SILESIAN UNIVERSITY OF TECHNOLOGY

FACULTY OF AUTOMATIC CONTROL, ELECTRONICS AND COMPUTER SCIENCE

COURSE: COMPUTER SCIENCE

BACHELOR THESIS

Eye Blinking Detection and Analysis

Author: Tomasz Jarzyński

Supervisor: Prof. Dr. Bogdan Smołka, Silesian University of Technology

Supervisor: Prof. Dr. Marcin Grzegorzek, University of Siegen

Consultant: M.Sc. Zeyd Boukhers, University of Siegen

Gliwice, Siegen, January 2014

UNIVERSITÄT SIEGEN

This work would not have been possible without the help and guidance from my professors, friends and family. Firstly I would like to personally thank Prof. Dr. Marcin Grzegorzek for helping me with my bachelor work even in the evenings. Prof. Dr. Bogdan Smołka for giving me a possibility to write my BSc thesis aboard. M. Sc. Zeyd Boukhers for being the first person to help me with problems. Dr. Maria Czarnecka for medical consultancy. Amie Crate and Ben Wrigley for help with my English. Also I would like to thank Universität Siegen for giving me the opportunity to work on my BSc thesis in Germany.

And finally to my parents for supporting me for many years.

Contents

Chapter 1

Introduction

The eye is one of the most important organs in human body. Thanks to it we can see everything that surrounds us. It is hard to image how difficult it must be to live without the possibility to observe the world.

1.1 Problem Statement

In this bachelor thesis I address the problem of automatic detection and analysis of human eye blinks in video data. For this, a software module is required that applies automatic image and video processing techniques.

1.2 Motivation and Challenges

If we have some knowledge about how to interpret the eye blinks in some situations, we can get more information about this person. Blinking of the eye depends on human emotions. Based on knowledge about frequency of the eye blinks we can recognise that somebody is bored, tired, thinking very intensively about something, lying or have eyes diseases.

For example, when people lie, their eyes blink less frequently because the brain works a lot more to create new facts so that things that had happened before had a logical connection to the past. After lying their eyes blink more frequently. When people do not lie, their eyes still blink with the same frequency [6] [13].

This challenge is hard to resolve because the appearance of each person is different:

- Color of the skin (Figure 1.1 a)). Let us say that we might have brighter or darker skins. These are two totally different problems to solve. It is necessary to do two various segmentations of an image.

- Some people have very widely or narrowly open eyes (Figure 1.1 b)).
- Sometimes it is hard to say if an eye is open or closed (figure 1.1 c)).
- Blinks are too fast for camera to capture the closed eye (Figure 1.1 d)).
- People can wear glasses, colors of their faces can be distorted (Figure 1.1 e)). I did not evaluate these cases in my bachelor work.

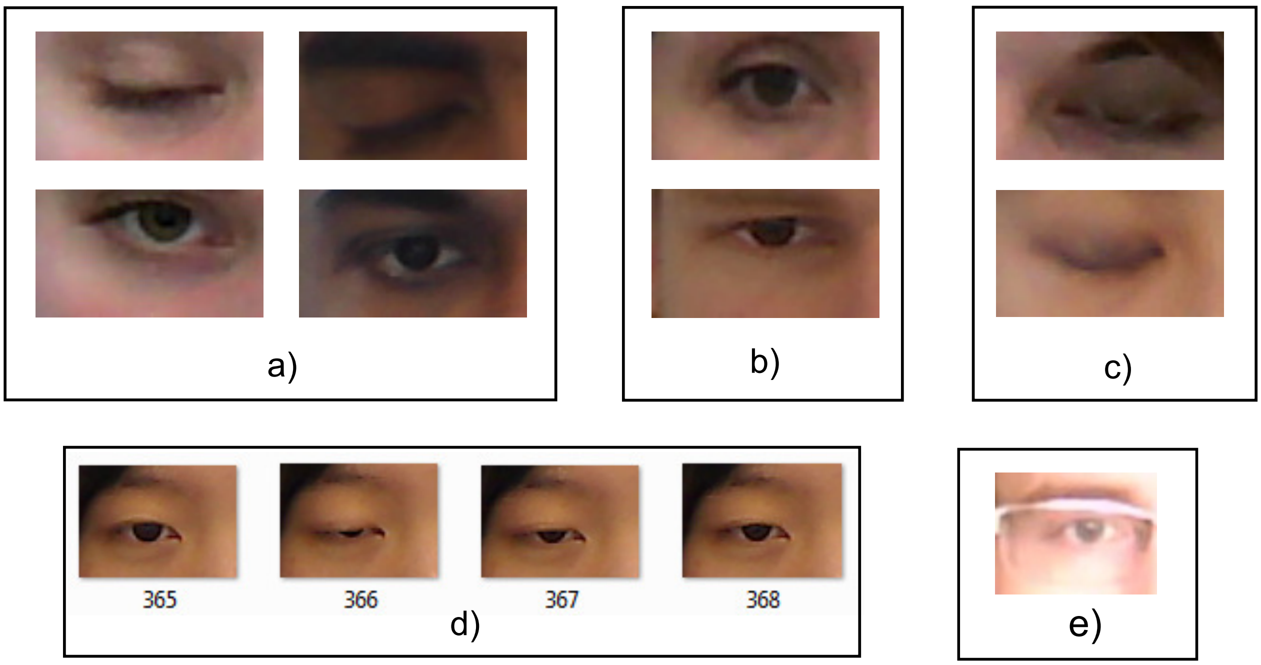


Figure 1.1: Exemplary frames with eyes.

1.3 Approach and main contribution

In order to resolve this problem, I have read many articles to become more familiar with algorithms already created. Later I created a database with short videos in which people are blinking. My next step was to extract every frame from videos only with the Region Of Interest (ROI) - eyes. Afterwards I made the golden standard and transcriptions which contain information if somebody in my videos has open, semi-closed and closed eyes. Then, I implemented two algorithms in Matlab: K-Nearest Neighbors and Support Vector Machines to check how good those methods are to classify open and closed eyes. On the basis of the transcriptions, I have created training sets and test sets to check how those methods work.

1.4 Outline of the Thesis

In Chapter 2 I will be focusing on the physiology of the human eye. In Chapter 3 I will be focusing on the basic knowledge from Pattern Recognition which is necessary to understand this thesis. In Chapter 4 I will presenting methods which I tried to to resolved this problem. In Chapter 5 I will showing the results from Chapter Image Classification and a experiment which I made. In the last Chapter I will be evaluating what I learnt and I commenting on my work.

Chapter 2

Physiology

Ophthalmology is a scientific field which is concerned with eyes. The eye is a very complex organ. Doctors have to study for many years to be able to treat people's eyes. In medical literature, we can find an expression *eyelids movement*. An expression *eye blinking* is found very rarely.

2.1 Why do people blink?

The cornea is one of the organs with the best blood supply. It must be constantly moistened, otherwise what we see becomes blurred. The lacrimal gland produces tears which lubricate eyes. Tears help us clear and to defend eyes against pathogens. When we blink, tears are piped to the nasolacrimal duct. When something irritates eyes, tears are secreted very fast. Then we blink to pipe them [10].

2.2 The Anatomy of the Human Eye

An eyelid is a thin fold of skin that covers an eyeball. An eye closes when a muscle which is called the *eyelid muscle* works. A palpebral fissue is the distance between eyelids. It is genetically conditioned. The Europeans or people of African descent have wider palpebral fisseus than the Chinese or Japanese people. In Walsh's opinion, standard palpebral fissues are $22 - 30mm$ high and $12 - 15mm$ wide. Palapal fissues of newborn babies are $18mm$ high and their shape is almost round.

According to Walsh, correct eye blinks could be divided into:

- Involuntarily movements, for example: to correct, depending on the cornea, as an effect of an excessively bright light, self-defence.

- Controlled movements, for example: squinted eyes.

2.3 Eye Blinking Rate

The frequency of correct eye blinks is dependent on the sympathetic nervous system. Adler noted that time between blinks takes 2.8s for men and 4s for women [10]. Time of blink takes from 0.3s to 0.4s and depends on emotions and what people see. The time when eyes are closed during blinking is called *pointer of obfuscation* [10].

2.4 Diseases of the Eye

I would like to present a few diseases connected to eye blinking [15]. The most common are:

- Ptosis. Its symptom is falling of the upper eyelid. It results from the damage caused to the muscle of eyelid (Figure 2.1 a)).
- Eyelid retraction. Its result is raising of the upper eyelid above sclera (Figure 2.1 b)).
- Apraxia. It is characterised by very slow opening of the upper eyelid to a normal position. It could take even thirty seconds.

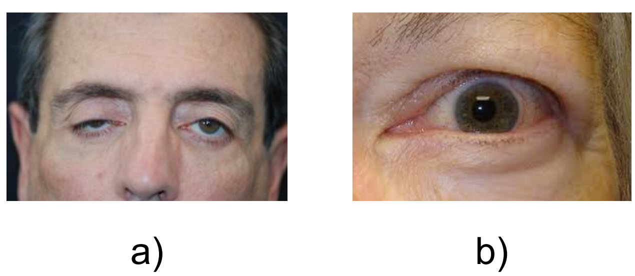


Figure 2.1: Exemplary illnesses.

Chapter 3

Image Classification

Pattern recognition is a scientific field where algorithms are used to classify images, signals, sounds and videos into appropriate groups. For example, if we want to recognise whether a photo was taken at night, we can check how many dark pixels are in this image. If the number of dark pixels in the photo is more than seventy percent, it can be assumed that this picture was taken at night. Initially, people were trying to classify objects with the help of statistics (like in the previous example). At present, we have many more mechanisms to classify objects. Currently, pattern recognition is being used everywhere: in medicine - to detect tumor, on the Facebook website - to detect faces, in security - to confirm access to some places by using eye scanning, all characters (letter or number).

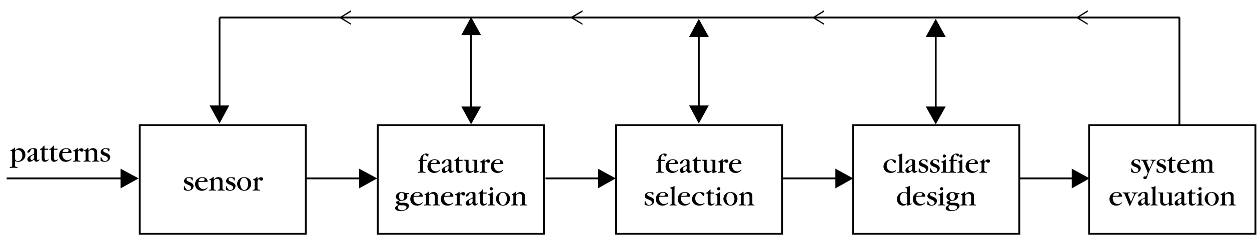


Figure 3.1: Main steps in the pattern recognition system.

To classify images correctly, it is important to do it in a right way. This process consists of five steps (Figure 3.1) [12]. We cannot skip any of these. In next paragraphs I will provide more information about each step.

3.1 Visual Sensors

In my bachelor work I used a built-in computer camera to detect eye blinking. In articles which I read about eye blinking detection the scientists also used web cameras which were built in laptops.

Figure 3.2: Comparison of faces from two different cameras.

I would like to compare images of the same situations which were taken with two different cameras. The first one is an integrated web camera from Acer Aspire 5750G and the second is a normal camera Panasonic LUMIX TZ18 (Figure 3.2).

The resolution of the webcam is 640 x 480 pixels and the LUMIX camera is 1280 x 720 pixels. The images shown above were taken after one of the preprocessing steps - face detection (Section 4.1). As it can be seen, the contrast and saturation is much better on the photo from the LUMIX camera. It shows that a problem might be easier to resolve if a normal camera was used. Additionally, I would like to mention that the resolution of a face in the webcam is 230 x 230 pixels and on the LUMIX camera 300 x 300 pixels.

After the second step of preprocessing - eyes detection (Section 4.1), my results of ROI are like in Figure 3.3.

It is easier to classify images if they are bigger [11].

I had one major problem with cameras (both build-in and LUMIX). Sometimes cameras made movies with fifteen frames. In technical documentation of these items it was only noted that cameras made videos with thirty frames per second (FPS). In file transcription there was information that the rate of the video was thirty FPS. It seemed obvious to me that I must have some problem with my program which extracted frames. I was looking for a solution on the Internet. Finally, I managed to find a program which extracted each frame. The program showed me that every second frame is duplicated. So I was reassured that my program works correctly. However, I was wondering why sometimes the rate of my

Figure 3.3: Comparison of the eyes from two different cameras.

movies was fifteen FPS. I started searching over through the Internet again, but I did not find anything satisfactory. I decided to make different videos in different environment and to find a feature which determined the frame rate of the video. I discovered a correlation. If a video is bright, the frame rate is thirty FPS and if a film is dark, the frame rate is fifteen FPS.

3.2 Image Features

In order to classify a part of a photo it is necessary to get certain features of it. They are the base for object classification. The most well-known and important visual feature extractions are:

Histogram It shows the distribution of data in a chart. It contains columns which are placed in a coordinate system. The axis X shows a range of a certain class and the axis Y shows how many times this class appears on objects. In image processing it is the most famous method of graphic data representation.

In a more general mathematical sense, a histogram is a function m_i that counts the number of observations that fall into each of the disjoint categories (known as bins), whereas the graph of a histogram is merely one way to represent a histogram. Therefore, if we let n be the total number of observations and k be the total number of bins, the histogram m_i meets the following conditions [2]:

$$n = \sum_{i=1}^{k} m_i \tag{3.1}$$

In Figure 3.4 there are exemplary histograms. I made histograms for a closed and open eye. On the basis of these histograms we can recognise that the eye is closed when values from red channel histogram are bigger than one hundred and from a grayscale image bigger then fifty. Unfortunately, it does not work for the whole data set. It functions only for these two images.

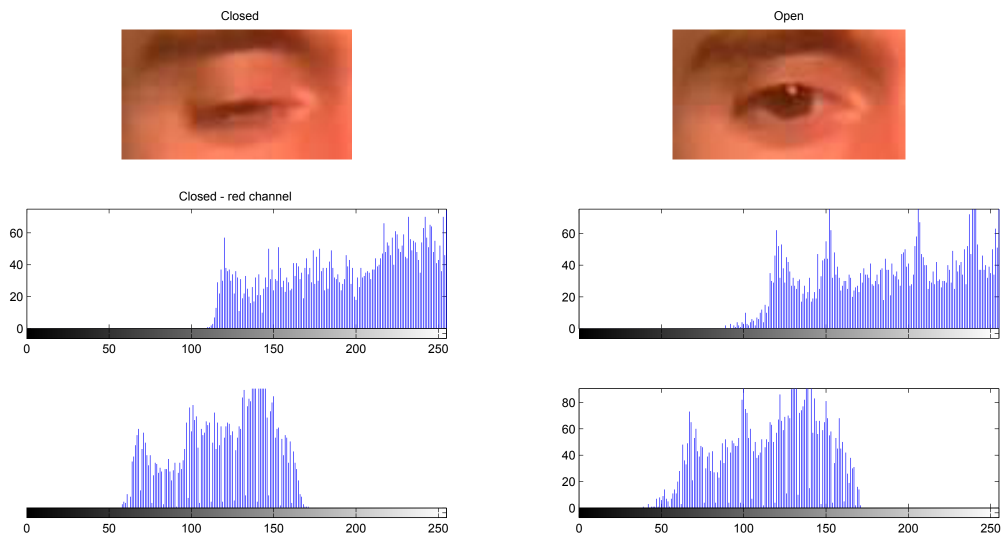


Figure 3.4: Exemplary histograms.

Fourier transformation Fourier transformation helps us to represent signals or images on a frequency scale. Fourier transformation of images is a very popular method nowadays. People who work in graphics use Fourier transformation to see things which are not easy to detect in an original image, for example disturbance on the picture. It is also very useful to create fast methods to image filtering.

The formula for Fourier transformation for image is the following [3]:

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi(ux+vy)}dxdy \tag{3.2}$$

In this case, where we use the Fourier transformation as the basis, it is hard to say if the eye is closed or open Figure 3.5.

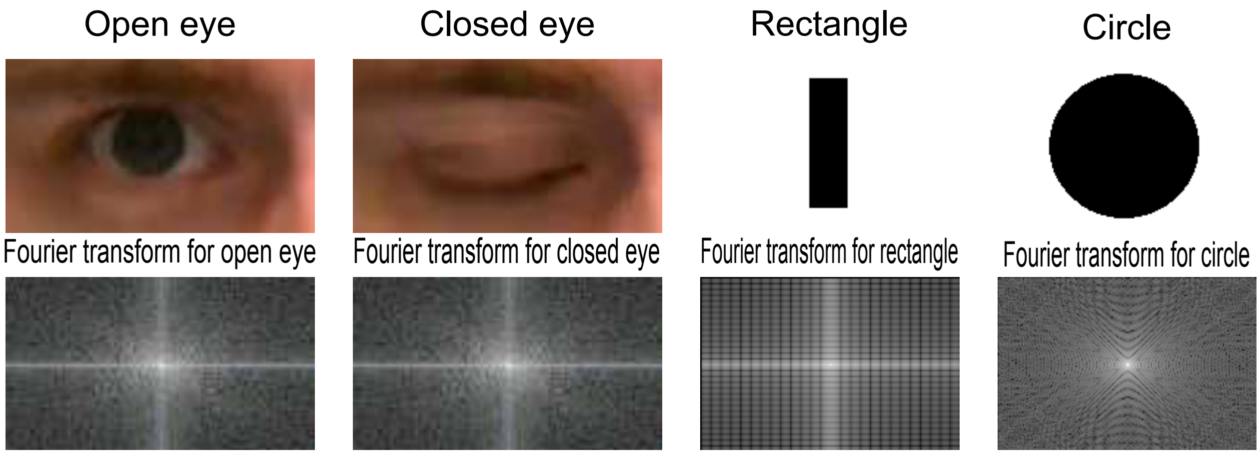


Figure 3.5: Exemplary images after Fourier Transformation.

Scale-Invariant Feature Transform (SIFT) Currently, it is one of the most fashionable features for multimedia retrieval. This algorithm is used for object recognition, navigation, 3D modelling, video tracking as well as in many others fields.

SIFT is looking for edge shapes in a local region, reasonably invariant to scaling, rotation, viewpoints changes and an illumination changes [5].

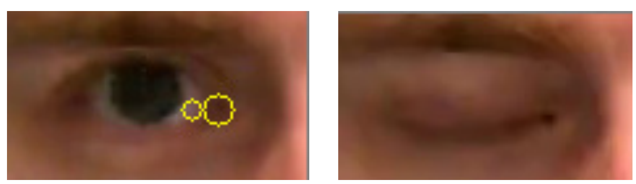


Figure 3.6: Exemplary SIFTs for an open and a closed eyes.

As we can see in Figure 3.6 for an opened eye algorithm found two features, but for a closed eye none. This method is useful for bigger images.

3.3 Classification

Neural Networks: One of the most popular classification techniques are neural networks. This algorithm is used in many fields, for example: medicine, price prediction, at airports to check if there are no dangerous items in the baggage, etc. A neural network is built from neurons. If we want to obtain results from this algorithm we put features to the program. These features go to neurons and do mathematical equalisation. Later, the outcome of this equalisation can go to the other neurons where other mathematical calculations can be done. An exemplary neuron is shown in Figure 3.7.

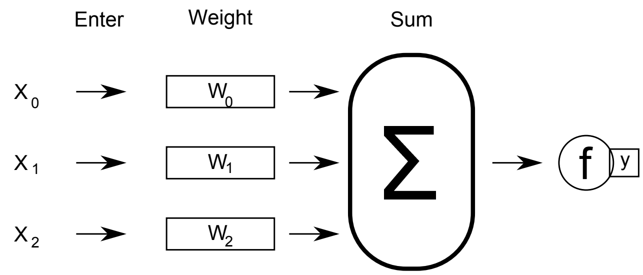


Figure 3.7: An exemplary neuron.

This neuron has three input values; one element which processes data: sums up input values and activates function; and one output value.

$$y = f(\sum_{i=0}^{n-1} w_i x_i) \tag{3.3}$$

where:

n - number of input values
y - output from neuron
x_i - i-th input
w_i - weight i-th input
$f()$ - activating function

Let us say that we have a special case in which we want to recognise that an eye is closed or open and we have only one neuron. An open eye is described by values: $2; 5; 5; 16$ and a close eye by: $-1, 0, 5, 2$. The weight of a neuron is 2. The neuron is activated when on exist the sum is bigger that 30. The calculation for the $c$losed and the $o$pen eye.

$$c = -1 \cdot 2 + 0 \cdot 2 + 5 \cdot 2 + 2 \cdot 2 = 12 \tag{3.4}$$

$$o = 2 \cdot 2 + 5 \cdot 2 + 5 \cdot 2 + 16 \cdot 2 = 56 \tag{3.5}$$

In the first case $56 > 30$ the neuron is activated on exit and its value equals 1. In the second case $12 < 30$ the output of neuron is not activated, so on exit of the neuron the value amounts for 0 [14].

Bayes Decision Theory From the probability theory we know that:

$$p(a/b) = \frac{p(a)p(b/a)}{p(b)} \tag{3.6}$$

On the basis of this equation Bayes created a method to check how to classify certain situations:

$$p(c_i/x) = \frac{P(c_i)p(x/c_i)}{p(x)} \tag{3.7}$$

where:
c_i is a hypothesis,
c is data (facts, observation) which can influence the classification of the hypothesis,
$P(c_i$ is a probability *a priori* of hypothesis h,
$p(c_i/x)$ is a probability *a posteriori* of hypothesis h.

Let us imagine that we have two baskets. In the first basket there are twenty red balls and twenty black balls. In the second basket there are thirty red balls and ten black balls. Let us choose a basket at random and consequently a ball from it. Before we say what color is a ball, we can say that the probability of choosing one of this basket is 0.5. It is *a priori* knowledge. Subsequently, we choose a red ball, it is *a posteriori* knowledge. Now we can say from which basket we choose this red ball.

The first basket:

$$p(A/\mathbf{r}) = \frac{P(A)p(\mathbf{r}/A)}{P(A)p(r/A) + P(B)p(\mathbf{r}/B)} = \frac{0.5 \cdot 0.5}{0.5 \cdot 0.5 + 0.5 \cdot 0.75 = 0.4} \tag{3.8}$$

The second basket:

$$p(B/\mathbf{r}) = \frac{P(B)p(\mathbf{r}/B)}{P(A)p(\mathbf{r}/A) + P(B)p(\mathbf{r}/B)} = \frac{0.5 \cdot 0.75}{0.5 \cdot 0.5 + 0.5 \cdot 0.75 = 0.6} \tag{3.9}$$

We can assume that the red ball was taken from the second basket [4].

Support Vector Machine This algorithm was invented by Vladimir N. Vapnik in 1995. The training examples are vectors in a measurement space. Each of them has a label, which says to which class this feature belongs. The main idea is to find a borderline which separates different classes $1, -1$ Let us say that we have situation like in Figure 3.8. Red circles are from the first class, blue squares are from the second class. A cross is a new feature vector, which we want to classify.

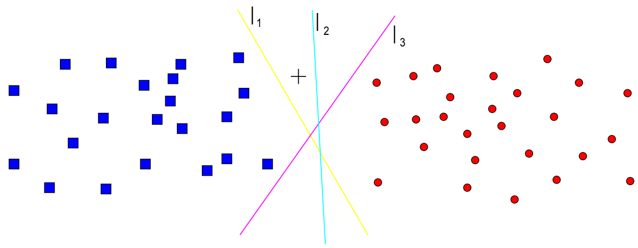


Figure 3.8: An example showing the idea of Support Vector Machine.

Lines l_1, l_2, l_3 are exemplary borders which can separate classes. Then, if we use l_1 as a border, the cross will belong to first class. For l_2 and l_3 the cross will belong to the second class.

In order to find a correct border, we need to find a decision boundary g($\mathbf{x}$):

$$g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b \tag{3.10}$$

The distance between the line and red circles or blue squares is called a margin. If the margin is bigger, the algorithm has better robust [12].

k-Nearest Neighbors algorithm This algorithm is a non-parametric method for classification. In the first step of this algorithm we use training examples to teach it how to recognise objects. The training examples are vectors in a multidimensional feature space. Each of them has a label which indicates to which class this feature belongs.

Let us assume that we have a situation like in Figure 3.9. Red circles are from the first class, blue squares are from the second class. The cross is a new feature vector which we want to classify.

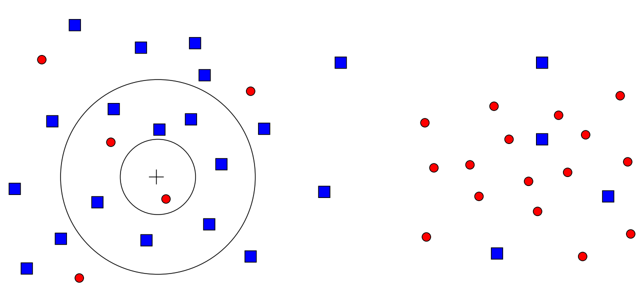


Figure 3.9: An example showing the idea of k-NN.

K is a parameter. It says how many nearest points should be taken into consideration in order to check to which class the new feature belongs. In our example k equals 1. Then our algorithm says that the cross belongs to the first class (red circle). If k equals 9, our cross belongs to the second class (blue square). If k is bigger, the algorithm is more robust [12].

3.4 Evaluation

To communicate results in a faster way we can use terms which will be represented in this section.

For example, we have a program to recognise closed eyes from an image. Our test database has one thousand pictures with closed eyes and one thousand with open eyes, two thousands images in total. Our program is able to recognise nine hundred and fifty from them as correct (closed eye), though fifty of them would be incorrect (open eyes).

Terms:

	Close eye	Open eye
Test outcome positive	True Positive (TP)	False Positive (FP)
Test outcome negative	False Negative (FN)	True Negative (TN)

Table 3.1: Useful terms

Recall - R This term says how many percent of correct answers the algorithm gives. In this case our program gives nine hundred correct answers from one thousand images. Then:

$$R = \frac{TP}{TP + FN} \cdot 100\% = \frac{900}{1000} \cdot 100\% = 90\% \tag{3.11}$$

Precision - P

$$P = \frac{TP}{TP + FP} \cdot 100\% = \frac{900}{900 + 100} \cdot 100\% = 90\% \tag{3.12}$$

Accuracy - A

$$A = \frac{TP + TN}{TP + FP + FN + TN} \cdot 100\% = \frac{900 + 50}{2000} \cdot 100\% = 47.5\% \tag{3.13}$$

Fallout - F

$$F = \frac{FP}{FP + TN} \cdot 100\% = \frac{100}{100 + 900} \cdot 100\% = 10\% \tag{3.14}$$

Chapter 4

Frame Classification for Eye Blink Detection

First of all, I decided that the most important thing for me to do was to detect an eye in an image. (Section 4.1). Only later I could focus on eye blinking detection. I decided to try the most intuitive ways for me. I write more on these methods in Section 4.2. The methods that I found worked best are described in Section 4.3.

4.1 Region of Interest

The region of interest determined with the method described in this section is used for further processing in Sections 4.2 and 4.3.

Face Localisation: In order to detect a face, I used an algorithm which is already implemented in OpenCV. This method makes uses of Viola - Jones object detection framework [1].

On the basis of an image, the algorithm calculates characteristic features which are created on Haara's wavelet base. Each feature is a single value obtained by subtracting the sum of pixels under white rectangles from the sum of pixels under black rectangles.

In order to create features, the algorithm uses an integral image. The value of this image in point (x,y) equals

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \tag{4.1}$$

where:

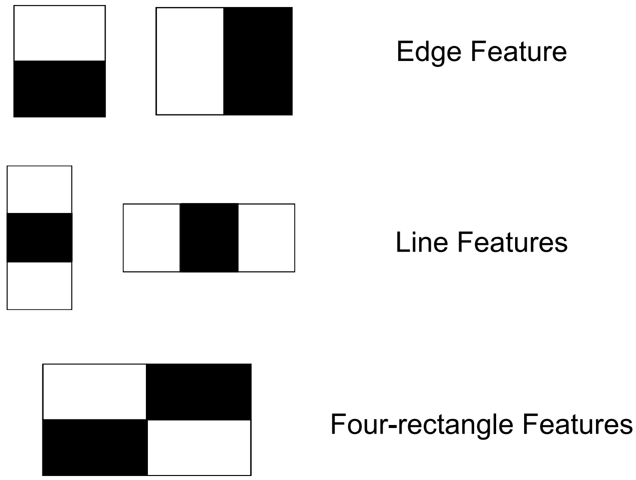


Figure 4.1: Exemplary characteristic features.

ii - integral image,
i - original image.

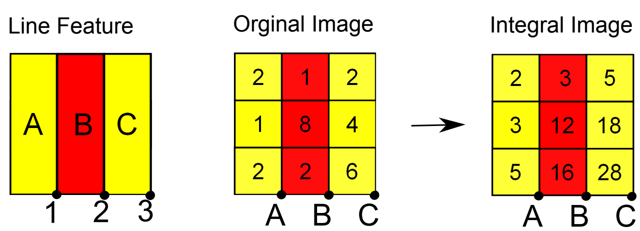


Figure 4.2: Exemplary characteristic features.

To calculate value for area:

$$\begin{aligned} I_C &= ii(x_3, y_3) - ii(x_2, y_2) = 28 - 16 = 12 \\ I_B &= ii(x_2, y_2) - ii(x_1, y_1) = 16 - 5 = 11 \\ I_A &= ii(x_1, y_1) = 5 \end{aligned} \tag{4.2}$$

Let us calculate Haare's feature for the example from Figure 4.2.

$$I_A + I_C - I_B = 5 + 15 - 11 = 9 \tag{4.3}$$

Viola and Jones used the AdaBoost algorithm to classify presence of a face in an image. It is necessary to have a training set with faces and other training set without faces. For each feature the program will classify faces positively or negatively. If some feature classifies an image correctly, the weight of this feature gets bigger [1].

Region of Interest (ROI): In this step, two rectangular regions of interest ($\boldsymbol{R}_{\text{left}}$, $\boldsymbol{R}_{\text{right}}$) corresponding to the left and to the right eye are determined based on the face region $\boldsymbol{I}$ computed in the last step. We consciously made the regions large enough so that they also include the eyebrows. The reason for this is the diversity of face sizes of our subjects. The eye regions can be expressed with the following matrices:

$$\boldsymbol{R}_{\text{left}} = \begin{pmatrix} \boldsymbol{I}(H-7H', W-27W') & \cdots & \boldsymbol{I}(H-7H', W-16W') \\ \vdots & \ddots & \vdots \\ \boldsymbol{I}(H-5H', W-27W') & \cdots & \boldsymbol{I}(H-5H', W-16W') \end{pmatrix} \tag{4.4}$$

$$\boldsymbol{R}_{\text{right}} = \begin{pmatrix} \boldsymbol{I}(H-7H', W-14W') & \cdots & \boldsymbol{I}(H-7H', W-7W') \\ \vdots & \ddots & \vdots \\ \boldsymbol{I}(H-5H', W-14W') & \cdots & \boldsymbol{I}(H-5H', W-7W') \end{pmatrix} , \tag{4.5}$$

where H and W denote the height and the width of $\boldsymbol{I}$ respectively. We consider $H' = \frac{H}{10}$ and $W' = \frac{W}{31}$, because these are the smallest values which precisely define the desired part (eye and eyebrow). The following processing steps are identical for the right and left eyes, therefore, we consider just one region of interest and denote it by the matrix $\boldsymbol{R}$.

Graphical representation of this method is on Figure 4.3.

4.2 Methods Based on Pixels and Histograms

4.2.1 Pixel-Based Movement Detection

Initially, I used the algorithm to detect a face and later to detect eyes. The main idea behind that was to take the first frame as a background. In the first frame eyes must be open. Every next frame is compared with the first one and the algorithm checks how many pixels have been changed (Figure 4.4). The coordinates of the area of interest of eyes are still the same, so the size of ROI also remains unchanged.

The following problem connected with using this method can appear: if somebody moves their head a little bit on the video, the results are incorrect, Figure 4.5. The yellow box is a ROI. As we can see, every pixel has a totally different value in this ROI.

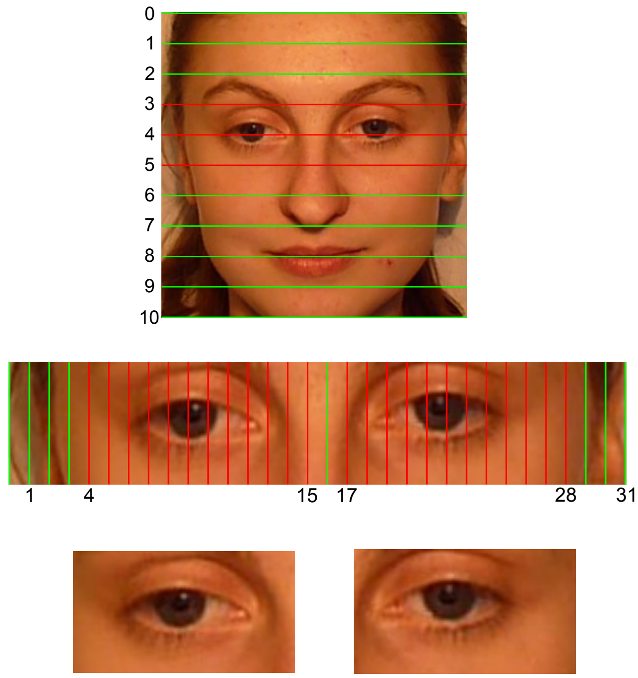


Figure 4.3: Detecting eyes.

4.2.2 Classification Based on Thresholding

In the beginning, I made a grayscale of this image. Each pixel has three values from three different channels: red, green and blue. To calculate a new value of a gray pixel I used the following equation:

$$N_{i,j} = 0.2126 \cdot R_{i,j} + 0.7152 \cdot G_{i,j} + 0.0722 \cdot B_{i,j} \tag{4.6}$$

where:

N - new value of a pixel in a gray scale image.

i, j - coordinates of a pixel in an image.

R, G, B - Values of red, green and blue channels from the original image.

Subsequently, I followed the same steps as in movement detection Subsection 4.2.1: getting ROI with eyes. What I did next, was to check how many bright pixels were in an open and a closed eye. The results of this operation can be seen in Figure 4.6.

The green color in this figure indicates the brightest pixels, red shows the darkest pixels while yellow marks values between red and green. From what can be seen, the brightest region is not in the eyeball but in the right part of an image, where there is a nose. We

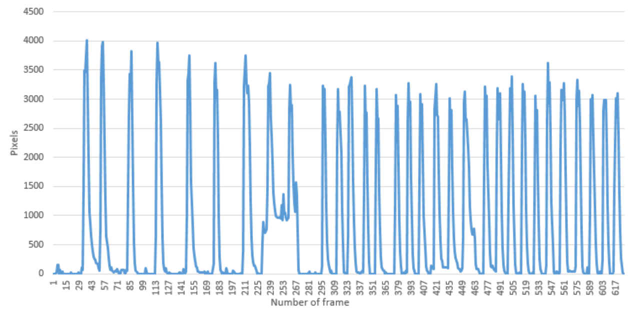


Figure 4.4: Exemplary results from detecting movement in video.

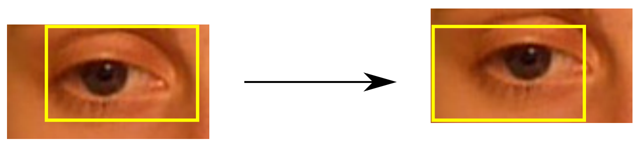


Figure 4.5: Problem with movements.

can see very clearly, that a pupil has a very dark region. I decided to write an algorithm which works like edge detection. The main idea was to find the center of an eye by finding adequate pixels which are next to each other and have appropriate values. In other words, at the beginning I had to do the thresholding. The problem was to find the threshold value. Hard coding value does not work because images can have different illumination.

$$O_i = (I_i - I_{\min}) \cdot \frac{O_{\max}}{I_{\max} - I_{\min}} \tag{4.7}$$

where:
O - new value of a pixel after normalization,
i - coordinates of a pixel in an image,
I - the original image,
$I_{\min}$ - minimum value of a pixel in the original image,
$I_{\max}$ - maximum value of a pixel in the original image,
$O_{\max}$ - new maximum value in the image Norm,

Even after normalization, hard code number does not work for a big number of images.

As we can see in Figure 4.7 even when a threshold has value +/- 7 the image is completely different. In my opinion, for this example the best one is the threshold in level 43. I tried to

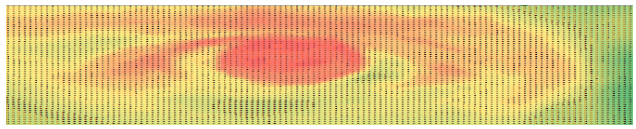


Figure 4.6: Grayscale of a left eye.

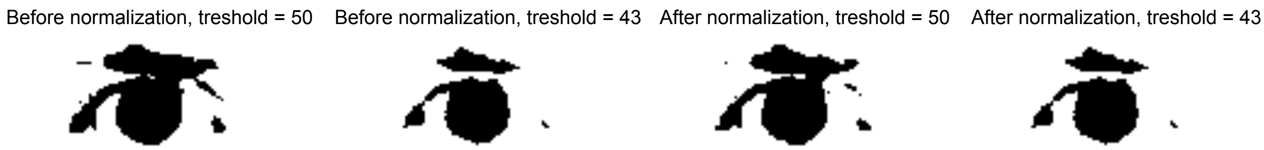


Figure 4.7: Tresholding in an original image and after normalization.

get value 43 from this image by using different statistical methods (mean, mode, histograms, variance). However, it turned out to be impossible.

4.2.3 Histogram-Based Classification

This idea is based on histograms of a channel of colors and a grayscale image. These histograms were features of images. After getting ROI with eyes,, I created two test datasets and two training datasets. These datasets contained images with open and closed eyes. In the next step I implemented the following algorithms: Support Vector Machine and k-Nearest Neighbors. The range of bins in histograms varied from two to two hundred fifty-five. For all the images I created new features for every number of bins. Subsequently, I taught SVM and kNN algorithms using these attributes. Later, I checked the recall and I calculated average values for each case. The results are as follows:

Open eye	red channel	green channel	blue channel	grayscale
KNN	58%	66%	62%	66%
SVM	58%	51%	57%	52%

Table 4.1: Classification rate for an open eye in different cases.

As we can see the results are bad. Red channel has the best results. In both cases: for open and closed eyes for kNN and SVM the recall is bigger that fifty seven percent.

Closed eye	red channel	green channel	blue channel	grayscale
KNN	57%	45%	45%	47%
SVM	57%	59%	47%	57%

Table 4.2: Classification rate for a closed eye in different cases.

Unfortunately, the outcome was only a little bit improved than the coin flip. It is necessary to make some better preprocessing of an image to obtain better results.

4.3 Robust Shape-Based Classification Method

4.3.1 Eye Segmentation

Below, we describe our method for eye segmentation. It consists of multiple steps depicted in Figure 4.8.

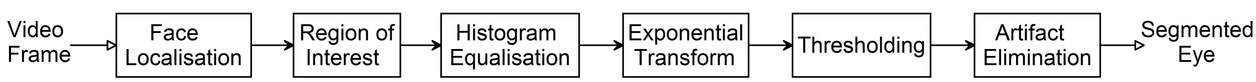


Figure 4.8: Processing steps for eye segmentation in video frames

For each video frame as an input, the method outputs a binary image with a segmented eye.

Histogram Equalisation: The goal of this step is the contrast enhancement in $\boldsymbol{R}_{\Phi\times\Lambda} = (r(\phi,\lambda))$. For this, we use histogram equalisation given by:

$$\begin{aligned} \boldsymbol{s} &= (s_1,\ldots,s_{256})^{\mathrm{T}} &\quad \text{with} \quad & \boldsymbol{s} = \tfrac{\boldsymbol{h}(\boldsymbol{R})}{\Phi\Lambda} \\ \boldsymbol{v} &= (v_1,\ldots,v_{256})^{\mathrm{T}} &\quad \text{with} \quad & v_j = \mathrm{floor}(256\sum_{i=1}^{j} s_i) \quad , \\ \boldsymbol{Q}_{\Phi\times\Lambda} &= (q(\phi,\lambda)) &\quad \text{with} \quad & q(\phi,\lambda) = v_{r(\phi,\lambda)+1} \end{aligned} \tag{4.8}$$

where $\boldsymbol{h}(\boldsymbol{R})$ represents the histogram of $\boldsymbol{R}$ with 256 bins and $\Phi = 2H'$, $\Lambda = 6W'$ represents the height and width of $\boldsymbol{R}$ respectively.

Exponential Transform: In order to improve the results of the thresholding conducted in the next step, we increased the difference amongst the intensity of pixels in $\boldsymbol{Q}$ according to the method presented in [9]. For this, the exponential transformation of each pixel in $\boldsymbol{Q}$ is computed as follows:

$$\boldsymbol{E}_{\Phi\times\Lambda} = (e(\phi,\lambda)) \qquad \text{with} \qquad e(\phi,\lambda) = \exp(q(\phi,\lambda)) \quad . \tag{4.9}$$

Thresholding: In this step we binarise the image $\boldsymbol{Q}$ by thresholding the values in the matrix $\boldsymbol{E}$ which results in:

$$\boldsymbol{T}_{\Phi\times\Omega} = (t(\phi,\omega)) \qquad \text{with} \qquad t(\phi,\omega) = \begin{cases} 0, & \text{if} \quad e(\phi,\omega) < \varepsilon \\ 255, & \text{otherwise} \end{cases} \quad , \tag{4.10}$$

where ε is the average element value of $\boldsymbol{E}$.

Artefact Elimination: The thresholded image $\boldsymbol{T}$ usually contains not only the desired elements (eyebrow and eyelashes for a closed eye; eyebrow and iris for an open eye), but also irrelevant parts and artefacts like hair, circles beneath the eye, or unnormalised lighting effects that are eliminated in this step. In addition to this, the eyebrow is also eliminated in this step, since it is not needed for further processing. The relevant image segments (the iris and the eyelashes) are supposed to be in the centre of the image $\boldsymbol{T}$. The elimination method proposed in this paper selects the nearest bloc of pixels to the centre of $\boldsymbol{T}$ as shown in Figure 4.9(e) and 4.10(e).

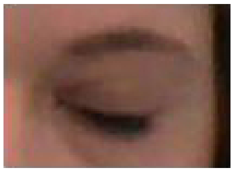
(a) Original Region of Interest (b) Result of Histogram Equalisation (c) Red Channel (d) Thresholding

(e) Result of Artefact Elimination

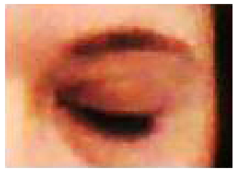
Figure 4.9: Opened eye segmentation

4.3.2 Feature Extraction

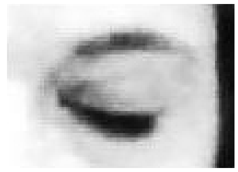
In this section we explain how we computed a three dimensional feature vector $\boldsymbol{c} = (c_1, c_2, c_3)^{\mathrm{T}}$ from the image $\boldsymbol{F}$ obtained in Subsection 4.3.1. Its first element c_1 is determined based on the outcome of a circle detection in the area of the segmented eye (circle detected $\rightarrow c_1 = 1$, circle not detected $\rightarrow c_1 = 0$). The second element c_2 is equal to the ratio of width to height



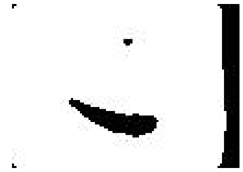
(a) Original Region of Interest



(b) Result of Histogram Equalisation



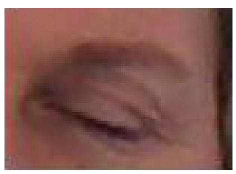
(c) Red channel



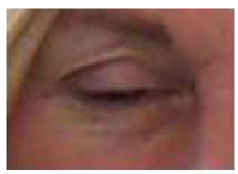
(d) Thresholding

(e) Result of Artefact Elimination

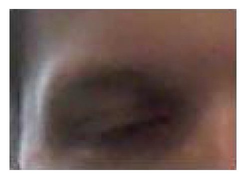
Figure 4.10: Closed eye segmentation



(a) Example 1



(b) Example 2



(c) Example 3

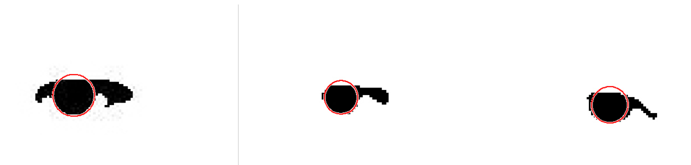
Figure 4.11: Example Images with Wrong Iris Detection

of the minimum bounding box enclosing the segmented eye. The third element c_3 is just the size of the segmented eye given in pixels.

According to the statistical analysis of the feature vector values obtained for different images, the first feature c_1 possesses the strongest discriminative power for distinguishing open and closed eyes. This is due to the iris visible in images of an open eye, that can only very rarely be found in closed eye images. However, the circle detection algorithm sometimes delivers wrong results, especially under drastically changing light conditions, while considering the diversity of ethnic groups and due to variation in people's age. Examples of such images can be seen in Figures 4.11(a), 4.11(b) and 4.11(c).

For computing c_1 a circle detection method with the sensitivity equal to 0.83, the minimum radius of six pixels and the maximum radius of twelve pixels have both been applied. An example of the results of this process can be seen in Figures 4.12(a), 4.12(b), and 4.12(c).

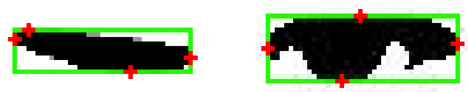
As mentioned above, the second element of the feature vector c_2 represents the quotient of the width to the height of the shape which corresponds to the iris or eyelashes in the image



(a) Example 1 (b) Example 2 (c) Example 3

Figure 4.12: Example Circle Detection Results

$\boldsymbol{F}$ as it is shown in Figures 4.13(a) and 4.3.2.



(a) Closed Eye (b) Open Eye

Figure 4.13: Quotient of the width to the length.

A normalised size of the segmented eye is the last element of feature vector c_3. More specifically, it corresponds to the fraction of the number of eye pixels against all pixels in the image:

$$c_3 = \frac{\sum_{i=1}^{\Phi} \sum_{j=1}^{\Lambda} (\boldsymbol{F})_{i,j} = 0}{\sum_{i=1}^{\Phi} \sum_{j=1}^{\Lambda} (\boldsymbol{F})_{i,j}} \tag{4.11}$$

After normalisation, all features in $\boldsymbol{c}$ values come from different ranges. For example c_2 is between 0.5 and 0.95, while c_3 takes values between 0.05 and 0.21. This makes c_3 negligible compared to c_1 and c_2. To equilibrate the feature vectors, it is necessary to normalise their values into the range $[0, 1]$. This is done as follows:

$$\begin{aligned} \boldsymbol{P}_{M\times N} &= (p(m,n)) \quad \text{with} \quad \boldsymbol{P} = ((\boldsymbol{c})_1, (\boldsymbol{c})_2, \ldots, (\boldsymbol{c})_N) \\ \boldsymbol{P}^*_{M\times N} &= (p^*(m,n)) \quad \text{with} \quad p^*(m,n) = \frac{p(m,n) - \min\limits_{1\leq n\prime\leq N}(p(m,n\prime))}{\max\limits_{1\leq n\prime\leq N}(p(m,n\prime)) - \min\limits_{1\leq n\prime\leq N}(p(m,n\prime))}, \end{aligned} \tag{4.12}$$

where N represents the number of feature vectors in the training set. The same normalisation procedure is used in the testing phase.

Chapter 5

Experiments and Results

In order to collect data with videos of eye blinking I did two experiments. People's reactions were very positive once they learnt I needed their help to write my BSc thesis. In Section 5.1 I showed how an algorithm from Section 4.3 works. In Subsection 5.2.1 I interpreted results from Section 5.1. Finally I wanted to focus on rate of blink base on Subsection 2.3.

5.1 Frame Classification

The main idea of the experiment was to check how often people blink when they see a boring movie and an action movie. I wanted to check this in two cases: when they do not know that I want to observe their eyes blinks and when they are informed of my intentions. Two of my friends helped me to conduct this experiment.

5.1.1 Video Dataset

We collected data at the University of Siegen, Germany. We constructed our work-place next to the exit from the building .

Fifty people contributed to this experiment. Each person watched four movies. While people were watching those short videos we were filming theirs faces. In the first step we informed people that we were collecting a database of human faces, because we were working on a face detection algorithm. We told them, that they would see two short videos. Each short movie was thirty seconds long. In the second step, we explained to them that the real theme of the research is eye blink detection. We asked them to watch another two videos, thirty seconds long each.

In the first step, people watched an action movie. It was a part of an advertisement for Top Gear program, with speeding, drifting, etc. In the second film there was only an elderly man who talked. After the first step, we showed people different movies, but all coming from the same category: an advertisement for Top Gear and an old man talking. The videos were played without sound.

	People informed	Video type
υ_1	no	action
υ_2	no	calm
υ_3	yes	action
υ_4	yes	calm

Table 5.1: Types of videos.

The database was created by using three cameras:

- The first web camera was built in a laptop. Resolution: 640 x 480 pixels, 30 FPS. The database contains 108 videos which were made using this camera (23 persons). Faces in videos have more than 200 x 200 pixels; frames are bright.
- The second web camera was also built in a laptop. Resolution: 640 x 480 pixels, 15 FPS. The database contains 56 videos which were made using this camera (13 persons). Faces in videos have more than 200 x 200 pixels; frames have a really good condition of brightness.
- The third camera was an external web camera. Resolution: 640 x 480 pixels, 15 FPS. Database contains 76 videos which were made using this camera (14 persons). Faces in videos have more than 140 x 140 pixels, frames have the best saturation of colors.

The laptops were placed on the table. People of average height had screens directly in front of their faces. We bent cameras for shorter and taller people to capture their whole face. There were white boards behind people which served as a background. We wanted to make sure that the algorithm of face detection would not have any problem with face detection. If somebody wore glasses we asked him or her to take off their glasses for the sake of this experiment.

5.1.2 Single Frame Classification

For single frame classification, we consider ω_1, ω_2, two classes which represent open and closed eyes respectively. To constitute these two classes we randomly chose segmented images from the set, 240 images for the training set which represents $\frac{2}{3}$, and 120 ($\frac{1}{3}$) images for testing,

Figure 5.1: Exemplary frames with faces from the first experiment.

	ω_1	ω_2
SVM	94.16%	92.5%
KNN	76.66%	75%

Table 5.2: Classification accuracy.

KNN is also used with the same training and testing set. The classification accuracy is shown in Table 5.2.

Our classification system is used to classify each eye separately, and from [8] we conclude that eye blinks are synchronized spontaneously, for this reason we decided to use the late fusion to get the final result. To do this step we needed to calculate the euclidean distances $d_{\text{left}}, d_{\text{right}}$ between the samples $x_k, x_{k'}$ (left and right eyes) and the SVM hyperplane $g(x)$, these distances are given by the orthogonal projection as shown in Equation 5.1, The final classification results (Table 5.3, Table 5.4) are decided according to the maximum distance.

$$d_{\text{left}} = \frac{(w^T x_k + w_0)}{||w||}, \qquad d_{\text{right}} = \frac{(w^T x_{k'} + w_0)}{||w||} \tag{5.1}$$

	Test Set	Left eye	Right eye	Fusion
SVM	60	95%	93.33%	95%
KNN	60	79.16%	74.16%	76.66%

Table 5.3: Final classification accuracy "Opened eye".

5.2 Eye Blinking Frequency Analysis

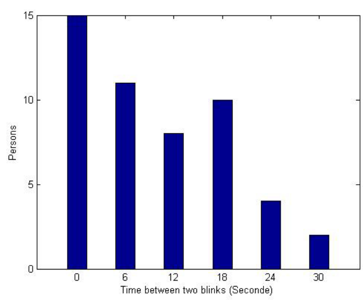
5.2.1 Eye Blinking Analysis

Figure 5.2 shows the number of persons against the average time between two blinks in υ_1, υ_2, υ_3 and υ_4, from these charts we see the similarities between υ_1, υ_3 and υ_2, υ_4. This means

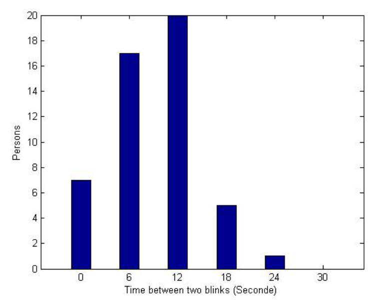
	Test Set	Left eye	Right eye	Fusion
SVM	60	95.83%	89.17%	94.16%
KNN	60	75.83%	74.17%	75%

Table 5.4: Final classification accuracy "Closed eye".

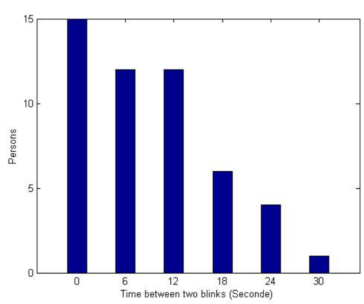
that informing people does not affect the number of blinks, unlike the type of video which was found to be very influential, υ_1, υ_3 charts show that 30% of people either don't blink at all or they blink once at most while watching action videos, this percentage does not exceed 14% in charts υ_2, υ_4. We conclude that people also blink faster when they are watching a calm video, more than 70% of people blink every 6-12 seconds, and only 30% of them blink faster during action videos.



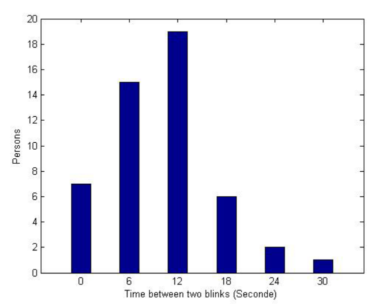
(a) υ_1



(b) υ_2



(c) υ_3



(d) υ_4

Figure 5.2: Histogram between two blinks

This result demonstrates that people try to keep their eyes open during action videos in order not to miss any part of a sequence, which is not so important for calm videos, where people blinked frequently to keep their eyes moist, especially while watching since their eyes are influenced by bright light.

In general, the average number of times a normal person blinks per minute is between 15 and 20, which means one blink every 3-4 seconds. This average increases when the person is watching TV because of there being a lot of bright light whilst they are trying to keep their eyes open during an important scene or a video featuring a lot of action for which the person has to concentrate [7].

5.2.2 The Rate at Which Subjects Blink

I collected this data in a students dormitory in Gliwice, Poland.

Twenty nine people participated in the experiment. I told them that I worked on face detection algorithms for my bachelor work. I asked each person to look at one point on the screen, which looked liked in Figure 5.3 and not to move their head for one minute.

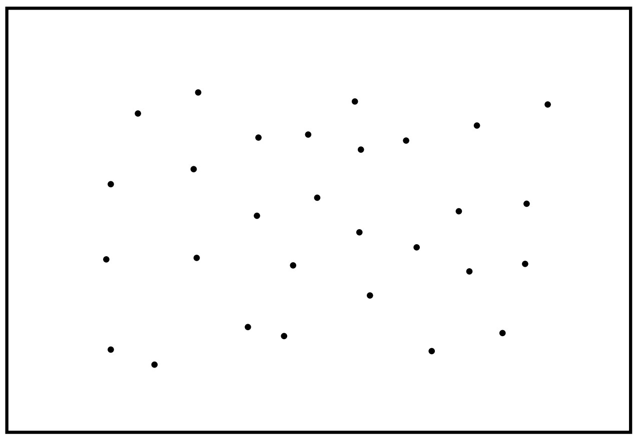


Figure 5.3: The white screen with dots from experiment.

The people were sitting on a chair. In front of their faces there was a laptop. The lamp was shining on theirs faces. Behind each person there was a big white piece of paper. I made videos by using an integrated web camera from Acer Aspire 5750G and a normal camera Panasonic LUMIX TZ18 (Section 3.1).

In the next step, I applied the gold standard to count how many times each person blinked. On the basis of my manual work, I created histograms which show how often blinks occur in this experiment Figure 5.5.

As we can see fifteen blinks dominate. If women and the person who blinked fifty times are removed from the database, the average number of blinks is equal to 12.4. The blink frequency equals $60/12.4 = 4.83$ blinks per minute. As it was mentioned in Section 2.3 men blink every 2.8 seconds.

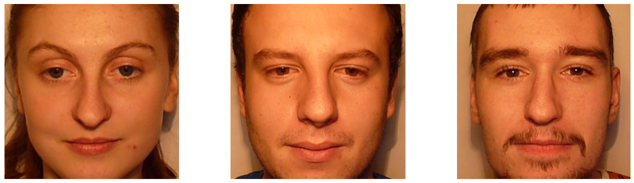


Figure 5.4: Exemplary frames with faces from the second experiment (LUMIX camera).

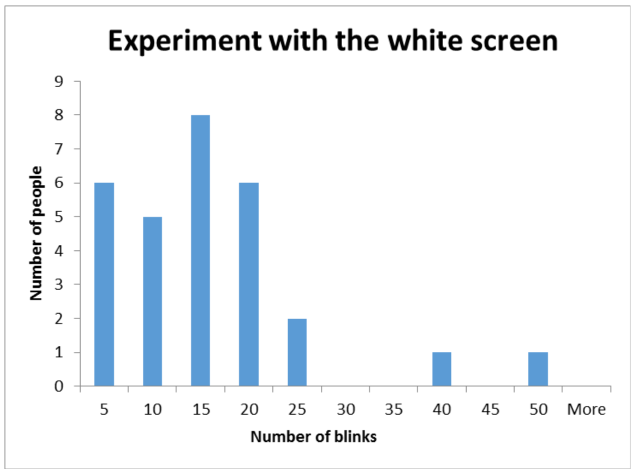


Figure 5.5: The histogram from the second experiment.

I can say that if men have certain tasks to perform, that they are focused on they blink less.

Chapter 6

Conclusion

My bachelor work has taught me much. I have become more aware of the fact that although scientists may work very hard on a certain topic, results of their work could still be unsatisfactory. I have learnt that it is crucial to conduct research in an appropriate way.

If somebody aspires to do some research in the field which they are not entirely familiarised with, it is necessary to read a lot about this topic in advance. Initially, my bachelor topic was meant to be concerned with detection of liars on the basis of eye blinks. I read books about lying. In order for the lie to be successfully detected on the basis of eye blinks, the liar needs to be preoccupied with this lie and needs to elaborate on it. Therefore, I became aware of the fact that it would be extremely difficult to collect videos with people who lie, so after having consulted this problem with my supervisor, we decided to focus on eye blinking detection.

The best idea is to consult problems with an expert in this domain. After I had collected my data, I was surprised why people did not close their eyes fully (Subsection 5.1.2). I consulted this problem with a doctor. She told me that people do not necessarily need to close their eyes fully.

A dataset is the most important thing. In my opinion, it is a good idea to create a very restricted database. In the next iteration I created a wider dataset to make the algorithm more robust.

The first results could be created very fast to verify the main idea. If this idea works, the algorithm can be checked on a wider test group. Firstly, I made three videos. At the beginning I checked each algorithm on these films. Next, I was thinking of advantages and disadvantages of this method in real data. Later, I applied this method for a larger dataset.

If a certain method does not work, it is important to take some time and think why it does not give us the result which we want to obtain. Very often it is due to a very silly mistake. Also, if the algorithm gives results which seems too good (like 98%) in the first iteration, we can assume that a mistake was made. When I was not able to find a mistake in my code, I asked other people to take a look at my code.

Experiments should be prepared very carefully, as every detail is important. At the beginning of my second experiment, people did not blink at all, which was not what I had expected. Surprisingly, when I was explaining the idea behind the experiment, a person asked me if he could blink. I replied that it was a normal function of human body. Consequently, I started to explain to everyone than blinking is allowed. *One person asked me if he could blink. Of course! You can blink. It is normal.* Therefore, I decided I would not take the previous videos into consideration.

While writing the research documentation, I was able to understand various things better.

Programming skills are very important in a research. During my programming, I made a few refactorisations of my code. In my opinion, knowledge about design patterns is very helpful for scientists as it can help to keep code clean. I think that in the future I will use POCO programming for this kind of research topics.

Obviously, there were problems or ideas that were not worth mentioning in this paper. For example, I did not mention my attempts to use the already implemented in OpenCV algorithm for eyes detection. I checked this method in a few examples, however the results were unsatisfactory. Figure 6.1.

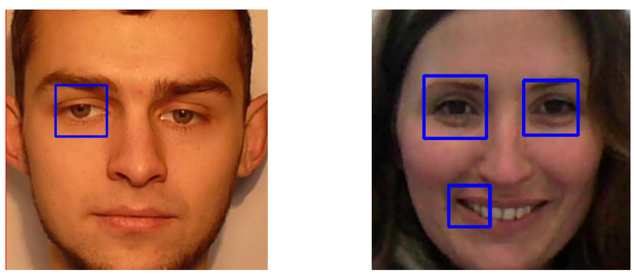


Figure 6.1: Exemplary eyes detection - OpenCV algorithm.

Bibliography

[1] Face detection using haar cascades. http://docs.opencv.org/trunk/doc/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html. Accessed: 2013-12-25.

[2] Histogram. http://en.wikipedia.org/wiki/Histogram. Accessed: 2013-12-25.

[3] Przekształcenie fouriera obrazów. http://eletel.p.lodz.pl/pstrumil/po/fft_0.pdf. Accessed: 2013-12-25.

[4] Rozpoznawanie obrazów. http://eletel.p.lodz.pl/pstrumil/po/rozpoznawanie.pdf. Accessed: 2013-12-25.

[5] Specific object recognition: Local features. http://www.pr.informatik.uni-siegen.de/sites/www.pr.informatik.uni-siegen.de/files/courses/Winter13-14/MMR/p07_sor1.pdf. Accessed: 2013-12-25.

[6] Ernest Bojek Cezary Bojek. *Wrocawskie Studia Erazmianskie. Zeszyty Studenckie, rodzial 17.* Katedra Doktryn Polityczych i Prawnych WPAiE UWr, 2008.

[7] D.-K. Cong, M. Sharikadze, G. Staude, H. Deubel, and W. Wolf. Spontaneous eye blinks are entrained by finger tapping. *Human Movement Science*, 29(1):1 – 18, 2010.

[8] Fukuda Ka. Analysis of eyeblink activity during discriminative tasks. 1994.

[9] A. S. Nixon, M. & Aguado. *Robust real-time face detection.* 2007.

[10] W.J. Orłowski and A. Bernardczykowa. *Okulistyka współczesna: podręcznik dla lekarzy specjalizujących się w chorobach oczu.* Number t. 2. Państ. Zakład Wydawnictw Lekarskich, 1977.

[11] John C. Russ. *The Image Processing Handbook Fifth Edition.* CRC Press, 2007.

[12] Konstantinos Koutroumbas Sergios Theodoridis. *Pattern Recognition Fourth Edition.* Elsiever, 2009.

[13] Aldert Vrij Sharon Leal. Blinking during and after lying. *Journal of Nonverbal Behavior*, 2008.

[14] Ryszard Tadeusiewicz. *Sieci Neuronowe.* Akademicka Oficyna Wydawnicza, 1993.

[15] J. Trobe. *Neurookulistyka. Szybkie rozpoznanie w okulistyce.* Elsevier Health Sciences Poland, 2012.