Person Detection in the Restaurant of the Future

Roeland A. van Batenburg
s1339273
batenb@ai.rug.nl

July 2010

Master Thesis
Artificial Intelligence
University of Groningen, The Netherlands

Internal supervisor:
Marco Wiering (Artificial Intelligence, University of Groningen)

External supervisors:
Nicole Koenderink (Agrotechnology & Food Science Group, Wageningen Research Center)
Mari Wigham (Agrotechnology & Food Science Group, Wageningen Research Center)

university of groningen / faculty of mathematics and natural sciences / artificial intelligence

FOOD & BIOBASED RESEARCH
WAGENINGEN UR

Abstract

Many computer vision techniques dealing with humans require that the location within an image of a person is known. In this research, we investigated into the subject of person detection in a real-world environment: the Restaurant of the Future. We evaluated several methods and selected one: person detection using face detection. In order to apply this we evaluated several known methods for face detection. Then we determined which aspects of the recordings from the Restaurant impacted the performance of the face detectors. All methods suffered strongly from the complex background in the Restaurant and the low quality of the recordings. The angled viewpoint had an impact on several of the methods while the video encoding used affected others. On the worst conditions, the Viola-Jones detector performed best. The analysis of the weaknesses of the face detectors will give future researchers a starting point in improving known or finding new methods.

Acknowledgements

Many people have contributed to me finishing my master project. First of all, I would like to thank my supervisor Marco Wiering of the department of AI at the University of Groningen, whose advise set me on the right track several times. I also want to thank Nicole Koenderink and Mari Wigham, my supervisors at Intelligent Systems, Wageningen UR, for their patience at our weekly meetings. Secondly, my thanks go to my colleagues at Intelligent Systems for making me feel welcome and part of the team. Special thanks go to Remco Gerlich, who made my internship at Intelligent Systems possible. Last but not least, I want to thank my parents Ellen Visser and Pim van Batenburg and my girlfriend Ingrid Wolfslag for supporting me along the way.

Table of Contents

Chapter 1

Introduction

1.1 The Restaurant of the Future

The Restaurant of the Future[1] is a facility founded to study consumer preferences and habits. It is a research facility of the department of Consumer Sciece of Wageningen UR Food & Biobased Research. The Restaurant consists of two parts, a company restaurant and a sensory consumer research lab. The catering company Sodexo[2] serves lunch to the employees of the Wageningen University and Research centre, provided they register themselves at the checkout. In the research lab the scientists from Consumer Science can set up specific conditions to evaluate their impact on the consumer experience. For an impression of the Restaurant see Figure 1.1[3] and 1.2, a plan of the company restaurant can be found in Figure 1.3.

(e)

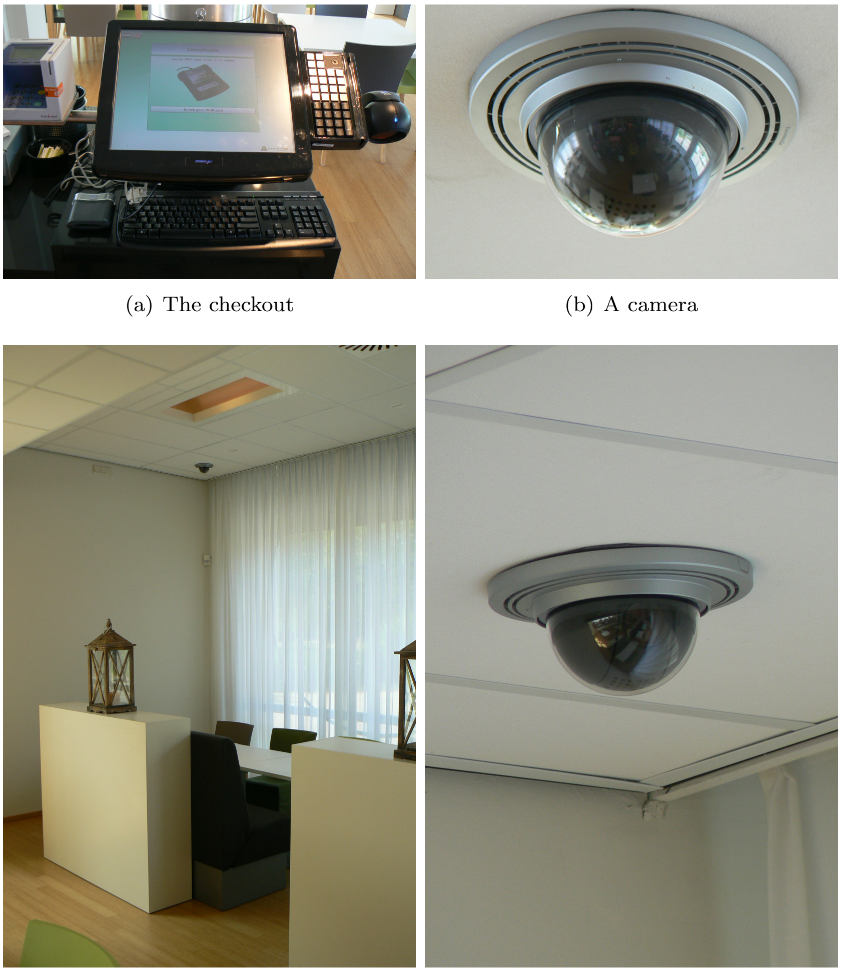
Figure 1.1: Some pictures from the Restaurant

With the Restaurant the Consumer Science researchers can study the visitors without influencing them while they no longer have to rely on verbal reports

[1] http://www.restaurantvandetoekomst.wur.nl/UK/

[2] http://www.sodexo.com

[3] http://www.restaurantvandetoekomst.wur.nl/NL/Media+en+Links/Fotogalerij/Het+Restaurant+in+bedrijf/



(a) The checkout

(b) A camera

(c) Camera on the ceiling

(d) Zoom on the camera

Figure 1.2: More pictures from the Restaurant

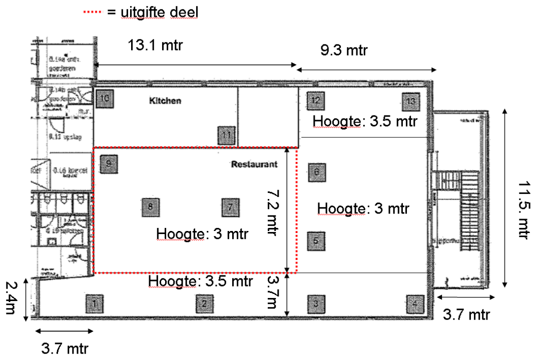


Figure 1.3: Plan of the Restaurant

[1]. Behavioural scientists usually have to rely on interviews, questionnaires or observing their subjects and because these techniques interfere with the experience of the subject the results are not always representative. In the Restaurant the consumers will find the products in a realistic situation. Due to the fact that the visitors are mostly regulars, they are no longer influenced by the fact that they are observed. Moreover, the Restaurant can be easily adapted for certain conditions, the lighting can be changed in intensity and colour and the furniture can easily be moved (see Figure 1.1(e)).

Several types of information of the visitors are stored: cameras record their behaviour, the checkout stores their choices and a hidden scale collects the weight of the visitor at the checkout. At registration, the visitors provide some information about themselves, such as age and gender and with this information the researchers can search for patterns in the checkout information. Additional information such as the weather and the weight of the visitors can make for more detailed food consumption patterns. However, there is more information that can be studied, such as the total visiting time in the Restaurant or the number of bites taken. To collect the information required, there are cameras mounted on the ceiling.

1.2 Recordings in the Restaurant of the Future

Due to the regular visitors of the Restaurant it is possible to collect large sets of recordings of eating behaviour, this brings us to the problem of organising this data. The researchers are often interested in a very specific behaviour or type of person, for instance people eating soup or visitors over 40. To easily find recordings that match the researchers' interests, the recordings need to be annotated. These annotations consist of marking who is visible and what this person is doing. Currently the researchers are working with the Observer XT[4] software. Observer XT was developed by Noldus[5] to annotate observations of many types, including video recordings. Using a pre-defined set of keys, the user can quickly annotate what is happening in the recordings.

A screenshot of an example of an annotation can be found in Figure 1.4. On top stills of four cameras can be seen, below are the annotation, each person has one row, containing the annotations relevant for this person. In this example, two things were annotated for each person, the initial view (red) and the visibility of the person (blue). The initial view is triggered when the person is clearly visible at the counter and defines whether the person is on camera 1 (the left most image) or camera 2 (the second image). The blue lines define whether a person is visible on one of the recordings and also which one (1, 2, 3 or 4).

There are several disadvantages to solely using the Observer software to organise the camera data: it is very time consuming, inconsistent and it is difficult to add new annotations to the system. Because every action needs to be annotated manually, the annotation can not be done in real-time. It takes longer as there are often multiple actors in the view. If an action has not been defined before the researcher starts annotating, it will not be included in the annotation and adding a new action later requires that all the recordings are processed again. Lastly, there is the problem of inconsistent annotations, for

[4] http://www.noldus.com/human-behavior-research/products/the-observer-xt

[5] http://www.noldus.com/

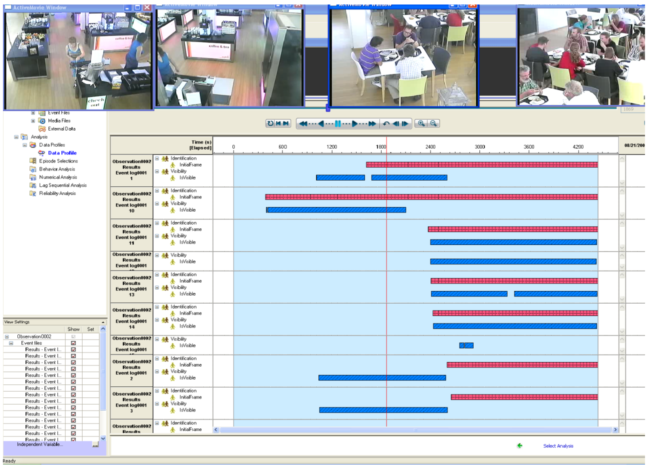


Figure 1.4: Example of a simple annotation of a set of recordings with Observer XT

example one annotator might classify eating a bit of soup as *drinking* while the other classifies it as *taking a bite*.

As a result of these disadvantages the researchers only collect data in very specific conditions and then annotate this data for their own purposes. As they will only annotate the behaviour that they are interested in, their data is probably not suitable for another research. This means that only a small part of the potentially available data is used. In the next section we will present a solution to this problem.

1.3 Automated Annotation of the Recordings

In this section we will describe an automated annotation system for the Restaurant, the advantages of such a system and then some background for each of the components. Finally we will describe some of the difficulties that are specific for the environment of the Restaurant. An automated annotation system would consist of three parts: person detection, person recognition and action recognition. The person detector would determine the location of a person in an image from the recording (as in Figure 1.5). This information can then be used to recognise each person, this can be combined with the information about a person that was stored at registration, such as the name, age or previous purchases. Lastly the action recognition system determines what the detected person is doing and annotates this.

Automating the annotation solves all of the disadvantages of the manual annotation mentioned at the end of the previous section. It is one system, so as long as the action recogniser works well the resulting annotation will be consistent. The system can analyse the recordings as many times as required and new interesting behaviours can be easily added later and then the whole set can be annotated again without requiring retraining. Furthermore, a system like this can be linked easily to other sources of information such as weather information, while the manual annotation system would require additional actions for each

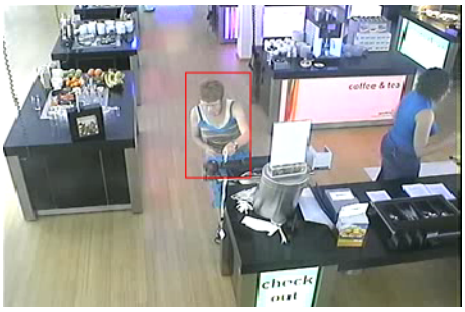


Figure 1.5: Example of person detection on one of the recordings from the Restaurant

recording.

Successful person recognition is also very interesting in the Restaurant, as this can automatically add a lot of information about the visitors. For example, the total visiting time can be determined by comparing the first and last time that a person is recognised. Or the time that a person spends sitting at a table by looking at the longest time a person is visible in one camera. Determining this is very difficult for a person as he or she would have to go back and forth to see when a person arrives or leaves.

The field of action recognition has gained more interest in recent years, partly caused by the increasing amount of security cameras. An advanced security system that does more than simple intrusion detection is Cassandra [2], this system detects aggression based on camera recordings and sound. More advanced action recognition systems can distinguish between multiple behaviours, examples are Liu *et al.* [3] and Luo *et al.* [4]. Currently Microsoft is developing a game system called Kinect[6] that aims to use a camera to allow the player to interact with the game by letting the character in the game follow the player's movement.

Similarly, person recognition is becoming more popular, besides recognising behaviour it is also often interesting to know who is doing the behaviour. In security systems [5] access can be granted or denied based on the identity of a person, or in large groups it can quickly find a certain person. When a system wants to interact with a person such as in intelligent environment [6], it needs to both see what the person is doing and, for a customised environment, recognise what he or she is doing.

Almost all methods in both the field of person recognition and the field of action recognition assume that the location of the target person within the view is known. Without this knowledge there is simply too much data to analyse, as faces can be found everywhere and at different scales as seen in Figure 1.6. Furthermore, action recognition usually uses a sequence of images from one or more persons, without information about what part belongs to whom. This makes it difficult to construct a correct model of the action. Consequently, a system must be in place that detects possible persons which gives the two recognisers a starting point.

Before the methods for person and action recognition can be applied the

[6] http://www.xbox.com/en-US/kinect

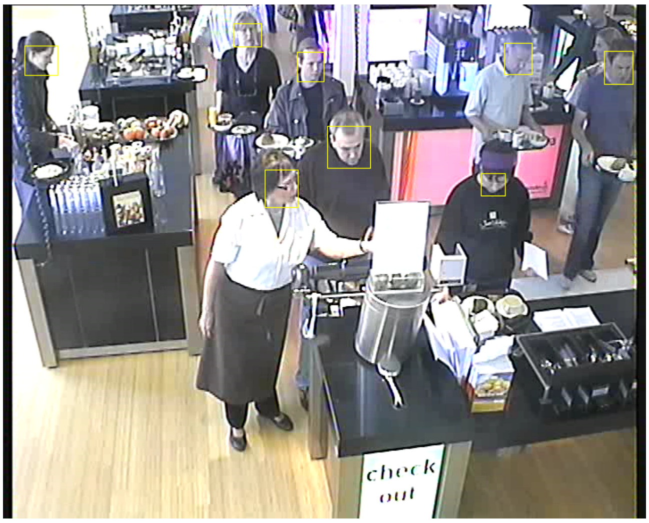


Figure 1.6: There are many faces in this image and at different scales, recognising faces at all positions and scales would be too time consuming

location of the person within the view needs to be found, therefore we look at person detection in the Restaurant in this research. Although person detection is required for most techniques in face recognition and action recognition it is often ignored. When applying person detection the researcher might only use samples which contain only the face or for action recognition the subjects may be required to stay in the centre of the screen.

The Restaurant is a rather difficult environment for automated image analysis, as the visitors are free to move through the Restaurant, the large windows greatly influence the light intensity and the cameras record at a rather low quality. The low quality of the recordings means that a person consists of a low number of pixels, in some cases the person might even start to lose some of his or her shape due to the lack of pixels. The changes in light intensity (See Figure 1.7) makes it hard to find fixed patterns unless the images are normalised in which case some information is lost. Beside changes in intensity the light also changes in colour due to the counters in the Restaurant (See Figure 1.8). Lastly, the unguided movement of the visitors results in many different positions and orientations of their bodies; finding a model that fits all of these might be difficult.

1.4 This Research

In this research we want to answer the question if automated person detection in the Restaurant is possible. We restrict ourselves to simply finding a person on the image without using information outside the current frame. First, we have to determine which method is most qualified for the environment of the Restaurant (Chapter 2). We do this by first presenting known methods of person detection and discussing some of the properties of the recordings in the Restaurant which influence the choice. Finally, we present the best match between methods and

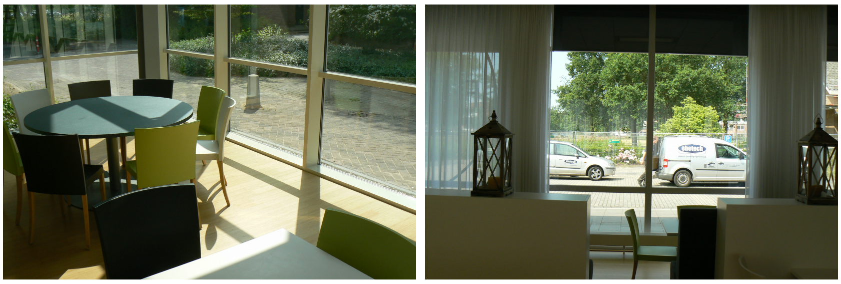


Figure 1.7: The Restaurant has large windows which causes strong fluctuations in the light intensity.

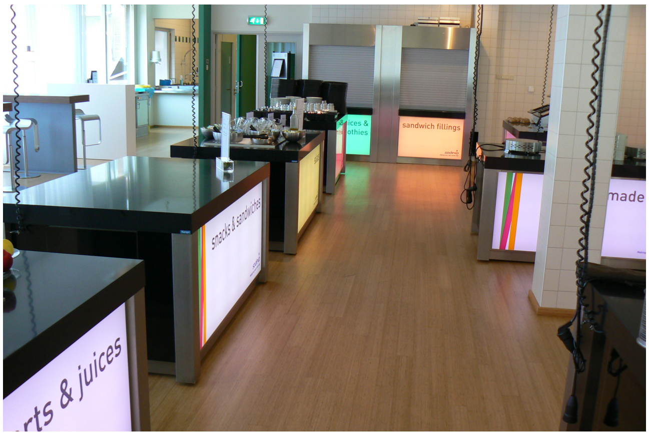


Figure 1.8: The food counter in the Restaurant can change in colour, which affects the perceived colours of the visitors as well.

features and discuss the selected method in more detail in section 2.2.

Then we examine in further detail on how this method works on the recordings. Training the automated person detectors takes a large amount of data, fortunately pre-trained detectors are available. However, they are trained on different data, in machine learning it is not always possible to generalise beyond the data [7]. To check if training on data from the Restaurant improved the detection we trained one method and in section 2.3 we present how the training works. To measure the effectiveness of the different methods, we use an evaluation method from the field of information retrieval, it is presented in section 2.4.

With these tools we can evaluate and train the chosen method, however this requires data. The collected data is presented in Chapter 3. With this data we can answer how the methods perform on the recordings from the Restaurant. Then we have done an additional experiment to see what properties of the Restaurant influences the performance. This will allow us to suggest how to improve the performance. The experiments and the results can be found in Chapter 4, the results are discussed in Chapter 5 where we conclude by answering the research questions from this section.

The research questions we just presented are:

- What is the best method for person detection with respect to the recordings from the Restaurant?
- How does this method perform on the recordings?
- What properties of the recordings influence the performance?
- Does a specialised detector outperform more general detectors?

Chapter 2

Methods

This chapter consists of three parts, in the first section (2.1), we will introduce several techniques for person detection, with this information we will pick one method based on the properties of the recordings from the Restaurant. We will elaborate on this technique in the next section, 2.2. As we are interested in the difference between pre-trained and newly trained detectors we will discuss the training method for one of the detectors in section 2.3. In the last section, 2.4, we will present common techniques used to evaluate detectors.

2.1 Active Person Detection

The field of person detection can be divided into two categories. One category focusses on single images, the second uses multiple images to construct a 3D model of the objects in the view. Making a 3D model puts some restraints on the recordings, they need to overlap as 3D requires at least two images of an object [8]. Furthermore, to obtain true 3D reconstruction the cameras need to be calibrated, after which they cannot be moved. The low number of cameras in the Restaurant and the fact that these cameras were designed to move around according to the researcher's wishes, makes 3D reconstruction a bad choice for the setup of the Restaurant.

Therefore, we looked at a single images approach. Person detection on single images can be divided into two subdomains. The first uses more information than just the current view, the other actively searches for person-like structures in the image. In the next paragraphs we will show some examples of the first approach and discuss the merits.

Examples of information that can be used to detect people are motion or background subtraction. This information is not always available or informative. Movement in this situation is not the best indication for a person, as people sit rather quiet when they are seated and eating. For an example, see Figure 2.1

One popular technique for object detection in computer vision is background subtraction [9]. Background subtraction usually compares a modelled background with the current view and then marks pixels that are somehow similar (for instance same RGB values) as background and the rest as foreground. With some noise removal techniques (such as dilation and erosion) this can lead to good results in simple environments. However, as mentioned the Restaurant

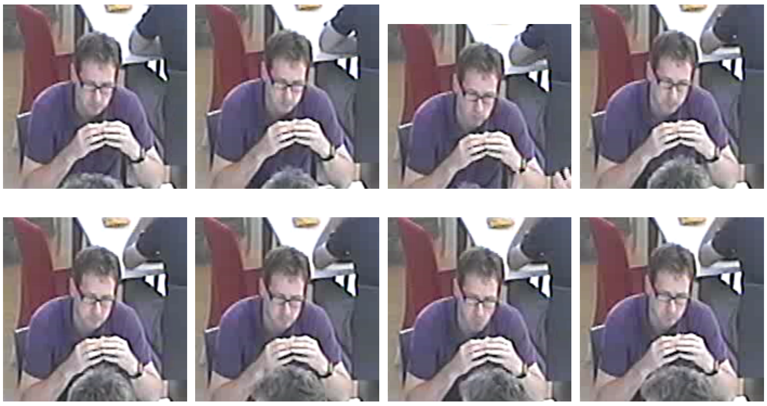


Figure 2.1: A sequence of a visitor eating in the Restaurant

has large windows, this means that the light conditions vary a lot during even one set of recording and more when comparing bright days with cloudy or rainy days. Furthermore, tables and chairs are moved around (on a different time scale), which also changes the background.

A dynamic background model [10, 11] updates the model according to the stability of an object and could be used to correct for these background changes. When a pixel stays the same for a given amount of time it is considered background and the model is updated accordingly. However, as mentioned people sometimes move very little in the Restaurant and this might lead to unwanted behaviour of the algorithm. It could cause the person to be absorbed into the background model. It is possible to adjust the update rule to wait longer before updating the background, but then the model would not be able to include the changes in light conditions and the moved furniture quickly. As the speed at which the light changes and the furniture is moved is on average higher than the movement of the visitors seated, it will be difficult to find a good balance that does not result in a lot of false positives.

Another approach to detection is the use of movement to detect intruders in a security system [12]. As we already discussed, movement is not a good indication in the Restaurant and we could find no other information source that said anything about the presence of a visitor. Therefore, we decided to look at static images and actively search in them for regions of pixels that match a person.

This approach is usually called active person detection. This is a trivial task for human beings who develop this skill very early in their lives. For computers, person detection is still a difficult task. In the rest of this section we introduce several popular, person detection methods and discuss their advantages and disadvantages, especially with respect to the environment of the Restaurant.

2.1.1 Skin-based Detection

A comprehensive overview of skin detection techniques is written by Vezhnevets *et al.* [13]. There are several approaches possible, almost all convert the RGB input images to a particular plane and then construct or learn some model of the skin distribution. The best results have been achieved with simply thresholding the I axis in the YIQ plane [14, 15], YIQ is a colour space introduced with the NTSC [16]. Because it was designed for television it uses very little bandwidth for chromatic information that the human eye cannot process [17]. This means that there is more detail in the colour region that the human eye can observe, which is useful because human sight evolved to easily perceive complex, natural scenes [18].

The Y in the YIQ space indicates the intensity. I encodes the chromatic information along a blue-green to orange scale, while Q encodes the chromatic information along a yellow-green to magenta vector. This is reflected in the YIQ to RGB conversion:

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1.00 & 0.95 & 0.62 \\ 1.00 & -0.28 & -0.64 \\ 1.00 & -1.11 & 1.73 \end{pmatrix} \begin{pmatrix} Y \\ I \\ Q \end{pmatrix}$$

The inverse of the matrix can be used to convert RGB to YIQ. The person detection method by Wang *et al.* [14] converts the normalised RGB values to the YIQ space and then only accepts pixels with values of I within a defined region. The I channel represents the *orangeness* of an image and Caucasian faces have a somewhat orange tint, this is why the I thresholding method works so well.

A more recent method has achieved higher recognition rates with lower false positives, compared with the YIQ method, with very little labelled data [19]. They trained Bayesian networks on patches of 3x3 pixels using three methods to create the network, Naive Bayes, Tree-Augmented Naive Bayes and Stochastic Structure Search. The 3x3 patch was translated to chromatic RG space, this space contains the same information as the RGB space, except for the intensity information. Sometimes it is called normalised RGB space. The normalisation is calculated using these formula's: $r = \dfrac{R}{R+G+B}$, $g = \dfrac{G}{R+G+B}$. Because of the normalisation we know that $b = 1-r-g$, and b is no longer necessary and the space is invariant to intensity. This method constructs a far more complex model of the colour it is detecting, compared to the I thresholding. In this case it can reject more non-skin as it has more information in the decision process.

Both methods obtained high detection rates with low false positives. On the Compaq database [20], the YIQ technique achieved 94.7% true positives with 30.2% false positives. The Bayesian network obtained up to 99.4% true positives with only 10% false positives. Skin is however, not the best indication of a person in the Restaurant, while selecting food they often wear their coats and hide most of their skin. And when they are seated they frequently sit with their back toward the cameras. The light from outside and the light emitted by the buffets affects the colours in the Restaurant and changes the hue of the skin. Lastly, some of the colours in the Restaurant background, such as some buffets and the floor, closely match skin tone, this could cause difficulties.

2.1.2 Shape-based Detection

A popular method in active object detection is using Scale Invariant Feature Transform (SIFT) features [21]. As their name indicates these features are scale invariant as well as invariant to orientation and somewhat robust against affine transformations. Aside from a large number of object recognition applications, it has been applied to face recognition [22] and an extended variant has been applied to person detection [23].

When detecting an object, SIFT features are calculated from an image. Each of these features is matched against the features from the training object and identical points are stored as matches. These matches are clustered over the pose they predict for the object. Finally, the clusters are ranked according to the likelihood that they are the result from the presence of the object.

SIFT features could be very useful as they are invariant to a lot of factors that occur in the Restaurant, people being far and near the cameras (scale), light changes (light invariant) and people standing at a different angle (orientation changes). Although object detection in SIFT is very robust against affine transformations, it is unable to cope with non-affine transformations. Unfortunately, this is exactly the kind of transformations that the people in the Restaurant display, they are bending over, taking a seat and moving their arms around. During these actions the body bends and changes shape, these are not affine transformations and will break the SIFT detection. Alternatives to SIFT are SURF [24] or GLOH [25], but none of these methods can adequately model the pose changes of the visitors of the Restaurant.

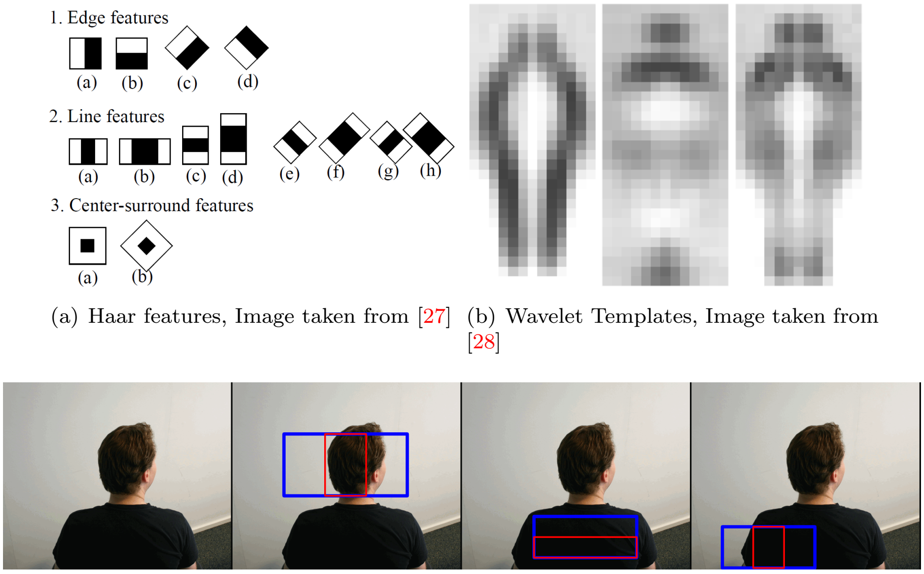
Lastly, these features are meant to recognise specific objects, in order to train a detector an additional generalisation step is required. This additional classifier would learn a model of feature points and their inner relations to model a human body. This generalisation from a specific person detector to a general person detector is tricky as you need a large set of different visitors to be sure that the detector is sufficiently generalised.

2.1.3 Appearance-based Detection

Appearance-based techniques construct a model of the target object using intensity information, examples include the Viola-Jones detector [26], which was used for person detection by Kruppa *et al.* [27]. Other examples are the Wavelet Templates from Oren *et al.* [28] and the binary models from Broggi *et al.* [29]. Broggi *et al.* construct a very simple model to detect head and shoulder. The Viola-Jones detector uses Haar-features [30] (Figure 2.2(a) & 2.2(c)) as features to learn a model from examples. The wavelet templates expand on the Haar-features to construct models with more detail [28], see Figure 2.2(b).

The Viola-Jones detector defines a model combining multiple Haar-features. Haar features compare the light intensity in the regions they cover. For instance the second image in Figure 2.2(c) looks at the space left and right of the head with respect to the intensity of the head. When you overlap the Haar-features in the model they form an outline.

The methods mentioned here are much faster compared to the previous subsection. They are robust against intensity changes as they look at the intensity difference between regions. This makes them well suited to deal with the changing lighting conditions in the Restaurant. However, they are even less invariant to pose changes as they do not model the relations between key points but try to model the entire outline. In the data from the Restaurant the outline of the visitors changes too much and multiple models would be required to model all the poses. This is not only difficult because a huge set of data would be required, it is also hard to determine the number of poses that are required.



(a) Haar features, Image taken from [27]

(b) Wavelet Templates, Image taken from [28]

(c) Haar features matched to the person they detected

Figure 2.2: Tools for two different appearance-based techniques

2.1.4 Face-based Detection

The previous two subsections discussed detectors that modelled the body, however the first approach just used information that was only present in case of a person, the skin. Another feature that is only present on a person is a face. The advancements in the field of face detection could be used in person detection as the face is in many situations indicative of where the body accompanying the face is. In the case of the Restaurant the body is almost always directly below the face. The face is far more static, the contours and configuration hardly changes. A person can move his arms or twist the upper body, such changes are rare in the human face and it is therefore easier to model. Some example methods for face detection have been mentioned above, Viola-Jones [26], skin-detection [31] and SIFT [32].

Although the model is simpler, this method will only work for some images, as it requires the face to be visible. In the Restaurant, visitors often take a seat with their back to the camera or at an angle. There visitors would not be detected with this method.

2.1.5 Overview of the Person Detection Methods

We can summarise the advantages and requirements of the approaches discussed in this section into a table to better compare the options. See Table 2.1. This is a rough evaluation, for example both face and skin detection have difficulties with occlusion, but a face is occluded easier than the skin.

All of the methods are somewhat invariant to scale and angle, which is important as both change often in the recordings of the Restaurant. Handling multiple people is also a requirement that all mentioned methods meet. Because

	Appearance	Background	Face	Skin	Shape
Robust to:					
Pose	-	+	-	+	-
Scale	+	+	+	+	+
Angle	+	+	+	+	+
Light	+	-	+	+	+
Occlusion	-	*	-	-	-
Multiple people	+	+	+	+	+
Over/underexposed	-	-	-	-	-
Requires:					
Additional classifier	-	+	-	-	+
Specific feature visible	-	-	+	+	-

Table 2.1: Overview of the approaches and the properties of the recordings.
+: is robust to/requires/can handle
-: is not robust to/does not require/cannot handle
*: depends on classifier

the background modelling is unable to handle the light changes combined with the movement of furniture it was rejected. This leaves the choice between two methods that are invariant to pose but require a specific feature (skin or face) to be visible and two methods that do not require such a feature but are less robust against pose changes.

Constructing a shape or appearance based model might be possible, but with the additional limits that the quality of the recordings sets, it will be very difficult. When we investigated into the successes that Kruppa *et al.* had with a Viola-Jones detector on person detection, it appears that the detector was only evaluated on images far simpler than those of the Restaurant, see Figure 2.3[1] & 2.4. Although the quality of the images from Kruppa *et al.* seems a bit lower their pose is not very varied, they are either facing toward or away from the camera, while standing straight. This gives them a much clearer profile to learn, compared with the images from the Restaurant in which no single profile will match each case.

Due to the difficulties that shape and appearance-based methods have we have limited our research to the recordings at the checkout, in which the visitors almost always show their face to the camera. Also on this data the visitors show many different poses which would make it difficult to construct a single model. Therefore, we did not work with appearance or shape-based methods. The decision between skin-based and face-based detection was rather easy, the changes in colour from the counters would make skin detection difficult and the location and size of a face gives us more information than a region of skin does, as we do not know whether skin resulted from the neck or arm. In the next section we will discuss some methods for face detection, which will serve as a base for the face-based person detector.

[1] http://alereimondo.no-ip.org/OpenCV/35.version?id=2

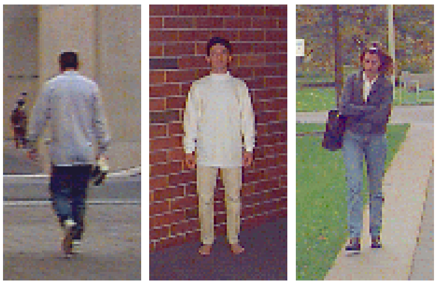


Figure 2.3: Images used for the upper body detector in OpenCV[27]

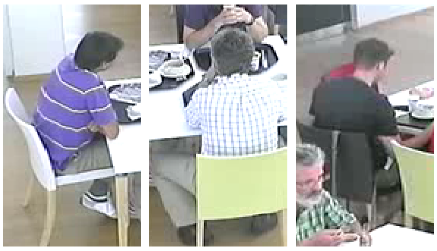


Figure 2.4: Images from the Restaurant, showing different poses of the visitors

2.2 Face Detection

In the field of face detection there are three types of approaches: knowledge based, template based and appearance based [33]. Knowledge based approaches are based on human knowledge and are constructed manually and might for instance have a rule about the relation between an eye and the nose. Template based methods match an image with a stored set of face patterns. Finally, appearance based approaches use a set of training images to learn a model representative for the set.

The knowledge and template based approaches have shown that it is difficult to construct a proper set of rules and the methods usually require detection of some features as eyes or mouth, which is a non-trivial problem. Furthermore, these methods are hard to extend to a larger set of faces, while the learning method would require only additional training samples. The methods with the best recognition rates have used an appearance based method and this research investigated only such methods. As such we only describe these methods here.

All of the methods discussed here have been tested on the MIT-CMU dataset [34][2] in Tsao *et al.* [**?**]. Figure 2.5 displays some images from the set. They have also been tested on the BioID[3] dataset by Tsao *et al.* [**?**]. The results can be seen in Figure 2.6.

[2] http://vasc.ri.cmu.edu/idb/html/face/frontal_images/index.html

[3] http://www.bioid.com

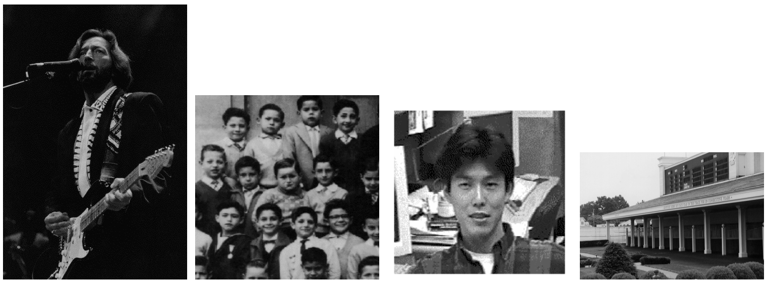
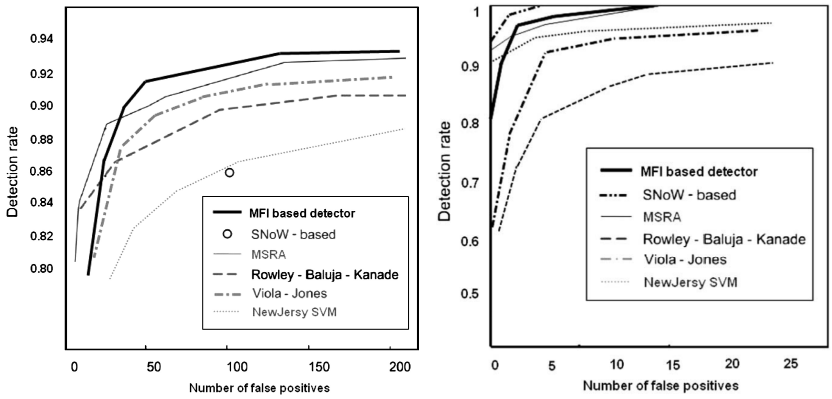


Figure 2.5: Four examples from the CMU-MIT database, three with positive faces and one without any.



(a) (b)

Figure 2.6: Detection rate versus number of false positives for several approaches. [?] On the left is the CMU-MIT dataset, which contained 483 faces. On the right is the BioID dataset, this contained 1521 faces.

2.2.1 Neural Network

Neural networks have often been applied to face detection, the implementation with the best performance was by Rowley *et al.* [34]. Their system consists of two stages, a preprocessing stage and the actual neural network. Windows of increasing size are taken from an image and then processed by the two stages. An overview can be seen in Figure 2.7. Each window is subsampled until it is 20x20 pixels. Preprocessing consists of lighting correction and histogram equalization. Then the pixels are used as input to the neural network.

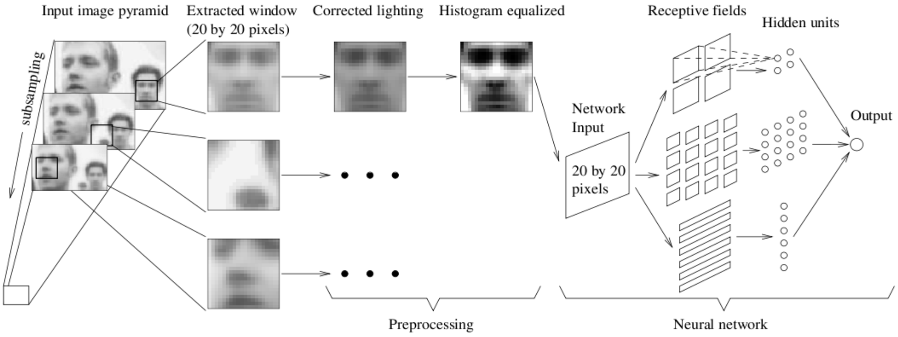


Figure 2.7: Face detection system with a neural network, image taken from [34].

The hidden layer consists of three types of hidden units, resembling the human receptive fields. There are a total of 26 hidden units. Four units look at 10x10 pixel subregions, dividing the region. Then there are sixteen, smaller regions of 5x5. These 20 look at the general shift of intensity across the whole image.

Finally, there are six 20x5 receptive fields looking for horizontal stripes. The authors suggest that the last group of hidden units are very useful to detect eyes, corners of the mouth and other important features in the human face. The result from the network is a boolean indicating whether the frame is a face or not.

On the MIT-CMU dataset this neural network detected up to 90% of the faces with about 100 false positives. A disadvantage of a neural network is that the model is hidden in the nodes, that means that it is hard to determine why it fails sometimes and even harder to know how to improve it. Furthermore, training it is difficult as the parameters have a large influence on a run being successful and this could mean a lot of experimental training.

2.2.2 SNoW

SNoW is a Sparse Network of Winnows introduced to face detection by Roth *et al.* [35]. The SNoW learning architecture was introduced earlier by Roth in 1999 [36]. It has been successfully trained on large training tasks in other, non-visual domains. The SNoW network consists of an input layer connected with weights to a class layer where the target nodes represent the possible classes. There are several possible update rules that can be used during training, according to Roth the Littlestone's Winnow update rule has been the most successful.

This rule is used in the implementation for face detection. The input layer consists of boolean features representing the input image. In [35], two types of features were tested, primitive features represented a position (x, y) and intensity $(I(x, y))$. This could also have been done without making them boolean, but the calculation time was smaller with this memory heavy approach. The other type of feature was multi-scale, here they included intensity mean and intensity variance over four sub-image scales (also encoded as boolean features).

SNoW detected up to 86% of the faces with about 100 false positives on the CMU-MIT database. As with the neural network this network is hard to analyse and therefore it is difficult to tell why it fails or succeeds.

2.2.3 Viola-Jones

The Viola-Jones detector [26, 37, 27] uses features that are very similar to Haar-features [30], see Figure 2.2(a) Each Haar-feature consists of two or three rectangles and a threshold. Each rectangle is defined by a position, size and orientation. Furthermore each has a weight. The value that a feature returns is given by:

$$\sum_{h \in \mathbb{H}} (I(h) * w_h) \begin{cases} > \theta & \text{true} \\ \leq \theta & \text{false} \end{cases}$$

with $\mathbb{H}$ the set of rectangles, $I(h)$ the total intensity of the part of the image covered by the rectangle h. w_h the weight of the rectangle h and θ the threshold.

These features are combined into a boosted cascade [38]. A boosted cascade is a binary decision tree where each node returns either “not a person” or “maybe a person”. See Figure 2.8. When the last node is reached and an image is still deemed as “maybe a person” it returns a positive signal. The resulting positive windows are combined using a simple averaging algorithm. When classifying the classifier takes windows from different sizes from the input image and runs it through the cascade.

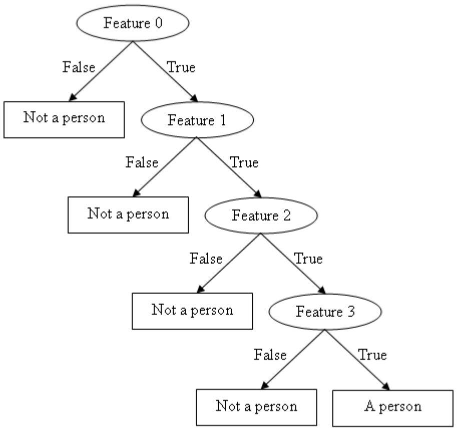


Figure 2.8: Example of a short boosted cascade

On the MIT-CMU dataset this method detected up to 92% of the faces with about 125 false positives. The Viola-Jones classifier by Kruppa *et al.* performed well on all kinds of faces, they do not specify a certain type of faces that is not recognised.

2.2.4 Edge Detection

Edge detection is very popular, it is effective and it is used in the visual system of mammals [39], which is a system that has proven itself for a long time. Two examples of face detection using edges are Jesorsky *et al.* [40] and a more recent article by Tsao *et al.* [**?**]. Both methods use Sobel's edge detection, which uses two 3x3 kernels to find the gradient of the image intensity [41]. A big change in intensity results in a high gradient, which is then used to determine the edges (which usually have high contrast). Jesorsky *et al.* then use the Hausdorff Distance to measure the similarity between a face model and the current window. They achieved up to 90% recognition rate on the BioID set.

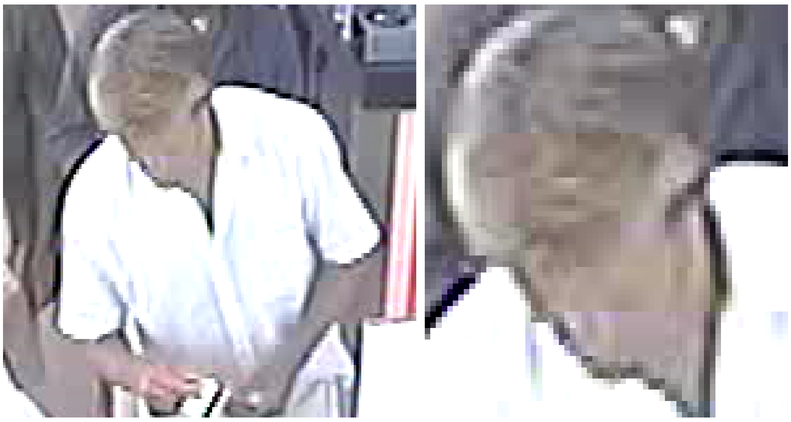
The second approach uses the MAFIA algorithm [42] to detect the most occurring positive and negative pattern in edge images. Positive here, means that it is frequent in the images with faces and negative that it is frequent in images without faces. MAFIA does this by constructing a search tree and then prunes infrequent nodes in a depth first search. The nodes of the three are the x,y-coordinates of the edges.

The features that are detected are combined in a cascaded classifier. To improve the computation efficiency a simple variance classifier analyses each window before the cascaded classifier. This classifier checks whether the variance in some regions, broadly defining the nose and mouth is not too high. After the cascaded classifier a kd-tree-based SVM classifier is used to increase the accuracy. The input for this classifier is all positives from the other two classifiers and it gives the final verdict of a window being a face or not. The method from Tsao *et al.* detected up to 93% of the faces on the CMU-MIT with about 150 false positives. This is better than the previous methods, but only by a small margin.

2.2.5 Overview of Face Detection Methods

Although all methods have been tested on the MIT-CMU database which seems to have low quality, this does not necessarily mean that it will work on the recordings from the Restaurant, as the quality is far worse. To show a typical example of the low quality and resulting difficulty, see Figure 2.9, where we present a close up of a person in the Restaurant.. The authors of the methods discussed in this section do not discuss specific features that their methods are able or unable to handle and with the small difference in performance that is measured on the CMU-MIT database it is hard to decide which method would be best suited for use on the Restaurant images.

In order to decide what method is suitable for the environment of the Restaurant we compared several methods on data from the Restaurant. This is described in further detail in Chapter 4. Using a pretrained classifier assumes that it is possible to generalise from faces in different situations to the specific environment for which the classifiers were not trained. In order to investigate this we picked a method that could be quickly trained and executed and compared the performance of the pre-trained classifier with classifiers trained on data from the Restaurant. We describe the training algorithm in the next section.



(a) zoomed on the person

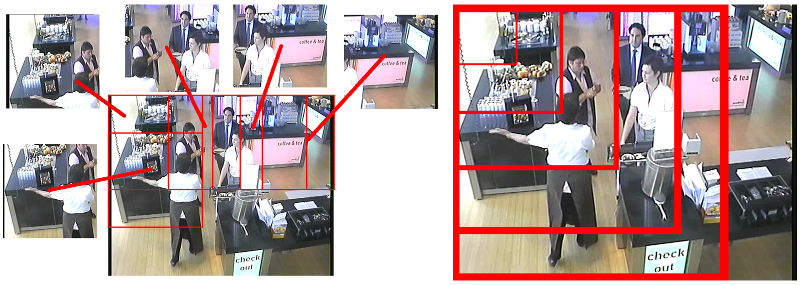
(b) zoomed further on the face

Figure 2.9: Example from the checkout recordings, zoomed in on the face

2.3 Training Viola-Jones Detector

Here we describe how to train the Viola-Jones detector introduced in subsection 2.2.3. Before going in detail two aspects are important, we already mentioned for each face detection method that it used windows of different scales, we will describe this process in detail here. Furthermore, Viola-Jones is much faster, compared to the other methods mentioned, we will also describe how this speed is achieved. After describing those two topics we will describe how to train the Viola-Jones detector.

Because the size of the person(s) in the image is unknown, the detector needs to take subsamples from the image and then for each determine whether it is a face or not. This subsampling is done by determining an initial size for the window and then moving this window over the image. When all possible windows have been extracted from the image, the size of the window is increased and it is moved over the image again. This is repeated until the window size is equal to the size of the image. See Figure 2.10 for illustration.



(a) windows are moved from left to right and then top to bottom

(b) window sizes are increased

Figure 2.10: Illustration of how windows are taken from a frame, intermittent windows are not shown for clarity.

The speedy classification by the Viola-Jones is caused by two factors, the cascaded tree (more on this later in this section) and the fact that Haar-features can be computed in constant time at any scale. For this, summed area tables are created. One set for the regular Haar-features and one set for the rotated rectangles. This means that the input window does not have to be scaled down, but that the features are scaled up to match the size of the window. This requires far less calculations.

Now we turn to the training of the detector. We already described the cascade, now we need to construct it as efficiently as possible. Each Haar-feature is a weak classifier [43], only able to reject a small amount of negative samples, combining many of these weak classifiers results in a strong classifier, as seen in Figure 2.11. When constructing the boosted cascade the Adaboost trainer [38] gives each negative training sample a weight. This weight is increased each time a new weak classifier classifies the sample wrong and set to zero once it is correctly rejected.

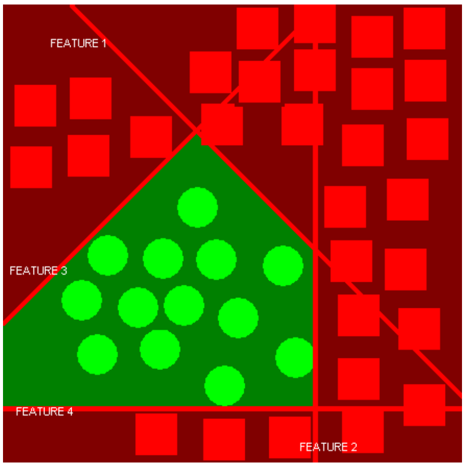
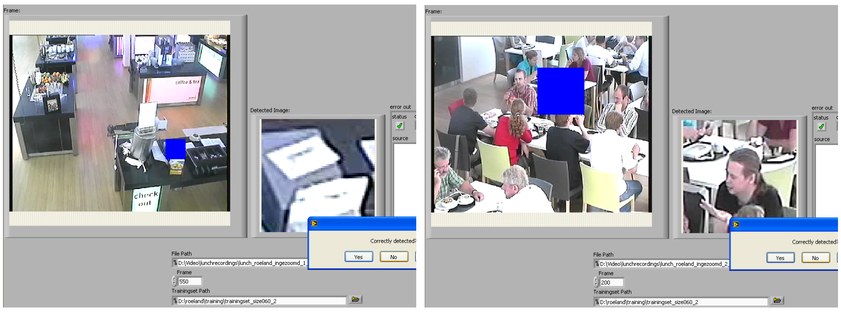


Figure 2.11: A schematic representation of a strong classifier made up by several weak classifiers. The green circles are positive samples, the red squares are negative samples. Feature 1 to 4 are weak classifiers that separate a small part of the negative samples from the positive ones. Together they separate all the negative from the positive samples.

During training the most significant feature is selected as top node, this means that it can already reject a large number of windows. The second node is the second most significant feature and can also reject a large number of windows, etcetera. Due to this hierarchy the cascade can analyse many windows per second. Significant means the feature that separates most of the negative samples from the positive ones.

Viola-Jones requires a large set of data to train, in order to speed up the process we used semi-online training. First we manually collected a small number of positive and negative samples, with this data we trained a classifier and then used this classifier on the training data to obtain more samples. We manually labelled these samples as positive, negative or unsure (discarded), see Figure 2.12.



File Path
D:\Video\lunchrecordings\lunch_roeland_ingezoomd_1_
Frame
550
Trainingset Path
D:\roeland\training\trainingset_size060_2

(a) Suggestions: bad

File Path
D:\Video\lunchrecordings\lunch_roeland_ingezoomd_2_
Frame
200
Trainingset Path
D:\roeland\training\trainingset_size060_2

(b) Suggestions: good

Figure 2.12: Supervised Training: the classifier presents detected ”persons” (blue box in the bigger image and then a close up on the right) to the researcher who votes positive, negatve or unsure

The advantage of this approach is that less manual selection is required and that the training data is more relevant. The last is quite intuitive, the wrong classifications that a classifier makes are more important to include in the training set compared to others. With this method those images are presented by the detector to the research and are then added to the list of negative samples.

2.4 Evaluation of Detectors

To evaluate the performance of the detectors we used precision, recall and F_1-measure [44]. For each testing image we measured three things: the number of true positives (tp), which are the correctly detected persons (see Table 2.2). The number of false positives (fp), the squares wrongly classified as a person and the number of false negatives (fn), the missed persons. Then precision $= \dfrac{\text{tp}}{\text{tp} \ + \ \text{fp}}$ and recall $= \dfrac{\text{tp}}{\text{tp} \ + \text{fn}}$. These measures can be combined into one number, an often used method is the F_1-measure, which is the harmonic mean of the precision and recall [44]: $F = 2 * \dfrac{\text{recall} * \text{precision}}{\text{recall} + \text{precision}}$.

		correct	
		Face	No Face
classification	Face	tp	fp
	No Face	fn	tn

Table 2.2: A table describing what types of error a detector or classifier can make. True positive (tp) and True negative (tn) are correct classifications. False positive (fp), also called Type I error, occurs when the detector marks something as a face while it is not. False negative (fn), Type II error, means that a face is rejected as not a face

Chapter 3

Experimental Setup

This chapter consist of two parts, in the first section we describe what datasets we collected and how we collected them. In section 3.2 we present the implementations of the face detection methods we discussed in chapter 2.

3.1 Data Aquisition

3.1.1 Hardware and Software

The cameras at the Restaurant are Panasonic WV-CS570 colour dome cameras, hanging from the ceiling. These cameras are compact 110mm diameter dome all-in-one colour units with a 22x optical zoom lens (3.79 - 83.4mm at F1.6) and auto focus. The recordings had a resolution of 720x576 pixels and were gathered by Noldus Recoder Software, which encoded the video in MPEG-4. Pictures (described in a later experiment) were taken at high quality (3888x2592 pixels) with a Canon EOS 400 camera.

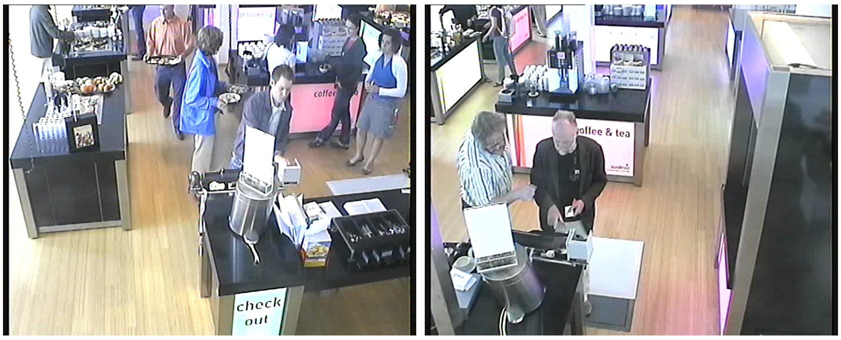
For processing a single machine was used. A Windows XP (SP3) system with an Intel Core 2 Quad Q6600 @ 2.4 GHz processor, 4 GB RAM. For C and C++ the compiler was Microsoft 32-bit C/C++ Optimizing Compiler Version 12.00.8168. Python was version 2.6.2 with Python Image Library 1.1.7. Matlab ran version 7.8.0 (R2009a). Finally, LabVIEW version 8.6 was used.

3.1.2 Recordings for Face Detection

To evaluate the different face detection approaches we used the recordings from two cameras, which were directed at the checkout. Recordings were made during lunch and consist of approximately 1.5 hour of video for each camera. On these videos, the visitors are standing behind the counter and it is viable to use the location of the face to determine the location of a visitor. At least 100 visitors visited the Restaurant and as almost everyone bought something at the Restaurant, most of them can be seen in the recordings. See Figure 3.1.

3.1.3 Photos for Face Detection Evaluation

To further investigate into the aspects of the recordings that cause problems for the face detectors, a second data set was created that consisted of high quality

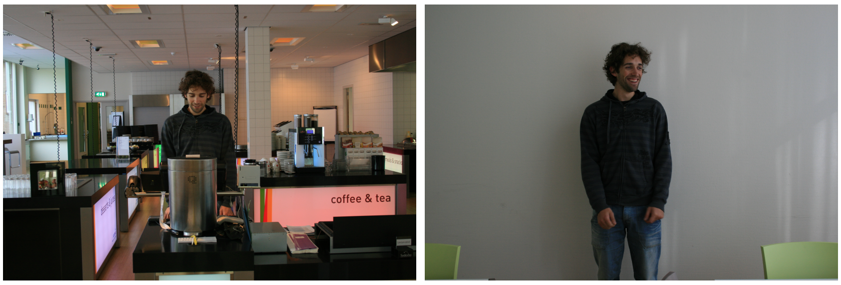


(a) Checkout 1

(b) Checkout 2

Figure 3.1: Stills of the recordings of the checkouts

pictures taken in the Restaurant. We identified five likely causes of performance deterioration: complex background, top view, light condition and video quality and encoding. Unfortunately, we were unable to take pictures without light influences, due to the opening hours of the Restaurant, but we systematically varied all other factors. The Restaurant is only open during the day and then the large windows cause fluctuations in the incoming light. We took pictures before a white wall and before the checkout (Figure 3.2) and with frontal view and top view (Figure 3.3).



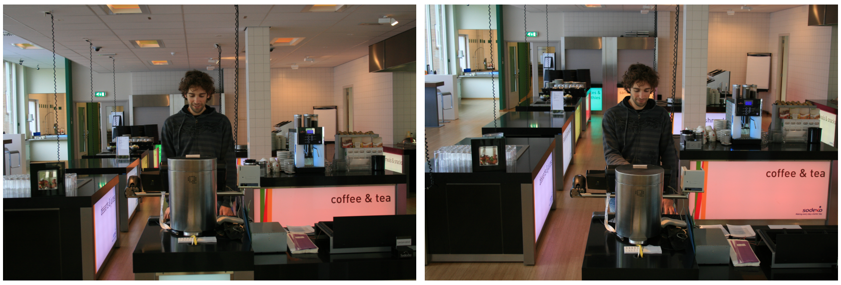
(a) Restaurant

(b) Wall

Figure 3.2: Example images with a cluttered background and a clean background

The resulting pictures were subsampled into three conditions: high, medium and low quality, respectively 1296x864, 788x525 and 394x263 pixels. The later two were chosen to reflect the number of pixels a face has in the recordings when close by and further away. Subsampling was done using the anti-alias method from Python Imaging Library[1] (PIL). We experimented with other methods for subsampling (nearest neighbour and linear interpolation) and noticed that there was little variance in the resulting quality. Therefore we selected the method which resembles the way the cameras translate real-life intensity values into a limited number of pixels (by combining a 'receptive' field into one pixel). Then

[1] http://www.pythonware.com/products/pil/



(a) Frontal (b) Top

Figure 3.3: Example images with a different viewpoint

a last condition was added, the MPEG-4 encoding and decoding, this was done using a binary encoder ffmpeg[2]. MPEG-4 was chosen as the recordings were encoded with the same encoding. The results can be seen in Figure 3.4, the faces are presented in detail in Figure 3.5.

In total 21 subjects were chosen to pose for the photos. Part of the pictures were used for a training set, see the next subsection. This left 14 subjects in 4 different conditions (two viewpoints and two backgrounds). For each condition 9 pictures were taken. Subsampling and encoding multiplies the number of conditions with 6 (three qualities and two encoding conditions). To get a clear overview of the different conditions see Table 3.1. This means that the evaluation set consists of $14 * 9 * 2 * 2 * 3 * 2 = 3024$ pictures. In some recordings there was more than one person visible as the employees at the Restaurant were already preparing the Restaurant for opening. These faces were also included in the evaluation, resulting in a total of 3113 faces.

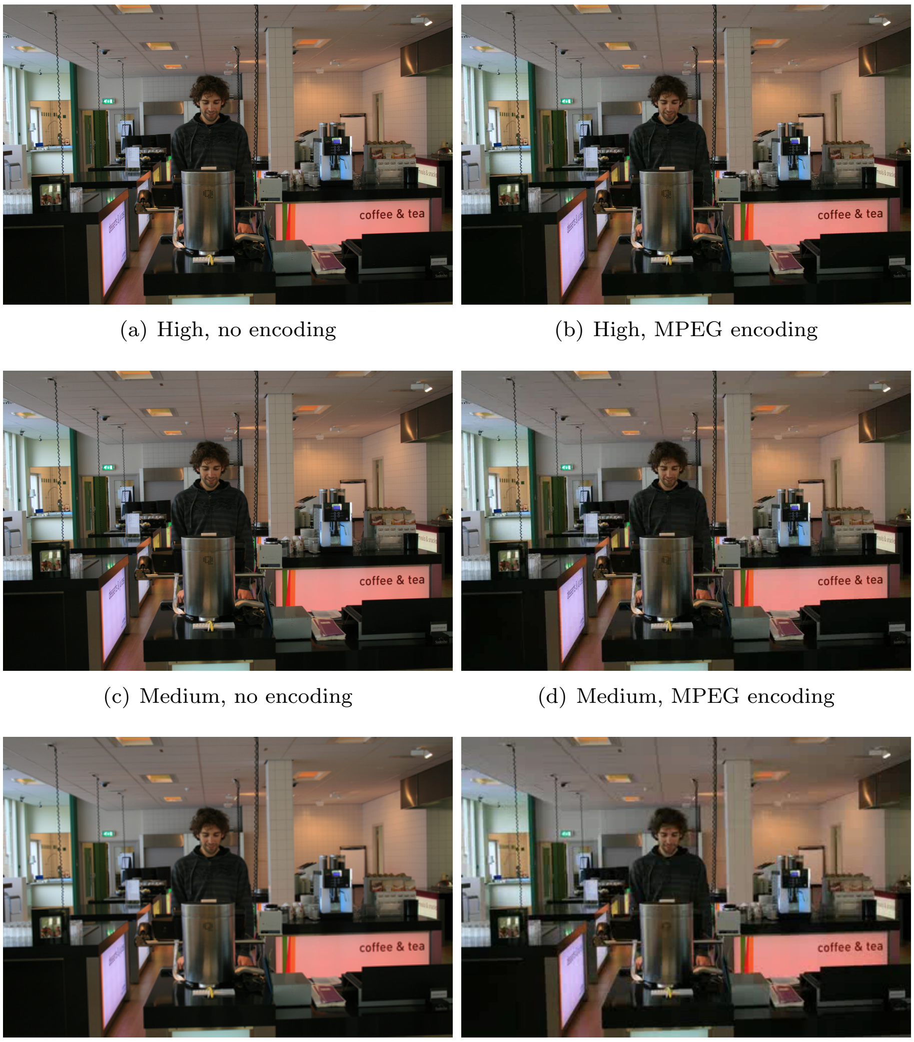
3.1.4 Photos for Face Detection Training

As mentioned in the previous subsection part of the high quality pictures was used to train a Viola-Jones detector. From the 21 subjects we randomly selected 7 to form the training set. This left sufficient subjects for evaluation. Two training sets were created from those 7 subjects. We chose for two different training sets in order to measure the effect of including the more difficult images. The 'clean' training set consisted only of the pictures at highest quality with no encoding, taken in with frontal view in front of the wall. The 'dirty' training set contained all pictures of the 7 subjects.

3.2 Tools

In this chapter we present the implementations of some of the methods for face detection discussed in the previous section. We have found freely available versions of those methods. We also included one additional method that was not discussed in the previous chapter, because the exact working is unknown.

[2] http://www.ffmpeg.org/



(a) High, no encoding

(b) High, MPEG encoding

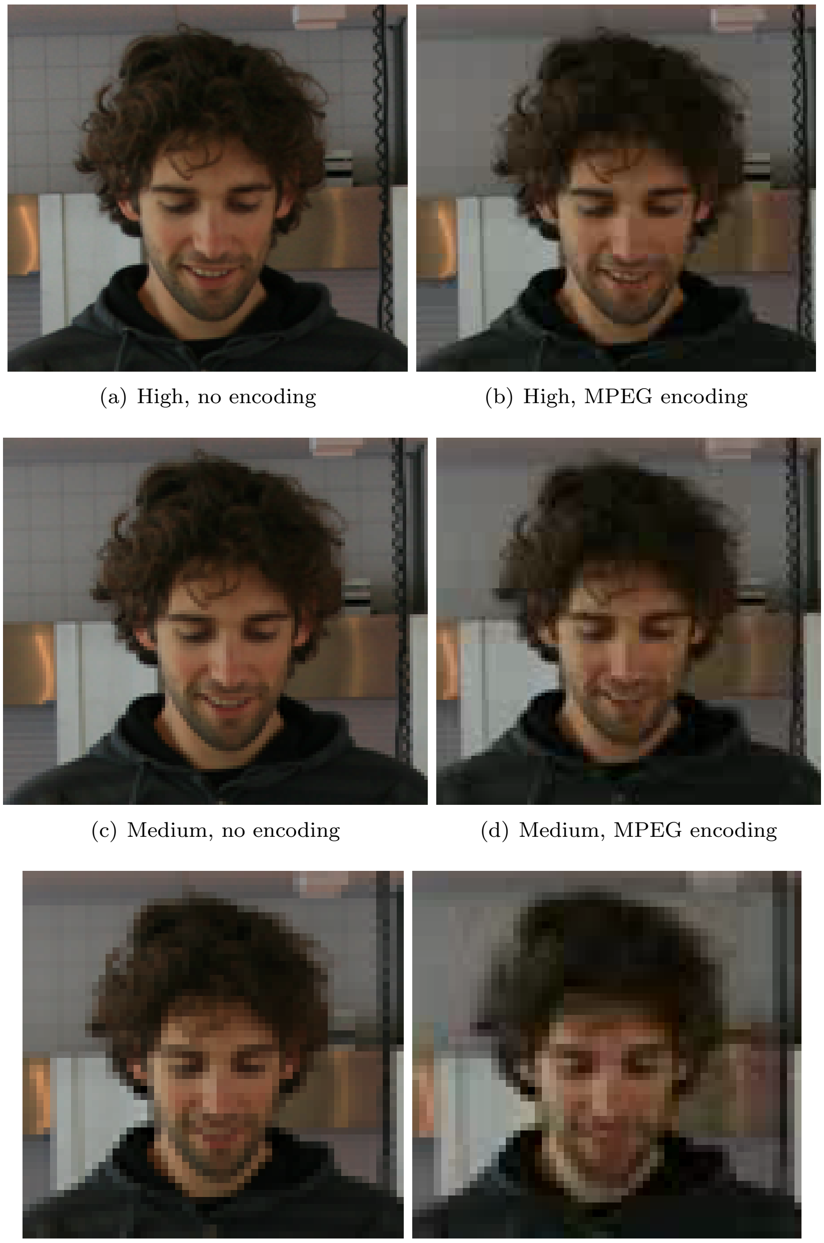
(c) Medium, no encoding

(d) Medium, MPEG encoding

(e) Low, no encoding

(f) Low, MPEG encoding

Figure 3.4: Example images with a different quality and encoding. These images were used to measure the effect of these factors on the performance of the different face detectors



(a) High, no encoding

(b) High, MPEG encoding

(c) Medium, no encoding

(d) Medium, MPEG encoding

(e) Low, no encoding

(f) Low, MPEG encoding

Figure 3.5: The same images as in Figure 3.4, but zoomed in on the face to better show the effect of the low quality and encoding

Background	Viewpoint	Quality	Encoding
Wall	Frontal	High	None
			MPEG-4
		Medium	None
			MPEG-4
		Low	None
			MPEG-4
	Top	High	None
			MPEG-4
		Medium	None
			MPEG-4
		Low	None
			MPEG-4
Checkout	Frontal	High	None
			MPEG-4
		Medium	None
			MPEG-4
		Low	None
			MPEG-4
	Top	High	None
			MPEG-4
		Medium	None
			MPEG-4
		Low	None
			MPEG-4

Table 3.1: Overview of the cases conditions that were used for evaluation. In each condition nine pictures each were taken from fourteen different people

3.2.1 Viola-Jones implementation

A Viola-Jones classifier and trainer is available in OpenCV[3]. Included in the package is a trained face detector and upper-body detector, which we used for evaluation. We developed a python script to collect negative and positive images and convert them to the correct format for the trainer. LabVIEW was used to call the classifier on the evaluation images. Evaluation of the full set took approximately 2 hours.

3.2.2 Neural Network implementation

The neural network from Rowley *et al.* was available in C[4]. A python script called this neural network on each of the pictures in the evaluation set and drew a rectangle around the detected faces. The script ran for approximately 30 hours on the full set of photos.

3.2.3 SNoW implementation

A trained SNoW face detector is available for Matlab[5]. To evaluate the performance a script was constructed to detect faces on each image and draw a rectangle around the detected windows. Evaluating the full set of photos took approximately 21 hours.

3.2.4 Picasa

Face detectors are very common in many applications these days, digital cameras often include some face detection software. We used Picasa[6] to evaluate one state of the art example. Unfortunately it is unknown what approach Picasa uses, it was therefore not introduced in the previous chapter. Because Picasa runs in the background it is also impossible to determine how long it took to evaluate the full set of photos.

[3] http://opencv.willowgarage.com/wiki/

[4] http://vasc.ri.cmu.edu/NNFaceDetector

[5] http://www.mathworks.com/matlabcentral/fileexchange/13701

[6] http://picasa.google.com

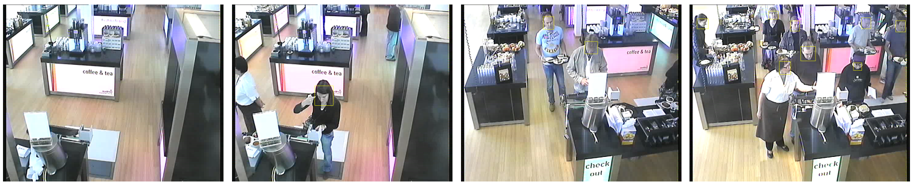
Chapter 4

Experiments

In this chapter we describe the three experiments that were done to answer the questions posed in Chapter 1. In these we used the methods discussed in Chapter 2 on the datasets we presented in Chapter 3. For each experiment we first describe the goal and the approach we took, then we describe how we evaluated the results and finally we present the results.

4.1 Evaluating Face Detection Methods on Recordings

The goal of this experiment was to determine the performance of the freely available face detectors that have achieved good results on the MIT-CMU database in our setting. We evaluated four detectors, a Viola-Jones detector [27], the neural network from Rowley *et al.* [34], the SNoW detector [35] and Picasa. See section 3.2 for more details. For this experiment we used the recordings presented in subsection 3.1.2. We selected 100 evaluation frames, 25 with nobody visible, 25 with one face visible. In 25 frames, two faces were visible and in the last 25 frames, three or more faces could be identified, see Figure 4.1. Those different situations were chosen to ensure that the algorithms were able to handle multiple persons.



(a) no persons (b) 1 persons (c) 2 persons (d) 3+ persons

Figure 4.1: Examples from the evaluation set created from the recordings of the checkout cameras in the Restaurant. Yellow rectangles mark the faces that should be detected

4.1.1 Results

To determine the performance of the different face detectors we measured the recall, precision and combined them into the F_1-measure. We also measured the running time, as the processing time becomes interesting when large sets of recordings are processed. All methods ran on the system described in the previous chapter. The running time for Picasa could not be determined as it runs in the background. The results are summarised in Table 4.1. Picasa, the neural network and the SNoW detector all detect less than 10% of the faces, Viola Jones detects almost 35% of the faces. All methods return many false positives resulting in a low precision and therefore F_1-measure. The results are further discussed in the next chapter.

	Recall	Precision	F_1-measure	Time (in s)
Picasa	0.063	0.140	0.09	-
Rowley	0.057	0.423	0.10	1449
SNoW	0.021	0.364	0.04	1684
Viola-Jones	0.349	0.090	0.14	60

Table 4.1: Recall, precision, F_1-measure and running time for the four approaches evaluated on the recordings from the counters in the Restaurant.

4.2 Determining Influences of Aspects from the Recordings on Face Detection Performance

To determine what aspect(s) of the recordings caused the face detectors to perform so badly, compared to the results reported on the CMU-MIT database we set up a follow-up experiment. We decided to take pictures of fourteen subjects in which we varied the background and viewpoint. For each subject we took nine pictures and then varied the quality and encoding of them. With this setup we aim to find the feature of the recordings that causes the low performance. See subsection 3.1.3 for more details. We evaluated each of the four face-detection methods that were available (Viola-Jones, neural network, SNoW and Picasa) on all of those pictures. This resulted in a recall, precision and F_1-measure for each condition, summarised in Table 4.2.

4.2.1 Results

Here we present the results from the experiment described above. There was a big difference in the precision, which affected the F_1-measure. Although this precision is not unimportant this can be improved on using other information, such as location within an image, colour information or a low person recognition rate later in the system. Because we were mainly interested in getting a high recall we focussed on the recall in the statistical analysis of the results. The recall, precision and F_1-measure of the main effects can be seen in Table 4.2. To discover if the main effects were significant t-tests were performed. A 2x2x3x2 ANOVA was used to discover any interaction effects.

	Recall				Precision				F_1-measure			
Condition	Pica	VJ	SNoW	NN	Pica	VJ	SNoW	NN	Pica	VJ	SNoW	NN
Overall	0.71	0.70	0.57	0.30	1.00	0.22	0.46	0.28	0.83	0.34	0.51	0.29
Background:												
Wall	0.92	0.86	0.73	0.44	1.00	0.39	0.73	0.28	0.96	0.53	0.73	0.36
Restaurant	0.52	0.55	0.43	0.17	1.00	0.14	0.29	0.27	0.68	0.22	0.34	0.21
Viewpoint:												
Frontal	0.75	0.72	0.58	0.31	1.00	0.23	0.48	0.27	0.86	0.35	0.53	0.29
Top	0.67	0.68	0.57	0.29	1.00	0.21	0.44	0.28	0.80	0.32	0.49	0.29
Quality:												
High	0.73	0.81	0.76	0.00	1.00	0.13	0.33	1.00	0.85	0.23	0.46	0.00
Medium	0.73	0.77	0.75	0.56	1.00	0.31	0.65	0.25	0.85	0.44	0.69	0.34
Low	0.67	0.53	0.21	0.35	1.00	0.56	0.75	0.35	0.80	0.54	0.32	0.35
Encoding:												
None	0.75	0.71	0.64	0.30	1.00	0.32	0.33	0.28	0.86	0.32	0.44	0.29
MPEG-4	0.67	0.70	0.50	0.30	1.00	0.24	0.85	0.28	0.81	0.35	0.63	0.29

Table 4.2: Recall, precision, F_1-measure of the main effects for the four approaches evaluated, Pica is Picasa, VJ is Viola-Jones and NN is the neural network.

Overall Picasa performed best, with the highest recall and precision and thus also F_1-measure. Viola-Jones performed not significantly worse with respect to the recall, but due to the very low precision the F_1-measure was also much lower. The SNoW detector detected over 50% of the faces with a precision just below 50%, resulting in a F_1 measure of 0.51. The neural network only detected one-third of the faces with a low precision and thus F_1-measure.

The neural network failed to detect any faces on the high quality images, this was probably due to the limited size of the subsamples it takes and this was not included in any of the tests. For background the difference in performance was significant for each method with $P < 0.001$. The performance difference on viewpoint was not significant for SNoW and the neural network, but for Viola-Jones with $P < 0.01$ and Picasa with $P < 0.001$. None of the methods showed a significant difference between the high and medium quality condition, however for each the difference in recall was significant for both high and low and medium and low with $P < 0.001$. The last main effect of MPEG-4 encoding was significant for both SNoW and Picasa with $P < 0.001$, but not for the neural network and Viola-Jones.

There was little interaction effect between the various factors that did not resemble the main effects in the previous paragraph. For Picasa the only significant interaction effect was between background and encoding with $P < 0.05$. The background and viewpoint as well as the performance difference on the background and quality was significant for Viola-Jones with $P < 0.001$. For the SNoW method there was a significant interaction effect between background and encoding and background, encoding and quality with $P < 0.001$. Furthermore, the difference in performance on the background and viewpoint was significant with $P < 0.05$. Finally, for the neural network there were no significant interaction effects (when controlled for the 0 detections on high quality).

All methods achieved their highest recall when the wall was the background, for Viola-Jones there was no difference between viewpoint and only a slight difference between high and medium quality as well as between no encoding en MPEG-4. With those conditions the Viola-Jones detector correctly detected 92% of the people in the pictures. Picasa performed optimal on the images with frontal view, high or medium quality and no encoding, achieving as much as 99% recall. For SNoW the viewpoint had no influence, with the wall as background, high quality and no encoding the method detected 94% of the faces. Finally the neural network worked best with no encoding on medium quality with frontal view with 75% recall.

The worst recall for each method was with the Restaurant as background, the recall for SNoW dropped to 0.01% with frontal view, low quality and MPEG-4 encoding. Picasa, Viola-Jones and the neural network performed worst on the low quality, MPEG encoded, topview pictures. The recall dropped to respectively 38%, 24% and 10%.

4.3 Comparing General Face Detector with a Specific Face Detector

The final experiment was designed to answer the question whether a face detector trained on faces from the Restaurant would perform better than the face detectors trained for the CMU-MIT database. To do so we trained several Viola-Jones detectors using the seven subjects that were removed from the evaluation data in section 3.1.4. Two sets were formed out of these pictures, the clean training set consisted of pictures with ideal condition: high quality, no encoding, frontal view and with the subject standing in front of a wall. Because this contained only a very small set of pictures that did not adequately represent the recordings we also created a dirty set, which consisted of all the pictures taken in the Restaurant. This set was processed with the same method used for the evaluation set, they were subsampled and encoded and decoded. This means that the dirty set contains both the ideal and the worst type of pictures.

The training process of a Viola-Jones detector requires a fixed width and height size to which the images are reduced before they are fed to the Adaboost algorithm. For ease of use we used squares and varied the size of the square between 10, 30, 50 and 60. The training required too much memory with a bigger size.

Because the clean set only had $7*9 = 63$ pictures, we cut the 63 faces visible on them and this was the initial training data for that set. For the dirty set this number was much larger, $7*9*2*2*3*2 = 1512$ pictures. Therefore, for each condition (background and viewpoint) and subject we randomly selected one original picture and cropped the faces from each of the six processed images that were the result from this original picture. In total this were $2*2*7*3*2 = 168$ pictures as initial training data.

Then the trained detector was used to generate more training data, according to the method presented in section 2.3. We repeated this until we had three evaluation methods and then checked for an improvement in the performance. The first evaluation determined that the sizes 10, 30 and 50 to which the images were reduced were far outperformed by size 60 (which obviously contains more information). Therefore, we only looked at this size, leaving a possible lower resolution for later optimisation.

4.3.1 Results

Here we present the results from the experiment described above. We evaluated the two trained detectors on the complete set but for clarity we only present the performance on the best images and the worst. Here the best images are with the clean background, frontal view, high quality and no encoding. On the other hand the worst images are those with a cluttered background, top view, low quality and MPEG-4 encoding. The results are plotted in Figure 4.2.

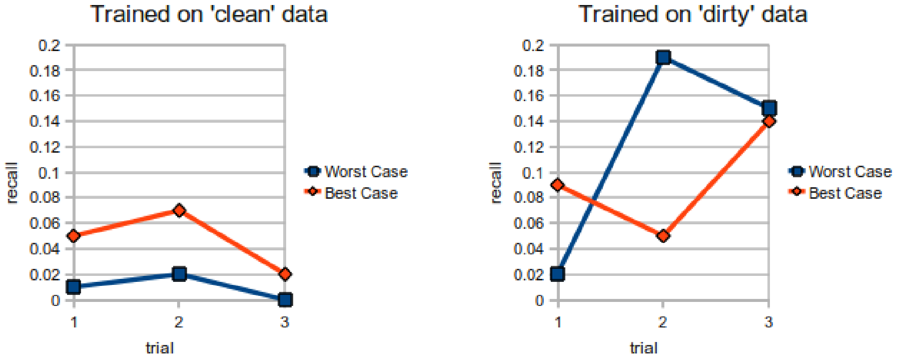


Figure 4.2: The results from the last experiment. On the left is the evaluation of the detector trained on the 'clean' data set, while the right graph displays the results of the detector trained on the 'dirty' set. On the y-axis the recall of the detector on the worst images (blue squares) and the best images (red diamonds) are shown. On the x-axis the trials are shown.

Chapter 5

Discussion

In the final chapter of this thesis we first answer the research questions posed in Chapter 1 in the conclusion, then we will present some discussion on the answers and finally we look at some interesting points for future work.

5.1 Conclusion

Due to the uncontrolled environment of the Restaurant all person detection techniques have some disadvantages. We selected person detection based on face detection as the best approach for a subset of the data. Face detection on still images has made more development compared to person detection on still images, person detection is often done in tracking systems. At the checkout the face is usually visible and then information about the appearance of a person can be used to find this person in the recordings later, for instance with the colour of hair and shirt. As such a starting point is useful for tracking systems we investigated face detection on the checkout recordings.

The performance of the face detectors was much lower than expected, we investigated further in several possible causes of this bad performance. Even with a very high number of false positives the best recall was below 35%. Therefore, we can conclude that while person detection in single images from the recordings of the Restaurant should be possible, with most common face detectors it is not. Possible features of the recordings that caused the low recall are the low quality of the recordings, the MPEG encoding, the angled view, the changes in light and the complex background. In a second experiment we controlled four of these conditions to see their impact, in the worst condition the algorithms performed below 40%.

Among the possible causes we investigated, the complexity of the background had the strongest influence on the performance. This almost halved the performance of most methods. This is not so strange when we look at the three known methods (Viola-Jones, SNoW and the neural network). Each uses context information, the constructed model from Viola-Jones does not only consist of the face, but compares background with the face. For instance the intensity of the background left and right of the face is compared to the intensity of the face. When the background is complex the background left and right of a face has a high chance to be different and thus has different intensity.

That the quality influences the performance is to be expected, however, between high and medium quality there is no significant effect, while there is between medium and low. The performance drop is non-linear, which means that some essential information is lost between medium and low quality. With so few pixels as in the low quality condition the faces no longer match the model.

On the encoding the methods reacted differently, Picasa and SNoW performed significantly worse with the MPEG encoding, while there was no significant effect for the neural network and the Viola-Jones detector. Although the method of Picasa is unknown, the fact that no wrong detections are made suggests that beside looking for patterns it also checks if the colour is right. As the colour is influenced by the MPEG encoding this might cause the drop in performance. The other methods work with grayscale images, which should remove most of the influence from the encoding, however SNoW is still affected. As the network learned in SNoW is inaccessible it is hard to determine why it is influenced by the MPEG encoding.

The methods also reacted differently to the viewpoint. Viola-Jones and Picasa performed significantly worse, while there was no impact on SNoW and the neural network. For SNoW it is easy to see why it is able to handle the changes in viewpoint as it uses no fixed ratio between width and height. This compensates for the distortion caused by the top view. However, the neural network uses a fixed ratio. One possible explanation would be that the detection rate is so low that the impact from the viewpoint is just too small.

Finally, the results of the second experiment do not seem to match the results from the first experiment in which the methods performed far worse. This is partly because Table 4.2 does not show the results of a combination of the conditions. At the end of section 4.2 we mention the worst recall and on which condition this occurred. Still, Picasa does far better in this worst case, compared to the first experiment. This is hard to explain because we do not know the inner workings of Picasa, but the colours on the pictures taken seem to be better compared to the colours on the recordings and this could possibly cause the very bad performance of Picasa on the recordings.

In a last experiment we attempted to improve the performance of the Viola-Jones face detector by training it on data from the Restaurant. To obtain sufficient data for training we used semi-online training to train more effectively. Unfortunately, the performance was very low and did not show clear improvement between sessions. One possible explanation is that there was not enough positive training data to learn a working model in the first training session. Without a somewhat working model this approach is unable to improve as it only adds negative data (or a few positive by chance) and is then still unable to construct a correct model.

Overall we can conclude that the popular techniques for face detection do not work very well on the recordings from the Restaurant. Several factors influence the performance. We analysed their importance the previous section. Picasa and Viola-Jones work the best, where Picasa works best on the best images and Viola-Jones outperforms Picasa on the imperfect images. However, the difference is very small and in the end the high precision of Picasa gives it the edge. In the next section we will discuss some possibilities to improve the face detectors so they can handle the features of the Restaurant that hinder the successful detection.

If the detection of faces is already difficult we expect that person detection is even harder as this is already a more difficult problem in simpler environments. The methods we discussed are probably not sufficient without many additional measures. Therefore, it seems like a good idea to look at a tracking system that uses information from other frames to determine the location of a person in one frame.

5.2 Future Work

Removing the influence from the complex background is not easy, because the environment needs to keep the current appearance. There are two types of solution, a software solution would consist of cleaning up the background so that it would no longer influence the face detectors. A hardware solution would change something in the setup (other than changing the background) that will simplify the task for the detectors. In the next two paragraphs we will describe a few of those solutions.

The hardware solution would include higher quality cameras (or more cameras and all zoomed in further) and maybe an unobtrusive camera at the checkout. The higher quality images gained from this camera will help, furthermore, the top-view problem will be removed. This does not solve the background problem, but if this camera is aimed slightly upwards the Restaurant will no longer be the background but the ceiling will be and the ceiling is a white, dropped ceiling. Which is of course far simpler and should help the performance. However, the slight tilt might introduce problems similar to the top-view problem, we will address this in a bit when we discuss possibilities to solve the top-view problem.

A software solution would require a clean up of the background, a possibility for this is to look at background modelling. We rejected this earlier as a method for person detection, but it can be used to remove certain parts of the background that are definitely background. This can be done by training on images of the background to learn certain patterns that occur only in the background. Some possibilities could be to look at an edge map and find straight edges or sharp corners and remove these structures as you will not find such lines in human faces.

The problem of top view can also be solved in two ways, training on top view images or unfixing the ratio between height and width. Training is not a very good solution as the faces are seen at different angles, if people are shorter or are standing farther away the angle changes. Therefore, relaxing the fixed ratio is the best option, unfortunately this means that a far larger amount of subsamples can be taken from a single image. This means that the processing will take longer and there will probably be more false hits.

It is difficult to conclude anything from the last experiment as the results appear quite random. In order to properly investigate in the training of a Viola-Jones detector a new experiment is needed with a much larger set of training images. Such an experiment could also investigate into the required number of data for a good result and in the use of semi-online training in combination with Viola-Jones.

Bibliography

[1] H. Schepers, R. de Wijk, J. Mojet, and A. Koster, “Innovative consumer studies at the Restaurant of the Future,” *Proceedings of Measuring Behavior*, p. 366, 2008.

[2] W. Zajdel, J. Krijnders, T. Andringa, and D. Gavrila, “CASSANDRA: audio-video sensor fusion for aggression detection,” in *Proceedings of the 2007 IEEE Conference on Advanced Video and Signal Based Surveillance*, vol. 0. IEEE Computer Society, 2007, pp. 200–205.

[3] C. Liu and P. Yuen, “Human action recognition using boosted EigenActions,” *Image and Vision Computing*, vol. 28, no. 5, pp. 825–835, 2010.

[4] Q. Luo, X. Kong, G. Zeng, and J. Fan, “Human action detection via boosted local motion histograms,” *Machine Vision and Applications*, vol. 21, no. 3, pp. 377–389, 2010.

[5] B. Achermann and H. Bunke, “Combination of Face Classifiers for Person Identification,” in *Proceedings of the International Conference on Pattern Recognition*, vol. 3. IEEE Computer Society, 1996, p. 416.

[6] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer, “Multi-Camera Multi-Person Tracking for EasyLiving,” in *Proceedings of the Third IEEE International Workshop on Visual Surveillance (VS’2000)*. IEEE Computer Society, 2000, p. 3.

[7] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. Wiley-Interscience, 2001.

[8] Q. Delamarre and O. Faugeras, “3D articulated models and multi-view tracking with silhouettes,” in *The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999*, vol. 2, 1999.

[9] A. Mcivor, “Background subtraction techniques,” *Proc. of Image and Vision Computing*, pp. 147–153, 2000.

[10] J. Heikkilä and O. Silvén, “A real-time system for monitoring of cyclists and pedestrians,” *Image and Vision Computing*, vol. 22, no. 7, pp. 563–570, 2004.

[11] P. Power and J. Schoonees, “Understanding Background Mixture Models for Foreground Segmentation,” in *Proceedings Image and Vision Computing New Zealand*, 2002, p. 267.

[12] P. Rosin and T. Ellis, "Detecting and classifying intruders in image sequences," in *Proc. British Mach. Vis. Conf*, 1991, pp. 24–26.

[13] V. Vezhnevets, V. Sazonov, and A. Andreeva, "A survey on pixel-based skin color detection techniques," in *Proc. Graphicon*, vol. 85, 2003.

[14] C. Wang and M. Brandstein, "Multi-source face tracking with audio and visual data," in *1999 IEEE 3rd Workshop on Multimedia Signal Processing*, 1999, pp. 169–174.

[15] J. Brand and J. Mason, "A comparative assessment of three approaches to pixel-level human skin-detection," in *International Conference on Pattern Recognition*, vol. 15, 2000, pp. 1056–1059.

[16] R. Hunt and E. Carter, *The reproduction of colour.* Fountain Press London, 1995.

[17] M. Schwarz, W. Cowan, and J. Beatty, "An experimental comparison of RGB, YIQ, LAB, HSV, and opponent color models," *ACM Transactions on Graphics (TOG)*, vol. 6, no. 2, p. 158, 1987.

[18] D. Coppola, H. Purves, A. McCoy, and D. Purves, "The distribution of oriented contours in the real world," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 95, no. 7, p. 4002, 1998.

[19] N. Sebe, I. Cohen, T. Huang, and T. Gevers, "Skin Detection: A Bayesian Network Approach," in *Proceedings of the 17th International Conference on Pattern Recognition*, vol. 2. IEEE Computer Society, 2004, p. 906.

[20] M. Jones and J. Rehg, "Statistical color models with application to skin detection," *International Journal of Computer Vision*, vol. 46, no. 1, pp. 81–96, 2002.

[21] D. Lowe, "Object recognition from local scale-invariant features," in *International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157.

[22] A. Majumdar and R. Ward, "Discriminative SIFT Features for Face Recognition," in *Proceedings of Canadian Conference on Electrical and Computer Engineering, 2009. CCECE09*, 2009, pp. 27–30.

[23] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE Computer Society, 2005, p. 893.

[24] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Computer Vision–ECCV 2006*, pp. 404–417, 2006.

[25] N. Dalai, B. Triggs, I. Rhone-Alps, and F. Montbonnot, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005*, vol. 1, 2005.

[26] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, 2001.

[27] H. Kruppa, M. Castrillon-Santana, and B. Schiele, “Fast and robust face finding via local context,” in *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, 2003, pp. 157–164.

[28] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio, “Pedestrian Detection Using Wavelet Templates,” in *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition.* IEEE Computer Society, 1997, pp. 193–199.

[29] A. Broggi, M. Bertozzi, A. Fascioli, and M. Sechi, “Shape-based pedestrian detection,” in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2000, pp. 215–220.

[30] R. Lienhart and J. Maydt, “An extended set of haar-like features for rapid object detection,” in *IEEE ICIP*, vol. 1, no. 2002, 2002, pp. 900–903.

[31] R. Hsu, M. Abdel-Mottaleb, and A. Jain, “Face detection in color images,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 5, pp. 696–706, 2002.

[32] J. Luo, Y. Ma, E. Takikawa, S. Lao, M. Kawade, and B. Lu, “Person-specific SIFT features for face recognition,” in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007. ICASSP 2007*, vol. 2, 2007.

[33] M. Yang, D. Kriegman, and N. Ahuja, “Detecting faces in images: A survey,” *IEEE Transactions on Pattern analysis and Machine intelligence*, pp. 34–58, 2002.

[34] H. Rowley, S. Baluja, and T. Kanade, “Neural network-based face detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23–38, 1998.

[35] D. Roth, M. Yang, and N. Ahuja, “A snowbased face detector,” in *Neural Information Processing*, vol. 12, 2000.

[36] D. Roth, “The SNoW learning architecture,” Technical Report UIUCDCS, Tech. Rep., 1999.

[37] P. Viola and M. Jones, “Robust real-time face detection,” *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[38] Y. Freund and R. Schapire, “A desicion-theoretic generalization of on-line learning and an application to boosting,” in *Computational Learning Theory.* Springer, 1995, pp. 23–37.

[39] D. Hubel and T. Wiesel, “Receptive fields of single neurones in the cat’s striate cortex,” *The Journal of Physiology*, vol. 148, no. 3, p. 574, 1959.

[40] O. Jesorsky, K. Kirchberg, R. Frischholz *et al.*, "Robust face detection using the Hausdorff distance," *Lecture Notes in Computer Science*, pp. 90–95, 2001.

[41] K. Engel, *Real-time volume graphics.* AK Peters Ltd, 2006.

[42] D. Burdick, M. Calimlim, J. Flannick, J. Gehrke, and T. Yiu, "Mafia: A maximal frequent itemset algorithm," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1490–1504, 2005.

[43] C. Ji and S. Ma, "Combinations of weak classifiers," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 32–42, 1997.

[44] C. Van Rijsbergen, *Information Retrieval.* Buttersworths, London, 1979.