ways of identifying human attributes such as gender from surveillance video is a fast growing area in machine vision today. This thesis presents, investigates and implements a system framework with the task of identifying gender from, relatively low resolution, detected faces in video sequences recorded in a natural environment, i.e. arbitrary poses, facial expressions, illumination, occlusion etc.

An individuals face is detected, tracked and classified during multiple frames to get a more robust and reliable classification result for each individual. A Support vector machine trained on a dataset of still facial images collected under more controlled settings is used as a classifier.

The proposed framework achieves a classification rate of 80% on the recorded video sequences and, while not perfect, could be successfully used to estimate the distribution of males and females entering e.g. a shopping mall.

Acknowledgements

This masters thesis was done in collaboration with Viametrics. I would therefore like to thank Dan Ekström, Development Manager, and the rest of the colleagues at Viametrics for the support and advice received during this project and for giving me the opportunity of doing my master thesis with them.

I would also like to thank my supervisor at Chalmers, Professor Mats Rudemo, for his support and ideas during the length of this project.

Johan Wiebe, Gothenburg 14/06/02

Contents

1

Introduction

1.1 Background

Viametrics AB is a fast growing company that specializes in visitor counters for various businesses, providing both measurement equipment and software. Their equipment provides stores and shopping malls with various statistics concerning their business. This can for example be data on how many people that visits a store during one day or a continuous update on the number of visitors during the day. Statistics like that can be used to adjust and plan opening hours and the amount of staff needed during different periods of a day, to evaluate effects of marketing strategies and to see the flows in, for example, a shopping mall.

Research for identifying human attributes such as gender or age using machine vision has increased during recent years. Such attributes is a vital part in many applications such as surveillance, demographic studies and targeted advertising and marketing. Providing data and statistics over such attributes is of great value for stores and shopping malls giving an additional dimension apart from just counting the number of visiting people. This is the main reason for Viametrics AB interest in the subject of gender identification from video sequences.

Many retail stores and shopping malls today have active video surveillance systems. These systems are mainly used for security, safety and forensic evidence collection and usually the collected data is analysed post event. Manually searching through collected data from video surveillance is very time consuming due to large amount of data. Therefore automatically extracting data giving, for example, the gender of persons is a quick and time saving process. Such a system is also non-intrusive and do not require human cooperation, physical contact or attention. Since the system is not required to recognise or identify individuals but merely the gender of a person the privacy of individuals is protected.

Studies have shown that a human can identify the gender of an individual with over

95% accuracy from facial images (see Bruce (1993)). Even from poor quality images and video sequences can this be done relatively easy by a human. For machine vision however this is a hard and challenging task and classification of attributes such as gender have not been as well studied as for example individual recognition. In this thesis a system for recognising human gender from real time video footage is to be developed, implemented and evaluated using ground truth data.

1.2 Aim

The aim of this masters thesis is to evaluate and implement a sufficiently robust and accurate method for human gender recognition from real time video footage.

1.3 Problem description

Identifying human gender from video footage poses many challenges. Since the data considered will, in many instances, be comprised of video sequences of more than one person at a time, and with some possible occlusion, a robust way to segment the frame/frames and single out each individual person in it needs to be found. Further a robust method for detecting and capturing the body, face and/or gait of each individual person has to be found. The same individual will be visible in more than one frame which means that detections of the same individual have to be associated over many consecutive frames. Visual traits or features have to be extracted from each detection. How many and which features are needed for classification? Which classification method gives the most accurate results? How high will the classification rate be?

1.4 Limitations

The main objective of the masters thesis is to identify gender from video sequences. In a real application only individuals walking in through a store, or mall, entrance would be considered. Thus, depending on the data obtained, only individuals walking towards the cameras field of view will be considered. Only one camera will be used and thus no overlap between two cameras field of view will be evaluated. Because of a relative low number of individuals appearing in the video sequences the classifier will be trained on still images obtained from existing datasets rather than on frames from the video sequences. The camera will be static and positioned in one angle which gives that different angles on the camera can not be compared.

1.5 Report structure

Chapter 2 offers an explanation and overview of the proposed system framework and how it is to be constructed along with an overview of previous work made in the field.

Chapter 3 describes the data used to evaluate and conduct the experiments in the thesis.

Chapter 4 gives a more detailed description of each step in the system framework. A theoretical explanation of the methods used in the system framework is given along with some basic illustrative examples. A small survey over previous research is also included.

In chapter 5 the results obtained from the system framework is evaluated and compared and possible problems/issues with each step in the framework are explained.

A discussion on the results obtained in this masters thesis is given in chapter 6 and finally some conclusions and thoughts on how to make some possible improvements to obtain better results in future work are given in chapter 7.

2

System framework

Most of the previous work made in gender classification with computer vision has been performed on still images (Lapedriza et al. (2006); Vankayalapati et al. (2011); Makinen and Raisamo (2008); Moghaddam and Yang (2002b)). Alrashed and Berbar (2013) proposed using only the eye and eyebrow region to classify gender from still images which gave promising results. Li et al. (2012) combined clothing, hair and facial features for identifying gender from still images. Toews and Arbel (2009) presented a framework for detecting, localizing and classifying face images in terms of soft biometrics, e.g. gender or age, from arbitrary viewpoints and with possible occlusion achieving an error rate of 16.3%.

When classifying gender from video sequences most of the previous research use face images detected and extracted from a frame in a video sequence. These extracted face images are then used for classifying gender. Shakhnarovich et al. (2002) used a framework similar to this and proposed tracking of a detected face over multiple frames in a video sequence and then use temporal integration for classification of gender achieving a misclassification error rate of 21%. Demirkus et al. (2010) attempted to achieve gender classification from face images acquired from totally unconstrained video sequences, i.e. subjects are unconstricted in terms of facial expression, viewpoint, illumination, occlusion etc., proposing a Bayesian framework to classify detected facial images over multiple frames.

Others have suggested using gait for recognising gender from a video sequence. Sudha and Bhavani (2012) and Yu et al. (2009) proposed using a side view of a walking subject for gender identification with the assumption that the camera is static and that the only moving object in the video sequence is the walking subject. Using gait for gender classification has received much attention from researchers in the last years but due to limitations it might not be very suitable for real world applications since they are often restricted to controlled environments such as no occlusion and only certain camera angles can be utilized making it hard to use it in e.g. a cluttered scene. A method proposed by Shan et al. (2008) was to combine gait and facial features for gender identification.

Ng et al. (2012) gives a review of approaches for classifying gender from either a still facial image or gait sequence concluding that while good results have been achieved under controlled environments much work can be done to improve the robustness of gender recognition under real life environments.

The system framework for automatic gender identification from video sequences proposed in this thesis can be divided into three main modules (see figure 2.1 for a flowchart of the system framework):

- **Background subtraction, face detection and tracking:** A video sequence is fed into the system frame by frame. A background subtraction method is applied to detect moving objects in each frame. A face detection algorithm is then used to find possible faces in each frame. In order to reduce the number of false face detections a detected face is discarded if it contains more background pixels than a given threshold. As soon as a face has been detected it is given a unique id and is tracked through subsequent frames until it passes out of the cameras view or crosses over a predetermined line.

- **Feature extraction:** When a face has been detected features are extracted from it which are then passed to a classifier. This is done for all detections and in all frames.

- **Classification:** Features extracted from a face are used to classify the face as either male or female. The classification is done for all detected faces in all frames. If a track of a particular face has ended the classifier will return a final identification according to a majority voting rule based on all classification results of that particular face, e.g. if the same person has been classified as a male in seven frames and as a female in three frames the person will be classified as a male.

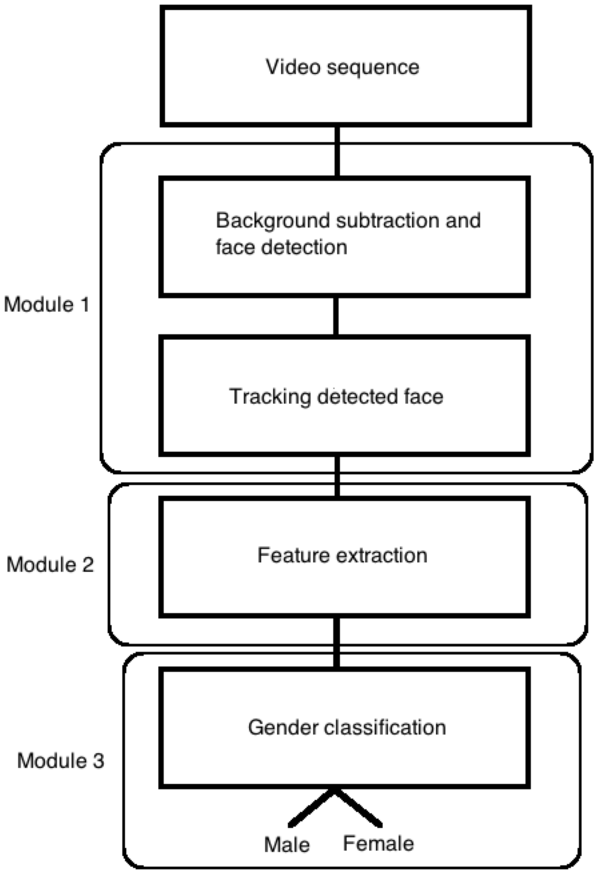
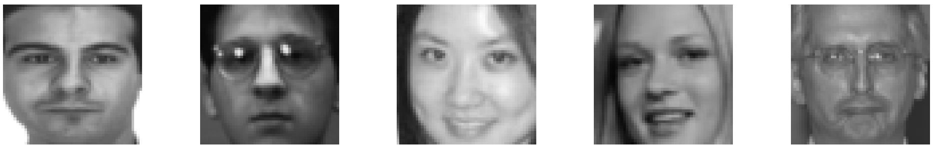


Figure 2.1: Flowchart over the automatic gender classification system.

3

Data

Portions of the research in this thesis use still images of faces taken from different image databases. These are: The AR face database (Martinez and Benavente (1998)), Face recognition data from the University of Essex (Spacek (2008)), the Kinship Verification Dataset (Fang et al. (2010)), Labeled Faces in the Wild (Huang et al. (2007)) and the FERET database of facial images collected under the FERET program, sponsored by the DOD Counterdrug Technology Development Program Office (Phillips et al. (1998); Phillips et al. (2000)). Each of these databases consists of multiple facial images taken under various conditions. Table 3.1 shows a summary of the different databases and figure 3.1 shows an example face image from each dataset. Note that some databases have multiple images of the same individual but in this thesis only images of unique individuals are considered.



(a) AR database.

(b) Essex University.

(c) Kinship dataset.

(d) Labeled Faces in the Wild.

(e) Color FERET.

Figure 3.1: Example of facial images from the AR, Essex University, Kinship, Labeled Faces in the Wild and Color FERET databases.

Dataset	# images	# unique individuals	# images used in thesis	Environment
AR face database (Martinez and Benavente (1998))	>4000	126	100 (50 male and 50 female)	Frontal view faces with different facial expressions, illumination conditions, and occlusions (sun glasses and scarf)
Face data from University of Essex (Spacek (2008))	7900	395	55 (34 male and 21 female)	Frontal view faces with controlled illumination. Some wearing scarfs and/or glasses
Kinship verification dataset (Fang et al. (2010))	288	288	286 (181 male and 105 female)	Frontal view faces taken from the web of parents and their children
Labeled Faces in the Wild (Huang et al. (2007))	> 13000	5751	2842 (1419 male and 1423 female)	Unconstrained face images taken from the web
Color FERET (Phillips et al. (1998); Phillips et al. (2000))	14126	1199	992 (589 male and 403 female)	Controlled settings with varying poses (frontal, profile etc.)

Table 3.1: Summary of Face datasets used.

The still images of faces are mainly used for training, testing and evaluating different feature extraction methods. They are also used for training the classifiers used.

To evaluate how well the proposed framework and gender identification system performs video sequences of people walking into a store entrance are used. These video sequences were recorded with a Axis M1014 Network Surveillance Camera at the office of Viametrics AB and at the entrance of a store inside a shopping mall. The camera resolution used when acquiring the video sequences was 1280 × 800 pixels. The camera was positioned to point downward in an approximately 45° angle. A frame from a video sequence is shown in figure 3.2.

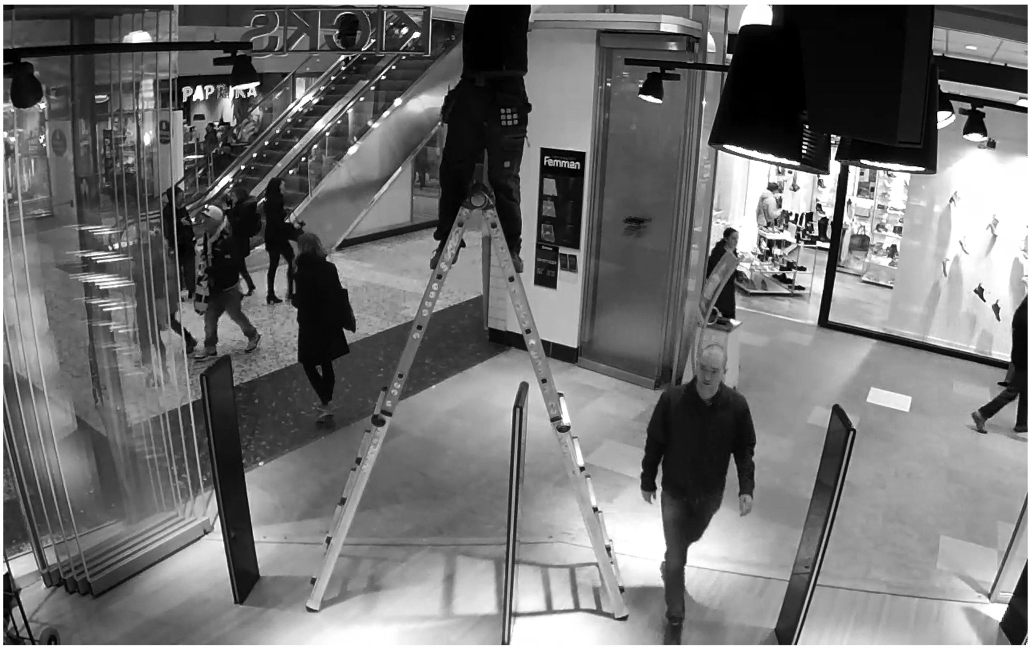


Figure 3.2: Example of a frame taken from video sequence filmed with resolution 1280×800 pixels at a store entrance inside a shopping mall.

4

Theory

This chapter describes in more detail the theory behind the methods used to construct the proposed system framework in chapter 2.

4.1 Background subtraction

The first step in the proposed framework for gender identification is detection of moving objects. Background modelling is commonly used in applications to detect moving objects in video sequences. The simplest way to do this is to acquire a background image of the scene that does not include any moving objects. In many applications and environments such an image is not available. Since the background can be changed in critical situations, by new objects being introduced or removed and by illumination changes, ways to model the background of a scene that are more robust and more adaptable has been introduced. Surveys over background subtraction methods can be found in Benezeth et al. (2010) and Piccardi (2004).

In this framework background modelling and foreground detection is used to reduce the number of false face detections. Since a detected face is discarded if the number of foreground pixels in the detected face is too low a background model that has minimal aperture problem is needed.

The chosen method for modelling the background in this thesis is Mixture of Gaussians (MoG). Many different variations on this approach for background modelling have been proposed over the years and a survey over such is given by Bouwmans et al. (2008). A widely used Mixture of Gaussians model was proposed by Stauffer and Grimson (1999) and offers a good compromise between robustness, computation time and memory requirement and is the model used for this framework. The principal for the method is described below.

4.1.1 Mixture of Gaussians (MoG)

Each pixel is characterised by its intensity in the RGB colour space and the value of a particular pixel over time is considered a time series. At every time t the history of a pixel (x,y) is known, i.e.

$$\{X_1,...,X_t\} = \{I(x,y,i) : 1 \leq i \leq t\} \tag{4.1}$$

where I is an image sequence.

With a static background and with static illumination the intensity of a pixel over time is relatively constant. The history of each pixel, $\{X_1,...,X_t\}$, is modelled by a mixture of K multivariate Gaussian distributions and the probability of observing the current pixel value is given by

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t}\eta(X_t,\mu_{i,t},\Sigma_{i,t}) \tag{4.2}$$

where K is the number of Gaussian distributions, $\omega_{i,t}$ is an estimate of the weight associated with the i:th Gaussian at time t (what proportion of the data that is accounted for by the i:th Gaussian), $\mu_{i,t}$ and $\Sigma_{i,t}$ is the mean and the covariance matrix of the i:th Gaussian at time t. η is a multivariate Gaussian probability density function given by

$$\eta(X_t,\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t-\mu)^T\Sigma^{-1}(X_t-\mu)} \tag{4.3}$$

where $n = 3$ to describe the red, green and blue colour channels. For computational purposes Stauffer and Grimson (1999) assumed that the covariance matrix is diagonal, i.e the different colour components are independent and have the same variance which gives that the covariance matrix is of the form

$$\Sigma_{i,t} = \sigma_{i,t}^2\mathbf{I} \tag{4.4}$$

where $\mathbf{I}$ is the identity matrix. The number of Gaussian distributions, K, is determined by available memory and computational power. Stauffer and Grimson (1999) proposed setting $K \in \{3,4,5\}$. The parameters $\omega_{i,t}$, $\mu_{i,t}$ and $\Sigma_{i,t}$ of the background model are initialized using an EM (Expectation Maximization) algorithm (Watanabe and Yamaguchi (2004)) which is essentially an iterative method for finding the maximum likelihood estimates of parameters in a statistical model. For real-time applications Stauffer and Grimson (1999) proposed using a K-mean algorithm. For modelling the background the K Gaussian distributions are first ordered according to the ratio

$$r_j = \frac{\omega_j}{\sigma_j} \tag{4.5}$$

A background pixel is assumed to have a large weight, ω_j, and a low variance σ_j (since the background is more often present and is almost constant compared to a moving object) which gives a large value for the ratio r_j. The first B Gaussians exceeding a threshold, τ, are chosen as the background model, where

$$B = \operatorname{argmin}_b(\sum_{i=1}^{b} \omega_{i,t} > \tau) \tag{4.6}$$

where τ is a measure representing the minimum portion of the image that is to be considered background. The other distributions are then assumed to represent the foreground. Every new pixel value, X_{t+1}, is then matched against all the K Gaussian distributions until a match is found. A pixel is matched with a Gaussian distribution if it lies within 2.5 standard deviations from the mean of a distribution, i.e

$$\sqrt{(X_{t+1} - \mu_{i,t})^T \Sigma_{i,t}^{-1} (X_{t+1} - \mu_{i,t})} < 2.5\sigma_{i,t} \tag{4.7}$$

If a match is found with one of the K Gaussian distributions and if the distribution is one of the B distributions identified as background the pixel is classified as a background pixel, otherwise it is classified as a foreground pixel. If no match is found among the K Gaussian distributions the pixel is classified as a foreground pixel. This gives a binary mask containing the foreground of the current frame. To make the next foreground detection the parameters $\omega_{i,t}$, $\mu_{i,t}$ and $\sigma_{i,t}^2$ must be updated. There are two cases to consider: Case one is when a match is found with one of the K Gaussian distributions. The matched components in this case are updated according to

$$\begin{aligned} \omega_{i,t+1} &= (1-\alpha)\omega_{i,t} + \alpha \\ \mu_{i,t+1} &= (1-\rho)\mu_{i,t} + \rho X_{t+1} \\ \sigma_{i,t+1}^2 &= (1-\rho)\sigma_{i,t}^2 + \rho(X_{t+1} - \mu_{i,t+1})^T (X_{t+1} - \mu_{i,t+1}) \end{aligned} \tag{4.8}$$

where α is a constant learning rate determining the speed with which the parameters of the distribution change and $\rho = \alpha\eta(X_{t+1},\mu_i,\Sigma_i)$. The unmatched components are updated as

$$\begin{aligned} \omega_{i,t+1} &= (1-\alpha)\omega_{i,t} \\ \mu_{i,t+1} &= \mu_{i,t} \\ \sigma_{i,t+1}^2 &= \sigma_{i,t}^2 \end{aligned} \tag{4.9}$$

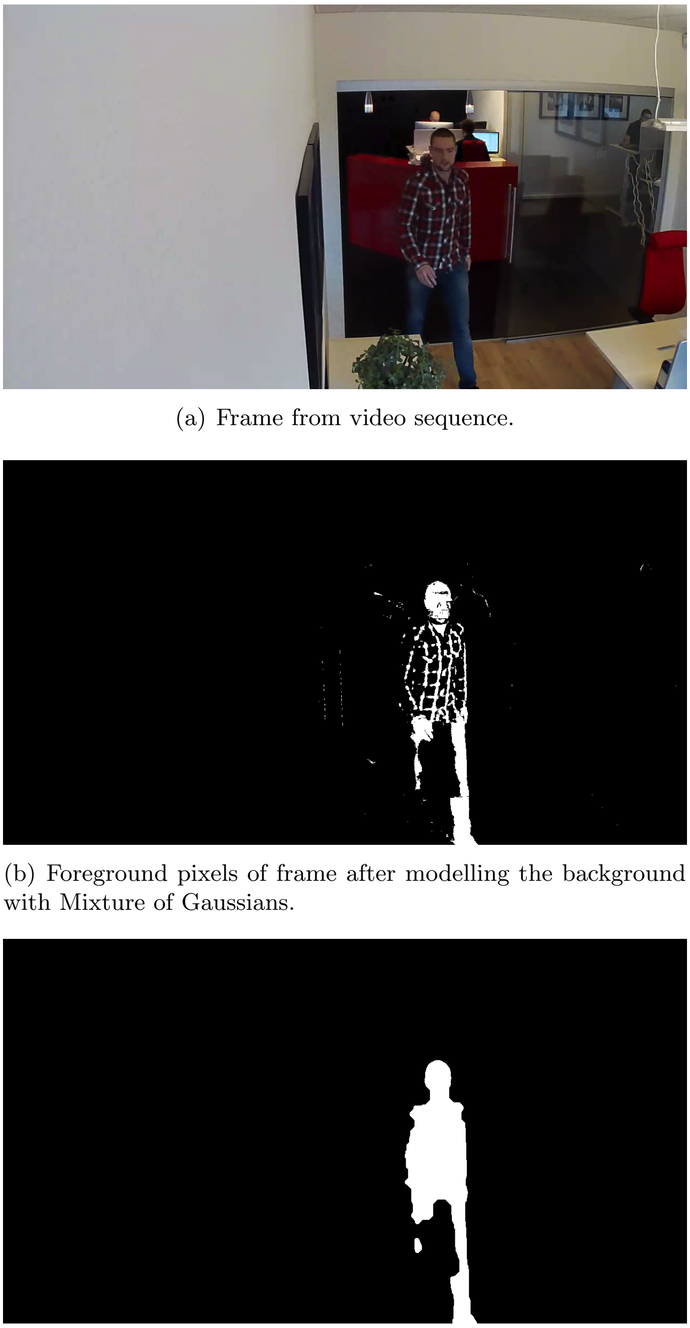
i.e. the mean and the covariance matrix are unchanged and only the weight is updated. Case two is when no match is found among the K distributions. In that case the least probable distribution, k, is updated to

$$\begin{aligned} \omega_{k,t+1} &= \text{Low prior weight} \\ \mu_{k,t+1} &= X_{t+1} \\ \sigma_{k,t+1}^2 &= \text{High initial variance} \end{aligned} \tag{4.10}$$

Updating the parameters according to equations (4.8), (4.9) and (4.10) allows for foreground detection throughout a video sequence.

Detecting foreground using Mixture of Gaussians is not perfect and the resulting foreground mask may contain some noise. Therefore each foreground mask frame acquired from the background modelling is processed by applying a median filter to remove so called *salt and pepper* noise. This is followed by two morphological operations, opening and closing. Opening is done to remove small clusters of foreground pixels that can be regarded as noise and closing is done to bridge gaps between nearby clusters of pixels assumed to be belonging to the same foreground object.

A example of applying the described method for detecting moving objects in a video sequence is shown in figure 4.1.



(a) Frame from video sequence.

(b) Foreground pixels of frame after modelling the background with Mixture of Gaussians.

(c) Foreground pixels after noise removal with a median filter and applying opening and closing morphological operations.

Figure 4.1: Frame from video sequence recorded at the office of Viametrics AB with resolution 1280 × 800 with the resulting foreground mask after modelling the background with Mixture of Gaussians and after removing noise and applying morphological operations.

4.2 Face detection

The second part in the proposed framework for gender classification from a video sequence is face detection. Many different methods for detecting faces has been suggested in previous research (Yang et al. (2002)). A widely used and successful face detection method was proposed by Viola and Jones (2004), and is the one used in this thesis (see also Englund (2003)), which describe a robust face detection framework capable of processing images in real time with high detection rates.

The face detection procedure classifies images based on features similar to Haar basis functions. There are three types of features used: two, three and four rectangle features. They are identified as the difference between the sum of the pixels inside two rectangular regions, the difference between the sum of the pixels inside two outer rectangles and the sum of the pixels inside a centre rectangle and last the difference between diagonal pairs of rectangles

The different features are computed at different scales and different orientations. Figure 4.2 shows the different rectangle features in different orientations. All rectangle regions are of the same size, have the same shape and are vertically or horizontally adjacent. The rectangle features are chosen to emphasize the differences in intensity in a face, e.g. eyes and mouth are usually low intensity regions while forehead and nose are high intensity regions.

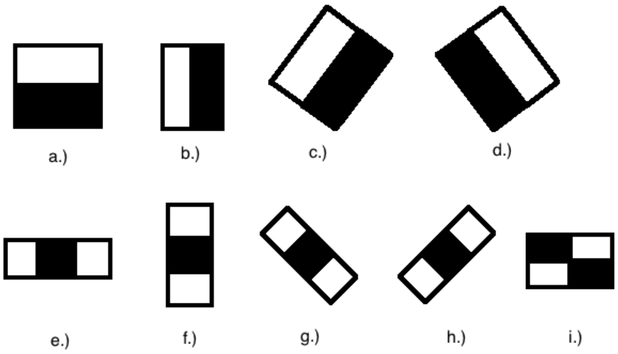


Figure 4.2: Example of three types of rectangle features in different orientations. The sum of the pixels that lie within the white rectangles are subtracted from the sum of the pixels in the black rectangles. a.)-d.) are two-rectangle features, e.)-h.) are three-rectangle features and i.) is four-rectangle features.

In order to efficiently compute the rectangle features a representation for the image called the integral image is used (see figure 4.3). The integral image at location (x,y) is the sum of all the pixels in the original image above and to the left of (x,y) and is given by:

$$\tilde{I}(x,y) = \sum_{i=1}^{x} \sum_{j=1}^{y} I(i,j) \tag{4.11}$$

where $\tilde{I}(x,y)$ is the integral image at location (x,y) and I is the original image.

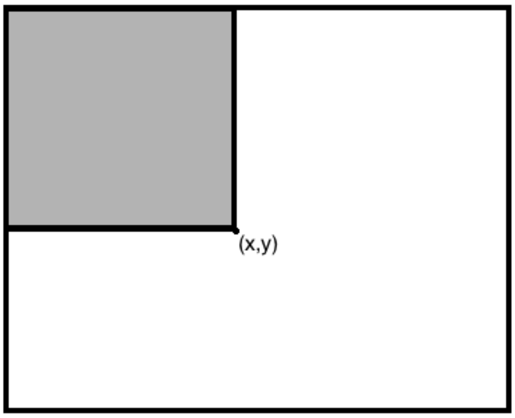


Figure 4.3: The integral image, $\tilde{I}(x,y)$, at location (x,y) is the sum of the pixels in the gray area.

This intermediate representation of an image gives that any rectangle sum can be computed by using the integral image in four different points, i.e. as:

$$\text{Rectangle sum} = \sum_{i=x_1}^{x_2}\sum_{j=y_1}^{y_2} I(i,j) = \tilde{I}(x_1-1,y_1-1)+\tilde{I}(x_2,y_2)-\tilde{I}(x_2,y_1-1)-\tilde{I}(x_1-1,y_2) \tag{4.12}$$

This is further visualized in figure 4.4

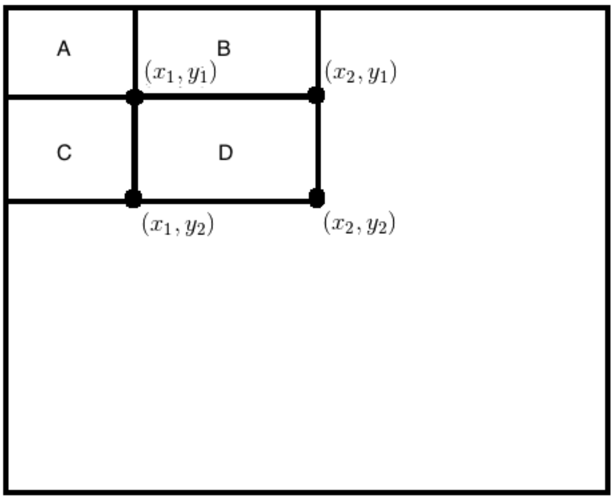
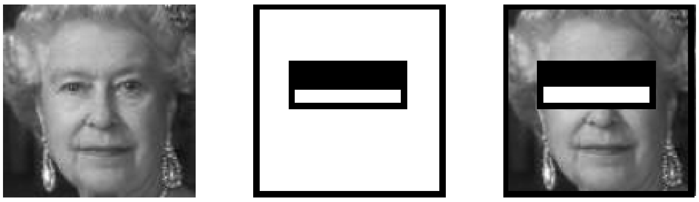


Figure 4.4: The integral image, $\tilde{I}$, at location (x_1,y_1) is the sum of all pixels inside rectangle A. Further the integral image at location (x_2,y_1) is the sum of all pixels inside rectangle A and B, the integral image at location (x_1,y_2) is the sum of all pixels inside rectangle A and C and the integral image at location (x_2,y_2) is the sum of all pixels inside rectangle A, B, C and D. Using equation (4.12) the sum of all pixels inside e.g. rectangle D can be calculated as $\tilde{I}(x_1-1,y_1-1)+\tilde{I}(x_2,y_2)-\tilde{I}(x_2,y_1-1)-\tilde{I}(x_1-1,y_2)$.

The detector scans an input image with a window of size 24 × 24 pixels. The rectangle features seen in figure 4.2 are placed at a suitable location inside this window and calculated. In order to detect faces at different scales the original image is rescaled to different sizes and the window of size 24×24 pixels is scanned across each of these images resulting in many sub images, x, of the original image. This is further visualised in figure 4.5.

The detector needs to be trained to have it work. There are two classes, positive and negative, where the detector labels all sub images, x, (of size 24 × 24 pixels) as belonging to one of these classes. A positive image is an image of a face (see e.g. figure

4.5). If the detector labels a sub image, x, as positive it should be a face. A negative sub image, x, is anything that is not a face. There are also false positives, i.e. a sub image is labeled positive when it really is negative, and false negatives, i.e. a sub image is labeled negative when it should be labeled positive. All the sub images are labeled using a classifier.



(a) Example of a training face image.

(b) Example rectangle feature shown relative to the enclosing window used to scan an image.

(c) Scanning window with rectangle feature overlaid on a training face image giving a sub image x.

Figure 4.5: Example of a two-rectangle feature used for face detection. It measures the difference in intensity between the eyes region and the upper cheek/nose region.

Viola and Jones (2004) proposes a variant of AdaBoost training (Freund and Schapire (1997)) to select the features, given from the rectangle transforms, and to train the classifier. As mentioned before the detector scans an image with a window of size $24 \times 24 pixels$ resulting in a number of sub images, x. The image is rescaled to different sizes to find faces in different scales. There are two types of classifiers used when training the detector on a set of positive and negative images. The first is a weak classifier, h, and the second is a strong classifier, H. The weak classifier is given by:

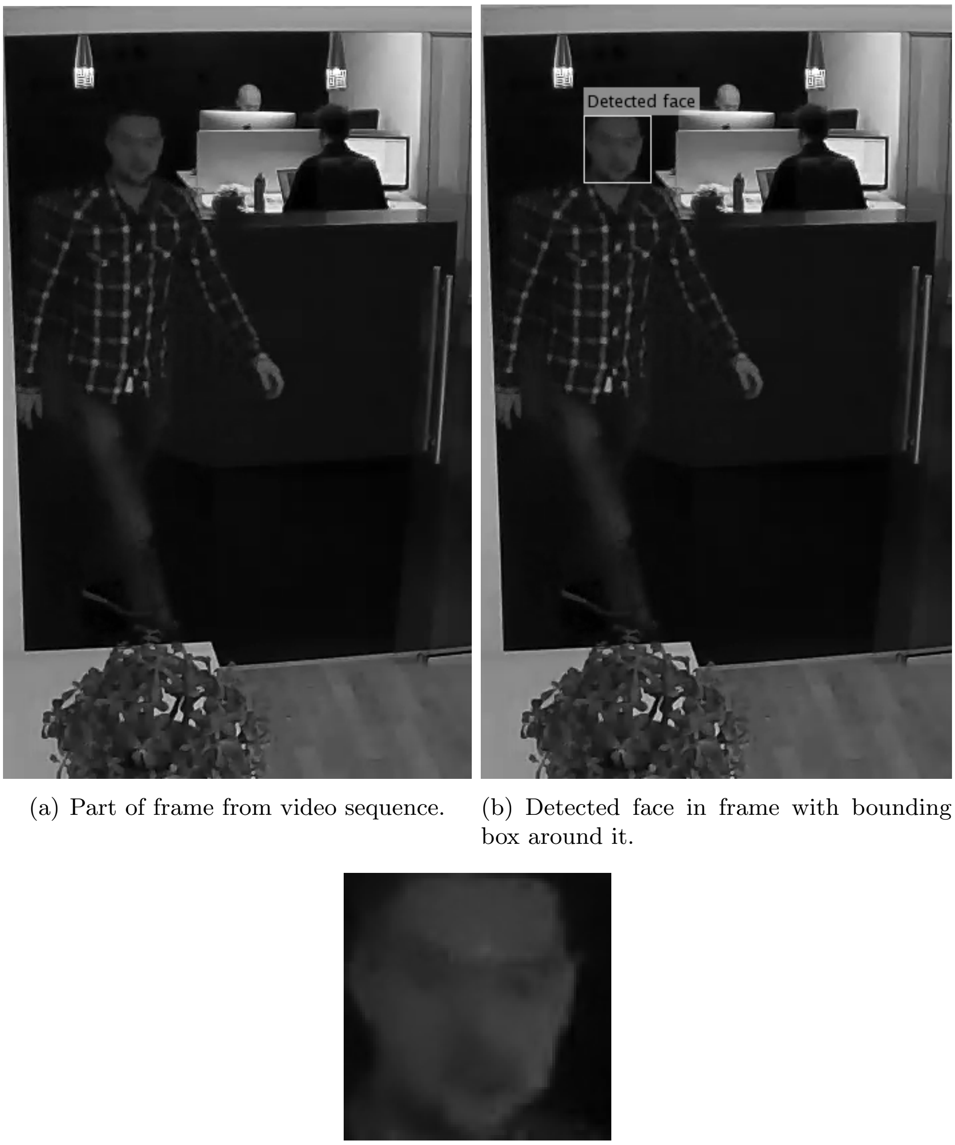
$$h(x,f,p,\theta) = \begin{cases} 1 & \text{if } pf(x) < p\theta \\ 0 & \text{otherwise} \end{cases} \tag{4.13}$$

where $f(x)$ is a feature given by a rectangle transform on a sub-image x, p is a polarity indicating the direction of the inequality and θ is a threshold. For each feature, $f(x)$, the parameters p and θ of the weak classifiers are chosen such that they minimize the number of misclassified sub images x. By choosing the T best weak classifiers, each using a single feature, and assigning a weight α to each of them a strong classifier can be assembled. The weight, α, corresponds to how well the weak classifier did compared to the other weak classifiers. Thus a strong classifier is given by a weighted combination of T weak classifiers, h_j, as

$$H(x) = \begin{cases} 1 & \text{if } \sum_{j=1}^{T} h_j(x)\alpha_j \geq \frac{1}{2}\sum_{j=1}^{T} \alpha_j \\ 0 & \text{otherwise} \end{cases} \tag{4.14}$$

Further information on the AdaBoost learning algorithm for face detection can be found in the thesis by Englund (2003)) and in the article by (Viola and Jones (2004).

An example of the face detection algorithm applied on a video sequence can be seen in figure 4.6.



(a) Part of frame from video sequence.

(b) Detected face in frame with bounding box around it.

(c) Detected face from frame.

Figure 4.6: Proposed face detection algorithm used on a video sequence.

4.3 Tracking

Tracking moving objects is an important aspect of motion-based recognition and classification. Objects, e.g. faces, might not be detected in every frame of a video sequence and since detected objects often needs to be associated between frames a method for tracking detected objects over multiple frames is very important.

There are a wide variety of methods for tracking moving objects and a review over different methods can be seen in the article by Patel and Mishra (2013). Many tracking models is based on dynamic Bayesian networks (Murphy (2002)) and use some kind of recursive Bayesian filter typically a Kalman filter (Patel and Thakore (2013); Villysson (2014); Stauffer and Grimson (2000)). A Kalman filter can be considered a generalisation of dynamic Bayesian networks using a recursive Bayesian filter with multivariate normal distributions (Barker et al. (1995)).

A tracking model often consists of three different steps:

- **Data assignment:** Here the measurements, i.e. detections, at the current time step are assigned to existing tracks of objects.
- **Track management:** Tracks that have new detections assigned to them are validated. Further, detections that are not assigned to any existing track are initialised as new tracks and tracks that have not had any detection assigned to them for a certain time are deleted.
- **Prediction and filtering:** The states of detected objects are predicted so that new detections can be assigned to tracks. Further, the state of the objects is updated.

4.3.1 Data assignment

Assigning new detections to existing tracks can be done using different methods. Patel and Thakore (2013) and Villysson (2014) uses a Nearest-Neighbour approach where the distance between the location of a detected object and the predicted location are calculated and those that lies closest, in a Euclidean distance sense, are associated with each other. Stauffer and Grimson (2000) uses multiple hypothesis tracking to assign new data to existing tracks. At each frame there are a pool of Kalman filters that could explain the new measurements. The most probable of these filters are matched to the tracked objects they can explain.

Another method, which are the one used in this project, is based on Munkres assignment algorithm described by Munkres (1957). First the cost of assigning a detection to a track is calculated by calculating the Euclidian distance between the location of the detected object and the predicted location of the object. This cost is represented as a cost matrix described by Munkres (1957) and is then used to solve Munkres assignment algorithm which associates the new measurement with the track that minimises the total cost.

4.3.2 Track management

In each frame there might exist detections that are not assigned to any existing track and there might also be tracks that do not get any detections assigned to them. Therefore new tracks have to be initialised for the unassigned detections. Tracks that have not had any detections assigned to them are either kept in memory and the predicted location is used as the current location of the tracked object or, if the track has remained unassigned for a certain period of time, it is deleted. Tracks that are assigned a new detection are validated and their predictions are corrected giving them a higher confidence.

4.3.3 Filtering and prediction

Each object is described by a state vector, $\mathbf{x}_k$, in each time step, $k \in \mathbb{N} = \{1,2,...\}$. The state vector of an object include relevant information about the object, e.g.

$$\mathbf{x}_k = [x_k, y_k, \dot{x}_k, \dot{y}_k]^T \tag{4.15}$$

where x and y are the coordinates of the object in the plane and $\dot{x}$ and $\dot{y}$ are the velocities in the x and y directions. In each time step, k, an observation (or measurement), $\mathbf{z}_k$, of the true state, $\mathbf{x}_k$, is made. The state of an object at time, k, is evolved according to a motion model given by

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{w}_{k-1}) \tag{4.16}$$

where $\mathbf{w}_{k-1}$ is a noise process describing the uncertainties in the motion model and $\mathbf{f}_{k-1}$ is a function describing the relationship between the previous and current state of an object. The measurements are made according to a measurement model given by

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{v}_k) \tag{4.17}$$

where $\mathbf{v}_k$ is a noise process describing the uncertainties in the measurements and $\mathbf{h}_k$ is a function describing the relationship between measurements and the state of an object.

When using a Kalman filter first described by Kalman (1960) the following assumptions are made:

- The functions $\mathbf{f}_{k-1}$ and $\mathbf{h}_k$ in equation (4.16) and (4.17) are assumed to be linear.
- The two noise processes $\mathbf{w}_k$ and $\mathbf{v}_k$ in equation (4.16) and (4.17) are assumed to be zero mean Gaussian and along with the initial state $\mathbf{x}_0$ mutually independent for all time steps, k.

These assumptions give the following settings for the motion and measurement models:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} \\ \mathbf{z}_k &= \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \end{aligned} \tag{4.18}$$

where $\mathbf{F}_{k-1}$ is a matrix describing the state transition model, $\mathbf{w}_{k-1}$ is the process noise assumed to be zero mean Gaussian with covariance matrix $\mathbf{Q}_{k-1}$, $\mathbf{H}_k$ is a matrix describing the measurement model and $\mathbf{v}_k$ is the measurement noise assumed to be zero mean Gaussian with covariance matrix $\mathbf{R}_k$.

The state of the Kalman filter is represented by two variables, $\hat{\mathbf{x}}_{k|k}$, denoting the a posteriori estimate of state $\mathbf{x}_k$ at time k given observations up to and including k and $\mathbf{P}_{k|k}$ denoting the a posteriori state estimate error covariance matrix, i.e. a measure of the estimated accuracy of the estimated state, $\hat{\mathbf{x}}_{k|k}$.

The predicted (a priori) state estimate, $\hat{\mathbf{x}}_{k|k-1}$, and the predicted (a priori) state estimate error covariance used for the data assignment (see chapter 4.3.1) is given by:

$$\begin{aligned}\hat{\mathbf{x}}_{k|k-1} &= \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1|k-1} \\ \mathbf{P}_{k|k-1} &= \mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1}\end{aligned} \tag{4.19}$$

By introducing the innovation residual, $\tilde{\mathbf{y}}_k$ and the innovation coveriance $\mathbf{S}_k$ given by equation (4.20), it is possible to derive an expression for the updated (a posteriori) state estimate, $\mathbf{x}_k$, and the updated (a posteriori) state estimate error covariance, $\mathbf{P}_{k|k}$ yielding equation (4.21).

$$\begin{aligned}\tilde{\mathbf{y}}_k &= \mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1} \\ \mathbf{S}_k &= \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k\end{aligned} \tag{4.20}$$

$$\begin{aligned}\hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\tilde{\mathbf{y}}_k \\ \mathbf{P}_{k|k} &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}\end{aligned} \tag{4.21}$$

Here $\mathbf{K}_k$ is the Kalman gain matrix given by

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T(\mathbf{S}_k)^{-1} \tag{4.22}$$

which makes sure that if the accuracy of the measurements is high, i.e. when $\mathbf{R}_k$ is small, then the update of the state vector is mostly influenced by the measurements. Conversely if $\mathbf{P}_{k|k-1}$ is small compared to $\mathbf{R}_k$ then the change in the state is small giving that the predicted estimate are given more weight than the newly assigned measure when updating the state. The updating scheme given by equation (4.21) use the state from the prediction step, $\hat{\mathbf{x}}_{k|k-1}$ (see equation (4.19)) and the newly assigned measurement, $\mathbf{z}_k$, and forms the optimal update for the state of an object.

4.4 Features

Feature extraction is a very important factor in any machine vision recognition system since they are used to separate images belonging to different classes from each other. Features are essentially a set of numbers for an image meant to describe the characteristics and texture of a particular image, or of a whole set of images belonging to the same class. There is no general method to know which, or how many, features that are to be used for a particular classification problem since there is potentially a very large number of features available. The usefulness of a particular feature varies depending on the problem but some good traits is that they should be robust to variabilities within each class and they should be scale and orientation invariant. The feature for each image is collected in a vector known as a feature vector.

A good approach for representing textural features in an image is with a gray-level co-occurence matrix (GLCM), which is a two dimensional histogram of the co-occurence of pixel values in a gray valued image. A GLCM differs from a more simple approach when describing textures in an image. For example using statistical moments of the intensity histogram of an image (or region in an image) only captures information about the distribution of intensities but it does not capture the relative position of the pixels in relation to each other which a co-occurence matrix does.

A GLCM can be used directly as a feature by taking the GLCM for an image and transform the matrix into a column vector. Haralick et al. (1973) introduced a set of 14 easily computable features that can be extracted from a GLCM which well describes the texture of an image. Soh and Tsatsoulis (1999) and Clausi (2002) introduced another 8 features (basically variations on some of the features introduced by Haralick et al. (1973)) making a total of 22 features to be extracted from a GLCM. Appendix A gives a more detailed definition of GLCM and the measures that can be obtained from them.

Gabor features have been shown to perform very well in e.g. face recognition (Messer et al. (2002); Haghighat et al. (2013); Shen et al. (2007); Liu and Wechsler (2002)) and fingerprint matching (Jain et al. (2006)). Gabor features are extracted from an image by convolving it with a Gabor filter of different scales and orientations. Research in neurophysiology have shown that Gabor filters fit the spatial response profile of certain simple cells in the visual cortex of mammalian brains (Daugman (1985)). Thus using Gabor filters for extracting features in image analysis is similar to perception in the human visual system. The greatest advantage of Gabor filters is their invariance to rotation, scale, illumination and translation. Appendix B offers a more detailed explanation of Gabor filters and how to obtain feature vectors from them.

In recent years Local Binary Patterns (LBP) has received increasing interest in many areas of image processing and computer vision and has been shown effective in many applications. Ojala et al. (2002) uses LBP for texture classification achieving good results. Huang et al. (2011) gives a survey over LBP and its application to facial image analysis e.g. face detection, face recognition and demographic classification. Shan (2012) uses LBP for classifying gender on unconstrained facial still images. Further details and the definition of the LBP operator is given in appendix C.

Both Gabor filters and LBP are well evaluated in previous scientific research and have both been shown to obtain good results in the area of facial image analysis. Zhang et al. (2005) proposed a feature combining local binary patterns and the magnitude part of Gabor filters for face recognition called Local Gabor Binary Pattern Histogram Sequence (LGBPHS). This approach is robust to noise and variations in illumination, occlusion and orientation. They achieved very high results on the Colour FERET face database. This approach does however suffer from feature vectors of very high dimensionality making it unsuitable for supervised learning problems due to limits in memory and time. Xia et al. (2008) extended the work by Zhang et al. (2005) and introduced a new feature called Local Gabor Binary Mapping Pattern (LGBMP) obtained by mapping local Gabor binary pattern features into a low-dimensional space. They also compared the performance of different features for gender classification from facial images using Support Vector Machines (SVM) concluding that LGBPM features performed better than both regular LBP features and Gabor features.

The process of extracting LGBMP features works as follows and is further illustrated in figure 4.7:

- Gabor magnitude pictures (GMP) are first obtained by convolving an input facial image with a series of Gabor filters of different scales and orientations.
- Each GMP is converted to a Local Gabor Binary Pattern (LGBP) image by using a Local Binary Pattern (LBP) encoding method.
- Each LGBP image is then divided into m non overlapping equal sized rectangular regions.
- Histograms are computed for each region of the LGBP image giving the LGBP feature.
- Each regional LGBP feature is mapped into a single value and all these are concatenated giving the final feature vector.

Note that an image is often pre-processed by normalisation and histogram equalisation before any features are extracted.

In the following chapter LGBMP are described in more detail.

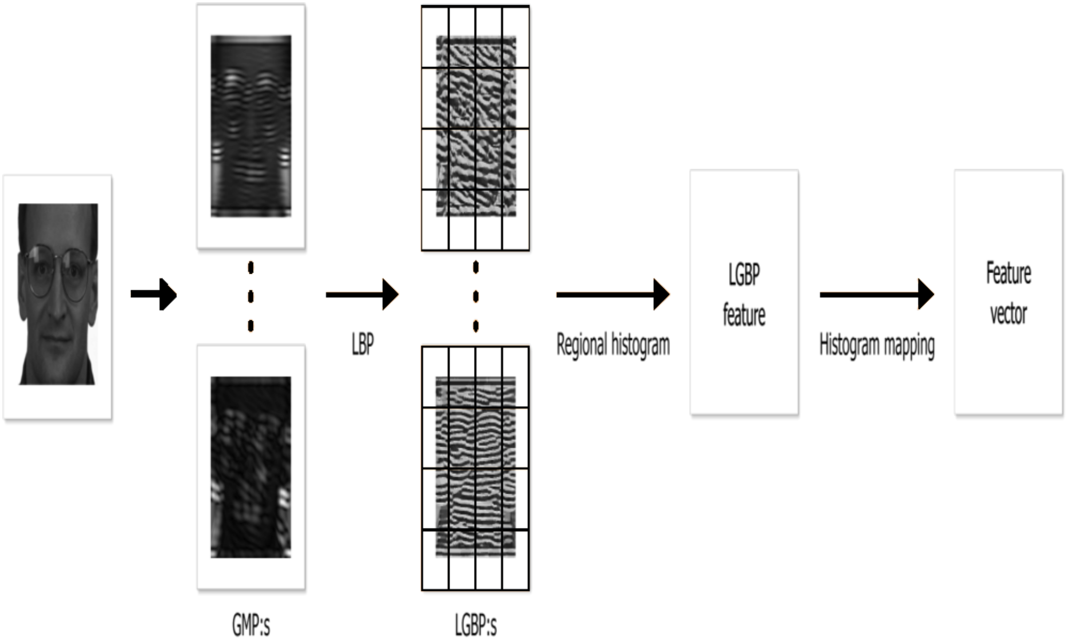


Figure 4.7: Overview of the LGBMP feature extraction process.

4.4.1 Local Gabor Binary Mapping Pattern (LGBMP)

A Gabor magnitude picture (GMP) is obtained by convolving a facial image with a series of Gabor filters at M scales and N orientations. Let $I(x,y)$ be a facial image, its convolution with a Gabor filter $\psi(x,y;f_\nu,\theta_\mu)$ is defined as

$$G(x,y;f_\nu,\theta_\mu) = \psi(x,y;f_\nu,\theta_\mu) * I(x,y) \tag{4.23}$$

where $\nu \in \{0,...,M-1\}$, $\mu \in \{0,...,N-1\}$ and the Gabor filter $\psi(x,y;f,\theta)$ is defined as

$$\begin{aligned} \psi(x,y;f,\theta) &= \frac{f^2}{\pi\gamma\eta} e^{-(\frac{x'+\gamma^2 y'^2}{2\sigma^2})} e^{j2\pi f x' + \phi} \\ x' &= x\cos\theta + y\sin\theta \\ y' &= -x\sin\theta + y\cos\theta \end{aligned} \tag{4.24}$$

where f denotes the tuning frequency of the filter (the frequency of a sinusoidal plane wave), γ and η are parameters controlling the bandwidth corresponding to the two perpendicular axes of the Gaussian, i.e. the spatial widths of the filter. θ denotes the rotation angle of both the Gaussian and the plane wave. ϕ is the phase offset and σ^2 is the variance of the Gaussian envelope. For more details on the Gabor filter see appendix B. Note that only the magnitude of the generated Gabor feature is used thus giving the GMP.

In order to enhance the information in the GMP:s the magnitude values are encoded with a LBP operator (Zhang et al. (2005)). Given a pixel at location (x_c,y_c), the LBP can be expressed as

$$\text{LBP}_{P,R}(x_c,y_c) = \sum_{n=0}^{P-1} s(i_n - i_c)2^P \tag{4.25}$$

where i_c is the gray level value of the central pixel and i_n, $n = 0,...,P-1$ is the gray level values of the P surrounding pixels in a circular neighbourhood with radius R. The function $s(x)$ is defined as

$$s(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \tag{4.26}$$

Different LBP operators can be used e.g. $\text{LBP}^{U2}_{P,R}$ which only accounts for the uniform patterns (see appendix C for more details). By applying a LBP operator on the GMP a Local Gabor Binary Pattern (LGBP) image is obtained. Each LGBP image is then divided into m non-overlapping equal sized rectangular regions $R_0,...,R_{m-1}$ and a histogram of each region is computed. The histogram of the j:th region in the μ:th orientation and the ν:th scale LGBP image is denoted $H_{\mu,\nu,j}$. Finally the histograms from all the LGBP images are concatenated giving the feature vector $X = \{H_{0,0,0},...,H_{0,0,m-1},H_{0,1,m-1},...,H_{N-1,M-1,m-1}\}$.

Since the dimension of the feature vector X can be very large its dimension is reduced by mapping each histogram, $H_{\mu,\nu,j}$, into a single value as described by Xia et al. (2008). For an overview of the LGBMP feature extraction process see figure 4.7.

4.5 Classification

Classification is a collection of different methods for categorizing data. The problem of classification can be illustrated in a scatter plot as seen in figure 4.8. To correctly classify the data one wishes to find features that separate each group of data into clusters.

Before classification it is common practice to normalize the data. Doing so will make the distances in different directions to carry the same weight, i.e. assuming that all features have the same importance.

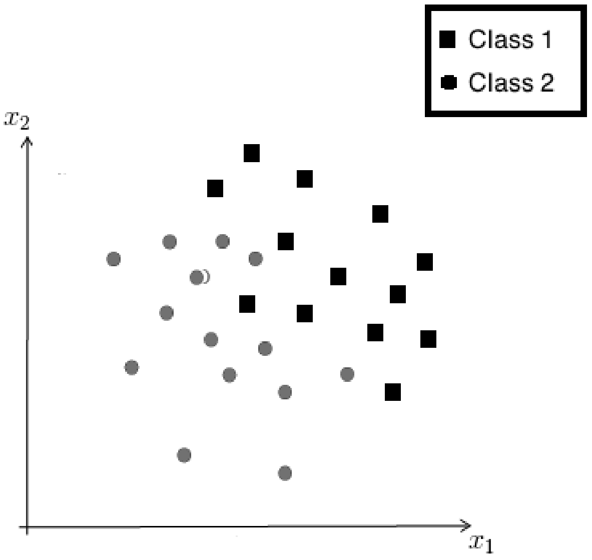


Figure 4.8: Example of a scatter plot showing two features x_1 and x_2 plotted against each other. Circles belong to one class and squares to another class.

Several classifiers have been used for gender classification, e.g. K-Nearest Neighbour (KNN), Neural networks (Cottrell and Metcalfe (1990)), Radial Basis Function Networks (RBF) (Moghaddam and Yang (2002a)) and Support Vector Machines (SVM). A commonly used classifier for gender classification is SVM as can be seen in the survey by Ng et al. (2012).

Moghaddam and Yang (2002a) compared the performance of different classifiers for gender classification from facial images. These included SVM, RBF networks, KNN and classical discriminant methods such as Bayesian (Quadratic) and linear discrimination analysis. According to their results SVM outperformed the other methods which is why it is the chosen method for this project.

4.5.1 Support Vector Machine (SVM)

A Support Vector Machine is a learning algorithm for pattern classification and regression estimation among others. Given a set of training samples (feature vectors) $\mathbf{x}_i \in \mathbb{R}^n$, $i = 1,...,m$ with corresponding binary class labels $y_i \in \{-1,1\}$, the basic principal behind a SVM classifier is to find the optimal hyperplane that correctly separates (classifies) the largest fraction of data points while maximising the distance of each class from the hyperplane as illustrated in figure 4.9

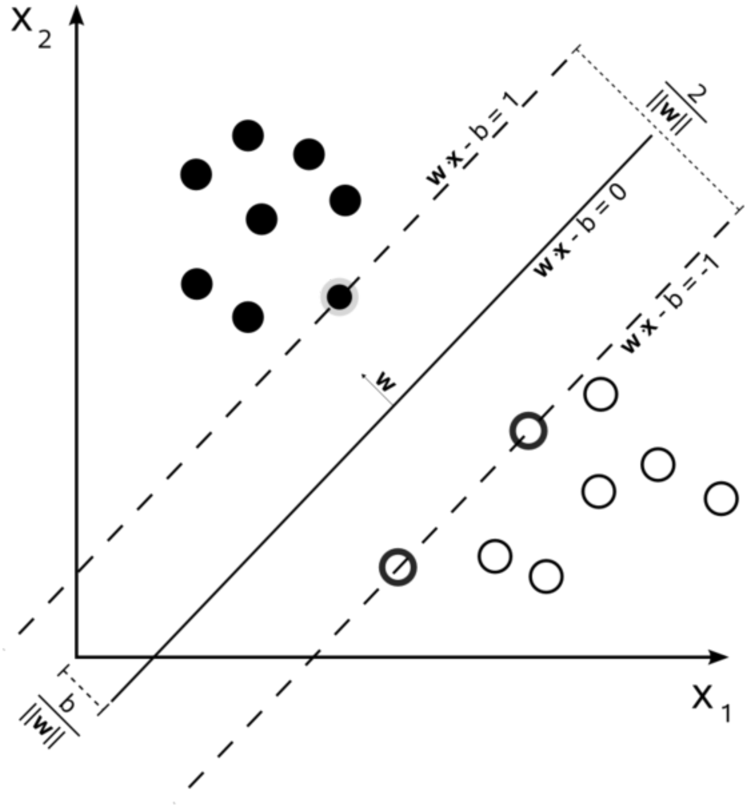


Figure 4.9: Maximum margin hyperplane (solid line) and margins (dashed lines) for an SVM trained on samples from two different classes. Samples on the margin are called support vectors.

The computation of the optimal hyperplane is posed as a quadratic optimaization problem:

$$\begin{aligned} \underset{\mathbf{w},b,\xi}{\text{minimize}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{m}\xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i,\ i = 1,\ldots,m \\ & \xi_k \geq 0,\ i = 1,\ldots,m \end{aligned} \tag{4.27}$$

where the training data, $\mathbf{x}_i$, are mapped to a higher dimensional space via the function $\phi : \mathbb{R}^n \to \mathbb{R}^t,\ t \gg n$. C is a penalty parameter on the training error (classification error) ξ_i and b is a bias term. Relaxing the constraints and moving to a dual formulation of the problem leads to a classifier defined as:

$$f(\mathbf{x}) = \sum_{i=1}^{m}\alpha_i y_i \phi(\mathbf{x}_i)^T\phi(\mathbf{x}) + b \tag{4.28}$$

Instead of working with ϕ it is common to replace it with a simpler kernel function satisfying the condition

$$k(\mathbf{x},\bar{\mathbf{x}}) = \phi(\mathbf{x})^T\phi(\bar{\mathbf{x}}) \tag{4.29}$$

yielding the classifier

$$f(\mathbf{x}) = \sum_{i=1}^{m}\alpha_i y_i k(\mathbf{x}_i,\mathbf{x}) + b \tag{4.30}$$

Many different kernel functions have been used successfully in training SVM:s and among the most common are

- Linear: $k(\mathbf{x},\bar{\mathbf{x}}) = \mathbf{x}^T\bar{\mathbf{x}}$
- Polynomial: $k(\mathbf{x},\bar{\mathbf{x}}) = (\mathbf{x}^T\bar{\mathbf{x}} + 1)^d$
- Gaussian radial basis function: $k(\mathbf{x},\bar{\mathbf{x}}) = \exp(-\gamma \left\|\mathbf{x} - \bar{\mathbf{x}}\right\|^2)$, for $\gamma > 0$

The kernel function used in this project is Gaussian radial basis function since it in many studies have performed as well or better than e.g. linear or polynomial kernels. An illustration of the SVM process is presented in figure 4.10

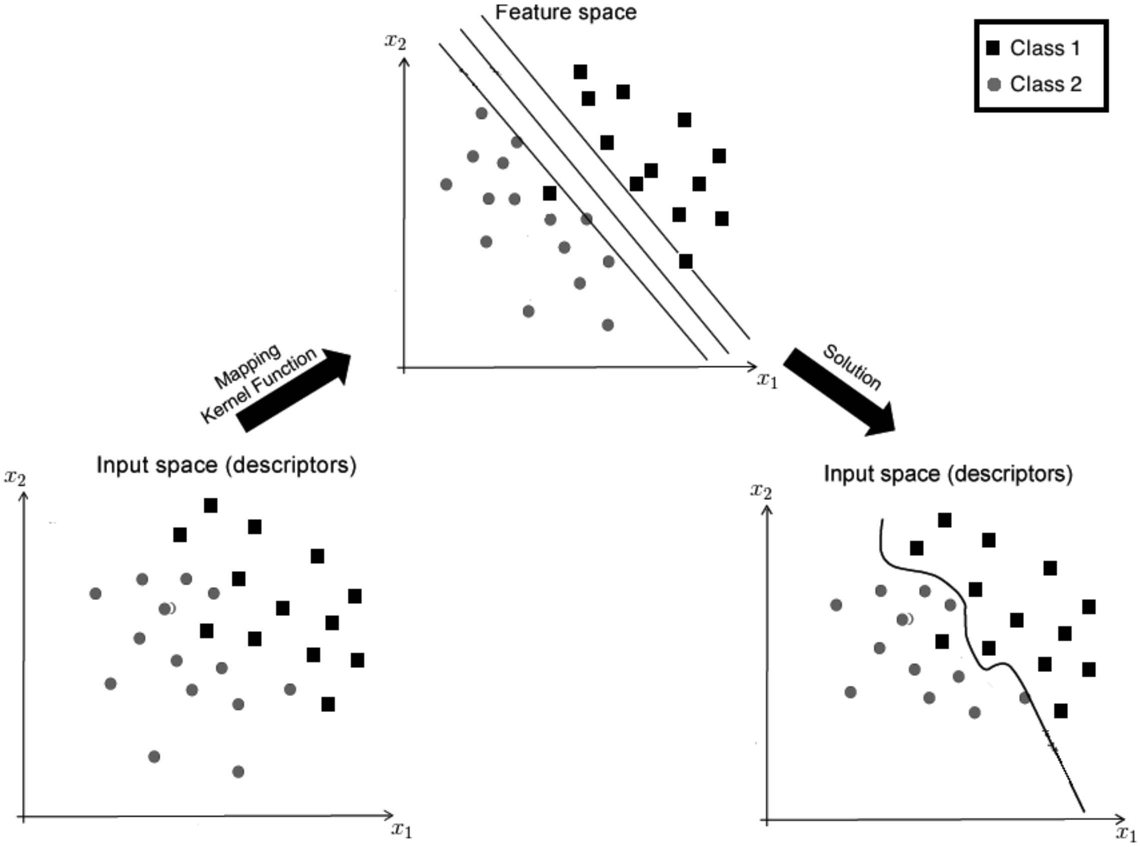


Figure 4.10: Overview of the SVM process for two features, x_1 and x_2, from a sample with two classes. Circles belong to one class and squares belong to another class.

5

Results

The performance of the gender classification framework described in chapter 2 was evaluated on video sequences obtained from a store entrance in an uncontrolled environment. Only people entering the store was considered since this is the main objective for the framework and also since the visibility of faces from people walking past the entrance or out from the store is very limited. Further only a certain area of the video frames are considered in order to reduce the number of false detections, better control of the size of the detected faces and to assure that the detected faces are from people walking towards and through the store entrance. A frame with the detection area can be viewed in figure 5.1.

To evaluate the features described in chapter 4.4 a dataset of facial images was created consisting of images from the different datasets described in chapter 3. Only images of unique individuals was used. This resulted in a dataset with 4270 facial images of unique individuals of which 2002 are females and 2268 are males. These images were then classified with a SVM using a Gaussian radial basis function as the kernel function. The best parameters for the kernel function were obtained by 5-fold cross validation. The classification rate for the facial images in the dataset was estimated by 5-fold cross validation.

The whole dataset with facial images were also used to train a SVM with kernel parameters obtained from the 5-fold cross validation. This classifier are used to classify detected faces in the unconstrained video sequences.

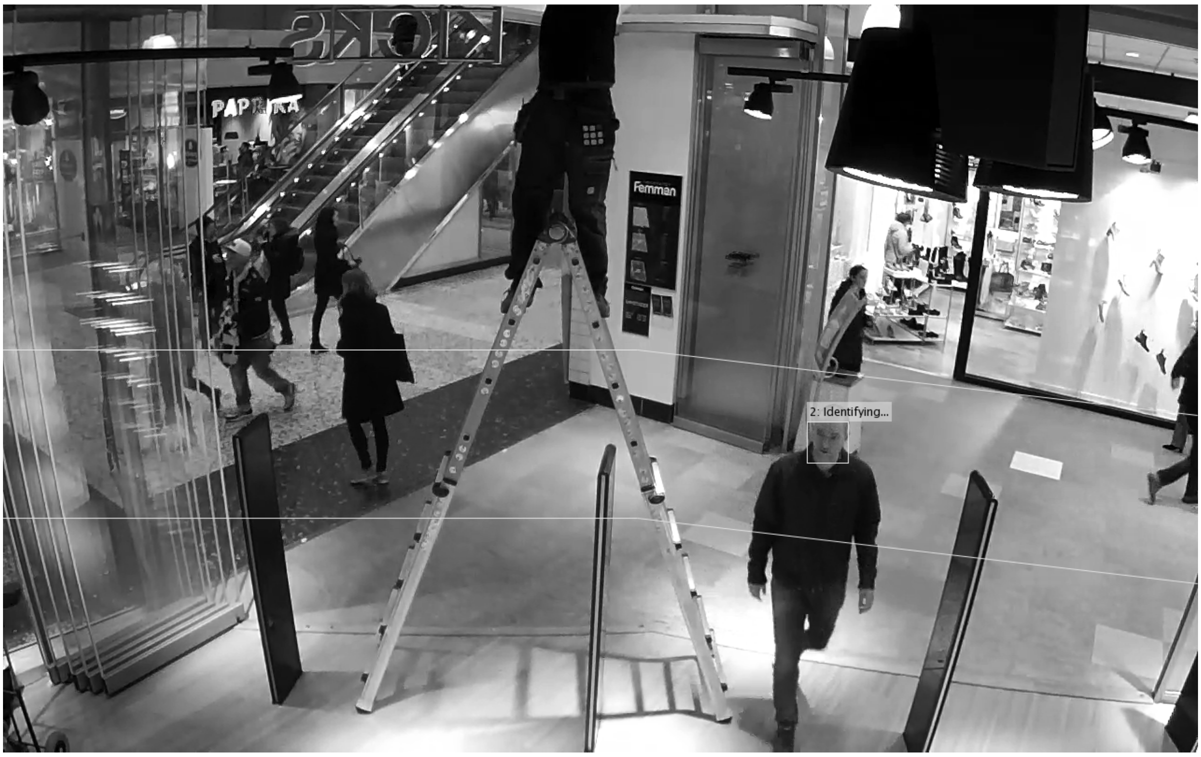
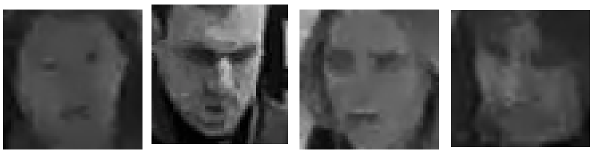


Figure 5.1: Frame from a video sequence with lines representing the detection area.

5.1 Face detection results

The face detection algorithm described in chapter 4.2 performs well on the evaluated video sequences and manages to detect many faces. One problem the face detector has is that it is trained on near frontal view faces and thus fails to detect profile faces or faces where the angle differs to much from the training data. Figure 5.2 shows an example of detected faces from the video sequences later used for classification.



(a) Detected face from frame in video sequence.

(b) Detected face from frame in video sequence.

(c) Detected face from frame in video sequence.

(d) Detected face from frame in video sequence.

Figure 5.2: Example of four detected faces from different frames in video sequence.

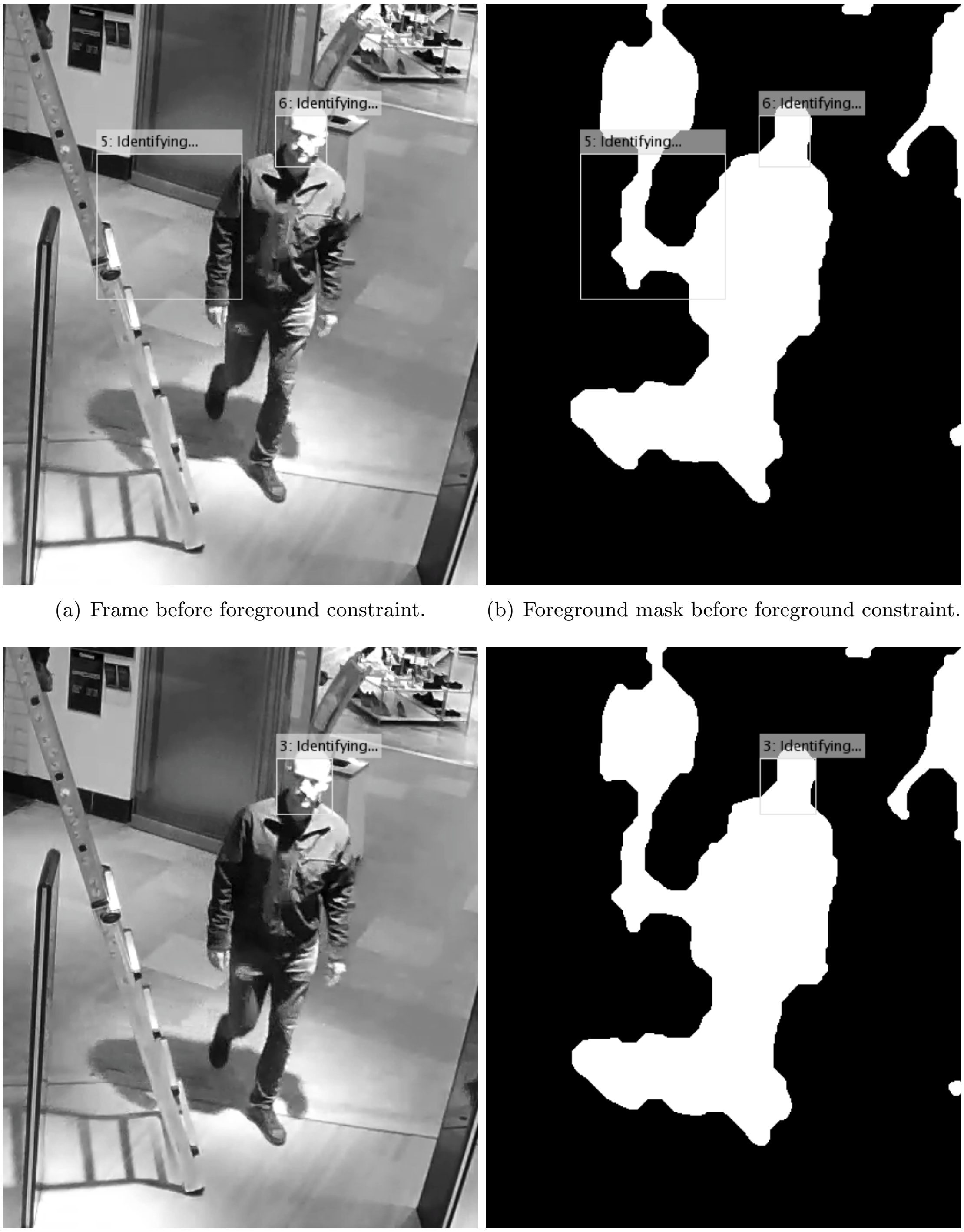
Many of the detected faces in the video sequences are far away and thus very small in size making them unsuitable for feature extraction and classification. Detection of faces on people along with false face detections are also a relatively bad result obtained from the face detection algorithm.

In order to minimise the amount of false face detections, the background model described in chapter 4.1 is applied. Only detected faces where its pixels consists of at least 50% foreground pixels are considered since they are then assumed to belong to a moving object, i.e a person. This fixes most of the false detections but, still some false detections will occur when only considering foreground pixels. Figure 5.3 shows the same frame before and after applying the foreground pixel constraint. It can be seen that one false detection has been removed when adding the constraint that at least 50% of the pixels in the bounding box are foreground pixels.

To address the two other problems (detected faces that are too small and far away and detected faces on people walking past the entrance) a detection area was defined, i.e a polygon representing the entrance. Only people whose faces are visible inside this area are considered. This solves the issue of people being classified that are not walking through the entrance but rather passing by it. Figure 5.1 shows the frame and the proposed detection area.

The main issue with the face detection scheme is that it fails to detect profile faces and angles where only a small part of the face is visible, e.g. a person looking down in the ground. The reason for this is that the face detector used is the one provided in MATLAB which is trained on near frontal faces and thus fails to detect faces that differs too much from the data it was trained on.

The ground truth for the recorded video sequences is 32 males and 132 females walking through the store entrance giving a total of 164 people whose faces should be detected in one or multiple frames inside the detection area. Of the total of 164 individuals 113 are detected and can be classified, i.e 68,9%.



(a) Frame before foreground constraint.

(b) Foreground mask before foreground constraint.

(c) Frame after foreground constraint.

(d) Foreground mask after foreground constraint.

Figure 5.3: Part of frame and its corresponding foreground mask, with bounding boxes showing detected faces, before and after the foreground pixel constraint.

5.2 Tracking results

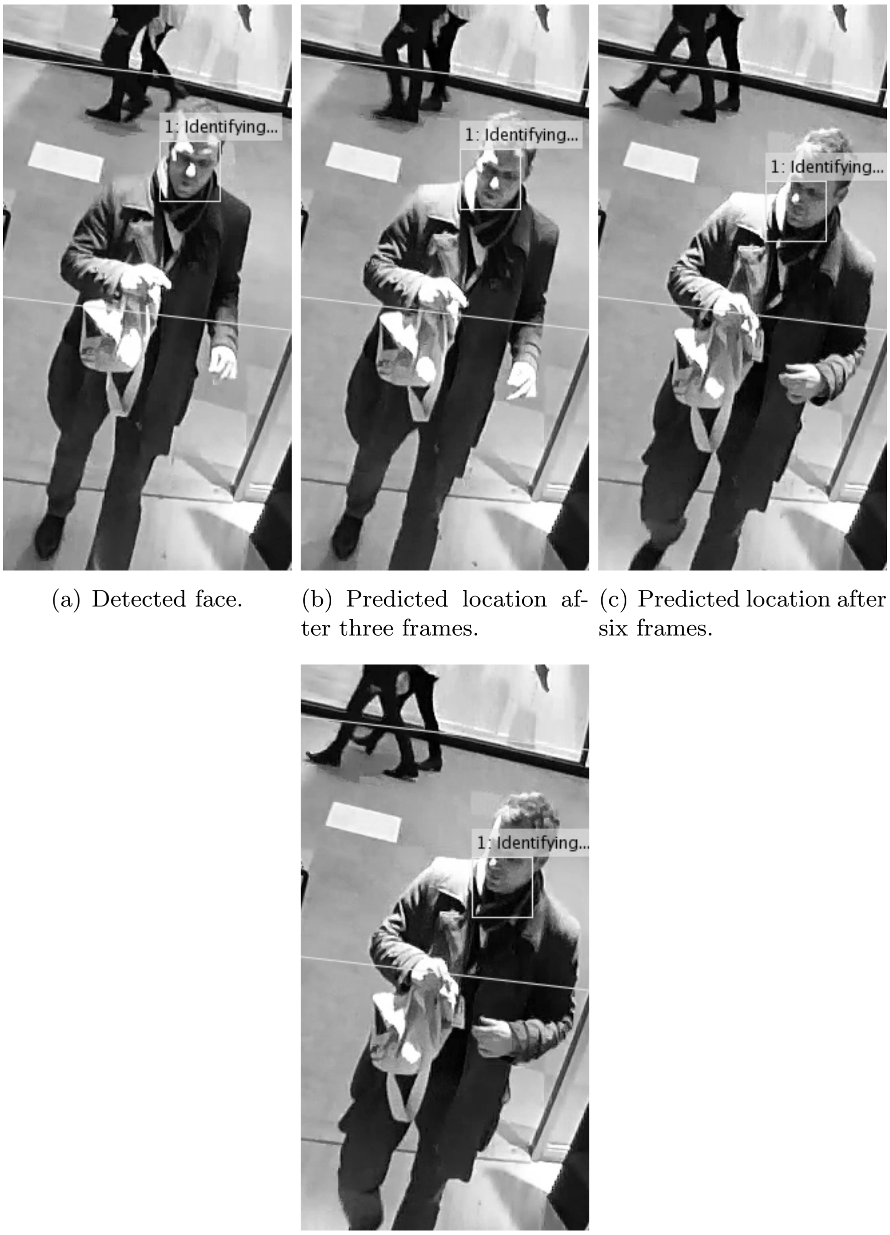
The tracker described in chapter 4.3 tracks the center of the bounding box around a detected face. Each detection is assigned a track that predicts the location of a detected face in consecutive frames. A constant velocity motion model was chosen, which means that the state of an object is described by its position and velocity and that the velocity is a process with independent zero-mean Gaussian increments with constant variance.

An object that the tracker has just started to track is uncertain and requires at least two consecutive detections to be associated to it before it will get a velocity. In order to minimize the amount of noisy detections, i.e. false detections, tracked and at the same time give stable tracks a track is not validated until it has had three consecutive detections associated with it. With less than three detections too many noisy detections was tracked and with more too many objects did not get a track at all.

Tracks are deleted when the center of the bounding box is outside the detection area or if the track have not had any detections assigned to it for ten consecutive frames. The amount of consecutive frames allowed since the last detection was chosen by trial. Tracks that were allowed to continue for more than ten frames since the last detection tended to deviate too much from the actual trajectory to be useful. An example of the performance of the tracker can be seen in figure 5.4.

As can be seen in 5.4 the tracking of a detected face works relatively well. The predicted location of the bounding box surrounding a face is good if it is only one, or very few, frames since the last detection. The more frames since the last detection the worse the prediction gets. For this reason the faces from predicted locations are not used for classification since they might not contain the whole face or contain too much background.

A scenario where the tracking fails is when a new face detection, not belonging to any existing track, gets assigned to an existing track instead of getting assigned a new track. This tends to happen if the new detection lies very close to an existing track that has not had any detections assigned to it for some time. The new detection is then confused as belonging to the old track. This is further visualised in figure 5.5.



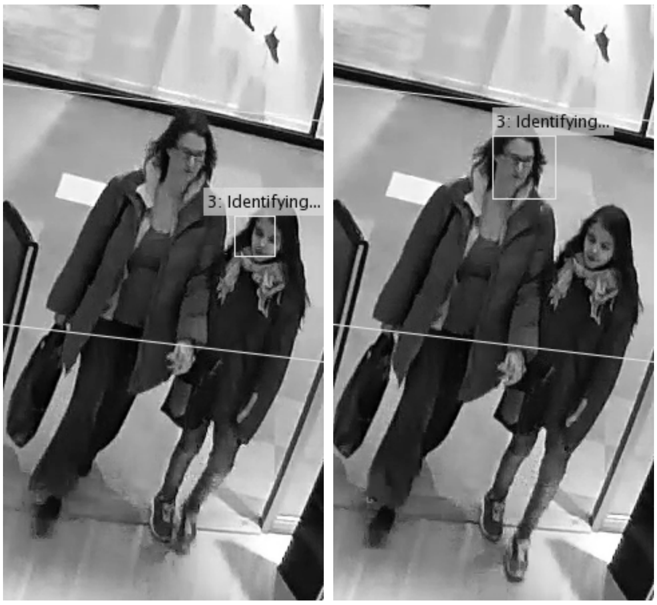
(a) Detected face.

(b) Predicted location after three frames.

(c) Predicted location after six frames.

(d) Predicted location after nine frames.

Figure 5.4: Detected face and its predicted location after 3, 6 and 9 frames.



(a) Predicted location of individuals face.

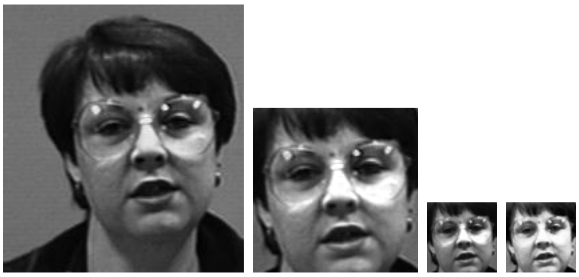
(b) New detection gets assigned to the same existing track.

Figure 5.5: An existing track gets assigned a new detection when instead the detection should get assigned a new track.

5.3 Gender classification

To evaluate the performance of the different features described in chapter 4.4.1 and in appendix A, B and C a dataset consisting of facial images from 4270 unique individuals, of which 2002 are females and 2268 are males, were used (see chapter 3 for more information on this dataset).

Before extracting features the facial images were pre-processed by first cropping them to show only the face region and no background. This cropping is done since the bounding box obtained from the face detection algorithm is limited to only the face. The images were then normalised and resized to the same size. The size of the bounding boxes around the detected faces from the recorded video sequences varies between approximately 26 × 26 pixels and 52 × 52 pixels. The face images were resized to 30 × 30 pixels and 50 × 50 pixels. Lastly the histogram of the pixel intensity values in each image were equalised to even out the contrast. The pre-processing and its result is shown in figure 5.6.



(a) Original image. (b) Cropped image. (c) Image normalized and resized to 50×50 pixels. (d) Image after histogram equalization.

Figure 5.6: Detected face and its predicted location after 3, 6 and 9 frames.

After pre-processing the facial images, features were extracted. Four different methods for feature extraction was evaluated on the still images of faces:

- **GLCM features**: Symmetrical GLCM:s were calculated for each image with $N_g =$ 64 gray levels, four different directions, $\theta = \{0°, 45°, 90°, 135°\}$ and 32 different distances, $d = \{1,...,32\}$, yielding offset parameters $(\Delta_x,\Delta_y) = (0,1),(-1,1),(-1,0),(-1,-1),...,(0,d),(-d,d),(-d,0),(-d,-d)$. 22 feature measures were extracted from each of these GLCM:s. The feature measures were averaged over the different directions, θ, to make them rotational invariant. This gives a feature vector of length $22 \cdot 32$. To obtain a more economical representation of these feature measures they were further averaged over the distances, d, giving a feature vector of length 22. The range of the distance was also calculated and used along with the average over direction and distance giving a total feature vector of length 44.

- **Gabor features**: Each facial image is convolved with Gabor filters of five different scales and eight different orientations. Given the assumption that adjacent pixels are highly correlated, redundant information is removed by downsampling the feature images resulting from the Gabor filters by a factor of four along each dimension. The size of the output feature vector is the size of the image ($N \times M$) multiplied by the number of scales and orientations (5×8) divided by the row and column downsamling factors (4×4). Depending on the size of the image this gives a large output feature vector. In order to reduce the dimension of the feature vector PCA analysis was used (**?**).
- **LBP**: Three different LBP operators were considered for feature extraction: uniform patterns, $\text{LBP}^{U2}_{P,R}$, rotational invariant patterns, $\text{LBP}^{ri}_{P,R}$, and rotational invariant uniform patterns, $\text{LBP}^{riU2}_{P,R}$, with $P = 8$ and $R = 1$.
- **LGBMP**: The LGBMP features are extracted from each facial image as described in chapter 4.4.1 with forty Gabor filters (five scales and eight orientations) and with one of three different LBP operators, $\text{LBP}^{U2}_{8,1}$, $\text{LBP}^{ri}_{8,1}$, $\text{LBP}^{riU2}_{8,1}$ giving three different sets of features denoted $\text{LGBMP}^{U2}_{8,1}$, $\text{LGBMP}^{ri}_{8,1}$, $\text{LGBMP}^{riU2}_{8,1}$. The number of non overlapping rectangle regions each image is divided into varies depending on the image sizes. If the image is of size 30×30 pixels it was divided into 100 3×3 pixel blocks and if it is of size 50×50 pixels it was divided into 100 5×5 pixel blocks.

The facial images were classified with an SVM using 5-fold cross validation. The kernel function for the SVM was a Gaussian radial basis function. The best parameters for the kernel function was also obtained by 5-fold cross validation. The classification rates achieved on the still facial images for different feature extraction methods is shown in table 5.1.

The best result was obtained using LGBMP features. Among them the best classification rate was achieved with images of size 50×50 pixels and with $\text{LGBMP}^{U2}_{8,1}$ features yielding a classification rate of 90.50%. As can be seen in table **??** the classification rate for males are slightly higher than for females, 91.23% for males compared to 89.76% for females.

The performance of the framework was evaluated on video sequences from an uncontrolled environment. For this, six SVM classifiers were trained on all the 4270 facial images in the dataset. These were three variations of the LGBMP features ($\text{LGBMP}^{U2}_{8,1}$, $\text{LGBMP}^{ri}_{8,1}$, $\text{LGBMP}^{riU2}_{8,1}$) for two different image sizes (30×30 and 50×50). These

		Classification accuracy		
Feature	Number of features	32 × 32	52 × 52	Average
Gabor features	34/46	78.64%	85.74%	82.19%
GLCM features	44/44	70.59%	71.01%	70.80%
$\text{LBP}_{8,1}^{U2}$	5900/5900	82.26	83.53	82.90
$\text{LBP}_{8,1}^{ri}$	3600/3600	82.38	83.74	83.06
$\text{LBP}_{8,1}^{riU2}$	1000/1000	82.79	83.65	83.22
$\text{LGBMP}_{8,1}^{U2}$	4000/4000	87.03%	90.54%	88.79%
$\text{LGBMP}_{8,1}^{ri}$	4000/4000	84.99%	88.62%	86.81%
$\text{LGBMP}_{8,1}^{riU2}$	4000/4000	84.94%	88.59%	86.77%

Table 5.1: Classification accuracy using Gabor, GLCM, LBP and LGBMP features on still facial images and SVM classifier with Gaussian radial basis kernel function.

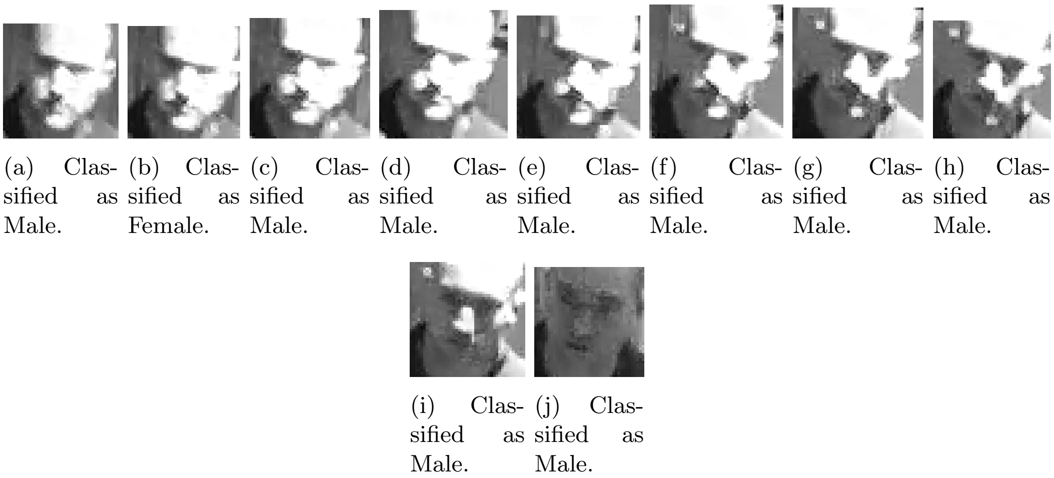
		Predicted class		
		W	M	Total
Actual class	W	1797	205	2002
	M	199	2069	2268
	Total	1996	2274	4270

Table 5.2: Confusion matrix for classification of gender from still face images using $\text{LGBMP}_{8,1}^{U2}$ features and SVM classifier with Gaussian radial basis kernel function. W - Women and M - Men.

SVM:s were used as classifiers for classifying the detected faces in the unconstrained video sequences.

After a face is detected in a video frame and a track is initialized the face image is resized to either 30 × 30 pixels or 50 × 50 pixels depending on the original size of the bounding box surrounding the detected face. It is normalised, its histogram is equalized and LGBMP features are extracted. In each frame where a face is detected it is classified as either male or female and when the track has ended or when the person has gone outside the predefined detection area a final classification result is given by majority voting. If the voting is equal the person is classified as undecided. Figure 5.7 shows ten detected faces from the same individual in ten consecutive frames and the result from the classifier in each frame.

Even though the $\text{LGBMP}_{8,1}^{ri}$ features performed worse than the $\text{LGBMP}_{8,1}^{U2}$ features on the still images it performed better on the video sequences and was thus used for the



(a) Classified as Male. (b) Classified as Female. (c) Classified as Male. (d) Classified as Male. (e) Classified as Male. (f) Classified as Male. (g) Classified as Male. (h) Classified as Male. (i) Classified as Male. (j) Classified as Male.

Figure 5.7: Example of detected faces from the same individual in ten consecutive frames and the result from the classifier in each frame. The individual was classified as male in 9 out of 10 frames.

experiment.

Of the total 164 individuals 113 were detected by the face detection algorithm and could be classified. Of these 113 individuals 23 were males and 90 were females. As can be viewed in the confusion matrix in table 5.3 70 women were correctly classified as women and 20 were misclassified as men giving a misclassification rate of 22.22% for women. For the men 19 were correctly classified and 4 were misclassified as women giving a misclassification rate of 17.39%. The overall misclassification rate for both men and women from the video sequences are 19.81%.

		Predicted class		
		W	M	Total
Actual class	W	70	20	90
	M	4	19	23
	Total	74	39	113

Table 5.3: Confusion matrix for classification of gender from video sequences using LGBMP$^{ri}_{8,1}$ features and SVM classifier with Gaussian radial basis kernel function. W - Women and M - Men.

6

Discussion

The results from the evaluation of the system framework were relatively good (80.19%) considering the data used and considering the fact that it was trained on still face image datasets. The detected faces in the video sequences were very small (between 26 × 26 pixels and 52 × 52 pixels) and with low resolution as shown in figure 5.2. The resolution of the training face images were much higher and the detected faces from the video sequences did not perfectly match the training data face images in resolution or face poses (compare figure 3.1 with figure 5.2).

As mentioned in chapter 2 several papers have been written on gender recognition but hardly any addresses the layout, setup and problems considered in this thesis. Most concentrate on still images of faces, and some claim their work can be extended to video sequences, making it hard to compare the results in this thesis with previous work.

The best results in gender classification from unconstrained video sequences found (Demirkus et al. (2010)) achieves a classification performance of 90%. While this results is better than what is obtained here it is not clear from the article what kind of resolution the video sequences are recorded in or what angle is considered. Resolution and angle of the camera are parameters that plays a vital role in this type of demographic classification and can greatly affect the performance of the system.

While using a majority voting rule for deciding the gender of an individual worked well it is possible that using a framework as proposed in Demirkus et al. (2010) would have been better.

The obtained result was slightly better than that of Shakhnarovich et al. (2002) who achieved an error rate of 21% compared to 20% in this thesis. Again it is hard to compare the results since the data used potentially differs very much in how it was obtained and in the environmental settings.

The results is slightly biased towards males in the classification but not enough to be significant and more data is needed to evaluate this further.

The amount of features used for the classification was quite large (4000). It is possible that by selecting fewer but more reliable features can result in a better performance.

Looking at the result obtained from classifying still images the performance was satisfying. While not as good as has been achieved on still facial images before, it was good enough for the task. The classifier that was used on the video sequences was trained on still face images. Since the detected faces in the video sequences differed some from the training images there is always a problem of the classifier being overtrained on the still images. This might be the reason that the classifier performing best on the video sequences (SVM+$\text{LGBMP}_{8,1}^{ri}$) was not the one that performed best on the still face images.

The face detector performed well in some instances but it was not optimal for detecting faces in the evaluated video sequences since it was trained on frontal faces. A face detector trained on faces under different illumination, viewpoints, scales etc. would have been preferred. Since the placement of the camera gives faces that are . The use of background subtraction and a detection area greatly helped to reduce the number of false detections.

The tracking worked well but suffered from some problems e.g. mismatching of detections (see figure 5.5). Since the face detector failed to detect faces in many instances a track did not last for much more than 20 to 30 frames. It was still long enough to get many useful frames for gender classification.

The system as implemented now processes a frame in approximately 1 frame per 0.5 seconds and is thus a bit away from working in real time. Speeding up the process might be very important in a real application depending on how often the data needs to be accessed. One part of the system that is very computational demanding is the MoG background modelling, which might be unnecessary complex for the setting. It is possible that a less demanding and less complex method can be used with little loss in performance but much gain in computational time.

7

Conclusion and Future work

The task of gender identification on surveillance video in an uncontrolled setting with varying illumination, occlusion, face poses, view angles etc. is a difficult one and the error rate of 19.81% is far from perfect. But achieving 100% is not only near to impossible but also not necessarily needed since the system in a real application can have a goal of giving an estimate of the distribution of males and females entering a store or a mall rather than giving an exact count of them.

During the work on the thesis there were some things that could have been done differently or investigated further with perhaps better accuracy as a result. Below are some ideas that could possibly improve the performance in the future:

- The face detector can be trained on more data including faces that are not only frontal but also a tilted up and down and varying face poses.

- The tracker can be improved by using some other scheme for data assignment and track management, e.g. multiple hypothesis tracking.

- In order to get more reliable classification results the classifier could be trained on facial images taken from surveillance footage or facial images more like the ones used for testing (see figure 5.2) rather than on relatively high resolution still images. Training data consisting on downward tilted faces would also be preferred and faces in more varying poses.

- A method robust and, computation wise, fast method for selecting features for the classification could be investigated.

- Another method than majority voting could be designed for the final decision on gender. Perhaps something similar to that described by Shakhnarovich et al. (2002) and Demirkus et al. (2010).

The above points are some of the ideas conceived while working on the project but lack of time unfortunately meant they were not further investigated.

The gender identification system developed during this project is good and could potentially be used in a real application as a way of estimating the distribution of males and females in for example a store, mall, subway etc.

Bibliography

H. F Alrashed and M. A Berbar. Facial gender recognition using eyes images. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(6), June 2013.

A. L Barker, D. E Brown, and W. N Martin. Bayesian estimation and the kalman filter. *Computers Mathematics with Applications*, 30(10):55–77, November 1995.

Y Benezeth, P. M Jodoin, B Emile, H Laurent, and C Rosenberger. Comparative study of background subtraction algorithms. *Journal of Electronic Imaging*, 19(3), July 2010.

T Bouwmans, F El Baf, and B Vachon. Background modeling using mixture of gaussians for foreground detection - a survey. *Recent Patents on Computer Science*, 1(3):219–237, November 2008.

V. Bruce. Sex discrimination: how do we tell the difference between male and female faces? *Perception*, 22(2):131–152, 1993.

D. A Clausi. An analysis of co-occurrence texture statistics as a function of grey level quantization. *Canadian Journal of Remote Sensing*, 28(1):45–62, February 2002.

Garrison W. Cottrell and Janet Metcalfe. Empath: Face, emotion, and gender recognition using holons. In *Proceedings of the 1990 Conference on Advances in Neural Information Processing Systems 3*, pages 564–571, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc.

J. G Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America A, Optics and image science*, 2(7):1160–1169, July 1985.

M Demirkus, M Toews, J. J Clark, and T Arbel. Gender classification from unconstrained video sequences. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 55–62, 2010.

S Englund. Realtime face detection. diploma thesis, Chalmers University of Technology, March 2003.

Ruogu Fang, Kevin D. Tang, Noah Snavely, and Tsuhan Chen. Towards computational models of kinship verification. In *IEEE Conference on Image Processing (ICIP)*, Hong Kong, China, September 2010. URL `http://chenlab.ece.cornell.edu/projects/KinshipVerification/`.

Y Freund and R. E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, August 1997.

D Gabor. Theory of communication. *Journal of the Institution of Electrical Engineers - Part III: Radio and Communication engineering*, 93(26):429–457, November 2006.

Mohammad Haghighat, Saman Zonouz, and Mohamed Abdel-Mottaleb. Identification using encrypted biometrics. In Richard Wilson, Edwin Hancock, Adrian Bors, and William Smith, editors, *Computer Analysis of Images and Patterns*, volume 8048 of *Lecture Notes in Computer Science*, pages 440–448. Springer Berlin Heidelberg, 2013.

R. M Haralick, K Shanmugam, and I Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6):610–621, November 1973.

Di Huang, Caifeng Shan, Moshen Ardabilian, Yunhong Wang, and Liming Chen. Local binary patterns and its application to facial image analysis: A survey. *IEEE Transactions on Systems, Man, and Cybernetics?Part C: Applications and Reviews*, 41(6), November 2011.

Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007. URL `http://vis-www.cs.umass.edu/lfw/`.

A Jain, Yi Chen, and M Demirkus. Pores and ridges: Fingerprint matching using level 3 features. In *18th International Conference on Pattern Recognition*, volume 4, pages 477–480, 2006.

R. E Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME?Journal of Basic Engineering*, 82:35–45, March 1960.

J-K Kamarainen, V Kyrki, and H Kälviäinen. Simple gabor feature space for invariant object recognition. *Pattern Recognition Letters*, 25(3):311–318, February 2004.

A Lapedriza, M. J Maryn-Jimenez, and J Vitria. Gender recognition in non controlled environments. In *18th International Conference on Pattern Recognition*, pages 834–837, 2006.

Bing Li, Xiao-Chen Lian, and Bao-Liang Lu. Gender classification by combining clothing, hair and facial component classifiers. *Neurocomputing*, 76(1):18–27, January 2012.

Chengjun Liu and Harry Wechsler. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 11(4):467–476, 2002.

E Makinen and R Raisamo. Evaluation of gender classification methods with automatically detected and aligned faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):541–547, March 2008.

A. M Martinez and R Benavente. The ar face database, cvc technical report. Technical Report 24, The Ohio State University, 1998. URL `http://www2.ece.ohio-state.edu/~aleix/ARdatabase.html`.

K Messer, J Kittler, M Sadeghi, and M Hamouz. Face authentication test on the banca database. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 4, pages 523–532, August 2002.

B. Moghaddam and Ming-Hsuan Yang. Learning gender with support faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):707–711, 2002a.

B Moghaddam and Ming-Hsuan Yang. Learning gender with support faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):707–711, May 2002b.

J Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, March 1957.

K. P Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning.* Phd thesis, Berkeley University of California, 2002.

Choon B. Ng, Yong H. Tay, and Bok M. Goi. Vision-based human gender recognition: A survey. 2012. URL `www.summon.com`.

T Ojala, M Pietikäninen, and T Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7), July 2002.

H. A Patel and D. G Thakore. Moving object tracking using kalman filter. *International Journal of Computer Science and Mobile Computing*, 2(4):326–332, April 2013.

S. K Patel and A Mishra. Moving object tracking techniques: A critical review. *Indian Journal of Computer Science and Engineering*, 4(2):95–102, April 2013.

P. J Phillips, H Wechsler, J Huang, and P. J Rauss. The feret database and evaluation procedure for face recognition algorithms. *Image and Vision Computing Journal*, 16 (5):295–306, April 1998. URL `http://www.nist.gov/itl/iad/ig/colorferet.cfm`.

P. J Phillips, H Wechsler, J Huang, and P. J Rauss. The feret evaluation methodology for face recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, October 2000. URL `http://www.nist.gov/itl/iad/ig/colorferet.cfm`.

M Piccardi. Background subtraction techniques: a review. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 4, pages 3099–3104, October 2004.

G Shakhnarovich, P Viola, and B Moghaddam. A unified learning framework for real time face detection and classification. In *Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 14–21, 2002.

Caifeng Shan. Learning local binary patterns for gender classification on real-world face images. *Pattern Recognition Letters*, 33(4), March 2012.

Caifeng Shan, Shaogang Gong, and Peter McOwan. Fusing gait and face cues for human gender recognition. *Neurocomputing*, 71(10):1931–1938, 2008.

LinLin Shen, Li Bai, and Michael Fairhurst. Gabor wavelets and general discriminant analysis for face identification and verification. *Image and Vision Computing*, 25(5): 553–563, 2007.

L Soh and C Tsatsoulis. Texture analysis of sar sea ice imagery using gray level co-occurrence matrices. *IEEE Transactions on Geoscience and Remote Sensing*, 37(2): 780–795, March 1999.

L Spacek. Face recognition data, university of essex, uk, CVC Technical Report 2008. URL `http://cswww.essex.ac.uk/mv/allfaces/index.html`.

C Stauffer and W. E. L Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 246–252, June 1999.

C Stauffer and W. E. L Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, August 2000.

L. R Sudha and R Bhavani. Gait based gender identification using statistical pattern classifiers. *International Journal of Computer Applications*, 40(8), February 2012.

M Toews and T Arbel. Detection, localization, and sex classification of faces from arbitrary viewpoints and under occlusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1567–1581, September 2009.

H. D Vankayalapati, R. S Vaddi, L. N. P Boggavarapu, and K. R Anne. Extraction of facial features for the real-time human gender classification. In *International Conference on Emerging Trends in Electrical and Computer Technology*, pages 752–757, 2011.

J Villysson. Robust tracking of dynamic objects in lidar point clouds. Masters thesis, Chalmers University of Technology, February 2014.

P Viola and M. J Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, May 2004.

M Watanabe and K Yamaguchi. *The EM algorithm and realted statistical models*. Marcel Dekker, 2004.

Bin Xia, Bin Xia, He Sun, He Sun, Bao-Liang Lu, and Bao-Liang Lu. Multi-view gender classification based on local gabor binary mapping pattern and support vector machines. In *IEEE International Joint Conference on Neural Networks*, volume 10, pages 3388–3395, 2008.

M. H Yang, D. J Kriegman, and N Ahuja. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34–58, January 2002.

Shiqi Yu, Tieniu Tan, Kaiqi Huang, Kui Jia, and Xinyu Wu. A study on gait-based gender classification. *IEEE transactions on image processing*, 18(8):1905–1910, August 2009.

Wenchao Zhang, Shiguang Shan, Wen Gao, Xilin Chen, and Hongming Zhang. Local gabor binary pattern histogram sequence (lgbphs): a novel non-statistical model for face representation and recognition. In *Tenth IEEE International Conference on Computer Vision*, volume 1, pages 786–791, 2005.

A

Gray Level Co-occurence Matrix (GLCM)

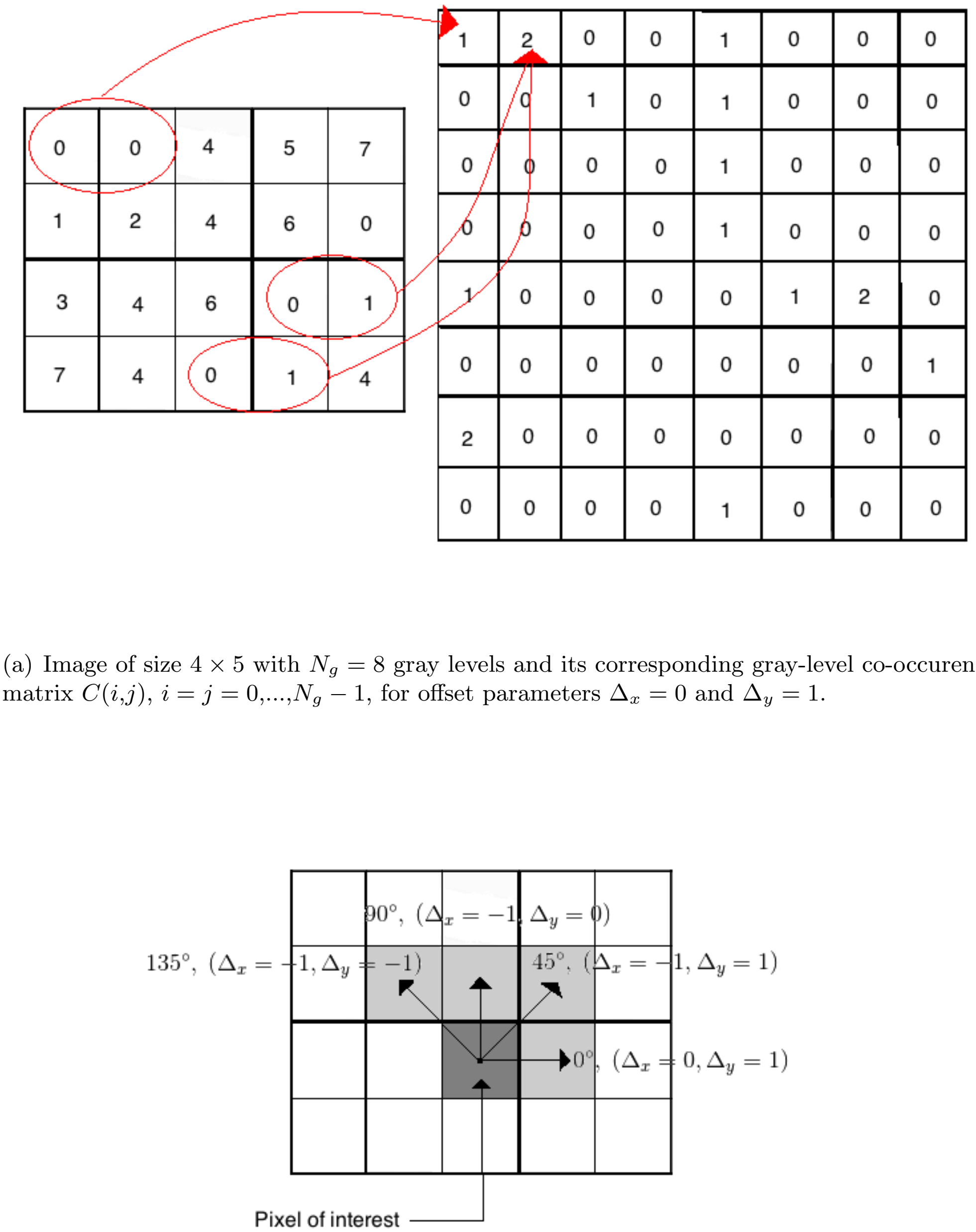
Suppose a gray valued image is of size $N_x \times N_y$ with N_g gray levels. The GLCM, $C(i,j,\Delta_x,\Delta_y)$ (written short as $C(i,j)$), give how frequent pixel intensity value i co-occurs with pixel intensity value j. The GLCM of an image I of size $N_x \times N_y$ is thus given by

$$C(i,j) = \sum_{x=1}^{N_x} \sum_{y=1}^{N_y} \mathbf{1}(I(x,y) = i \text{ and } I(x+\Delta_x, y+\Delta_y) = j) \tag{A.1}$$

where i and j are the pixel intensity values of image I ranging from $0,...,N_g - 1$, x and y are the spatial positions in image I. Δ_x and Δ_y the offset parameters denoting a geometrical relationship between image pixels which depends on the direction, θ, and the distance, d, at which the matrix is computed and $\mathbf{1}$ the indicator function defined as

$$\mathbf{1}(p) = \begin{cases} 1 & \text{if } p \text{ is true,} \\ 0 & \text{if } p \text{ is not true.} \end{cases} \tag{A.2}$$

For an 8-bit gray level image (with $N_g = 256$) the size of the GLCM is 256×256. It can often be convenient to reduce the dimension of the matrix by reducing the number of gray levels, N_g. For example the number of gray levels can be reduced to $N_g = 8$. This gives a GLCM of size 8×8 which is computationally more convenient. Figure A.1 shows an example of a GLCM along with an illustration of the meaning of the offset parameters Δ_x and Δ_y.



0	0	4	5	7
1	2	4	6	0
3	4	6	0	1
7	4	0	1	4

1	2	0	0	1	0	0	0
0	0	1	0	1	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	0
1	0	0	0	0	1	2	0
0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0

(a) Image of size 4 × 5 with $N_g = 8$ gray levels and its corresponding gray-level co-occurence matrix $C(i,j)$, $i = j = 0,\ldots,N_g - 1$, for offset parameters $\Delta_x = 0$ and $\Delta_y = 1$.

$90°,\ (\Delta_x = -1, \Delta_y = 0)$
$135°,\ (\Delta_x = -1, \Delta_y = -1)$
$45°,\ (\Delta_x = -1, \Delta_y = 1)$
$0°,\ (\Delta_x = 0, \Delta_y = 1)$
Pixel of interest

(b) Illustration of offset parameters Δ_x and Δ_y.

Figure A.1: Illustration of an image and its corresponding gray-level co-occurence matrix along with an illustration of the effect of the offset parameters.

Haralick et al. (1973) proposed that the matrix should be symmetrical, i.e. both (0,1) and (1,0) pairings are counted when calculating the number of times the value 0 is adjacent to the value 1. Often a number of features containing useful information about the texture of an image are extracted from the GLCM which are then used to form a feature vector. Some of these were proposed by Haralick et al. (1973) and some by Soh and Tsatsoulis (1999).

The direction and distance (given by the offset parameters) in a GLCM are important when computing the features. Soh and Tsatsoulis (1999) compared different implementations of the GLCM, for example feature measures from a symmetrical GLCM for four different directions, $\theta = \{0°, 45°, 90°, 135°\}$ are averaged in order to make them rotational invariant. These feature measures are then further averaged over a range of displacement values in order to obtain a reliable and economical representation.

22 features that can be extracted from a GLCM is described next (Haralick et al. (1973); Soh and Tsatsoulis (1999); Clausi (2002)). First some useful notations and equations are described.

Let $C_x(i)$ and $C_y(j)$ be the ith and jth entry respectively in the marginal probability matrix obtained by

$$C_x(i) = \sum_{j=1}^{N_g} C(i,j) \tag{A.3}$$

and

$$C_y(i) = \sum_{i=1}^{N_g} C(i,j) \tag{A.4}$$

Further let

$$C_{x+y}(k) = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} C(i,j), \quad i+j=k, \, k = 2,3,...,2N \tag{A.5}$$

and

$$C_{x-y}(k) = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} C(i,j), \quad |i-j|=k, \, k = 0,1,...,N-1 \tag{A.6}$$

Let μ_x and μ_y denote the mean of the rows and columns of $C(i,j)$ given by equation (A.7) and (A.8). Further let σ_x and σ_y denote the standard deviation of the rows and columns of $C(i,j)$ given by equation (A.9) and (A.10).

$$\mu_x = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} iC(i,j) \tag{A.7}$$

$$\mu_y = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} jC(i,j) \tag{A.8}$$

$$\sigma_x = \sqrt{\sum_{i=1}^{N_g}\sum_{j=1}^{N_g}(i-\mu_x)^2 C(i,j)} \tag{A.9}$$

$$\sigma_y = \sqrt{\sum_{i=1}^{N_g}\sum_{j=1}^{N_g}(j-\mu_y)^2 C(i,j)} \tag{A.10}$$

With the help of the equations above a set of features can be extracted from a GLCM as follows:

$$\text{Autocorrelation} = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} ijC(i,j) \tag{A.11}$$

$$\text{Contrast} = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g}(i-j)^2 C(i,j) \tag{A.12}$$

$$\text{Correlation1} = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g}\frac{(i-\mu_x)(j-\mu_y)C(i,j)}{\sigma_x\sigma_y} \tag{A.13}$$

$$\text{Correlation 2} = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g}\frac{(ij)C(i,j)-\mu_x\mu_y}{\sigma_x\sigma_y} \tag{A.14}$$

$$\text{Cluster Prominence} = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g}(i-\mu_x+j-\mu_y)^4 C(i,j) \tag{A.15}$$

$$\text{Cluster Shade} = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g}(i-\mu_x+j-\mu_y)^3 C(i,j) \tag{A.16}$$

$$\text{Dissimilarity} = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g}\left|i-j\right| C(i,j) \tag{A.17}$$

$$\text{Energy} = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} C(i,j)^2 \tag{A.18}$$

$$\text{Entropy} = -\sum_{i=1}^{N_g}\sum_{j=1}^{N_g} C(i,j)\ln(C(i,j)) \tag{A.19}$$

$$\text{Homogeneity } 1 = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} \frac{C(i,j)}{1+|i-j|} \tag{A.20}$$

$$\text{Homogeneity } 2 = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} \frac{C(i,j)}{1+(i-j)^2} \tag{A.21}$$

$$\text{Maximum Probability} = \max_{i,j} C(i,j) \tag{A.22}$$

$$\text{Sum of Squares: Variance} = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g}(i-\mu)^2 C(i,j) \tag{A.23}$$

where μ is the mean of $C(i,j)$

$$\text{Sum Average} = \sum_{k=2}^{2N_g} k C_{x+y}(k) \tag{A.24}$$

$$\text{Sum Variance} = \sum_{k=2}^{2N_g}(k - \sum_{k=2}^{2N_g} k C_{x+y}(k))^2 C_{x+y}(k) \tag{A.25}$$

$$\text{Sum Entropy} = -\sum_{k=2}^{2N_g} C_{x+y}(k)\ln(C_{x+y}(k)) \tag{A.26}$$

$$\text{Difference Variance} = \sum_{k=0}^{N_g-1}(k - \sum_{l=0}^{N_g-1} l C_{x-y}(l))^2 C_{x-y}(k) \tag{A.27}$$

$$\text{Difference Entropy} = \sum_{k=0}^{N_g-1} C_{x-y}(k)\ln(C_{x-y}(k)) \tag{A.28}$$

$$\text{Information Measure of Correlation } 1 = \frac{HXY - HXY1}{\max(HX,HY)} \tag{A.29}$$

$$\text{Information Measure of Correlation } 2 = \tag{A.30}$$

$$= \sqrt{1-\exp(-2(HXY2 - HXY))} \tag{A.31}$$

where HXY is the entropy of $C(i,j)$ given by equation (A.19) and HX and HY is the entropy of C_x and C_y respectively:

$$HX = -\sum_{i=1}^{N_g} C_x(i)\ln(C_x(i)) \tag{A.32}$$

$$HY = -\sum_{j=1}^{N_g} C_y(j)\ln(C_y(j)) \tag{A.33}$$

$HXY1$ and $HXY2$ is given by:

$$HXY1 = -\sum_{i=1}^{N_g}\sum_{j=1}^{N_g} C(i,j)\ln(C_x(i)C_y(j)) \tag{A.34}$$

$$HXY2 = -\sum_{i=1}^{N_g}\sum_{j=1}^{N_g} C_x(i)C_y(j)\ln(C_x(i)C_y(j)) \tag{A.35}$$

$$\text{Inverse Difference Normalized} = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} \frac{C(i,j)}{1+\frac{|i-j|^2}{N_g^2}} \tag{A.36}$$

$$\text{Inverse Difference Moment Normalized} = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} \frac{C(i,j)}{1+\frac{(i-j)^2}{N_g^2}} \tag{A.37}$$

B

Gabor features

Features based on Gabor filters can be extracted by convolving an image with a 2D-Gabor filter function. The filter can be described as an elliptical Gaussian multiplied by a complex plane wave (a 2-D Fourier basis function). The filter in the 2-D spatial domain is defined as (Haghighat et al. (2013)):

$$\begin{aligned} \psi(x,y;f,\theta) &= \frac{f^2}{\pi\gamma\eta} e^{-(\frac{x'+\gamma^2 y'^2}{2\sigma^2})} e^{j2\pi f x' + \phi} \\ x' &= x\cos\theta + y\sin\theta \\ y' &= -x\sin\theta + y\cos\theta \end{aligned} \tag{B.1}$$

where f denotes the tuning frequency of the filter (the frequency of a sinusoidal plane wave), γ and η are parameters controlling the bandwidth corresponding to the two perpendicular axes of the Gaussian, i.e. the spatial widths of the filter. θ denotes the rotation angle of both the Gaussian and the plane wave. ϕ is the phase offset and σ^2 is the variance of the Gaussian envelope.

Note that the Gabor filter in equation (B.1) is not the general form of the 2-D Gabor filter but rather a form, devised by Daugman (1985) from Gabor (2006) original 1-D function, whose properties are the most useful in image analysis.

Gabor features are constructed from the response of convolving a Gabor filter $\psi(x,y;f,\theta)$ with an image $I(x,y)$, i.e.

$$G(x,y;f,\theta) = \psi(x,y;f,\theta) * I(x,y) \tag{B.2}$$

The response is calculated for multiple frequencies f_m and orientations θ_n.

The frequency corresponds to the scale and Kamarainen et al. (2004) proposed that they should be exponential defined as

$$f_m = k^{-m} f_{\max},\ m = \{0,...,M-1\} \tag{B.3}$$

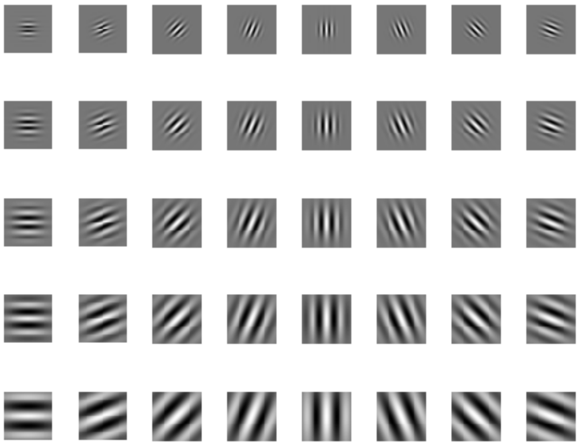
where f_m is the m:th frequency, M is the number of scales to be used, $f = f_{\max}$ is the highest frequency desired and $k > 1,\ k \in \mathbb{R}$ is the frequency scaling factor. The different orientations are given by

$$\theta_n = \frac{n2\pi}{N}, n = \{0,...,N-1\} \tag{B.4}$$

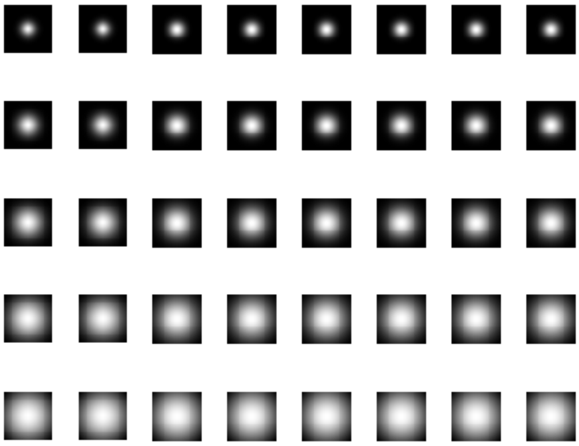
where θ_n is the n:th orientation and N is the number of orientations to be used. Figure B.1 shows the real part and the magnitude of the Gabor filter at five different scales and 8 different orientations.

Given the fact that adjacent pixels in an image are highly correlated redundant information can be removed by downsampling the feature images resulting from the Gabor filters (Shen et al. (2007); Liu and Wechsler (2002)). The output feature vector given from applying Gabor filter on an image is the size of the image multiplied by the number of scales and orientations divided by the row and column downsampling factors. Depending on how many different scales and orientations of the Gabor filter that are used the output feature vector can be very large even after downsampling. Therefore a dimensionality reduction method (e.g. general discriminant analysis (GDA), principal component analysis (PCA), GDA) is often applied afterwards to reduce the size of the vector. These can for example be general discriminant analysis (GDA), principal component analysis (PCA) or linear discriminant analysis (LDA).

An example of the response from applying a Gabor filter in five different scales and eight different orientations on a facial image is illustrated in figure B.2.



(a) Real part of the Gabor filter for $M = 5$ scales and $N = 8$ orientations.



(b) Magnitude of the Gabor filter for $M = 5$ scales and $N = 8$ orientations.

Figure B.1: Real part and magnitude part of the Gabor filter at five different scales, f_m, $m = \{0,...,4\}$, and eight different orientations, θ_n, $n = \{0,...,7\}$.

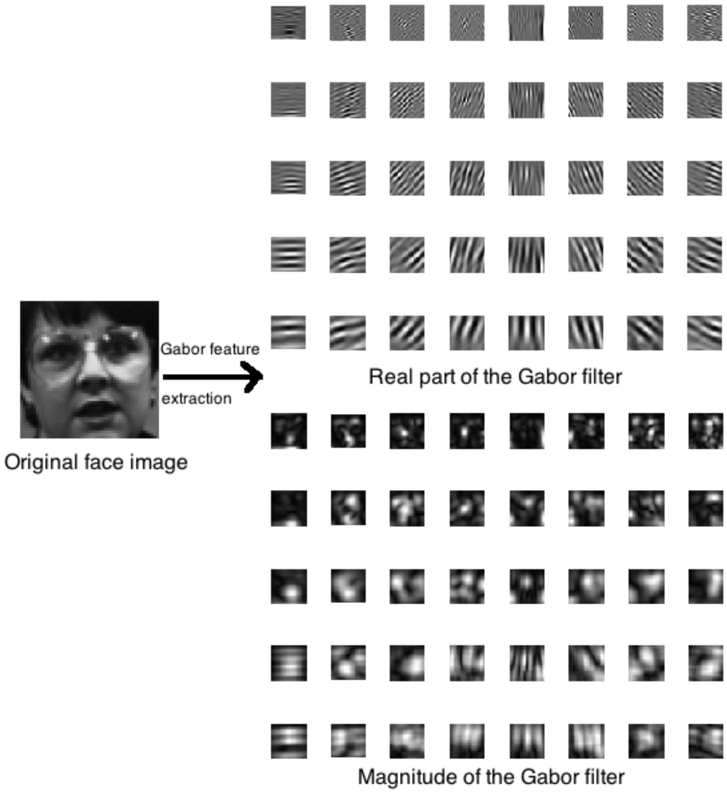


Figure B.2: Response from applying the Gabor filter at five different scales, f_m, $m = \{0,...,4\}$, and eight different orientations, θ_n, $n = \{0,...,7\}$ on a facial image.

C

Local Binary Patterns

The original LBP operator labels each pixel in an image with a decimal number called LBP codes which encode the local structure around each pixel. It proceeds as shown in figure C.1. Each pixel is compared with its neighbours in a 3 × 3 neighbourhood by subtracting the center pixel value. All negative values are encoded with 0 and all positive with 1 resulting in a binary number obtained by concatenating all these values in a clockwise direction. This is done for all pixels in the image. The original LBP operator can be extended to neighbourhoods of different sizes in order to deal with texture at different scales as illustrated in figure C.2. Formally, given a pixel (x_c,y_c), the LBP can be expressed as

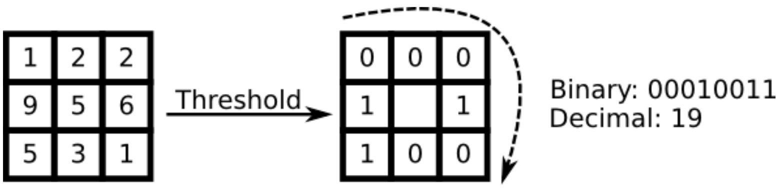
$$\mathrm{LBP}_{P,R}(x_c,y_c) = \sum_{n=0}^{P-1} s(i_n - i_c)2^P \tag{C.1}$$

where i_c is the gray level value of the central pixel and i_n, $n = 0,...,P-1$ is the gray level values of the P surrounding pixels in a circular neighbourhood with radius R. The function $s(x)$ is defined as

$$s(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \tag{C.2}$$

The operator $\mathrm{LBP}_{P,R}$ produces 2^P different binary patterns formed by P pixels in the neighbourhood, i.e giving 2^P different labels. The histogram of LBP labels calculated over a region is often used as a texture descriptor (feature).

Many variations of the LBP operator has been proposed and an extensive survey of LBP methodology and its application to facial image analysis is given by Huang et al. (2011). One commonly used variation of the LBP operator is to only use a subset of the 2^P binary patterns to describe the texture of an image. Ojala et al. (2002)



1	2	2
9	5	6
5	3	1

0	0	0
1		1
1	0	0

Figure C.1: Example of the basic LBP operator.

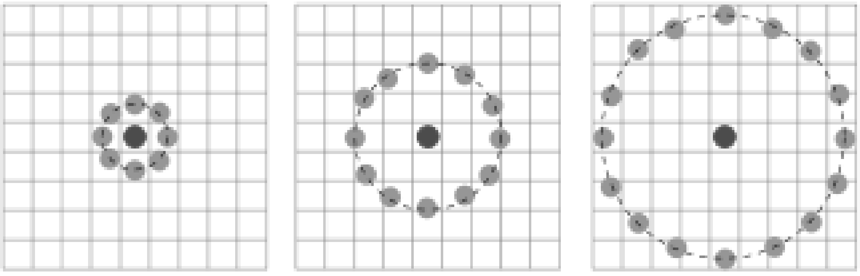


Figure C.2: Example of the LBP operator with different circular neighbourhoods. From left to right: $R = 1$ and $P = 8$ (LBP$_{8,1}$), $R = 2$ and $P = 16$ (LBP$_{16,2}$), $R = 3$ and $P = 24$ (LBP$_{24,3}$).

called these patterns uniform patterns denoted LBP$^{U2}_{P,R}$. An LBP is called uniform if it contains at most two bitwise transitions from 0 to 1 or vice versa when the corresponding bit string is considered circular. For example 00000000 (0 transitions) and 01110000 (2 transitions) are uniform patterns while 11001001 (4 transitions) and 01010011 (6 transitions) are not considered uniform. It has been observed that uniform patterns account for approximately 90% of all the patterns in a (8,1) neighbourhood (Ojala et al. (2002)). As an example the number of labels given from a (8,1) neighbourhood is 256 for the standard LBP operator but only 58 if only the uniform patterns are used. Usually all the non-uniform patterns are accumulated into a single bin giving that the number of labels for an LBP^{U2} operator is 59 (58 labels for the uniform patterns and one for the remaining patterns).

Other variations include rotation invariant LBP operator LBP$^{ri}_{P,R}$, Elongated LBP, rotation invariant uniform LBP operator LBP$^{riU2}_{P,R}$ etc.