UNIVERSITÄT KOBLENZ · LANDAU
Fachbereich 4: Informatik

Classification of Facial Expressions Based on Visual Features

Bachelorarbeit
zur Erlangung des Grades
BACHELOR OF SCIENCE
im Studiengang Computervisualistik

vorgelegt von

Alruna Veith

Betreuer: Dipl.-Inform. Viktor Seib, Institut für Computervisualistik, Fachbereich Informatik, Universität Koblenz-Landau
Betreuer: M.Sc. Susanne Thierfelder, Institut für Computervisualistik, Fachbereich Informatik, Universität Koblenz-Landau
Erstgutachter: Prof. Dr.-Ing. Dietrich Paulus, Institut für Computervisualistik, Fachbereich Informatik, Universität Koblenz-Landau
Zweitgutachter: Dipl.-Inform. Viktor Seib, Institut für Computervisualistik, Fachbereich Informatik, Universität Koblenz-Landau

Koblenz, im Februar 2013

Kurzfassung

Autonome Systeme, wie Roboter, sind bereits Teil unseres täglichen Lebens.

Eine Sache, in der Menschen diesen Maschinen überlegen sind, ist die Fähigkeit, auf sein Gegenüber angemessen zu reagieren. Dies besteht nicht nur aus der Fähigkeit zu hören, was eine Person sagt, sondern auch daraus, ihre Mimik zu erkennen und zu interpretieren.

In dieser Bachelorarbeit wird ein System entwickelt, welches automatisch Gesichtsausdrücke erkennt und einer Emotion zuordnet. Das System arbeitet mit statischen Bildern und benutzt merkmalsbasierte Methoden zur Beschreibung von Gesichtsdaten. In dieser Arbeit werden gebräuchliche Schritte analysiert und aktuelle Methoden vorgestellt.

Das beschriebene System basiert auf 2D-Merkmalen. Diese Merkmale werden im Gesicht detektiert. Ein neutraler Gesichtsausdruck wird nicht als Referenzbild benötigt. Das System extrahiert zwei Arten von Gesichtsparametern. Zum einen sind es Distanzen, die zwischen den Merkmalspunkten liegen. Zum anderen sind es Winkel, die zwischen den Linien liegen, die die Merkmalspunkte verbinden. Beide Arten von Parametern werden implementiert und getestet. Der Parametertyp, der die besten Ergebnisse liefert, wird schließlich in dem System benutzt.

Eine Support Vector Machine (SVM) mit mehreren Klassen klassifiziert die Parameter. Das Ergebnis sind Kennzeichen von Action Units des Facial Action Coding Systems (FACS). Diese Kennzeichen werden einer Gesichtsemotion zugeordnet. Diese Arbeit befasst sich mit den sechs Basis-Gesichtsausdrücken (glücklich, überrascht, traurig, ängstlich, wütend und angeekelt) plus dem neutralen Gesichtsausdruck.

Das vorgestellte System wird in C++ implementiert und an das Robot Operating System (ROS) angebunden.

Abstract

Autonomous systems such as robots already are part of our daily life.

In contrast to these machines, humans can react appropriately to their counterparts. People can hear and interpret human speech, and interpret facial expressions of other people.

This thesis presents a system for automatic facial expression recognition with emotion mapping. The system is image-based and employs feature-based feature extraction. This thesis analyzes the common steps of an emotion recognition system and presents state-of-the-art methods.

The approach presented is based on 2D features. These features are detected in the face. No neutral face is needed as reference. The system extracts two types of facial parameters. The first type consists of distances between the feature points. The second type comprises angles between lines connecting the feature points. Both types of parameters are implemented and tested. The parameters which provide the best results for expression recognition are used to compare the system with state-of-the-art approaches.

A multi-class Support Vector Machine classifies the parameters. The results are codes of Action Units of the Facial Action Coding System. These codes are mapped to a facial emotion. This thesis addresses the six basic emotions (happy, surprised, sad, fearful, angry, and disgusted) plus the neutral facial expression.

The system presented is implemented in C++ and is provided with an interface to the Robot Operating System (ROS).

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Die Vereinbarung der Arbeitsgruppe für Studien- und Abschlussarbeiten habe ich gelesen und anerkannt, insbesondere die Regelung des Nutzungsrechts.

Mit der Einstellung dieser Arbeit in die Bibliothek bin ich einverstanden.	ja ☒	nein □
Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.	ja ☒	nein □

Koblenz, den 18. Februar 2013

Contents

List of Tables

List of Figures

Chapter 1

Introduction

A system capable of automatic facial expression recognition could be used in all aspects of our daily lives. Machines would have the skill to recognize our behavior in certain situations. This is useful for assisting systems in cars, games and animation, psychology, health care, robotics, and in all other assisting systems. A system which helps the robot to understand and react to human emotions is required especially in service robotics. Human-computer interaction (HCI) interfaces are designed to interact more instinctively, and with the focus on the human. Zeng et al. [ZPRH09] called this Affective Computing. They conducted a survey of affect recognition methods and their use in HCI. With the skill of emotion recognition, a HCI interface can be controlled even more instinctively. Facial expression recognition is a great extension to all of the aforementioned assisting systems. In work with handicapped people, a computer could react to different emotions. For example, a smile could be a double-click on an item. People could say yes and no by means of a happy or angry facial expression. In addition, a service robot could recognize if something is wrong with its owner and if he does not feel well. With this ability, the robot is able to help its owner according to his needs. A fully automatic system is required because the user should not need to be an expert in order to use it. The system has to work robustly and without any human support. Hence, this thesis develops a fully automatic recognition system. It should be easy to use and easy to embed in existing systems. Another important factor is the objectivity of the facial expression recognition. The system should not depend on culture, age or gender [TKC05, KCT00]. To achieve this goal, the Facial Action Coding System (FACS) is used as a basis.

The system presented deals with the six basic emotions. Recognizing facial expressions does not implicate recognizing emotions. Facial expressions are only one part of nonverbal communication [TKC05]. Emotions involve the whole human being, more than just facial expressions, for example gesture, pose, gaze direction, and voice [TKC05, FL03]. For purposes of simplification, this thesis describes emotions resulting only from facial expressions.

Posed expressions are easier to recognize and it is difficult to find a presentable database with non-posed expressions [Bet12, TKC05]. For this reason, this thesis examines posed expression recognition.

The system presented is developed to work without the neutral face as a reference image. No training step before emotion recognition is needed. Thereby, the system behaves economically in terms of memory and computation time.

The outline of this thesis is as follows. Chapter 2 presents two types of facial features and further information about FACS. Chapter 3 describes the basic structure of a facial expression recognition system (FERS) and lists related solution methods for every step. The first part of each Section describes why this step is important and necessary for further adaptation. The next parts explain basic methods for this step and their current use in related work. Each Section concludes with advantages and disadvantages of the methods presented. Detailed information about important methods which are frequently used in emotion recognition is given in Chapter 4 where Haar-like features, Support Vector Machines (SVM), and AdaBoost are explained. Chapter 5 describes the approach of an automated recognition system developed in this thesis. Chapter 6 contains experiments and results of the approach presented. Chapter 7 concludes with a summary and an outlook on future work.

Chapter 2

Facial Features and Coding System

This Chapter gives detailed information about two types of facial features and the Facial Action Coding System (FACS).

2.1 Facial Features

To work with facial features it is important to understand the two different types, permanent and transient features [FL03, TKC01, TKC05]. Permanent features give information about the current shape of the face. In contrast, transient features can give important information about the emotion demonstrated [TKC01].

2.1.1 Permanent Features

Permanent features are those features of the face which are always visible [FL03]. They mark the most relevant feature points for expression recognition. Fasel et al. asserted that these features can be deformed whilst performing an emotion. Permanent features persist permanently, such as eyebrows, eyes, nose, mouth, and cheeks [Bet12]. Bettadapura et al. [Bet12] established that the nose and mouth carry the most information of all features, whereas the mouth carries more. Observing the mouth can definitely identify a smile and hence a happy face. In contrast, the eyebrows mostly remain constant during happy and neutral emotions. Figure 2.1 visualizes the permanent features of a person from the database used in this thesis. See Chapter 6 for more information about the database. The blue zones in the Figure approximate regions, an emotion recognition system has to concentrate on.

2.1.2 Transient Features

Transient features are those features which are not always visible [Bet12]. They occur while moving facial muscles, for example wrinkling the nose. Wrinkles can

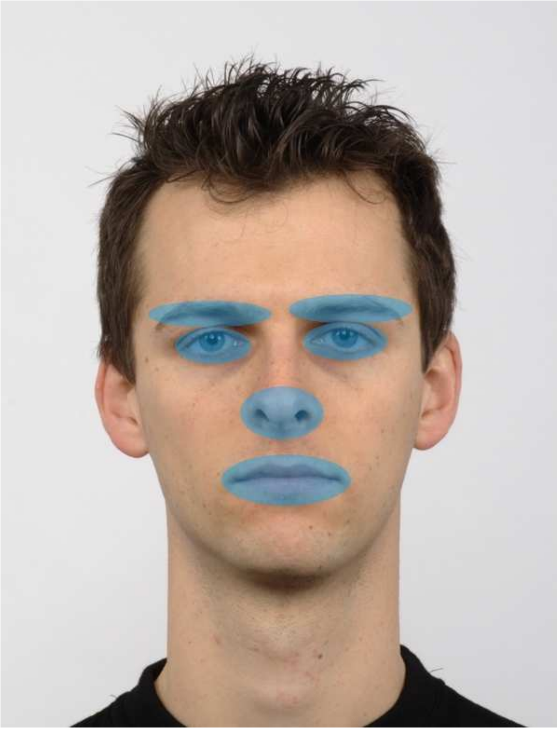


Figure 2.1: Permanent facial features

appear on the forehead, in the outer eye regions, between the eyebrows, around the nose, in the outer mouth regions, and on the chin [FL03]. Transient features do not appear on the neutral face [Bet12], but in other expressions, especially evident in disgust and anger. Figure 2.2 visualizes the transient features of the person. The orange zones in the Figure approximate regions of wrinkles, an emotion recognition system can concentrate on.

2.1.3 Summary of Facial Features

It is important to distinguish between two types of features. For emotion recognition it is not mandatory to detect transient features. Permanent features provide the most information about the face. However, including transient information may enhance recognition results for some expressions [TKC05]. Experiments of the system presented show that wrinkles detected between eyebrows mostly occur on angry and disgusted faces, whereas wrinkles between the nose and mouth usually occur on happy and disgusted faces. By using this additional information, recognition results may be improved. Hence, the system introduced in this thesis

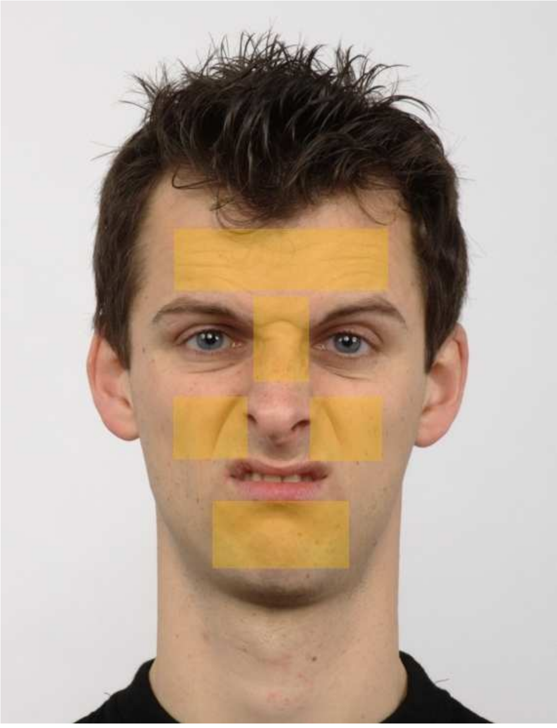


Figure 2.2: Transient facial features

utilizes both types of features. Wrinkle regions are between the eye brows and between the nose and mouth.

2.2 Facial Action Coding System

This Section describes FACS and its use in recognition systems. FACS serves as a theoretical foundation for expression recognition.

2.2.1 Construction of a Coding System

In the 1970s, Paul Ekman et al. began working on psychological studies of facial expressions [EF77]. Their work was a milestone and still has great influence on facial expression recognition [Bet12]. Prior to their study, the classification of facial expressions was obtained by subjective judgments of observers. Ekman et al. [EFH02] suggested that recognition should be completely objective, as observers could be influenced by context and their cultural interpretation. In 1977, Ekman and Friesen developed a system called Facial Action Coding System [EF77]. Contractions of facial muscles compose facial expressions [FL03]. On grounds of this knowledge, emotions are not regarded as being a subjective opinion of a person. The system provided an objective solution for emotion recognition. Coders of FACS had to pass a test [1] and had to recognize the slightest facial actions in a face. Hence, they did not observe the entire face, but muscles thereof. Bettadapura et al. [Bet12] and Zhang et al. [ZJZY08] pointed out that FACS has become an important milestone and the de-facto standard for facial expression recognition to date. Every possible facial movement is coded in an Action Unit (AU). AUs are presented in the next Section.

2.2.2 Action Units

As described above, all facial movements are coded in AUs. Thus, AUs include the corresponding muscles related to facial expressions. However, several AUs do not have their origin in facial muscles [KCT00]. For example, moving the head to the right or moving the eyes upwards have no related facial muscle.

FACS describes 44 different AUs. One code is attributed to every AU. Table 2.1 lists some examples of AUs with an image and the related description. Appendix A presents a full Table of 42 AUs assembled by Cohn et al. [CAE07] and the robotics institute of the Carnegie Mellon University [2]. Kanade et al. [KCT00] lists all 44 AUs.

2.2.3 Action Unit Combinations

A combination of AUs can be additive or non-additive [CAE07, KCT00]. This definition describes whether the AUs can be combined with each other without any influence. Non-additive AUs alter their respective appearances.

Cohn et al. [CAE07] presented examples of both combination types with AU1, AU2, and AU4. An additive combination of AU1 and AU2 is shown in Figure 2.3. Inner and outer corners of the brows are raised. The appearance of one AU has no effect upon the other. This combination often arises in a surprised expression. In contrast to this, a non-additive combination of AU1 and AU4 is presented in Figure 2.4. With AU1, the inner corners of the brows are raised. AU4 shows that the brows move closer and lower. The appearance of one AU affects the other. The inner corners are pulled together but raised. This combination often arises in a sad expression.

On the basis of this, emotions can be handled as combinations of AUs [Bet12]. For example, happy is composed of AU6 + AU12 + AU25. This is cheek raise, lip

[1] `http://face-and-emotion.com/dataface/facs/fft.jsp`

[2] `http://www.cs.cmu.edu/~face/facs.htm`

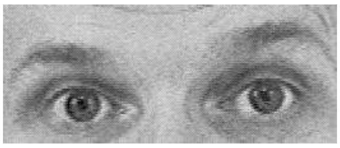


Figure 2.3: Additive combination of AU1 + AU2 by [TKC01]

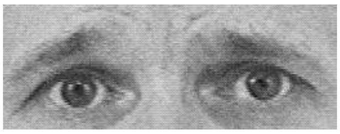


Figure 2.4: Non-additive combination of AU1 + AU4 by [TKC01]

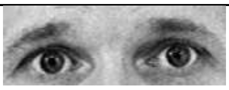
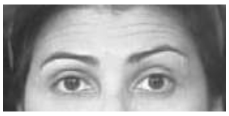
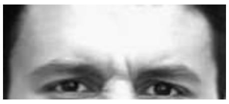
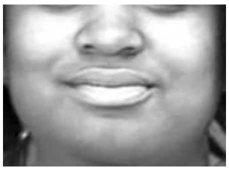
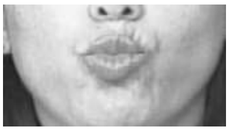
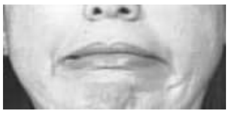
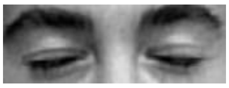
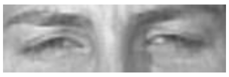
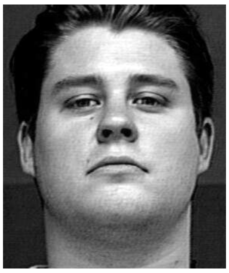
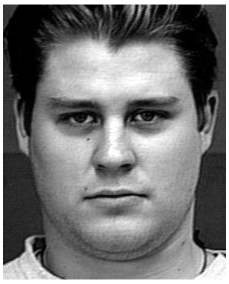
corner pull, and lips part. All combinations can be seen in Figure 5.25 of Chapter 5.

2.2.4 Summary of Facial Action Coding System

FACS has become an excellent foundation for facial expression analysis. The system can be utilized both in sequence-based as well as in image-based recognition systems which can be seen in the next Chapter. One disadvantage of FACS is its descriptive manner [KCT00]. There are no defined values which can be applied by analysis systems. Systems have to extract parameters for each emotion using different approaches. Since every system utilizes different extraction and classification methods, it is difficult to compare them with other systems [FL03, TKC01]. In any case, FACS is a common foundation and provides a step between facial analysis and emotion interpretation [FL03]. This is why it is used in the presented approach. It makes facial expression recognition more measurable and objective [BLL$^+$04].

The next Chapter depicts a common structure of recognition systems. It describes each step and common approaches utilized.

Table 2.1: Action Units of the Facial Action Coding System

AU	Example Image	Description
1		Inner Brow Raiser
2		Outer Brow Raiser
4		Brow Lowerer
12		Lip Corner Puller
18		Lip Puckerer
20		Lip stretcher
43		Eyes Closed
44		Squint
53		Head up
54		Head down

Chapter 3

Facial Expression Recognition Systems

This Chapter describes the basic structure of a facial expression recognition system (FERS) and lists related solution methods for every step.

Image-based and sequence-based related work is mentioned in this Chapter, but this thesis concentrates on static images. Both types of systems are relevant for understanding the entire range of FERS.

3.1 Basic Structure

A FERS consists of different stages. The common system is divided into three main steps: face acquisition, feature detection and representation, and expression recognition [TKC05, FL03]. The main steps comprise different sub-steps. Figure 3.1 illustrates these steps and their relationships. The image normalization step is not mandatory because facial data can also be normalized on the verge of classification [FL03]. Thus, parameters are normalized as opposed to the face itself. This occurs after the feature extraction. Normalization of contrast and lighting can also be disregarded if it is ensured that input images are always well illuminated. Related work mentioned in this Chapter demonstrated that region segmentation is also not required. Appearance-based systems in particular utilize the face as a whole. Feature representation on the basis of distances or angles is only used in feature-based systems and is not mandatory for appearance-based FERS. In the last step, most systems focus either on the classification of AUs or emotions. This can be seen in the overview Section of this Chapter. Hence, AU recognition is not mandatory for emotion recognition.

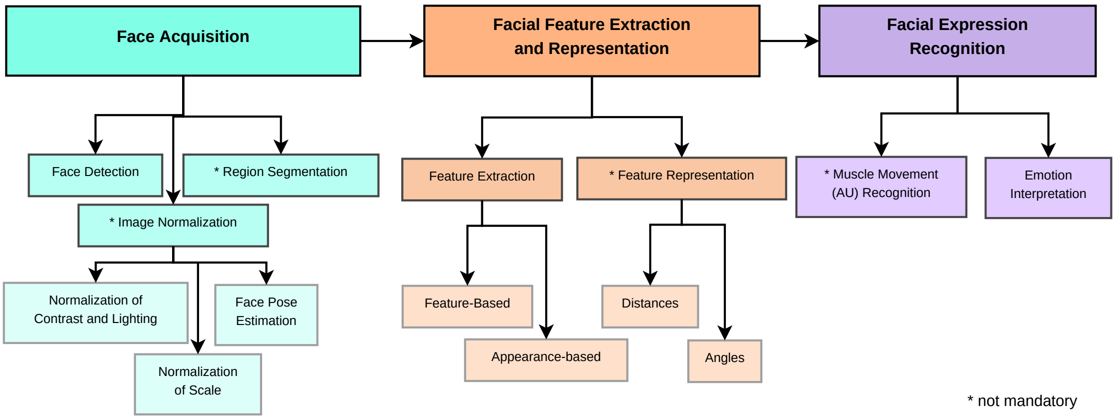


Figure 3.1: Organization of facial expression recognition systems

3.2 Face Acquisition

The first step is face acquisition. In this step, the face is detected and extracted from the whole image [TKC05]. Additionally, lighting and contrast are normalized and the face region is scaled to a standard resolution [FL03].

3.2.1 Face Detection

Face detection is the most important step of facial expression recognition as well as of many other problems of computer vision such as face localization, facial feature detection, face recognition, face authentication, face modeling, head or face tracking, gender recognition, facial expression recognition, etc. [ZZ10, YKA02].

A face on an image is not always placed centrally, nor does the face always fill the full frame. If feature detection and extraction starts before face extraction, all small regions in the full frame must be searched and this involves considerable time and memory performance. To resolve this problem, the face region must be detected in order to identify the full region of interest to be considered in the next steps. The face region starts on the uppermost line which touches the head. This does not need to be the hair line, but can also be the highest forehead line, depending on the face detector used. For purposes of emotion recognition, the face region ends on the lowest chin line. Figure 3.2 shows the face region for emotion recognition systems.

A great many of approaches have been developed for face detection [ZZ10]. Yang et al. [YKA02] classified methods in four different types. Table 3.1 illustrates this classification of the approaches used in facial expression recognition.

Figure 3.2: Face region for emotion recognition

Table 3.1: Classification of face detection methods

Classification method	Feature detection methods	Description
Feature invariant	Skin Color Recognition	These methods find structural features which describe the face
Template matching	Face Template Matching	Facial templates are stored and used to compare with the input image
Appearance-based	Neural Network, Support Vector Machines, Haar-like features, Gabor features	These methods need a training part in which facial images are learned to detect them later

Knowledge-based class is not included because is it not common in emotion recognition.

The next paragraphs explain these common basic methods and their current use in related work.

Skin Color Recognition

Skin color recognition is a heuristic-based method. It analyzes the color of the image. All regions are searched for containing a color equal to the facial colors. Esau et al. [EWKK07] used skin color recognition for detecting largest skin region in the input image. Srivastava et al. [Sri12] employed it as a last step verifying face detection. Pantic et al. [PR04] utilized this method to detect face region. Watershed segmentation is done for face extraction.

Pupil Localization

Pupil Localization is an algorithm which detects pupils in an image. Esau et al. [EWKK07] used this approach for face tracking. They first detected the pupils. Depending on the location, they detected other feature points.

Template Matching

Template Matching is, as the name suggests, a template-based method. Several patterns of faces are stored [YKA02]. The agreement of the template and the input image is computed. When the template matches to a region in the image, a face is detected. Esau et al. [EWKK07] used this approach for face localization. Tian et al. [TKC01] made use of template matching for feature detection. They utilized different templates for eyebrows, eyes, lips, and cheeks.

Neural Network

Neural Network is a machine learning algorithm. It computes the Eigenvectors of the autocorrelation matrix of an image, also called Eigenfaces [YKA02]. Rowley et al. [RBK98] suggested that at least one Neural Network must be trained to handle variations of faces. They developed a system to detect faces with Neural Networks.

Support Vector Machine

A Support Vector Machine is a machine learning algorithm. Osuna et al. [OFG97a] used this method for face detection. To see how Support Vector Machines work in detail, refer to Chapter 4.

Haar-like Features

Haar-like features are used for machine learning. The Open Source Computer Vision (OpenCV) library utilizes them for face detection. AdaBoost is commonly used for classification or selection of Haar-like features. For a detailed description of Haar-like features and AdaBoost, refer to Chapter 4. Srivastava et al. [Sri12] utilized Haar-like features for face detection.

Gabor Filter Features

Gabor filter features are also used for machine learning. These features can be utilized for a classification-based face detection method [HSK05]. Gabor features are computed on the entire face region. The extracted features are compared to trained facial images, for example by means of SVMs [LFBM02]. Hence, a region of the input image is classified if it contains a face. Usually Gabor features are like Haar-like features selected with AdaBoost. Huang et al. [HSK05] utilized Gabor filter features for robust face detection.

3.2.2 Conclusion of Face Detection

Skin color recognition varies between different cultures and depends highly on illumination. Hence, it is not suitable for the proposed system. Pupil localization is an adequate approach for systems using the pupils for further feature point processing. However, this approach is more effective for sequence-based FERS. Template matching is commonly used for sequence-based FERS as well. All machine-based methods seem to achieve best results for face detection. This can be seen in Table 3.3 in the overview Section of this Chapter. A method using Haar-like features combined with AdaBoost turns out to be the best for the approach presented. The OpenCV library implements its face detector on the basis of these two methods. Hence, this preexisting algorithm is used in this thesis.

3.2.3 Image Normalization

Image normalization is a recommended step before further working on the image [FL03]. The normalization has influence on results of feature detection and thus on expression recognition.

Normalization of Contrast and Lighting

Normalization of contrast and luminance is advisable for further steps [FL03]. FERS should be used in everyday life where lighting is not always optimal. Using normalization, faces are detected in dark places as well. This is useful in places where illumination depends on nature lighting. For instance systems used in apartments where lighting is very different depending on the apartment. In several systems like [EWKK07] this kind of normalization is a step before face detection to make it more robust. This depends on the face detection method which is used. In other approaches like [ZJZY08] the normalization comes after face detection.

Normalization of Scale

To compare extracted parameters of facial expressions, the scale of the face has to be the same through all images [TKC05]. Because every frame has a different face scale, there has to be a normalization of the scale. Bartlett et al. [BLL$^+$04] used a size of 48 x 48 pixels for every detected face. Tian et al. [TKC05] pointed out that with a size of 96 x 128 or 69 x 93 pixels, more emotions are recognized than on 48 x 64 or 24 x 32 pixels. The system of Velusamy et al. [VKA$^+$11] worked with normalized images of 96 x 96 pixels. Esau et al. [EWKK07] utilized image sizes of 320 x 240 pixels. Pantic et al. [PR04] worked with 720 x 576 pixels.

Face Pose Estimation

To handle a wide range of facial image types, it should also be possible to work with rotated faces. After the detection of the face, it is possible to detect whether the face looks straight to the camera or if it is rotated. The face has to be rotated to the normal position to be passed to next steps. Therefore, the current pose has to be estimated. This step has to be prior to feature extraction, to use correct parameters instead of rotated ones with false positions [TKC05]. It can be done in the face detection part as Rowley et al. did with neural networks [RBK98]. Similarly, pose estimation can be a part of the face normalization as a step after face detection.

There are two types of rotations, in-plane rotations and out-of-plane rotations [FL03]. In case of in-plane rotations, the face still looks frontal to the camera. Fasel et al. suggested that out-of-plane rotations are more difficult to solve. Parameters of the face are distorted. To obtain correct parameters, the warped ones have to be converted.

3.2.4 Conclusion of Image Normalization

Taken together, image normalization is an useful step for achieving robust results. Hence, the system in this thesis uses a normalization of contrast and lighting as a step previous to face detection and several normalization functions after the detection. The functions which are used are described more in detail in Chapter 5. The normalized sizes of related systems seem to be very small. The approach described in this thesis operates with a bigger size like Esau et al. in order not to lose information. Face pose estimation can be done after face detection to handle in-plane and out-of-plane rotations. This approach is common in systems which use automatic face detectors which are already implemented. A disadvantage can be the fact that the face detector does not even detect a rotated face. In this thesis face rotations are not covered. In future work they will be treated after normalization of scale.

3.2.5 Region Segmentation

To work with AUs, a FERS may divide the whole facial region in smaller regions. These regions are each called a Region Of Interest (ROI) [PR04]. In these ROIs, the facial components are located which are relevant for facial expression recognition. These parts are the eyebrows, the eyes, the nose, and the mouth. There are different methods to extract the ROIs out of a face region. Whitehill et al. [WO06] pointed out that feature detection on subregions greatly reduces time and possible features. They used squares with a width of 24 pixels around the mouth and each of the brows and eyes. Srivastava et al. [Sri12] noticed that face segmentation into smaller regions improves frame rate. They divided faces geometrically.

3.2.6 Conclusion of Region Segmentation

The proposed recognition system considers the furrows between nose and mouth and measures parameters between eyes and brows on the one hand and between nose and mouth on the other hand. Hence, it makes use of two ROIs. One ROI consists of the upper part of the face with both eyebrows and eyes. The second ROI consists of the lower part of the face including the nose and mouth. This region division is optimal for this system, because the respective facial parts in each ROI interact with each other. It would not be useful to have four ROIs: one for the brows, eyes, nose, and the mouth. The components of two ROIs do not influence each other for AU recognition [TKC01]. Hence, only ROI1 and ROI2 can be handled separately.

To divide the facial image in ROIs, the proposed system uses the geometrical approach of Leonardo da Vinci. Every face consists of several parts with the same length. Figures 3.3 and 3.4 visualize the division of a face. The system presented takes advantage of this discovery.

3.3 Feature Detection, Extraction and Representation

In this step, facial feature points are detected and facial parameters are extracted. Feature vectors represent a facial expression.

3.3.1 Facial Feature Points

MPEG-4 is an ISO-standard for facial features, basically used for facial animation, as described by Pandzic et al. [PF02]. They declared that the standard defines the location of 84 facial feature points in the neutral face. These feature points provide a basis for 68 face animation parameters (FAPs). FAPs are closely related to movements of facial muscles as it is described in the AUs of FACS. Zhang et al. [ZJZY08] pointed out that FAPs represent these of AUs descriptive defined movements quantitatively. In their paper, they described the relationship between the

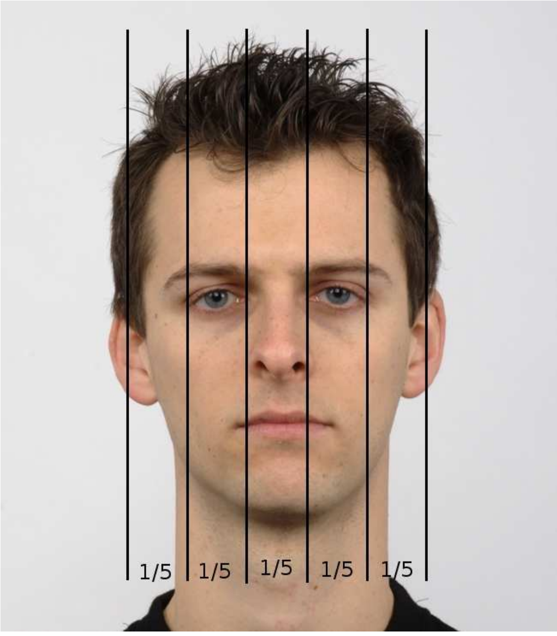


Figure 3.3: Vertical face division

feature points of the MPEG-4 standard and the AUs of FACS. Further information is given in the Section of facial feature representation in this Chapter. Figure 3.7 shows all defined feature points.

Esau et al. [EWKK07] made use of 13 facial feature points. These points can be seen in Figure 3.5.

Cerezo et al. [CH06] used 10 feature points. The points were marked manually. Hence, the approach is not an automatic FERS. Figure 3.6 shows the feature points of their system.

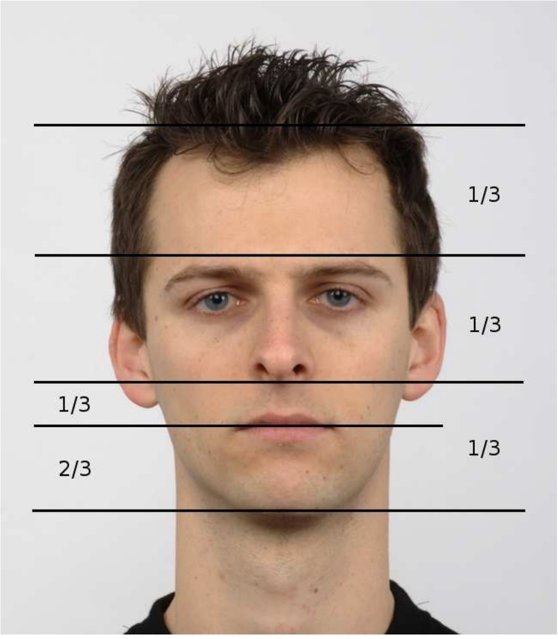


Figure 3.4: Horizontal face division

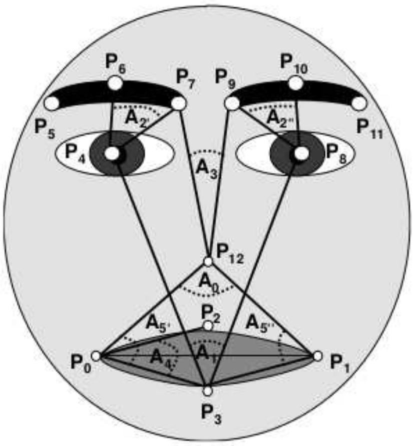


Figure 3.5: Feature Points of Esau et al. [EWKK07]

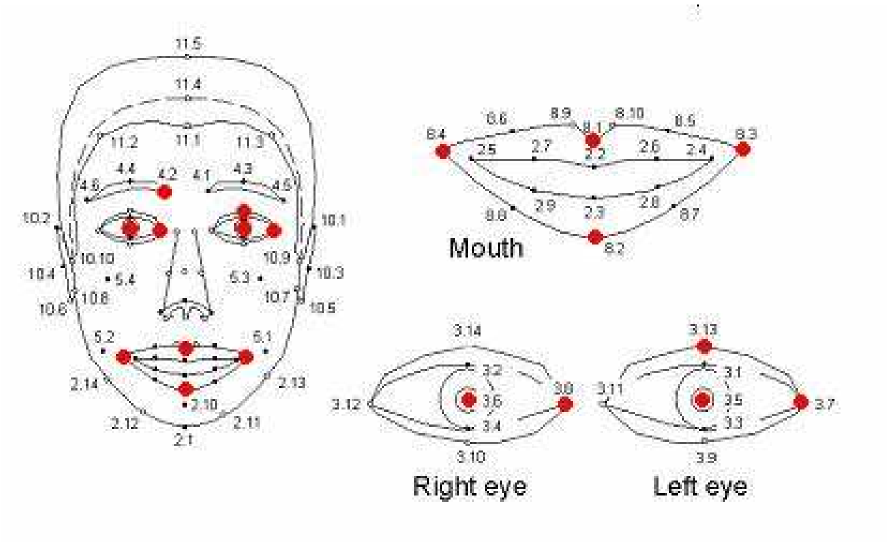


Figure 3.6: Feature Points of Cerezo et al. [CH06]

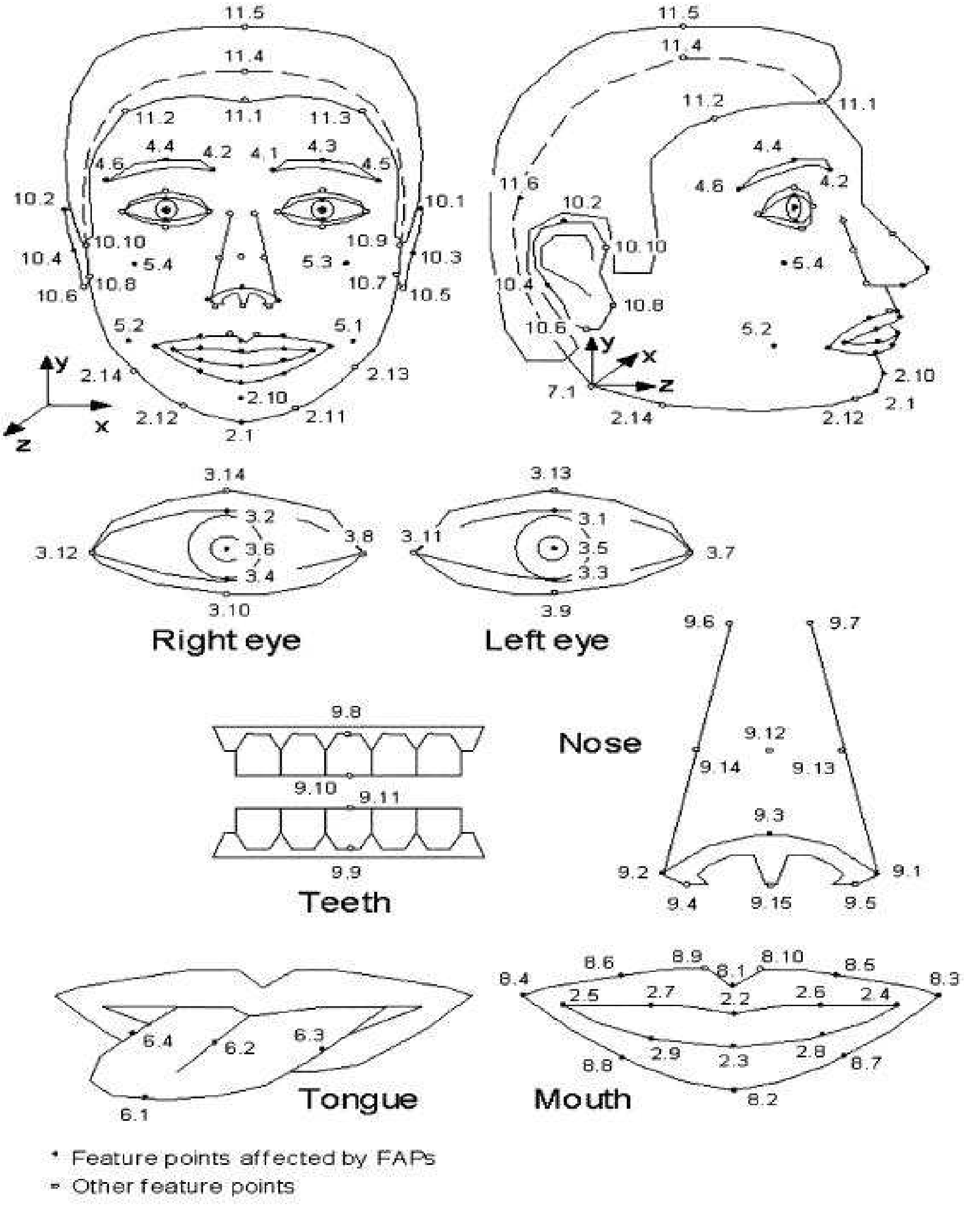


Figure 3.7: Feature Points of MPEG-4 standard by [AP99]

3.3.2 Facial Feature Extraction

There are two main approaches for facial feature detection and extraction [TKC05, FL03, WO06].

Feature-based Approach

The feature- or geometric-based approach acts locally [FL03]. Geometric locations of facial features are extracted [WO06]. Information about the face is given in these positions. Depending on the location of feature points, a statement about the shape of facial components can be made. Algorithms compute different facial parameters depending on the position feature points have in each emotion. The parameters are extracted into a feature vector. In this vector, face geometry is typified [TKC05]. The feature vector is compared to feature vectors of each emotion.

Feature-based approaches use methods like high gradients, contour detection, edge detection, and corner detection for feature detection and extraction.

Kotsia et al. [KP07] used geometrical information for a grid-based feature extraction. Tian et al. [TKC01] utilized the Canny edge detector [Can83] for the detection of transient features. Esau et al. [EWKK07] first detected the pupils. Depending on the location, they localized the nose tip. Afterwards the corners of the mouth were detected. By means of this step, they determined other needed feature points. Srivastava et al. [Sri12] employed the Sobel derivative [SF68] and corner detection to find facial feature points.

Appearance-based Approach

The appearance-based approach acts holistically [FL03]. The face is considered as a whole [TKC05]. Appearance-based approaches use image filters like Gabor wavelets for feature detection and extraction. They have a high computation time [EWKK07]. Further appearance-based methods are described in [FL03] and [TKC05]. Tian et al. [TKC05] pointed out that the recognition of AUs is more accurate when using geometric features instead of Gabor wavelets.

Velusamy et al. [VKA$^+$11] used a Gabor filter bank with seven scales and eight orientations. The feature vectors had a size of 96x96x56. Therefore, AdaBoost was needed for feature selection. Littlewort et al. [LFBM02] utilized a Gabor representation of each face for further recognition steps. They made use of 40 Gabor filters. They used eight orientations and five spatial frequencies. Bartlett et al. [BLL$^+$04] used Gabor filters as well. They utilized eight orientations and nine spatial frequencies for the recognition of AUs and seven spatial frequencies for the recognition of emotions. Srivastava et al. [Sri12] employed Haar-like features to classify facial regions into eyes, nose, and mouth region. Whitehill et al. [WO06] used Haar-like features for feature extraction and AdaBoost to select 500 of these features for classification.

3.3.3 Facial Feature Representation

Facial features are represented by a set of parameters combined into one feature vector. There are two basic methods of parameterization to describe the deformations in a face after a feature-based extraction method [EWKK07].

The first approach only uses distances as parameters. These distances are between the detected feature points. Zhang et al. [ZJZY08] described the relationship between the MPEG-4 feature points and the AUs of FACS. Table 3.2 illustrates this relationship. On the basis of this relationship, relevant distances can be measured. Distance $D_x(p_1, p_2)$ is measured in the x-direction between the feature points p_1 and p_2. The value is computed as $D_x(p_1, p_2) = |p_1.x - p_2.x|$, for D_y equivalent. AU25 and AU26 rely on the same distances. Zhang et al. clarified that for AU25 the distances are small and for AU26 they are medium-sized. The Table lists only the AUs used in the presented system. For more AUs, refer to [ZJZY08].

The second approach only uses angles as parameters. Esau et al. [EWKK07] developed a fuzzy rule-based FERS called VISBER. They extracted eight angles, three in the upper face region, four in the lower face region and one belonging to both. Figure 3.5 visualizes the angles of their system. The angles are defined as A_0 - A_5. A_5 and A_2 each consist of two angles. These angles should have the same values because they are symmetrical. Angle A_0 describes happiness with a high value and sadness with a low one. A small value of A_1 specifies fear or happiness with an open mouth. A_2 and A_3 assign anger or fear. A_4 and A_5 depict fear when the mouth is opened widely. Esau et al. used angles because they are size invariant and the step of face normalization can be left out. The method to represent a face only with angles is described as robust against individual changes. In contrast to this, Esau et al. depicted the variation of distance parameters between different persons as significant. However, they made use of the neutral face of a person for training individual characteristics. They clarified that it is not mandatory, but improved their recognition results.

Cerezo et al. [CH06] made use of both types of parameters. They employed two different angles to extract the mouth shape. The angles can be seen in Figures 3.8 and 3.9. Their other parameters were computed with distances.

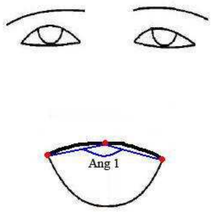


Figure 3.8: Additional information by means of a mouth angle by [CH06]

Figure 3.9: Additional information by means of a second mouth angle by [CH06]

Table 3.2: Relationship between MPEG-4 Feature Points and AUs

Distance of two Feature Points	AU
$D_y(4.2, 3.8)$	AU1
$D_y(4.1, 3.11)$	
$D_y(4.6, 3.12)$	AU2
$D_y(4.5, 3.7)$	
$D_y(4.2, 3.8)$	AU4
$D_y(4.1, 3.11)$	
$D_x(4.4, 3.8)$	
$D_x(4.3, 3.11)$	
$D_y(3.6, 3.2)$	AU5
$D_y(3.5, 3.1)$	
$D_y(3.6, 3.2)$	AU6
$D_y(3.5, 3.1)$	
$D_y(5.4, 3.12)$	
$D_y(5.3, 3.11)$	
$D_y(3.4, 3.6)$	AU7
$D_y(3.3, 3.5)$	
$D_y(9.14, 3.8)$	AU9
$D_y(9.13, 3.11)$	
$D_y(8.4, 3.12)$	AU10
$D_y(8.3, 3.11)$	
$D_y(8.4, 3.12)$	AU12
$D_y(8.4, 3.11)$	
$D_x(8.4, 9.15)$	
$D_x(8.3, 9.15)$	
$D_y(8.4, 9.15)$	AU15
$D_y(8.3, 9.15)$	
$D_y(8.2, 9.15)$	AU17
$D_x(8.4, 8.3)$	AU20
$D_x(8.3, 8.4)$	
$D_y(8.2, 9.15)$	
$D_x(8.4, 8.3)$	AU23
$D_x(8.3, 8.4)$	
$D_y(8.1, 9.15)$	AU24
$D_y(8.2, 9.15)$	
$D_y(8.2, 8.1)$	AU25
$D_y(8.2, 9.15)$	
$D_y(8.2, 8.1)$	AU26
$D_y(8.2, 9.15)$	

3.3.4 Conclusion of Feature Extraction and Representation

The approached system defines 19 feature points of the MPEG-4-standard. These points seem to be sufficient to provide facial parameters needed for AU-recognition.

A feature-based approach is used for facial feature detection. This approach has certain advantages in contrast to appearance-based methods. The system presented divides the face into regions and considers each region separately. Each region is split into facial parts. Hence, a feature-based approach is used to define feature points on each facial part. On the basis of this approach, the face can be regarded in detail. This is important to distinguish between emotions based on FACS because AUs consist of slight muscle movements, as described in Chapter 2. An appearance-based approach may evoke errors considering different emotions which appear to be the same. For example, fearful and surprised both imply widely opened eyes and an open mouth. Like Tian et al. [TKC01], the system presented in this thesis uses the Canny edge detector for the detection of transient features. The feature-based approach is used because only 19 feature points have to be detected. These points should be detected quickly with feature-based methods as they lie on corners and edges. Hence, the feature-based approach is efficient and fast. Methods like Gabor features or Haar-like features would be too complex and expensive for these slight computations. The system presented should have an adequate performance of time. This is important especially for the employment in robotic systems.

There is no research to decide whether one of the two parameterization methods is better than the other. Hence, both approaches are tested and evaluated. The parameters providing best results for expression recognition are finally used in the system.

Furthermore, in this thesis there is no neutral face needed for comparison. The neutral expression can be recognized as every other emotion. Related systems like Esau et al. [EWKK07] used a reference image to improve their recognition results. The system presented in this thesis investigates if it is truly necessary.

3.4 Facial Expression Recognition

Two approaches are distinguished in facial expression recognition. Velusamy et al. [VKA^{+}11] pointed out that emotion recognition can be single-phase or two-phase. A single-phase method directly recognizes emotions from facial data. A two-phase method first recognizes AUs in the face and then interprets the emotion from them.

3.4.1 Action Unit Recognition

The action unit recognition classifies the extracted facial data into one specific AU or into a combination of AUs.

The next paragraphs only list several classification methods to provide an overview. Detailed information is above the scope of this thesis and for example presented in [TK09]. Sebe et al. [SLS$^+$07] described certain classifiers in a more detailed analysis. Bettadapura et al. [Bet12] listed several classifiers as well. Theodoridis et al. [TK09] distinguished between three basic classification approaches for pattern recognition: classifiers based on Bayes Decision Theory, linear classifiers, and nonlinear classifiers. This thesis denotes a fourth class called probabilistic graphical models. Additionally, there are certain systems using rule-based approaches.

Classifiers based on Bayes Decision Theory

Classifiers based on Bayes Decision Theory are Naive Bayes, Tree Augmented Naive Bayes, Stochastic Structure Search, and k-Nearest-Neighbor.

Linear Classifiers

A SVM is a linear classifier. There are binary or multi-class SVMs. An in-depth description of SVMs can be found in Chapter 4. Bartlett et al. [BLL$^+$04] used one binary SVM for each AU for a context-independent recognition. Velusamy et al. [VKA$^+$11] made use of 15 SVMs to classify each of their 15 AUs. Kotsia et al. [KP07] used multi-class SVMs. A six-class SVM was used for recognizing the six basic expressions. Seventeen binary SVMs were used for the recognition of 17 AUs. Kotsia et al. pointed out that SVMs deliver good performance in pattern recognition. Littlewort et al. [LFBM02] utilized SVMs for classification as well. They recognized emotions directly without considering AUs. They figured out that SVMs are fast to train and accurate in recognition.

Other linear classifiers are Neural Networks (NN), Artificial Neural Networks (ANN), and Linear Discriminant Classifier. Tian et al. [TKC01] utilized two ANNs for AU recognition. One ANN was for the upper facial region and one was for the lower region.

Non-Linear Classifiers

Non-linear classifiers are generalized linear classifiers, polynomial classifiers, probabilistic Neutral Networks, and Decision Trees.

Probabilistic Graphical Models

Probabilistic graphical models are Bayes Networks, Single Hidden Markov Models, and Multi-Level Hidden Markov Models. Detailed information about probabilistic graphical models is presented in [KF09].

Rule-based Classifiers

In 2004, Pantic et al. [PR04] used a rule-based method for the recognition of 32 AUs in static images. In 2005, Pantic et al. [PP05] defined 27 rules for AU-dynamics recognition on video sequences. In 2006, Pantic et al. [PP06] described a new rule-based method to recognize AUs on profile image sequences. Esau et al. [EWKK07] made use of a fuzzy rule-based classification. The extracted angles were defined as large, medium or small. Four emotions (happiness, sadness, anger, and fear) were classified with help of the combination of these angle-states.

3.4.2 Emotion Interpretation

AUs describe facial movements. Facial expressions are composed of movements in different facial regions. Hence, an emotion is characterized by a set of AUs. Emotion interpretation receives the extracted feature vectors or the recognized AU-codes as input and delivers an emotion as output.

Velusamy et al. [VKA$^+$11] clarified that a two-phase method for emotion recognition is more practical, as only a set of 44 AUs of FACS have to be detected which are used in emotions. Not every state of emotion has to be learned because this method relies on AUs and therefore on muscle movement. For example, the system does not have to define a threshold for smile but only identify a movement of the related muscle. Additionally, Velusamy et al. pointed out that a two-phase system is more culturely independent.

In their paper, Velusamy et al. presented a set of 15 AUs which built a foundation for a learned statistical relationship for emotion mapping. They presented an algorithm which allocates each AU to every emotion. A detected AU could either improve or downgrade the probability of a specific emotion. The emotion with highest probability was recognized in the input image. Langner et al. [LDB$^+$10] introduced a rule-based method for expression classification. They presented a set of 16 AUs which built a base for the six basic emotions. Each emotion consisted of at least one and at most six AUs, which were explained by certified FACS experts.

3.4.3 Conclusion of Facial Expression Recognition

This thesis utilizes a SVM as classification method. SVMs form a common method for facial expression recognition. Most of the systems developed recently used SVM classifiers. This can be seen in the Table of the next Section. SVMs can be used as a binary SVM or multi-class SVM as described in Chapter 4. Recognition systems based on SVM delivered stable performance results. Therefore, it is also used in this thesis.

In the majority of cases, other systems like [TKC01, PR04, PP05, PP06, WO06] only recognized AUs in the input image. Or they recognized emotions without considering AUs like it is done in [LFBM02, CH06, EWKK07, Sri12]. These systems are single-phase. The system presented goes a step further and uses AU-information for emotion mapping. Hence, this system is two-phase.

The system introduced utilizes the Radboud Faces Database (RaFD). It is based on the AUs of FACS. Certified experts observed recording of the database. Hence, the mapping algorithm for emotion interpretation is created on the basis of this database. More information is given in Chapters 5 and 6.

3.5 Overview

Table 3.3 gives an overview of state-of-the-art expression recognition systems. It summarizes all information of this Chapter.

Sign "-" means there is no information about this topic in the paper or the topic is not important for this system. For example, in several systems the scale size is not important because the extracted parameters are normalized later on. Sign "X" denotes that this topic is not handled in this system. Sign "Y" means yes and "N" means no. Commonly, the neutral face of a person is treated as the reference face.

This thesis presents the references of this Table as state-of-the-art in terms of 2D features. Only systems since 2001 are covered. Several recent systems are mentioned to show the methods which are used, and highlight that recognition rate has not been improved significantly. Most of the systems developed in the last few years were either based upon the systems presented or dealt preferably with 3D features and / or video sequences as input. A comprehensive survey of static and dynamic facial expression recognition based on 3D features is presented in [SZPY12].

The overview shows that the majority of related systems (10 of 12) recognized either AUs or emotions. Their emotion interpretation was merely single-phase. Actually, emotions based on FACS are more presentable and facile to extend. However, two-phase systems consist of two recognition steps and hence are more error-prone. The use of one step clarifies high recognition results of 86.16% averaged. The majority of related systems worked on sequences when not considering single frames of sequences as input. On average, sequence-based systems achieved recognition rates of 87.91% higher than image-based with 84.35%. Half of the systems mentioned were fully automatic and half were not. Recognition results of fully automatic systems were inferior. FERS which were not fully automatic, averaged 90.23%. On the contrary, fully automatic systems averaged 83.57%. Each of the feature-based systems needed a reference face for comparison.

3.6 Summary

The basic structure of a FERS consists of different stages. The common system is divided into three main steps and certain sub-steps. Face acquisition is built up of face detection, image normalization, and region segmentation. To obtain information of the face, facial features are detected and extracted. Facial feature representation typifies the gained information. Facial expression recognition means AU recognition and / or emotion classification. This last step differs in several FERS. The system presented first recognizes AUs in the image. The emotion demonstrated is classified with information of these AUs.

The next Chapter gives more in-depth information about important methods frequently used in emotion recognition. Haar-like features, SVMs, and AdaBoost are explained.

Table 3.3: State-of-the-art

Reference	Input	Face Detection	Scale	Rotation	Feature Extraction	AU Recognition	Emotion Interpretation	AU Output	Emotion Output	Reference Face Needed	Fully Automatic	Recognition Rate
Tian et al., 2001, [TKC01]	sequences	Template Matching	-	in-plane + limited out-of-plane	feature-based	ANN	X	16 AUs + neutral + combinations	X	Y	N	96%
Littlewort et al., 2002, [LFBM02]	single frames of sequences	AdaBoost	48 x 48	X	appearance-based	X	SVM	X	6 emotions + neutral	-	Y	91.5%
Bartlett et al., 2004, [BLL^{+}04]	single frames of sequences	GentleBoost	48 x 48 or 192 x 192	X	appearance-based	SVM	SVM	18 AUs + combinations	6 emotions + neutral (either AUs or emotions)	-	Y	94.6% or 93.3%
Pantic et al., 2004, [PR04]	images	Skin Color Recognition + Watershed Segmentation	720 x 576	out-of-plane (front + profile)	feature-based	rule-based	X	32 AUs + combinations	X	Y	Y	86%
Pantic et al., 2005, [PP05]	sequences	manual selection of feature points	-	X	feature-based	rule-based	X	27 AUs + combinations	X	Y	N	90%
Pantic et al., 2006, [PP06]	sequences	manual selection of feature points	-	out-of-plane (profile)	feature-based	rule-based	X	27 AUs + combinations	X	Y	N	87%
Cerezo et al., 2006, [CH06]	images	-	-	X	feature-based	X	rule-based	X	6 emotions + neutral	Y	N	71.4%
Whitehill et al., 2006, [WO06]	images	manual selection of ROIs	width of 64 pixels	X	appearance-based	AdaBoost	X	11 AUs	X	N	N	92.4%
Esau et al., 2007, [EWKK07]	single frames of sequences	Skin Color Recognition + Template Matching	320 x 240	X	feature-based	X	fuzzy-rule-based	X	4 emotions + neutral + combinations	Y	Y	72%
Kotsia et al., 2007, [KP07]	sequences	manual selection of feature points	-	out-of-plane (front + profile)	feature-based	SVM	rule-based (after AU recognition) or SVM (only emotions)	17 AUs	6 emotions (either alone or with AUs)	Y	N	95.1% (after AU recognition) or 99.7% (only emotions)
Velusamy et al., 2011, [VKA^{+}11]	images + sequences	Bartlett et al.	96 x 96	out-of-plane (+-10°)	appearance-based	SVM	learned statistical relationship	15 AUs	6 emotions	Bartlett et al.	Y	87.6%
Srivastava et al., 2012, [Sri12]	sequences	Haar-like features + Skin Color Recognition	-	out-of-plane (front + profile)	appearance-based + feature-based	X	SVM	X	3 emotions + neutral	Y	Y	60%

Chapter 4

Important Algorithms

This Chapter presents Haar-like features, SVMs, and AdaBoost. These algorithms are frequently used in emotion recognition.

4.1 Haar-Like Features

This Section gives more detailed information about Haar-like features. This thesis focuses on the basic concept of Haar-like features. Further information can be found in [VJ01b, VJ01a].

Haar-like features is a machine learning algorithm for object detection. Viola and Jones [VJ01b] developed the algorithm in 2001. Their work consists of three main parts. The first part is called integral image and is a representation of the image. Here, the features for object detection can be detected and evaluated very quickly at any scale. The second part consists of an AdaBoost learning algorithm. AdaBoost is described in the last Section of this Chapter. The third part combines complex classifiers in a cascade. By means of this cascade, subregions which do not include the object are discarded very quickly. Hence, the classifier can concentrate on object-like regions.

4.1.1 Feature Types

Viola and Jones introduced three types of features. In general, the value of a feature is computed as the difference of sum of pixels which lie in different rectangular regions. The regions have the same shape and size.

The first type is called two-rectangle feature. Figure 4.1 shows this Haar-like feature. The sum of the white pixels is subtracted from the sum of the gray pixels.

The second type of feature is called three-rectangle feature. Its value is calculated by the sum of two rectangles being subtracted from the sum of values of

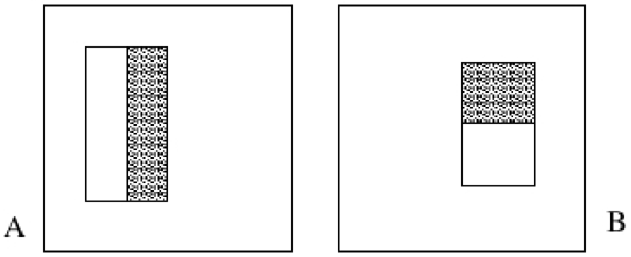


Figure 4.1: Two-rectangle Haar-like feature by [VJ01b]

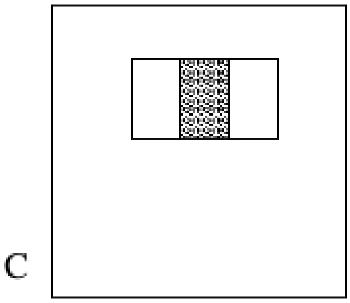


Figure 4.2: Three-rectangle Haar-like feature by [VJ01b]

the pixels which lie in a central rectangular region. Figure 4.2 shows this kind of Haar-like feature.

The last type of feature is called four-rectangle feature. Its value is computed by the difference of the sum of two rectangles and the sum of two other rectangles. Figure 4.3 illustrates the rectangles with black and white colors.

4.1.2 Integral Image

The integral image is a representation of the image [VJ01b]. Values of rectangles can be calculated very quickly using this image. The whole integral image is computed with some operations on every pixel. Viola and Jones defined the value at pixel (x, y) in the integral image as follows:

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

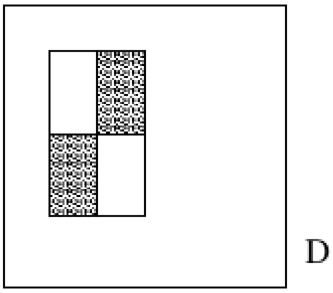


Figure 4.3: Four-rectangle Haar-like feature by [VJ01b]

The value at $ii(x, y)$ is the sum of the values above and left from the pixel in the original image $i(x, y)$. The following equations are used to compute each value of the integral image ii.

$$s(x, y) = s(x, y - 1) + i(x, y) \tag{4.2}$$

$$ii(x, y) = ii(x - 1, y) + s(x, y) \tag{4.3}$$

Equation 4.2 defines the cumulative row sum with the following predefinition:

$$s(x, -1) = 0 \tag{4.4}$$

Equation 4.3 defines the value in ii as the cumulative row sum, plus the value at pixel $(x - 1, y)$ with:

$$ii(-1, y) = 0 \tag{4.5}$$

By means of recurrence in equations 4.2 and 4.3, each value of ii is calculated in one run over the image i.

Figure 4.4 illustrates a computation example adopted from Viola and Jones. The sum of pixel-values in rectangle D can be calculated as $4 + 1 - (2 + 3)$. Every number represents a location on the integral image. By means of equation 4.1, the value of 1 is calculated as the sum of pixel-values in rectangle A. The value of 2 is $A + B$. The value of 3 is computed as $A + C$ and the value of 4 is $A + B + C + D$. The value of D can be used for the evaluation of Haar-like Features.

Viola and Jones described that once the integral image is calculated, the algorithm can charge Haar-like feature at any scale or location. This is done in constant time. The window in which Haar-like features are calculated can be resized. Therefore, objects are detected independent of size.

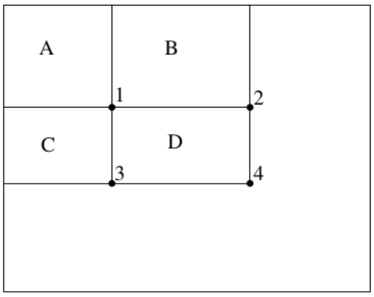


Figure 4.4: Integral image by [VJ01b]

4.1.3 Feature Selection by AdaBoost

Viola and Jones used AdaBoost to reduce the number of features to important ones. Complex classifiers are combined to a cascade. The cascade focuses on regions which often include the object of interest. Hence, it increases the speed drastically. An initial classifier discards subregions with small probability of involving the object. Remaining subregions are executed by a cascade of classifiers. Each classifier in this sequence is more complex than its precursor. If one classifier disallows a subregion, this region is discarded and the next is processed.

4.1.4 Summary of Haar-Like Features

Haar-like features play an important role in object detection. By means of the integral image, Haar-like features can be computed very quickly. The use of AdaBoost additionally improves the speed. The system presented utilizes Haar-like features for face detection implemented in a preexisting face detector. Face detection is the common application area of the algorithm developed by Viola and Jones [VJ01a].

4.2 Support Vector Machine

This Section gives more detailed information about Support Vector Machines (SVMs). This thesis focuses on the basic concept of SVMs. Further information can be found in [Bur98].

A SVM is a machine learning algorithm which is used for classification. In 1995, Cortes and Vapnik [CV95] extended the original algorithm for non-separable data.

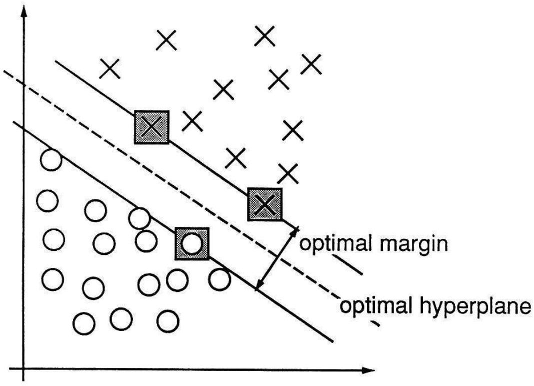


Figure 4.5: Feature space of a binary SVM by [CV95]

4.2.1 General Structure

In general, a SVM receives a feature vector and maps it into a high-dimensional feature space [CV95]. A linear decision surface is computed to separate label classes. For generalization testing, the feature vector of a test object is inserted in the feature space of training data.

4.2.2 Binary Support Vector Machine

A binary SVM, also known as two-class SVM, discriminates between two classes. The input vectors are mapped to a feature space with the same dimension as are the vectors. A linear hyperplane is stretched to separate the two classes in an optimal way [HCL08]. As Cortes and Vapnik [CV95] declared, an optimal hyperplane has a maximum margin to support vectors of both classes. Support vectors are the vectors of each class which lie on the boundaries between the two classes. They are the nearest points to the hyperplane. Figure 4.5 illustrates the feature space of a binary SVM. The support vectors are highlighted in gray.

4.2.3 Multi-Class Support Vector Machine

A multi-class SVM distinguishes between multiple classes. In general, a multi-class SVM is realized as multiple two-class SVMs [WW$^+$98, DK05]. Weston et al. [WW$^+$98] and Duan et al. [DK05] explained two methods for implementing this approach: the one-versus-all and the one-versus-one method. The first method uses a winner-takes-all strategy and the latter method uses max-wins voting. One-versus-all makes use of the training examples of one class trained as positive. The vectors of all other classes are trained as negative. One-versus-one compares each

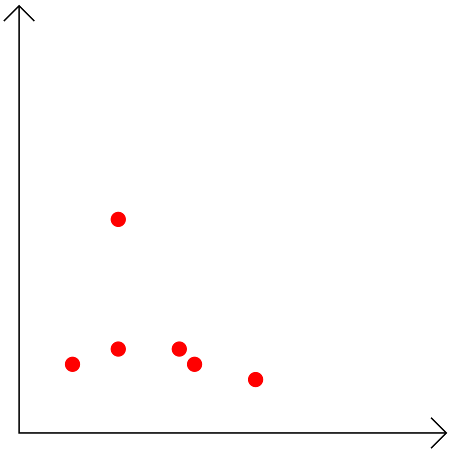


Figure 4.6: Training of the first class

class with each other. If one class is selected to be correct for an input test vector, the vote of this class is incremented. After all binary SVMs have compared two classes and made their vote, the class with maximal vote is chosen.

4.2.4 Training

Hsu et al. [HCL08] presented a practical guide for training and testing data with SVMs. They clarified that data has to be split in training and testing data. A set of training data must be learned so that the SVM can make a decision. Hsu et al. expalined that each feature vector which is trained has to be labeled with one class. The SVM can classify new input vectors with this training. If more vectors are learned to belong to a specific class, the output is more precise. It is also important to train vectors of border cases so that the SVM is robust against them.

Figures 4.6 and 4.7 illustrate the training of feature vectors of two different classes. Feature vectors are demonstrated with colored dots. The color indicates the related class.

Figure 4.8 visualize the construction of the optimal hyperplane in green. The hyperplane separates the trained classes. The width of the margin to the support vectors is illustrated with orange lines. The support vector of each class are marked with a yellow dot in the middle.

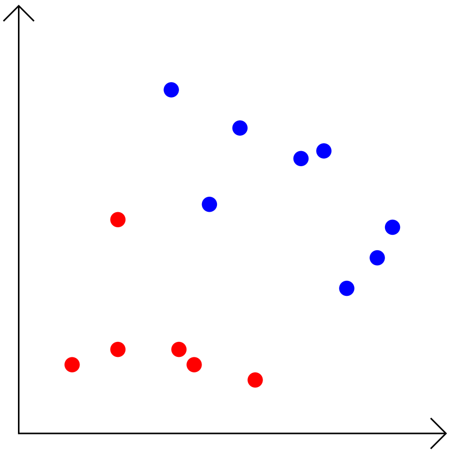


Figure 4.7: Training of the second class

4.2.5 Classification

Input vectors which have to be classified are mapped to the same feature space which was constructed by the training vectors [CV95]. By means of the stretched hyperplane, the position of the input vector shows its correspondent class. Depending on which side of the hyperplane the input vector is located, the SVM classifies it as one of the two classes.

Figure 4.10 shows the feature space of the training data. A feature vector of test data is illustrated as a lilac dot. The vector is classified as the same class like the vectors in blue because it is located in the same area cunstructed by the hyperplane.

4.2.6 Non-Linear Support Vector Machine

Several datasets may not be distinguishable with a linear hyperplane. Osuna et al. [OFG97b] explained that a non-linear classifier is most useful. The feature vectors are mapped to a higher dimensional feature space [OFG97b, HCL08]. Thus, the classes can be separated by a linear surface. Afterwards the vectors are mapped to the prior feature space with a resulting non-linear hyperplane. This can be seen in Figure 4.10. In this example, training data is ordered circularly to demonstrate a non-linear classification problem.

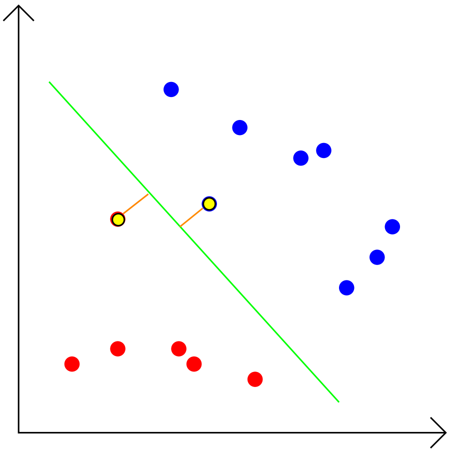


Figure 4.8: Construction of the optimal hyperplane

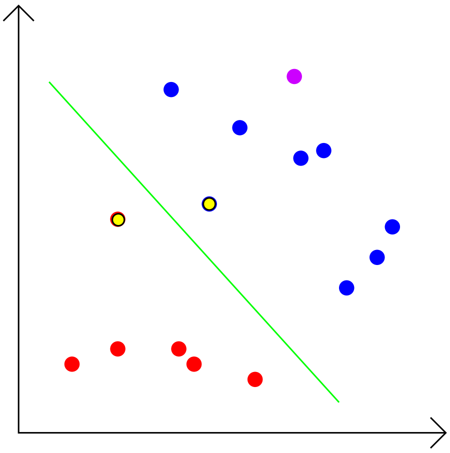


Figure 4.9: Classification of test data

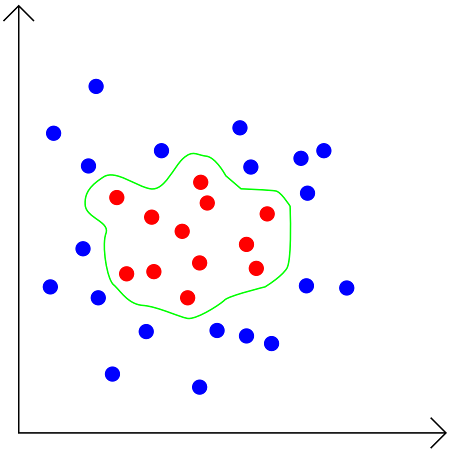


Figure 4.10: Non-linearly separable training data

4.2.7 Summary of Support Vector Machines

SVMs play an important role in classification. They can be used for one-class and multi-class problems. The system introduced utilizes SVMs for AU recognition and emotion classification. Extracted facial parameters are classified to an AU or an AU-combination. The set of all AUs recognized in the face is classified to an emotion.

4.3 AdaBoost

This Section gives more detailed information about Adaboost. This thesis focuses on the basic concept of Adaboost. Further information can be found in [FS95, Sch03].

AdaBoost is a machine learning algorithm which is used for classification and feature selection. Freund and Schapire [FS95] developed the algorithm in the year 1995.

4.3.1 General Structure

AdaBoost is based on a weak learning algorithm which has a slightly smaller error rate than random guessing [Sch03]. AdaBoost invokes this weak learner frequently. On every step the weak learner is invoked, examples of the training set have different weights. Most weight is placed on those examples which are harder to classify. Freund and Schapire explained that this method allows each step to focus on examples which were hard to classify in the step before. After many steps, a strong classifier is built up to a single prediction rule. This is done by taking the weighted majority vote of the predictions of the rules from the weak learners.

A weak learner can be any learning algorithm like Haar-like features or Gabor features. Thereby, AdaBoost is a general boosting algorithm [Sch03].

4.3.2 Binary Classification

Algorithm 1 describes the structure of a binary classification of AdaBoost.

The algorithm receives a sequence of labeled data as input. Freund and Schapire defined x_i as a feature vector and y_i as its label of either -1 or $+1$. A weak learning algorithm and a number of iteration are chosen. A first distribution is initialized to set weights of each training data equally. In every iteration step, one weak learner is trained on the basis of the weight distribution. The hypothesis of this weak learner is then used to calculate its error rate. The algorithm chooses the importance of the weak learner using this information. The smaller the error, the bigger its importance. The weight distribution is updated with Z_t

Algorithm 1 Classification algorithm of AdaBoost after [Sch03, FS95]

```
Input:
Data $(x_1, y_1), ..., (x_m, y_m)$ with $x_i \in X, y_i \in Y = \{-1, +1\}$
Weak learning algorithm
Integer T as length of iteration

Pseudocode:
Initialize distribution $D_1(i) = \frac{1}{m}$
for t = 1 to T do
    Train weak learner with distribution $D_t$
    Obtain hypothesis of weak learner $h_t : X \longrightarrow \{-1, +1\}$
    Calculate error of weak learner $\epsilon_t = Pr_{i\tilde{}D_t}[h_t(x_i) \neq y_i]$
    Choose importance of weak learner $\alpha_t = \frac{1}{2} ln(\frac{1 - \epsilon_t}{\epsilon_t})$
    Update distribution $D_{t+1}(i) = \frac{D_t(i) exp(\alpha_t y_i h_t(x_i))}{Z_t}$
end for

Output:
Final hypothesis $H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x_i))$
```

as a normalization factor as a last step. After this step, training examples which were classified falsely receive a higher weight. Freund and Schapire explained that thereby the weak learner of the next iteration step focuses on examples hard to classify. At the end of the algorithm, the final hypothesis is calculated as a weighted majority vote. To achieve this, every of the T weak learners is weighted with its importance.

4.3.3 Multi-Class Classification

Several extensions of AdaBoost exist to provide a multi-class classification [Sch03, FS95, FSA99, FS$^+$96]. They are defined as follows.

AdaBoost.M1

AdaBoost.M1 is used when the weak learning algorithm supplies not less than 50% accuracy on every step of the boosting algorithm. Each training example is assigned to a label of finite set. The multi-class problem has to be handled by the weak learning algorithm.

AdaBoost.M2

In the algorithm of AdaBoost.M2, the weak learner provides a vector as an output instead of only one label. The vector contains 1 on the position of a label which may be correct for the training example. It includes 0 when the label on this position is improbable. Thereby, the weak learner must not decide in favor of only one label to be correct. The weak learner centers not only on hard examples. It focuses on incorrect labels which are hard to define as well. This is done by regarding the distribution over pairs of examples instead of single ones. More detailed information is for example presented in [Sch03].

AdaBoost.MH

AdaBoost.MH uses binary classification for multi-class cases. This is done on the basis of the one-versus-all method. This method is also used for multi-class SVMs, as described above.

4.3.4 Summary of AdaBoost

AdaBoost plays an important role in several analysis methods. It can also be used for feature selection by its selection of hard training examples. The system presented in this thesis utilizes AdaBoost in combination with Haar-like features in a preexisting face detector.

The next Chapter describes the recognition system developed in this thesis.

Chapter 5

Automatic Facial Expression Recognition based on 2D Features

This Chapter describes the approach of an automated facial expression system developed in this thesis. To separate an automated recognition system from a non-automated, it is called AFERS like in [RCL$^+$09].

The system presented is a full AFERS, which means it is fully automatic. The user working with the interface of the system merely has to load an image. The user who integrates this AFERS to his own project, for example a robot system, singly invokes the starting function with an image path as parameter. All particular steps are invoked automatically from prior steps.

The system is implemented in C++ and built in the Robot Operating System (ROS) [1]. By means of ROS, the system can be easily embedded in existing robot systems. Several functions are adapted from the Open Source Computer Vision library (OpenCV) [2]. OpenCV is suitable for real time computer vision and integrated in ROS.

5.1 Robot Operating System

This Section gives a short introduction to the ROS architecture.

ROS is a meta-operating system for robots [3]. It offers plenty of libraries, hardware drivers, and tools. ROS is open-source and licensed under the BSD license.

The code system of ROS is divided into stacks. Stacks consist of packages. Packages again are built up of nodes. Nodes are executable files. These programs

[1] `http://www.ros.org/`

[2] `http://opencv.willowgarage.com/wiki/`

[3] `http://ros.org/wiki/ROS/Introduction`

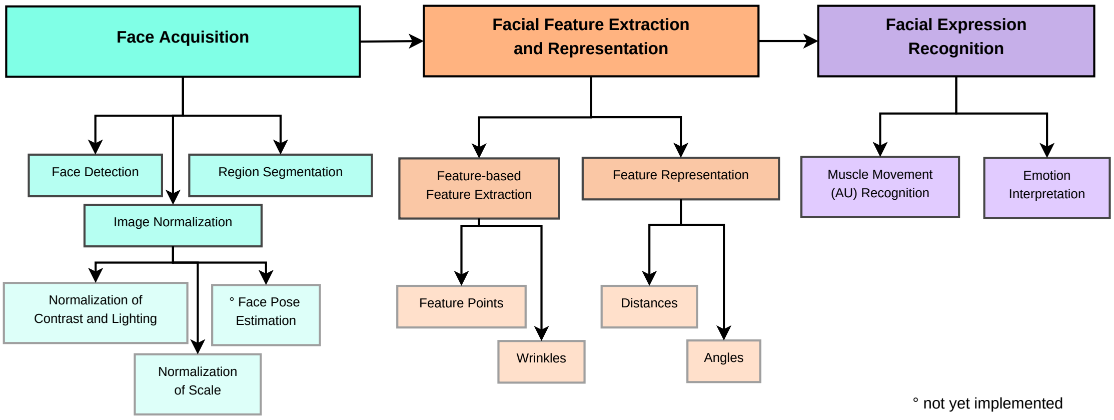


Figure 5.1: Organization of the system presented

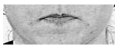
can communicate which each other through messages. This distribution offers the ability to add packages from outside to a preexisting system. The ROS core manages all nodes. Every node has to log on the core to communicate with other nodes. Each node is executed as a separate process.

This thesis utilizes the ROS release “fuerte”. On the basis of this system, the AFERS presented can be integrated in every system based on ROS.

5.2 Automatic Facial Expression Recognition System

The system presented in this thesis is a full AFERS. Figure 5.1 demonstrates its steps and their relationships. A special thing about this system is that it first recognizes the AUs in an image and then classifies these to emotions. Using this ability, it can be extended without large modification. Table 5.1 lists the AUs used in this system. Another special thing that differs from related work is the ability to recognize emotions on the actual image without any prior knowledge. Merely the SVMs have to be learned before tested. This can be done once. The advantage of this approach is that this system recognizes person independent emotions because different people are used for training.

Table 5.1: AUs used in the system presented

AU	Example Image	Description
1		Inner Brow Raiser
2		Outer Brow Raiser
4		Brow Lowerer
5		Upper Lid Raiser
6		Cheek Raiser
7		Lid Tightener
9		Nose Wrinkler
10		Upper Lip Raiser
12		Lip Corner Puller
15		Lip Corner Depressor
17		Chin Raiser
20		Lip stretcher
23		Lip Tightener
24		Lip Pressor
25		Lips part
26		Jaw Drop

5.2.1 Face Acquisition

The step of face acquisition detects and extracts the face region from the entire input image. Furthermore, image normalization and region segmentation is done.

Face Detection

Face detection with the OpenCV face detector is the first step of the proposed AFERS. The face detector uses the Viola-Jones algorithm [VJ01b] as it is implemented in the OpenCV Haar feature-based cascade classifier for object detection [4]. The Viola-Jones algorithm is further extended by Lienhart et al. [LM02] [5]. AdaBoost is used in a small modification selecting the most important features. In-plane and out-of-plane rotations are detected to a certain degree. The OpenCV face detector *cv::CascadeClassifier::detectMultiScale* [6] works on gray scale images. Hence, the input image is converted into gray scale at first and then normalized with the OpenCV histogram equalization function [7]. A normalization as a step before face detection improves detecting results. Additionally, the output of the OpenCV face detection is extended to detect the chin as well.

Multiple faces may occur in the input image. For this reason, an algorithm is developed which chooses the closest face for further processing. Algorithm 2 specifies this detection. The OpenCV face detector provides a vector of detected faces. The first element of this vector is selected as reference. The algorithm compares width and height of each detected face with the reference face. If width and height are greater than these of the reference, the actual face is allocated as a new reference.

[4] `http://opencv.willowgarage.com/documentation/object_detection.html`

[5] `http://pr.willowgarage.com/wiki/Face_detection`

[6] `http://docs.opencv.org/modules/objdetect/doc/cascade_classification.html#cascadeclassifier`

[7] `http://docs.opencv.org/modules/imgproc/doc/histograms.html#equalizehist`

Algorithm 2 Detection of the closest face

```
Input:
Vector of detected faces ← $faces$

Pseudocode:
$max$ ← first element of $faces$
$w_{\texttt{max}}$ ← width of first element of $faces$
$h_{\texttt{max}}$ ← height of first element of $faces$
for all Faces $f$ of $faces$ do
  $w \leftarrow f.width$
  $h \leftarrow f.height$
  if $w > w_{\texttt{max}} \wedge h > h_{\texttt{max}}$ then
    $w_{\texttt{max}} \leftarrow w$
    $h_{\texttt{max}} \leftarrow h$
    $max \leftarrow f$
  end if
end for
Closest face ← $max$

Output:
Closest face
```

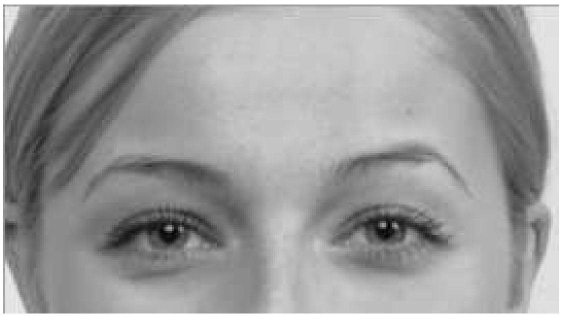


Figure 5.2: First region of interest (ROI1)

Image Normalization

Image normalization as a next step starts with facial size normalization. The approached system uses the OpenCV scaling function [8]. The size normalization resizes all detected faces to 270 x 300 pixels. This size is checked as an optimal size for not losing information and not being too tall for further operations.

Face pose estimation and rotation handling is not treated in this thesis. An approach to deal with this problem is described in the future work Section of Chapter 7.

Region Segmentation

Region segmentation is predicated on model-based rules. For all region segmentation steps used in this system, the proportion doctrine of Leonardo da Vinci is used as foundation. With these rules, the face can be divided into certain parts, as mentioned in Chapter 3. The detected face is segmented into two regions in this thesis. The upper part of the face contains eyebrows and eyes (ROI1). The lower part consists of nose, mouth, and chin (ROI2). Figures 5.2 and 5.3 picture the two ROIs. The ROIs are separated at the half size of the detected face.

The two ROIs are further segmented to avoid false feature point detection. This division conduces only to the feature point and wrinkle detection. After these steps, it is abolished and the feature vectors are computed on the basis of one whole ROI. ROI1 first consists of eyebrows and eyes, as well as the hair and face outline. The two latter components distort feature detection. The system presented removes them by finding a permitted zone which includes only eyebrows and eyes. The same is done with ROI2 to find the nose and mouth. The permitted zones are built up of facial parts shown in Figure 5.4. The regions are separated by

[8] `http://docs.opencv.org/modules/imgproc/doc/geometric_transformations.html#resize`

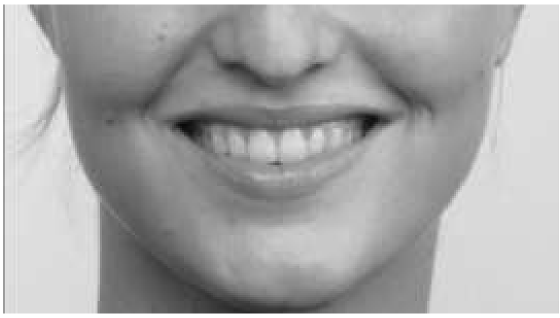


Figure 5.3: Second region of interest (ROI2)

the middle green line which is located in the half of the face height. The permitted zones lie $\frac{1}{3}$ of face height above and below the separation line.

Half of the permitted zone is computed to split ROI1 into eyebrows and eyes. Components above this line pertain to the eyebrows. Components below the line form the eyes. The same division is done in ROI2. Figures 5.6 and 5.7 show lines and components of each ROI. Splitting the zones at half height comes from the geometrical approach of Leonardo da Vinci as well as the ROI segmentation. Figure 5.5 illustrates the division computing. As described above, the face is first divided at half height. One third of face height above and below this line are the boundaries of the permitted zones. Both zones are divided at half of their height.

In order to find wrinkles in the face, facial components are further split. The most obtrusive wrinkles which provide information about emotion occur between the eyebrows as well as between the mouth and nose. This information is gained by observation of the emotions presented in the database. Relying on the model-based approach, the wrinkle zones are located as Figure 5.8 visualizes.

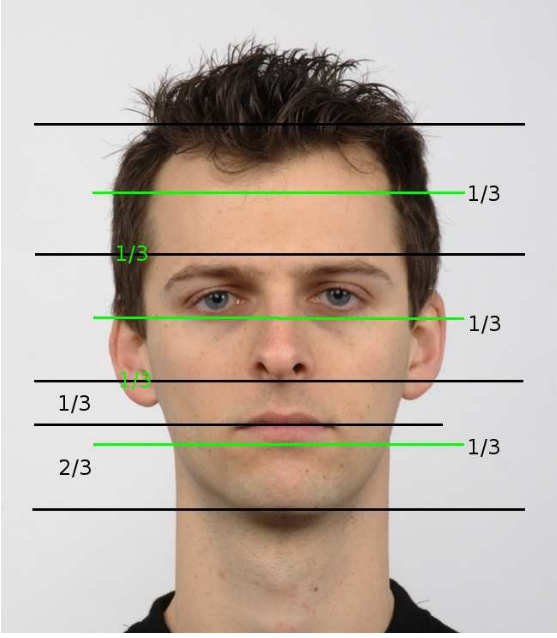


Figure 5.4: Permitted zones in the face

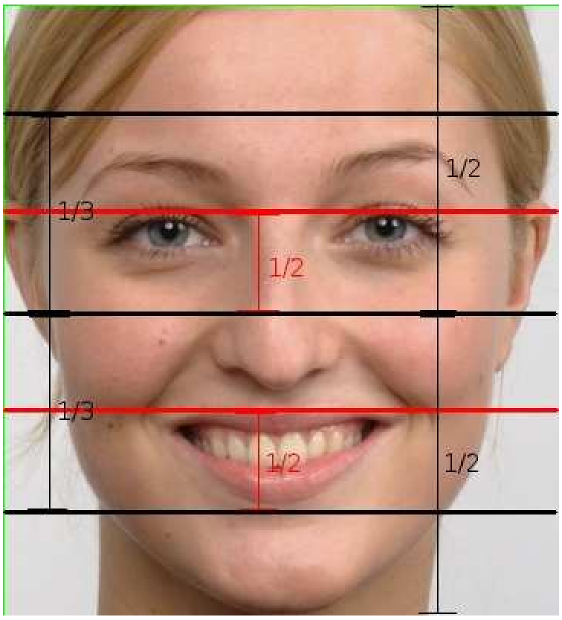


Figure 5.5: Division computing of the face into ROIs and their components

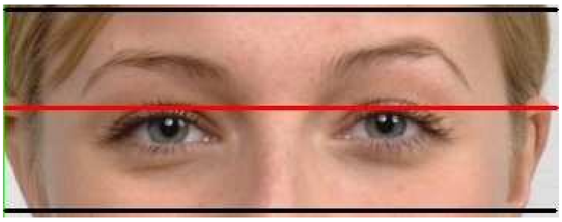


Figure 5.6: Division of ROI1 into eyebrows and eyes

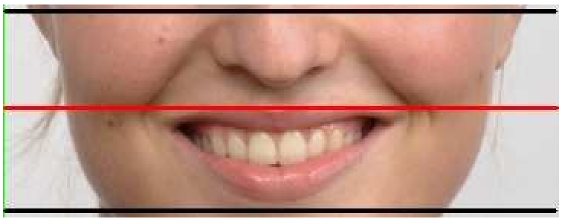


Figure 5.7: Division of ROI2 into nose and mouth

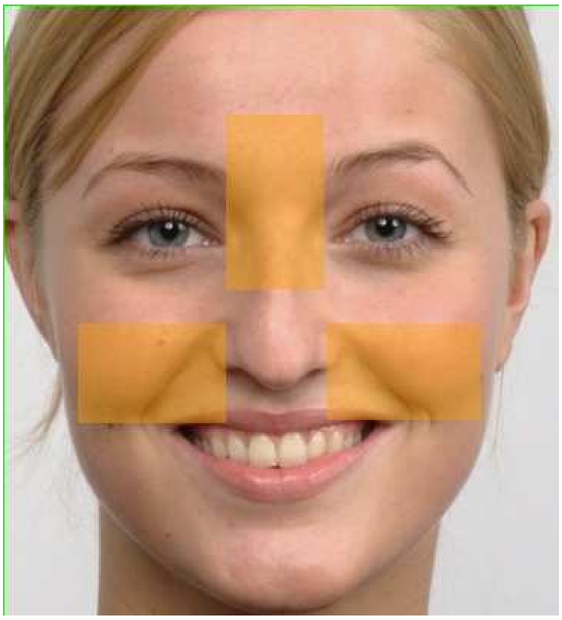


Figure 5.8: Zones in the face where wrinkles are detected

5.2.2 Facial Feature Detection, Extraction and Representation

This step of the system detects facial feature points and extracts facial parameters. A feature vector assembles these parameters to represent a facial expression.

Facial Feature Points

As described in Chapter 3, Zhang et al. [ZJZY08] specified the relationship between the distances of the MPEG-4 standard and the AUs of FACS. On the basis of the 16 AUs used in this system, Table 5.2 shows which of the feature points are relevant for this approach. Considering the fact that this thesis regards each ROI separately, some distances of Zhang et al. have to be modified. This is described in the Section of facial feature representation in this Chapter. Furthermore, this system does not use the points 3.5 and 3.6 which are located in the center of the pupil. The points 5.3 and 5.4 are the left and right cheek bone. These points are also not used, and raising cheeks is measured another way. At last points 9.13 and 9.14 which lie on the lower edges of the nose bones are not employed.

The introduced AFERS uses 19 facial feature points of the MPEG-4-standard. Figure 5.9 visualizes the used feature points on a picture of the standard. Figure 5.10 shows the same points on a person of the database used in this system.

Figure 3.7 of Chapter 3 shows the feature points of the MPEG-4 standard. The standard specifies point 4.3 and 4.4 as the uppermost point of the eyebrow [PF02]. It defines the x-coordinate of point 4.4 as

$$4.4.x = \frac{(4.2.x + 4.6.x)}{2} \tag{5.1}$$

or the x-coordinate of the uppermost eyebrow point, for $4.3.x$ equivalent. The introduced system utilizes the first definition. Equation 5.1 defines point 4.4 to lie in the middle of point 4.2 and point 4.6 considering the x-axis. The second definition may evoke errors especially considering the sad emotion where the uppermost point may be equal to the innermost point.

This approach consults both eyebrows and both eyes although all seven facial expressions are symmetric. Feature points are detected automatically and this may evoke errors. To avoid detecting errors, the system handles both sides. The corresponding parameters are compared and the median computed. Experiments show if this approach achieves more accurate results than without this comparison.

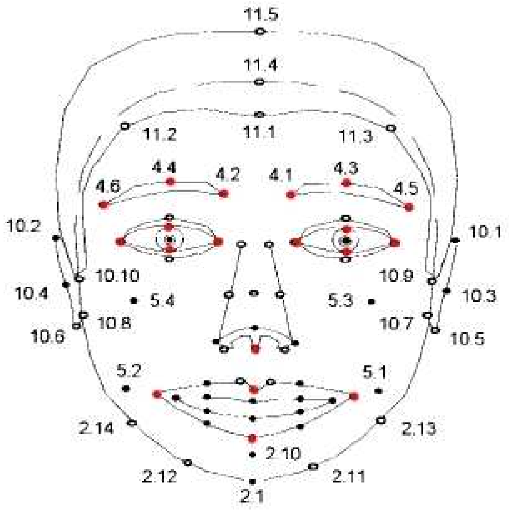


Figure 5.9: Marking of the MPEG-4 feature points used in this system on a picture as presented in [CH06]

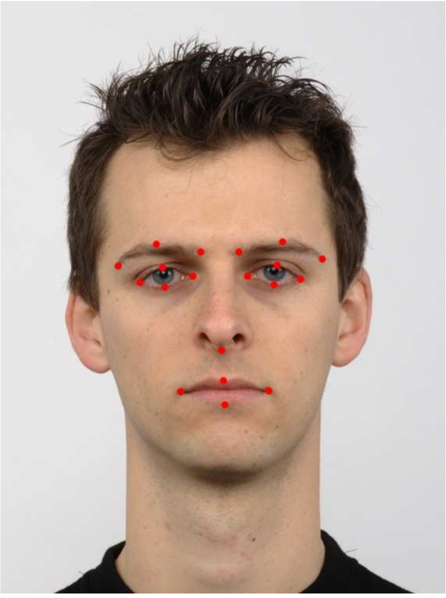


Figure 5.10: Marking of the MPEG-4 feature points used in this system on a person of the database

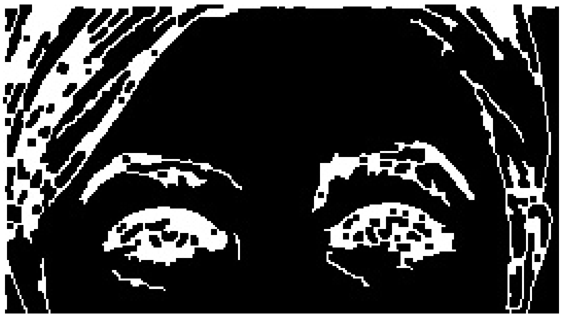


Figure 5.11: Canny and Closing executed on ROI1

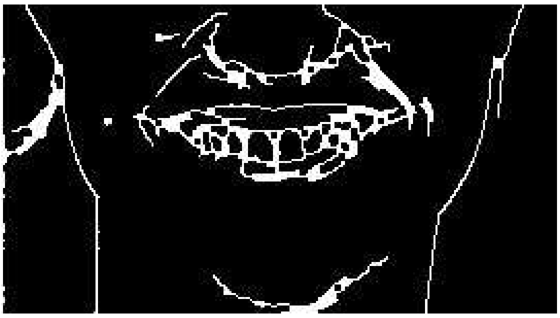


Figure 5.12: Canny and Closing executed on ROI2

Facial Feature Detection and Extraction

In the second step, facial feature detection, extraction, and representation are done. The following part describes the methods used for every step. First, the feature points have to be detected in the face.

ROIs are converted into a binary image and edges are detected by the OpenCV Canny function [9] for image preprocessing. Contour enhancement is achieved by the morphology closing function [10]. Figures 5.11 and 5.12 show the result of both functions.

The OpenCV findContours algorithm [11] provides every contour detected by the Canny operator. The algorithm was developed by Suzuki et al. [SA85] in the year 1985. Contours are detected by boundary tracing. Figures 5.13 and 5.14 show the resulting images of ROI1 and ROI2.

[9] `http://docs.opencv.org/modules/imgproc/doc/feature_detection.html#canny`

[10] `http://docs.opencv.org/modules/imgproc/doc/filtering.html#morphologyex`

[11] `http://docs.opencv.org/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html#findcontours`

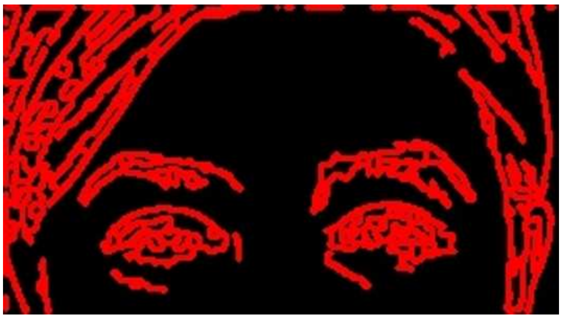


Figure 5.13: FindContours algorithm executed on ROI1

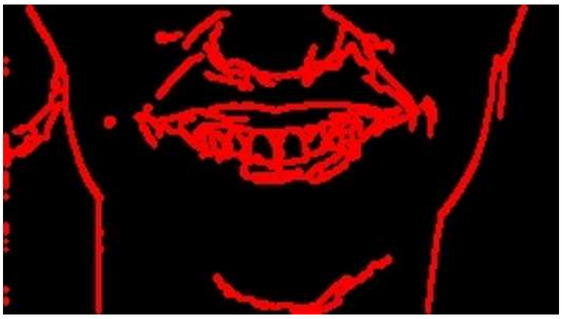


Figure 5.14: FindContours algorithm executed on ROI2

An algorithm is developed to remove hair which may occur in the permitted zone of ROI1. Algorithm 3 specifies the detection of permitted contours in one ROI. Every contour is tested if the majority of its points lie in or beyond the permitted zone. By means of this algorithm, hair is removed and eyebrows or eye wrinkles which poke out of the zone are tolerated. Figures 5.15 and 5.16 visualize the permitted zones of ROI1 and ROI2.

Remaining facial components are classified by means of the model-based approach described above. In order to detect and analyze transient features, the wrinkle regions of the two ROIs are extracted as well. At this point, the face consists of contours belonging either to the left eyebrow, right eyebrow, left eye, right eye, wrinkles between the brows, nose, mouth or to the wrinkles between nose and mouth. Facial features are detected on each of these components. Figures 5.17 and 5.18 illustrate the facial components of ROI1 and ROI2. The upper components are colored in purple and the lower ones in blue. Wrinkles which lie in the wrinkle zones visualized by orange rectangles are colored in yellow. The red line marks the division line by which the components are separated. Green colored contours are not considered in next steps. Thereby, small components like birthmarks or

Algorithm 3 Detection of permitted contours in one ROI

```
Input:
Vector of detected contours ← $contours$
Permitted zone ← $zone$

Pseudocode:
Counter of points of one contour ← $counter$
for all Contours $c$ of $contours$ do
  $counter \leftarrow 0$
  for all Points $p$ of $c$ do
    if $p.x \geq zone.x \wedge p.x \leq zone.x + zone.width$ then
      if $p.y \geq zone.y \wedge p.y \leq zone.y + zone.height$ then
        $counter \leftarrow counter + 1$
      end if
    end if
  end for
  if $counter > contours.size * 0.5$ then
    Permitted contour ← $c$
  end if
end for

Output:
Permitted contour
```

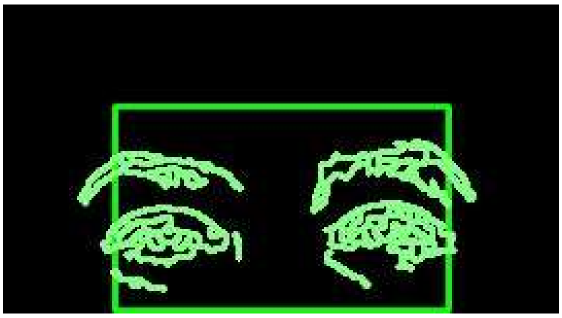


Figure 5.15: Permitted zones defining permitted contours of ROI1

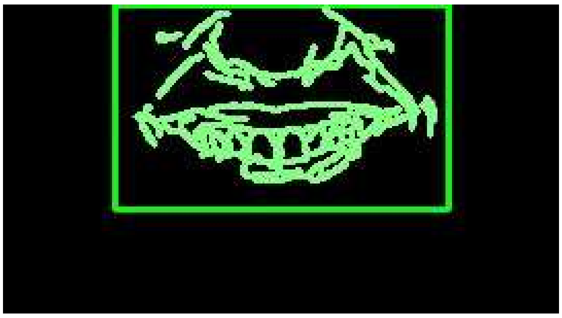


Figure 5.16: Permitted zones defining permitted contours of ROI2

impurities are eliminated. A threshold is chosen by experiments to regulate false contours. Merely contours with over nine points provided by the findContours algorithm are classified to facial components. In an analogous manner, contours are associated with wrinkles. Merely contours which consist of over 40 points and which are located in the wrinkle zones are classified as wrinkles.

After the extraction of all facial components, feature points are detected on each. This paragraph provides an example of the geometric-based feature detection. To detect point 4.4 in the face, the uppermost point in the region of the right eyebrow is found. Therefore, the lowest value on the y-axis is computed running through all contours belonging to this component. Algorithm 4 specifies the detection of point 4.4. At this point of computation, the feature points 4.2 and 4.6 are known. First, any point of the eyebrow is allocated to point 4.4 and the y-value is defined as the number of columns in the image matrix. Thereby, the reference point is a point anywhere on the x-axis with a maximum y-value. For every point on the contours of the right eyebrow, the x-value is observed to lie in the middle of the two outer eyebrow points 4.2 and 4.6. A variance of two ensures that at least

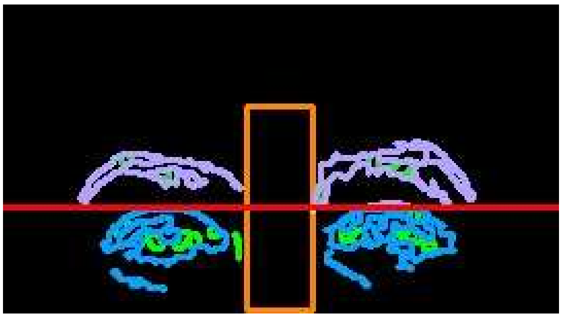


Figure 5.17: Extracted facial components and wrinkles of ROI1

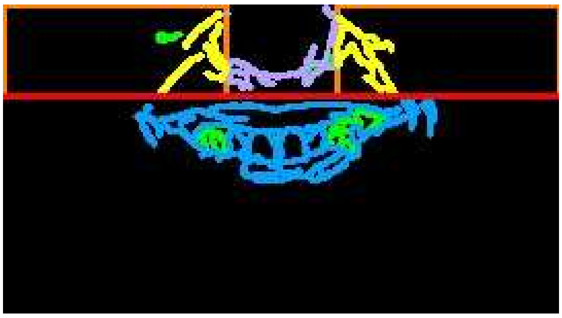


Figure 5.18: Extracted facial components and wrinkles of ROI2

one point is found. The y-value of this point is compared to the reference point. Thus, the point with a minimum y-value is found.

Figures 5.19 and 5.20 show the detected feature points of ROI1 and ROI2. Figure 5.21 illustrates the same feature points marked manually for a comparison of exactness. The colors serve as auxiliary means for the comparison.

Algorithm 4 Detection of feature point 4.4

```
Input:
Contours of right eyebrow ← $eyebrow_{\texttt{right}}$
Point 4.2 ← $p_{42}$
Point 4.6 ← $p_{46}$
Pseudocode:
$p_{44}$ ← first point of first contour
$p_{44}.y$ ← number of columns in the image
for all Contours $c$ of $eyebrow_{\texttt{right}}$ do
  for all Points $p$ of the contour $c$ do
    if $p.x \leq 0.5 * (p_{42}.x + p_{46}.x) + 2 \wedge p.x \geq 0.5 * (p_{42}.x + p_{46}.x) - 2$ then
      if $p.y < p_{44}.y$ then
        $p_{44} \leftarrow p$
      end if
    end if
  end for
end for
Output:
Point 4.4 ← $p_{44}$
```

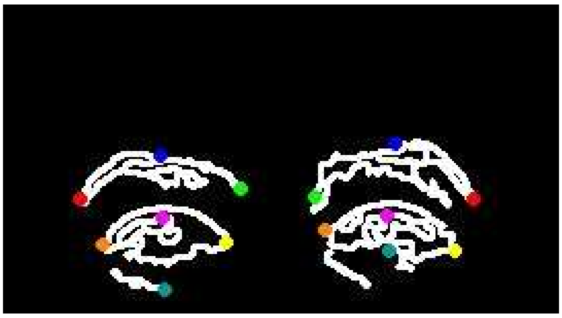


Figure 5.19: Detected feature points of ROI1

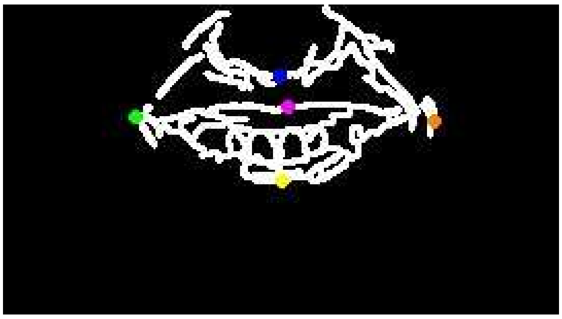


Figure 5.20: Detected feature points of ROI2

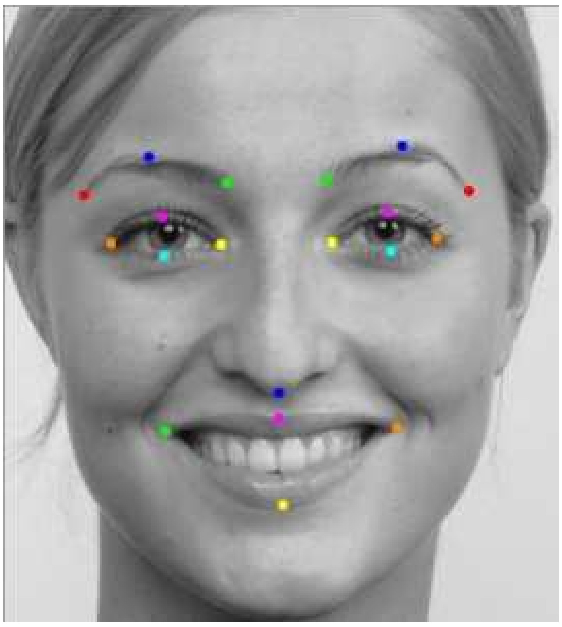


Figure 5.21: Feature points of a face

Facial Feature Representation

As described in Chapter 3, Esau et al. [EWKK07] suggested the method to represent a face only with angles as robust against individual changes. This approach is compared to distance parameters which are used commonly.

The first approach uses only distances as parameters. These distances are between the feature points detected in the step before. In the proposed system, 16 AUs are used for emotion classification. Table 5.2 shows relevant distances for detecting these AUs. Distance $d_{\mathtt{n}}.x$ is measured in the x-direction and $d_{\mathtt{n}}.y$ in the y-direction of the same two points.

There are nine basic distances. Five distances on the second facial half serve either as a comparison or as further distance parameters. Figure 5.22 illustrates the distances of the face. Although the distances are measured in the x or y direction, for simplification the figure shows one straight line per distance. Distances with the same color are symmetrical. The color indicates the name of the distance which can be seen in the legend. All names with associated computation can be found in Table 5.2. Seven distances are modified from Zhang et al. to fit to the requirements of this system. The approach presented utilizes distances in each ROI. The ROIs are considered separately. For AU5 and AU7, the entire eye height is used. It is computed by means of distances d_{31} and d_{32}. AU10 and AU12 are measured only in ROI2. The computation of AU24 is simplified by computing only d_{61}. Furthermore, AU6, AU9, and AU17 are basically treated as wrinkles. Only particular distances are used to improve wrinkle detection results.

The appearance of wrinkles is represented as a 1-0-state. This means the wrinkles are either present or absent like in the recognition system of Tian et al. [TKC01, TKC05].

The second approach uses only angles as parameters. The feature points stay the same, but this time no distances are measured. Six basic angles are extracted on the basis of the angles of the approach of Esau et al. [EWKK07]. One additional angle helps to avoid errors. Five angles on the second facial half serve either as a comparison or as further angle parameters. Figure 5.23 illustrates the angles of the face.

The positions of the angles A_0, A_2, and A_4 from Esau et al. are adopted. The angle a_{51} of the system presented matches A_0. a_{01} and a_{02} are consistent with A_2. The angles a_{41} and a_{42} correspond to A_4. The other angles are chosen to detect relevant information about the 16 AUs. For example, the emotion fear does not always imply an open mouth. Furthermore, Esau et al. utilized no angle to detect open eyes in a surprised facial expression. The system introduced provides this information by computing the angles a_{21}, a_{22}, a_{31}, and a_{32}.

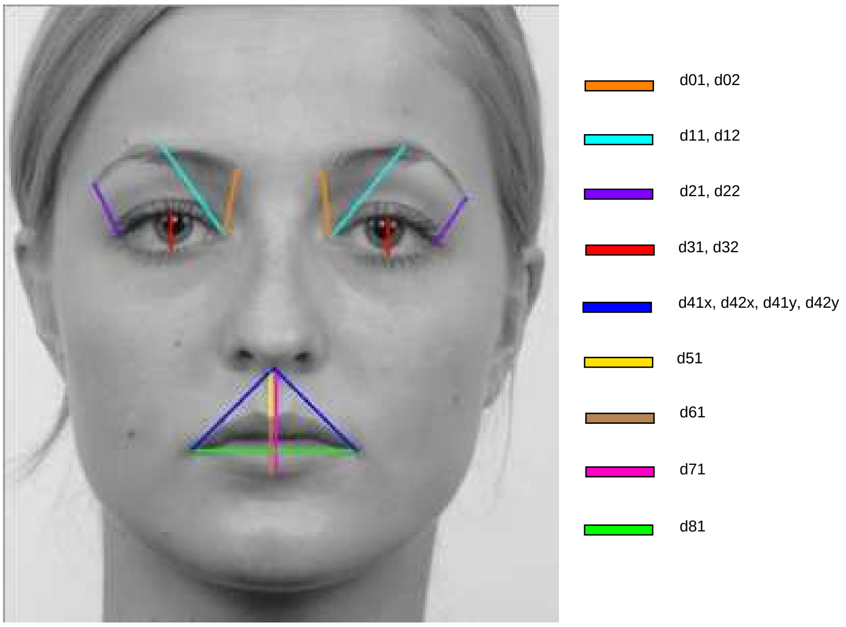


Figure 5.22: Distance parameters of a face

Angles are computed by means of dot product between two vectors. The vectors are computed on distances between feature points. Figure 5.24 visualizes this relationship.

For the computation of the angle a', a function is developed which receives three objects of OpenCV Point [12] as parameters. The first point P_1 should be the starting point of the two vectors. The angle at this corner will be computed. With the second and third parameter P_2 and P_3, both vectors $\boldsymbol{a} = \overline{\mathsf{P}_1\mathsf{P}_2}$ and $\boldsymbol{b} = \overline{\mathsf{P}_1\mathsf{P}_3}$ are calculated. The algorithm computes the dot product of both vectors and each magnitude. The function to calculate the angle from these values is as follows:

$$a' = \arccos\left(\cos\left(\frac{\boldsymbol{a}\cdot\boldsymbol{b}}{|\boldsymbol{a}| * |\boldsymbol{b}|}\right)\right) \tag{5.2}$$

The approach presented utilizes angles in each ROI. The ROIs are considered separately. As in the first approach, the appearance of wrinkles is represented as a 1-0-state.

The output of both approaches is a feature vector which represents the actual facial expression. This feature vector will be analyzed in the next step.

Both approaches of feature representation are tested to work without use of a reference face. A common method is to use the neutral expression as a reference,

[12] http://docs.opencv.org/modules/core/doc/basic_structures.html#point

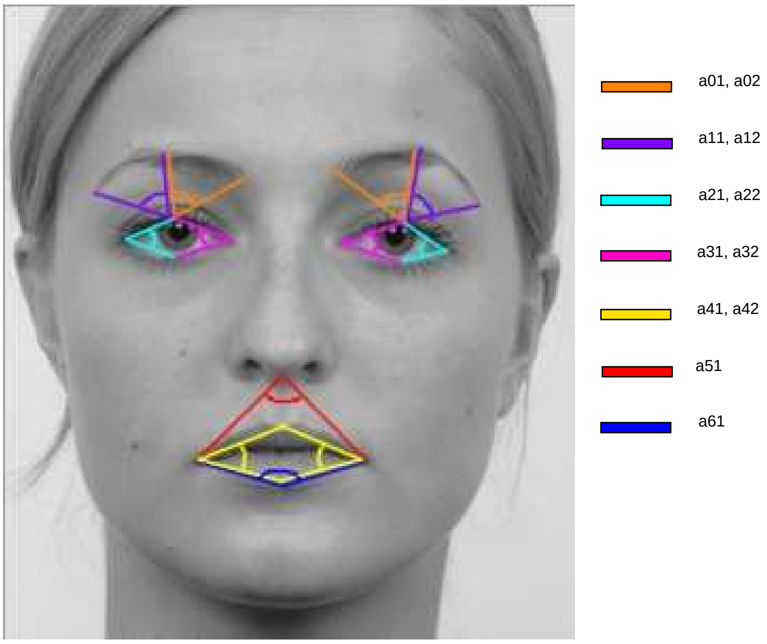


Figure 5.23: Angle parameters of a face

as it is done in [EWKK07]. With this method, the extracted parameters are compared to the parameters on the neutral face. The advantage of this method is that emotions of even very diverse people are recognized. Fasel et al. [FL03] pointed out that wrinkles which appear permanently in the face are not detected as transient features by means of a reference image. With this ability, a person with pronounced frown lines is not wrongly analyzed as angry or disgusted. On the other hand, a disadvantage of this method is that the neutral face has to be learned for every person who will be analyzed. Hence, emotion recognition needs more time and does not work on strangers. In addition, a database has to be built up for the system to save neutral faces. Otherwise the system has to learn the neutral expression of every person every time new. Without required neutral faces, the neutral expression is handled like every other emotion. Furthermore, the system recognizes emotions like humans. If a person has pronounced frown lines, another person would consider him as angry although he is not. If a person has a friendly neutral face with raised lip corners, another person would consider him as happy although he is in a neutral mood. In this thesis, a system is developed which behaves like humans looking at strangers and recognizes the current emotion without any previous knowledge.

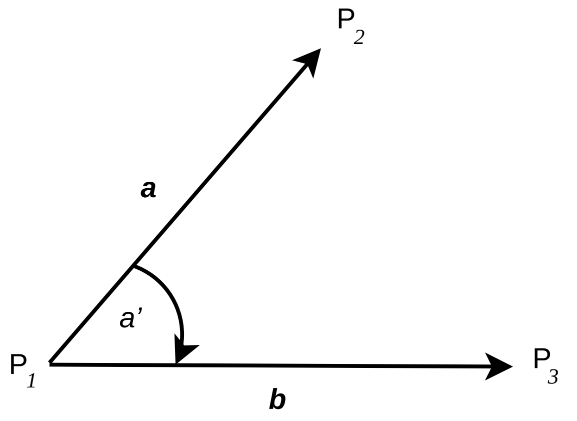


Figure 5.24: Relationship between three points, two vectors and an angle

Table 5.2: Distances between Feature Points and AUs used in this Thesis

Distances	Distance of Two Feature Points	AU
$d_{\mathtt{01}}$	$D_{\mathtt{y}}(4.2, 3.8)$	AU1
$d_{\mathtt{02}}$	$D_{\mathtt{y}}(4.1, 3.11)$	
$d_{\mathtt{21}}$	$D_{\mathtt{y}}(4.6, 3.12)$	AU2
$d_{\mathtt{22}}$	$D_{\mathtt{y}}(4.5, 3.7)$	
$d_{\mathtt{01}}$	$D_{\mathtt{y}}(4.2, 3.8)$	AU4
$d_{\mathtt{02}}$	$D_{\mathtt{y}}(4.1, 3.11)$	
$d_{\mathtt{11}}$	$D_{\mathtt{x}}(4.4, 3.8)$	
$d_{\mathtt{12}}$	$D_{\mathtt{x}}(4.3, 3.11)$	
$d_{\mathtt{31}}$	$D_{\mathtt{y}}(3.4, 3.2)$	AU5
$d_{\mathtt{32}}$	$D_{\mathtt{y}}(3.3, 3.1)$	
$d_{\mathtt{31}}$	$D_{\mathtt{y}}(3.4, 3.2)$	AU6
$d_{\mathtt{32}}$	$D_{\mathtt{y}}(3.3, 3.1)$	
$d_{\mathtt{31}}$	$D_{\mathtt{y}}(3.4, 3.2)$	AU7
$d_{\mathtt{32}}$	$D_{\mathtt{y}}(3.3, 3.1)$	
$d_{\mathtt{41}}.y$	$D_{\mathtt{y}}(8.4, 9.15)$	AU9
$d_{\mathtt{42}}.y$	$D_{\mathtt{y}}(8.3, 9.15)$	
$d_{\mathtt{51}}$	$D_{\mathtt{y}}(8.1, 9.15)$	AU10
$d_{\mathtt{41}}.y$	$D_{\mathtt{y}}(8.4, 9.15)$	AU12
$d_{\mathtt{42}}.y$	$D_{\mathtt{y}}(8.3, 9.15)$	
$d_{\mathtt{41}}.x$	$D_{\mathtt{x}}(8.4, 9.15)$	
$d_{\mathtt{42}}.x$	$D_{\mathtt{x}}(8.3, 9.15)$	
$d_{\mathtt{41}}.y$	$D_{\mathtt{y}}(8.4, 9.15)$	AU15
$d_{\mathtt{42}}.y$	$D_{\mathtt{y}}(8.3, 9.15)$	
$d_{\mathtt{71}}$	$D_{\mathtt{y}}(8.2, 9.15)$	AU17
$d_{\mathtt{81}}$	$D_{\mathtt{x}}(8.4, 8.3)$	AU20
$d_{\mathtt{81}}$	$D_{\mathtt{x}}(8.3, 8.4)$	
$d_{\mathtt{71}}$	$D_{\mathtt{y}}(8.2, 9.15)$	
$d_{\mathtt{81}}$	$D_{\mathtt{x}}(8.4, 8.3)$	AU23
$d_{\mathtt{81}}$	$D_{\mathtt{x}}(8.3, 8.4)$	
$d_{\mathtt{61}}$	$D_{\mathtt{y}}(8.2, 8.1)$	AU24
$d_{\mathtt{61}}$	$D_{\mathtt{y}}(8.2, 8.1)$	AU25
$d_{\mathtt{71}}$	$D_{\mathtt{y}}(8.2, 9.15)$	
$d_{\mathtt{61}}$	$D_{\mathtt{y}}(8.2, 8.1)$	AU26
$d_{\mathtt{71}}$	$D_{\mathtt{y}}(8.2, 9.15)$	

5.2.3 Facial Expression Recognition

The presented two-phase system first recognizes AUs in the face and then interprets the emotion from them.

AU Recognition using SVMs

The facial feature extraction delivers two different feature vectors of one face. The first vector contains all parameters belonging to the first ROI. The second includes the parameters of ROI2. One feature vector is analyzed by one SVM. The advantage of this approach is the modularity, as described by Bettadapura et al. [Bet12]. They pointed out that the failure of one classifier does not imply a wrong classification. Therefore, occlusion errors can be avoided.

The first parameter vector is the input of a seven-class SVM. The second vector is classified by a second seven-class SVM. Seven classes indicate seven emotions. Six AUs of the 16 used in this thesis lie in ROI1. AU 9 belongs to both ROIs. The other nine are located in ROI2. Each SVM classifies each assigned vector to a specific AU or to an AU-combination. The output of the first SVM can be either 0, 6, 9, 14, 125, 457, or 1245, related to the AUs of ROI1 for every emotion. The numbers are composed of single AUs where 14 does not mean AU14, but AU1 + AU4. The output of the second SVM can be either 0, 26, 1225, 1517, 2025, 91025, or 172324. The combinations can be seen in Table 5.3. AU combinations are handled as one AU like in the systems summarized in [TKC01], for example [DBH+99]. This is done because the database used in this thesis does not provide single AU images. Only combinations can be trained and tested.

The AUs which can be recognized in this system are only a part of the whole mass of FACS. To achieve accurate performance, only these AUs are considered which construct the six basic emotions. Langner et al. [LDB+10] defined 16 AUs which represent the six emotions the best. Table 5.3 shows this assignment. The neutral expression consists of no AU (AU0) because the face is relaxed and no muscle movement takes place.

Table 5.4 lists the wrinkle-states of both ROIs for each emotion. The states are gained by observation.

For implementation of the introduced approach, the OpenCV SVM function [13] is used. It is based on LibSVM, a library for support vector machines [CL01].

As described above, the output of each SVM in the presented system is a set of AU-codes, for example 1245. Altogether, AU recognition provides two sets of codes. The first defines the AUs detected in ROI1 and the second consists of the AUs detected in ROI2. These two vectors of AU-codes are passed to the emotion interpretation step.

[13] http://docs.opencv.org/modules/ml/doc/support_vector_machines.html

Table 5.3: 16 AUs representing basic emotions

Emotion	AU-codes of ROI1	AU-codes of ROI2
Neutral	AU0	A0
Happy	AU6	AU12 + AU25
Surprised	AU1 + AU2 + AU5	AU26
Sad	AU1 + AU4	AU15 + AU17
Fearful	AU1 + AU2 + AU4 + AU5	AU20 + AU25
Angry	AU4 + AU5 + AU7	AU17 + AU23 + AU24
Disgusted	AU9	AU9 + AU10 + AU25

Table 5.4: Wrinkle states representing basic emotions

Emotion	Wrinkle-state of ROI1	Wrinkle-state of ROI2
Neutral	0	0
Happy	0	1
Surprised	0	0
Sad	0	0
Fearful	0	0
Angry	1	0
Disgusted	1	1

Emotion Interpretation

Facial expression interpretation classifies recognized AUs into emotions. As mentioned before, this thesis focuses on the six basic emotions plus the neutral face. As described in Chapter 3, there are certain methods which can be used to classify input vectors of AU-codes. Langner et al. [LDB$^+$10] presented a rule-based method for expression classification. They assigned sets of AUs to eight emotions. As their database is used in this thesis, the mapping algorithm for emotion interpretation is created on the basis of this database. Figure 5.25 illustrates each emotion with respective AUs. Table 5.3 summarizes the sets of AUs for every emotion covered in this thesis.

A third SVM is used to avoid allocation errors. The SVM is trained with the mapping rules of Langner et al. mentioned above. It receives the AU vectors as input and defines the emotion. With the SVM, a wrong classification of one of the former SVMs should be handled correctly. For example, the output of the SVM of ROI1 is actually wrong, but the output of the SVM of ROI2 is accurately. However, the third SVM should classify the AUs correctly. The recognized emotion should be correct more times than the output of rule-based methods. SVMs classify the input vector to the most probable emotion. Experiments test this assumption in the next Chapter. Additionally, there is always an emotion output on the basis of a SVM, whereas a rule-based method may output no emotion when a matching is not found. The usage of this system in robotics should always guarantee an output which is most probable. An output supposably is better than no output.

ROI1 and ROI2 are analyzed in regard to their AUs in the AU recognition. Finally, the emotion interpretation assembles the AU vectors of each ROI. This entire feature vector is then classified with the third SVM to a specific emotion.

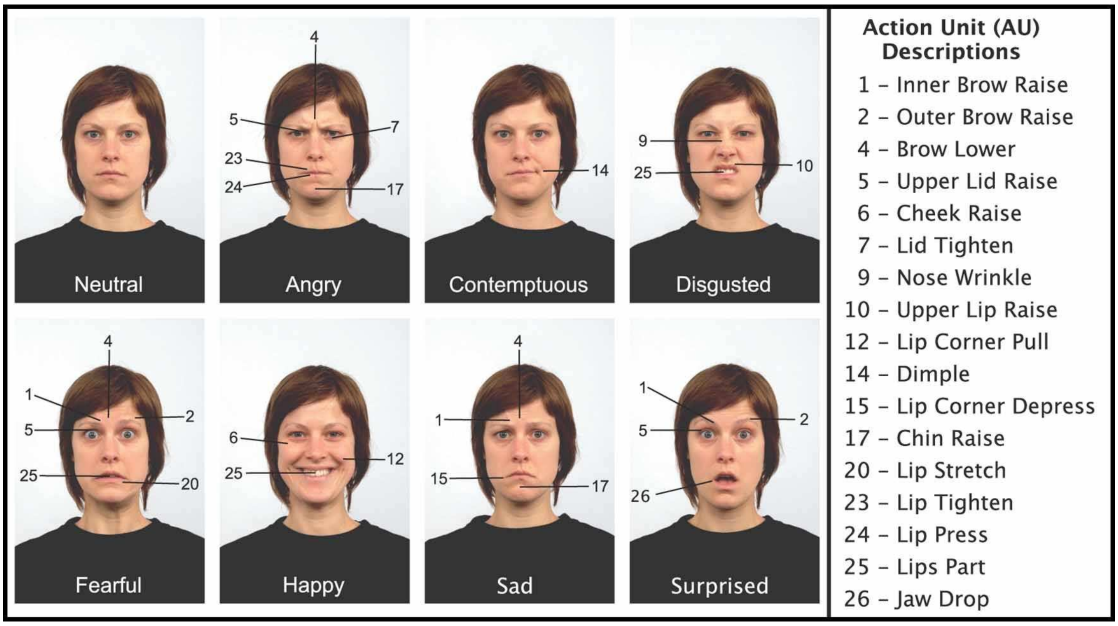


Figure 5.25: Mapping of AUs to emotions by [LDB+10]

Classification Example

This paragraph gives an example in order to understand the outputs and combinations of the three SVMs used for AU recognition and emotion interpretation.

The first SVM receives parameters of ROI1 which correspond to the combination AU1 + AU2 + AU4 + AU5. Hence, the output is 1245. In ROI2, parameters are extracted which are assigned to the combination AU20 + AU25. Thus, the output of the second SVM is the label 2025. The two outputs are used to fill a new feature vector $\boldsymbol{v}$.

$$\boldsymbol{v} = 1, 2, 4, 5, 20, 25 \tag{5.3}$$

This vector $\boldsymbol{v}$ is filled with zeros to have the same length over all emotions. In this example, a zero is placed on the back of $\boldsymbol{v}$. Four AUs can maximal be recognized in ROI1 and three in ROI2. If there were an AU-combination of only three AUs be recognized in ROI1 in this example, a zero would be placed behind the AUs of ROI1. The resulting vector of the example mentioned above is as follows:

$$\boldsymbol{v} = 1, 2, 4, 5, 20, 25, 0 \tag{5.4}$$

The last SVM receives vector $\boldsymbol{v}$ as input. It is trained on the basis of the mapping of the database which can be seen in Figure 5.25. Hence, the SVM should classify $\boldsymbol{v}$ as fearful.

5.3 Graphical User Interface

The user has to start the launchfile *start.launch* of the ROS node *emrec* to use or work with the introduced system. It starts a graphical user interface (GUI). This application consists of four views.

In the first view, the user loads an image into the application. The input image is shown in the GUI. The recognized expression is displayed below the image. Additionally, computation time of emotion recognition appears.

The second view is a detailed view. The user can see output images of several steps of the system. The first image is the input image. The second shows the output of face detection. Next images are each divided in ROIs. Normalized ROIs are shown as well as the preprocessing step with Canny and Closing. Additionally, results of the findContours algorithm are illustrated. The next images visualize the permitted zones with permitted contours. The penultimate images demonstrate the division of each ROI to find facial components and wrinkles. The last frames display detected facial feature points.

The third view is used for training mode. The user can load a directory where training images are located. The status of training is displayed. After training, the number of images and computation time appear.

An evaluation of the system can be done in the last view. The user can load a directory with testing images. Status and result of evaluation is displayed. After testing, the number of images and computation time appear.

Appendix B presents screenshots of the GUI.

Appendix E describes an installation and user guide.

5.4 Summary

The recognition system developed in this thesis is fully automatic. It detects 19 facial feature points on static input images. Furthermore, it differentiates between permanent and transient features and detects both types. Transient features are represented in a 1-0-state. Permanent features are represented as distances and angles between the feature points. These parameters on the face are classified as AUs. By means of the recognized AUs, a set of AU-codes is classified as one of seven emotions. The system works without a reference image.

The next Chapter contains experiments and results of the presented AFERS. Further information about the database is given. The two types of parameters are evaluated and the most effective method is selected.

Chapter 6

Experiments and Results

This Chapter contains experiments and results of the presented approach.

Several experiments compare the distance-based approach with the angle-based approach. The approach providing best results is used to analyze the whole approach. These results shown in diagrams and tables are compared to results of the related systems mentioned in Chapter 3.

6.1 Database

The functionality needs to be tested on a presentable database in order to compare the introduced system. It is useful to utilize a database which is used in many other systems. Thus, the results are easy to compare with those of others.

Table 6.1 describes common databases for facial expression recognition. The most common database is the Cohn Kanade Database (CK or CK+) [KCT00, LCK$^+$10] which was developed in 2000 and extended in 2010. Pantic et al. [PP05] described disadvantages of the CK. CK and CK+ contain only image sequences and thus are not suitable for the system presented. The MMI Facial Expression Database [PVRM05] has certain advantages like AU and emotion labeling, but does not include out-of-plane rotations. Rotations are not handled in this thesis but are a main part of future work. Not only is the Japanese Female Facial Expression Database [LAKG98] not up-to-date, it only contains images of Japanese women. Training and testing with this database would be biased. The Radboud Faces Database (RaFD) [LDB$^+$10] is a quite novel database and hence not very often used in emotion recognition systems to date. Nevertheless, it seems to be optimal for the introduced system.

Table 6.1: Databases for emotion recognition

Database	Images / Sequences	Posed / Non-posed	AUs / Emotions	Details
Cohn Kanade Database CK, 2000, [KCT00]	Sequences	Posed expressions	AUs of six basic emotions	486 videos, 97 subjects
Cohn Kanade Database Extension CK+, 2010, [LCK$^+$10]	Sequences	Posed and non-posed expressions	AUs and labels of seven emotions	593 videos, 126 subjects
Radboud Faces Database RaFD, 2010, [LDB$^+$10]	Images	Posed expressions	AUs and labels of eight emotions	8040 images, 67 subjects
MMI Facial Expression Database, 2005, [PVRM05]	Images and sequences	Posed expressions	AUs and labels of emotions	740 images, 848 videos, 19 subjects
Japanese Female Facial Expression Database JAFFE, 1998, [LAKG98]	Images	Posed expressions	Seven emotions	219 images, 10 subjects

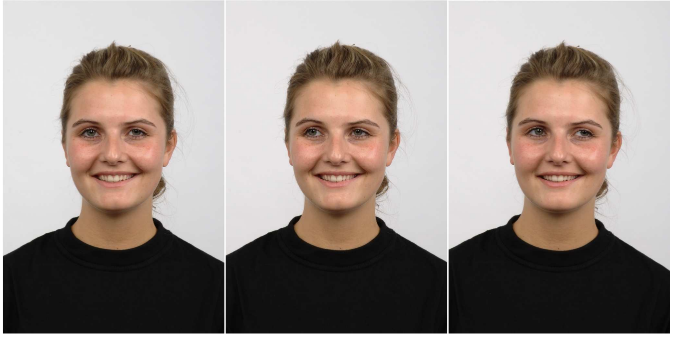


Figure 6.1: Three different gaze directions of the RaFD

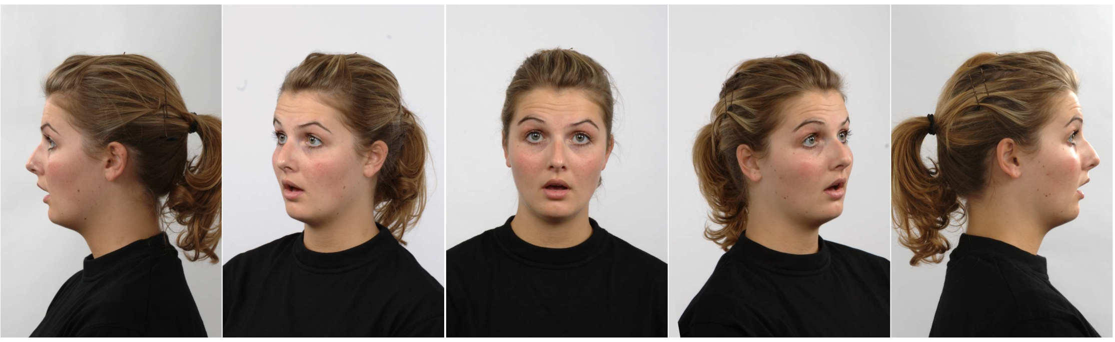


Figure 6.2: Five different face rotations of the RaFD

The introduced approach works with rigid images. The RaFD, presented by Langner et al. [LDB+10], is a new database with certain advantages. It was developed in 2010 at the Radboud University Nijmegen in Netherlands. It consists of 8040 static images. A number of 67 various models show facial expressions: 20 Caucasian male adults, 19 Caucasian female adults, four Caucasian male children, six Caucasian female children and 18 Moroccan male adults. Trained by a FACS coder, each model shows eight different expressions: happy, surprised, fearful, sad, disgusted, angry, neutral as well as contemptuous. Appendix C includes several images of the seven emotions used in this thesis. Langner et al. presented every model of the database showing every expression in three different gaze directions: left, frontal, and right. Figure 6.1 illustrates them. Five different camera angles demonstrate face rotations. They can be seen in Figure 6.2. All models trained with the FACS manual and were observed by certified FACS experts. Langner et al. used five Nikon cameras with 10 to 12 mega pixels. All images have the size 1024x681 and are colored. The images are labeled with the emotion presented and further information which is not used in this thesis. Everything can be found

Table 6.2: Partitions evaluated of the database

Number	Training	Testing
1	40%	60%
2	45%	55%
3	50%	50%
4	55%	45%
5	60%	40%

in support material for the database. The targeted AUs for every emotion are displayed in Figure 5.25 of Chapter 5.

It is important to verify the reliability of the database and the FACS coding [TKC05]. The shooting of the RaFD was observed and the models were coached by a FACS specialist [LDB$^+$10]. All emotions presented were based on prototypes of FACS. Hence, the required reliability of ground truth is given.

All 57 adults looking frontal are considered for the introduced system. This implies Caucasian women, Caucasian men, and Moroccan men. Altogether 399 images are used for training and testing.

6.2 Experiments

The experiments are made on the RaFD. The SVMs have to be trained to test the two methods of representing facial parameters. Therefore, several images are extracted from the database for training and testing. The percentage differs to investigate how results change. Table 6.2 shows proved partitions of the database. For the analysis of the system, images are chosen at random. It is ensured that every emotion is trained and tested with equal quantity. For example, 50% training images and 50% testing images imply 29 persons used for training set and 28 for testing set. Hence, the emotion presentation of one person is not split.

6.2.1 Training and Testing

This Section explains training and testing mode of the system presented. Both modes are necessary for analysis.

Training

The introduced AFERS possesses a training mode. Figure 6.3 illustrates the training flow. All images located in a specified directory are read in. Emotion recognition starts for each image. A face is detected, normalized, and segmented in ROIs.

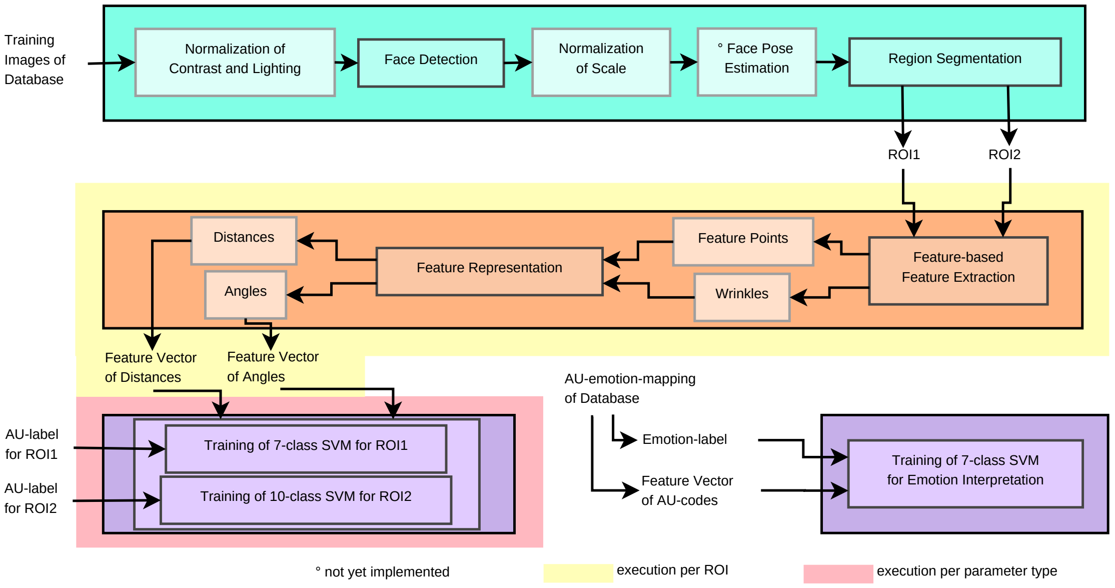

Training Images of Database
Normalization of Contrast and Lighting
Face Detection
Normalization of Scale
° Face Pose Estimation
Region Segmentation
ROI1
ROI2
Distances
Feature Points
Feature Representation
Feature-based Feature Extraction
Angles
Wrinkles
Feature Vector of Distances
Feature Vector of Angles
AU-emotion-mapping of Database
AU-label for ROI1
Training of 7-class SVM for ROI1
AU-label for ROI2
Training of 10-class SVM for ROI2
Emotion-label
Feature Vector of AU-codes
Training of 7-class SVM for Emotion Interpretation
° not yet implemented
execution per ROI
execution per parameter type

Figure 6.3: Training mode of the AFERS presented

Feature points are localized and both types of parameters extracted. A total of four feature vectors store the parameters of both types and both ROIs. Altogether, five feature vectors are initialized with the parameter data of the aforementioned four vectors plus emotion data for the last SVM. Another five vectors are initialized with the label data. The system receives emotion data and label through names of images. All images should contain the correct emotion label in their name. Thus, the label vector for the third SVM is filled. Corresponding AU-codes are stored in the data vector by means of the label. The system determines correct AU-codes with the help of the mapping of RaFD, see Figure 5.25 of Chapter 5. These AU-codes for a specific emotion are used to fill the label vectors of ROI1 and ROI2 of both parameter types. The associated data vectors store the extracted parameters, as mentioned above. The SVM for classification of distance parameters of ROI1 is trained with label vector and data vector. Accordingly, the SVM for ROI2 is trained. Both SVMs for classification of angle parameters are trained respectively. The SVM for emotion classification is trained with label vector and data vector of emotion information.

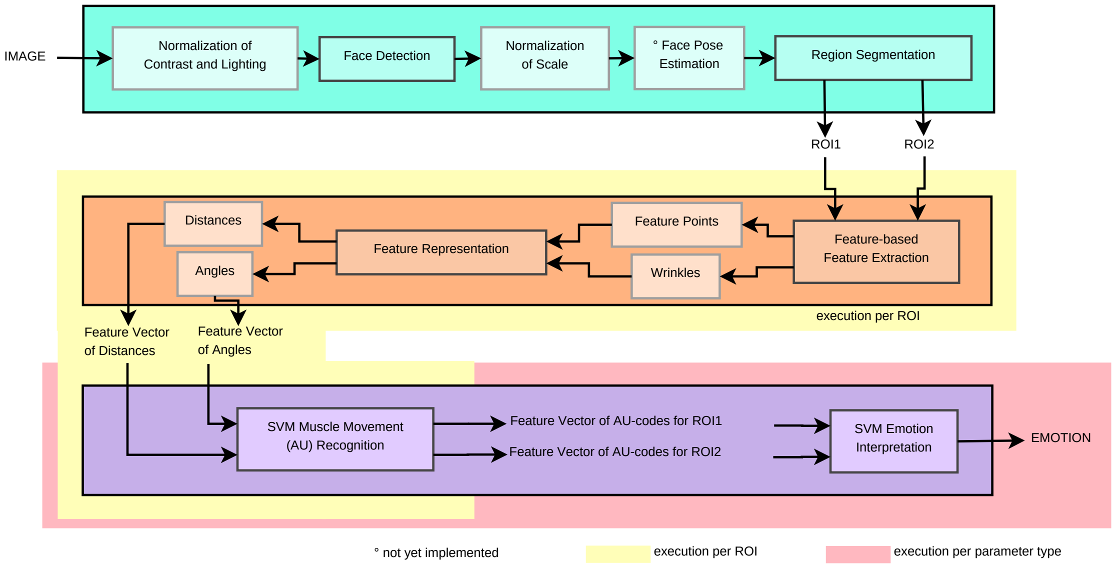


IMAGE
Normalization of Contrast and Lighting
Face Detection
Normalization of Scale
° Face Pose Estimation
Region Segmentation
ROI1
ROI2
Distances
Angles
Feature Representation
Feature Points
Wrinkles
Feature-based Feature Extraction
execution per ROI
Feature Vector of Distances
Feature Vector of Angles
SVM Muscle Movement (AU) Recognition
Feature Vector of AU-codes for ROI1
Feature Vector of AU-codes for ROI2
SVM Emotion Interpretation
EMOTION
° not yet implemented
execution per ROI
execution per parameter type

Figure 6.4: Testing mode of the AFERS presented

Testing

The presented system possesses an evaluation mode. Figure 6.4 illustrates testing flow. Testing images are placed in a specified directory and read in the system. Each image runs through the testing flow. A face is detected, normalized, and segmented in ROIs. Feature points are localized and both types of parameters extracted. These parameters are passed to the AU recognition step. Each ROI provides one feature vector of distances and one of angles. The AU recognition outputs one vector of AU-codes per ROI and per parameter type. The vectors of AU-codes are passed to the emotion interpretation step. Both vectors of ROI1 and ROI2 are combined and classified to one emotion. Altogether, the AFERS provides two emotions as output. One is generated by distance parameters and one by angle parameters. The evaluation mode compares recognized emotions to the emotion label in the name of the input image. All images should contain the correct emotion label in their name. By means of this compromise, precision, recall, specificity, and accuracy are computed.

6.2.2 Face Detection

The OpenCV face detector is tested on certain images. It provides robust results on straight frontal faces. In-plane rotated faces are mostly detected up to 25°. Several faces are detected with out-of-plane rotations of 45°, but the majority of faces is disregarded. The face detector delivers good results for the actual AFERS. For future work with rotated faces, the face detector should be extended.

6.2.3 Feature Point Detection

Figure 6.5 illustrates results of facial component division and feature point detection. The images are chosen at random. From left to right, each face is split into facial components and feature points are detected on these parts. Features are described from the point of view of the person.

The first image expresses anger. The system separates ROI1 optimally. In ROI2 a fraction of the nose is clipped and seems to be a wrinkle. A threshold makes sure that small fractions are not counted among wrinkles. Feature point detection performs nearly optimal. The lowermost point of the mouth, which is point 8.2 of the MPEG-4 standard, is not detected correctly. The OpenCV findContours algorithm provides linked points of a contour. If the contour is approximated as a straight line, no points are located on the linear slope. Hence, no feature points can be detected on this line.

The second face expresses disgust. The lower lip contour lies outside the permitted zone. Hence, the mouth height is extracted shorter as it actually is. The left wrinkle of the person stretches to the sides of the mouth. Additionally, the person has a birthmark on the right side of his mouth. It is too large to be avoided by the specified threshold of the system. Hence, both outer feature points are shifted and moth width is extracted as being longer as it truly is. The lowermost point of the left eye of the person, point 3.3, is wrongly located on a skin fold under the eye.

The next face expresses happiness. The lower lip contour is not detected by the findContours algorithm. Hence, point 8.2 is extracted as the inner point of the lower lip. Furthermore, the nose point 9.15 is not detected as the lowermost point. However, on this example this point fits better than the lowermost point of the nose contour, which is actually evoked by shadow.

The next image of these examples expresses the neutral emotion. The lower lip contour lies outside the permitted zone. Additionally, the lowermost contour is approximated as a straight line. Hence, point 8.2 is wrongly detected too high.

The penultimate image shows a happy face. The eyebrows stretch to the sides of the eyes and poke out of the eyebrow zone. Thereby, both outer points of the eyes are located too high. Point 3.3 is wrongly located on a skin fold under the eye.

The ultimate face expresses sadness. By reasons of dark hair and shadow between and under the eyebrows, there are wrinkles detected misleadingly in ROI1 and feature points are shifted. The upper lip contour lies above the splitting line and counted among the nose. Both middle lip points are located side by side, because above and underneath are just straight lines. This falsifies parameters extracted essentially. Not trained, the emotion is recognized by distance parameters as angry. In an angry face, lips are tightened and pressed, and wrinkles are present in ROI1.

These examples indicate, that in certain cases the lower lip contour is not linked with the entire mouth and hence not detected because it lies outside the permitted zone. Thereby, the mouth height is measured as being shorter than it actually is. The presence of shadow influences feature point detection, especially under the eyes and between the eyebrows and eyes. Thereby, the innermost points of the eyebrows and the lowermost points of the eyes are detected too far downwards.

In general, facial component division and feature point detection provide stable results. In most cases, feature points are detected nearly optimal position. Splitting facial contours by means of a model-based approach delivers facial parts accurately. Sometimes fractions of contours are clipped wrongly. However, the system presented is able to handle small fractions correctly.

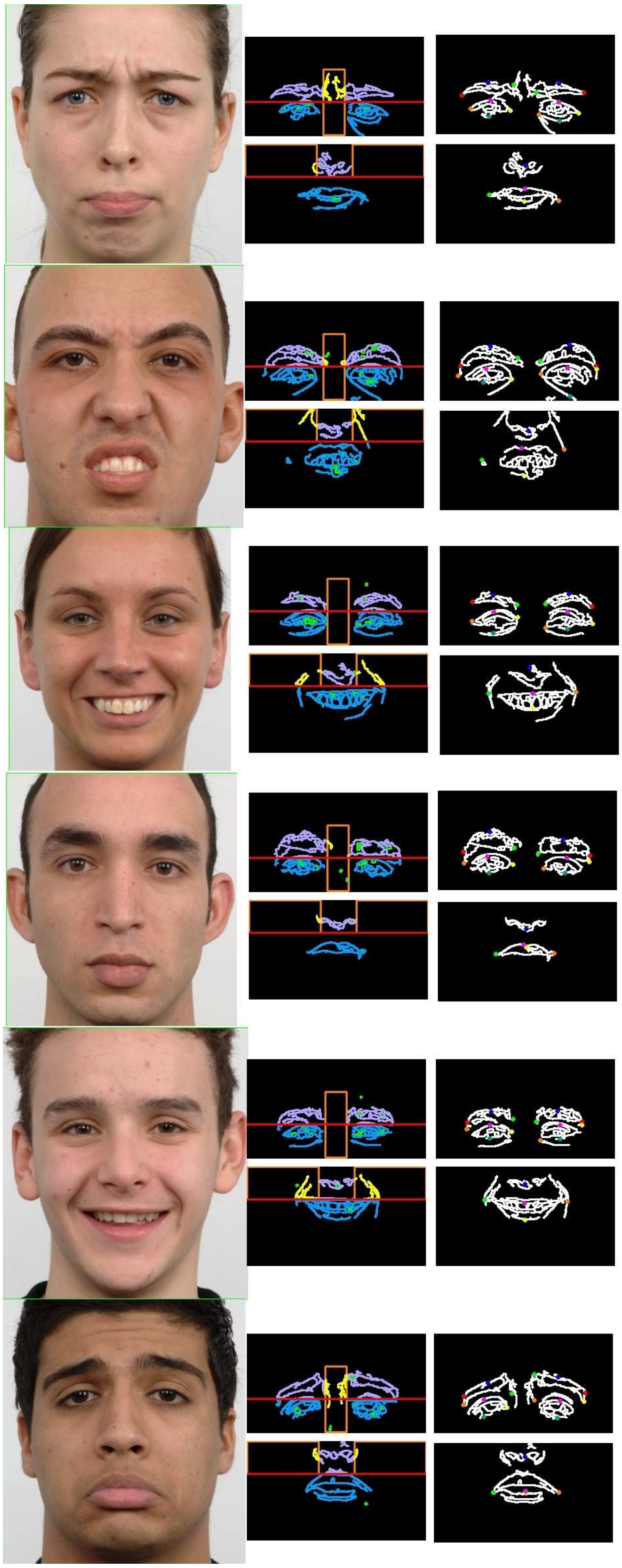


Figure 6.5: Results of facial component division and feature point detection

6.2.4 Distance and Angle Parameters

This Section analyzes the system introduced in certain cases. The extraction of distance and angle parameters achieve different results.

Parameter Scaling

Hsu et al. [HCL08] asserted that it is important to scale parameters prior to classification. They advised a range of [-1, +1] or [0, 1]. Parameters of training and testing have to be scaled equally. Distance parameters of the system presented range from 0 to 250. Hence, all data is divided by 250. The wrinkle state is either 0 or 250, so that after scaling it is either 0 or 1. Angle parameters of the system range from 0 to 360 maximum. Hence, all data is divided by 360. The wrinkle state is either 0 or 360, so that after scaling it is either 0 or 1. On evaluation partition number 5 of Table 6.2, accuracy is 60.87% when working with scaled distance parameters and 43.48% when working with scaled angle parameters. Without scaling it is 57.76% based on distance parameters and 42.86% based on angle parameters. Hence, the feature vectors of both parameter types are scaled before training and testing. Scaling the feature vector of AUs does not improve results of the last SVM.

Symmetric Parameters

As described in Chapter 5, this approach consults both sides of the face, although all seven facial expressions are symmetric. The average accuracy over all partitions evaluated is computed. Distance and angle parameters achieve better results without a comparison with the symmetric parameters. Averaging and hence reducing the number of distance parameters provides an accuracy of 56.72%. Passing all distance parameters of the face to the SVMs provides an accuracy of 60.21%. The two approaches on angle parameters provide accuracies with a difference of 0.09%. Hence, the symmetric parameters are not used for computing averaged parameters, but as further parameters.

SVM Parameters

The SVMs are trained with the OpenCV train_auto function [1]. Parameters are chosen automatically. Wagner et al. [Wag12] presented a guide for machine learning with OpenCV. They recommend to use the train_auto function for OpenCV versions after 2.0. Additionally, they explained that the approach to find optimal

[1] `http://docs.opencv.org/modules/ml/doc/support_vector_machines.html#cvsvm-train-auto`

Table 6.3: Parameters automatically chosen to be optimal for 238 training images

SVM	C	γ
ROI1 (d)	62.5	0.50625
ROI1 (a)	62.5	0.50625
ROI2 (d)	312.5	0.50625
ROI2 (a)	62.5	0.50625
Emotion	0.5	0.00225

Table 6.4: Parameters automatically chosen to be optimal for 217 training images

SVM	C	γ
ROI1 (d)	62.5	0.50625
ROI1 (a)	312.5	0.50625
ROI2 (d)	62.5	0.50625
ROI2 (a)	312.5	0.50625
Emotion	0.5	0.00225

parameters is based on grid-search and k-fold cross-validation. More detailed information can be found in [HCL08]. In the proposed system, training of 60% images provokes parameters C and γ to be as shown in Table 6.3. SVM "ROI1 (d)" means the SVM which receives distance parameters of ROI1. SVM "ROI2 (d)" represents the SVM receiving distance parameters of ROI2, for "(a)" equivalent. "Emotion" SVM is the last SVM classifying AUs to emotions. Training of 55% images results in the parameters listed in Table 6.4. Training of 50% images involves parameter values shown in Table 6.5.

Evaluation partitions number one and two cannot be trained with the OpenCV train_auto function. The function throws an error on these training sets. In general, minimum 200 images can be trained with train_auto. For less training

Table 6.5: Parameters automatically chosen to be optimal for 203 training images

SVM	C	γ
ROI1 (d)	312.5	0.50625
ROI1 (a)	62.5	0.50625
ROI2 (d)	62.5	0.50625
ROI2 (a)	62.5	0.50625
Emotion	0.5	0.00225

images, the same parameters of partition number three are used in the OpenCV train function [2].

SVM Classification

Distance and angle parameters are each passed to two SVMs, one per ROI. By means of distance parameters, the SVM for ROI1 achieves an accuracy of 43.48% on the evaluation of partition number 5. The SVM for ROI2 reaches an accuracy of 60.87%. By means of angle parameters, the SVM for ROI1 achieves an accuracy of 38.51%. The SVM for ROI2 reaches an accuracy of 43.48%. In both cases, SVMs of ROI2 achieve better results. This does not depend on the partition evaluated. ROI2 achieves up to 20% better accuracy than ROI1. Bettadapura et al. [Bet12] confirmed that nose and mouth carry the most information of all features. Based on this knowledge, all AU combinations which are not assigned to an emotion by the mapping of RaFD, can be trained as well. These combinations are classified to the emotion built up of the AUs recognized in ROI2. Experiments show that ROI2 is more unfailing than ROI1. Hence, if an AU-combination like AU6 + AU20 + AU25 is recognized, the SVM in the last step classifies it as fearful, which consists of AU20 + AU25 in ROI2. Otherwise AU6 + AU20 + AU25 could also be classified as an emotion which is neither built up of AU6, AU20, nor AU25. By means of this, mapping errors of the SVM are avoided and accuracy of distance parameters is improved about 3%. Accuracy of angle parameters stays the same. Additionally, since the system favors the output of ROI2, occlusion errors of ROI1 should be avoided. Experiments should test this assumption in future work.

Computation Time

Computation time is measured on an Intel Core i5-3570K processor with 4x 3.40GHz and 16GB RAM. The system is executed single threaded. Computation time for training of 238 images is 188.32 sec., thus 0.79 sec. per image. Testing of 161 images takes 90.47 sec., which is 0.56 sec. per image. Recognizing an emotion on a single image also needs about 0.56 sec.

Confusion Matrix

Precision, recall, specificity, and false positive rate are computed for each emotion class. A confusion matrix is assembled for each partition presented in Table 6.2. They can be seen in the last Section of Appendix D. Figure 6.6 shows an instance of a confusion matrix. Recognition results of distance parameters on partition

[2] `http://docs.opencv.org/modules/ml/doc/support_vector_machines.html#cvsvm-train`

recognized class

		happy	surprised	sad	fearful	angry	disgusted	neutral
true class	happy	21	0	0	0	0	2	0
	surprised	0	18	1	4	0	0	0
	sad	0	0	1	8	7	4	3
	fearful	0	2	1	17	1	1	1
	angry	0	0	1	2	13	2	5
	disgusted	0	1	0	0	1	21	0
	neutral	0	1	1	7	6	1	7

Figure 6.6: Confusion matrix of 60% images trained and 40% images tested

number 5 are illustrated. Figure 6.7 shows the confusion matrix of angle parameters on partition number 5. Each emotion is tested 23 times. The major diagonal of the matrix illustrates correct recognitions [Faw06]. A comparison of both confusion matrices shows that happy is recognized superiorly by angle parameters, but distance parameters cause no false positives of this emotion. A great difference can be seen on surprised, fearful and sad. By means of distance parameters, the two former emotions are recognized correctly 18 and 17 times respectively of 23 images. Sad is recognized correctly merely one time. Based on angle parameters, these results are swapped. Sad is recognized correctly 11 times, but surprised and fearful degrade to 4 and 1 respectively. Other emotions are recognized more times correctly with the extraction of distance parameters.

Precision, recall, specificity, false positive rate, and accuracy can be computed on the values of the confusion matrix. In general, they are computed for one class as described by Fawcett et al. [Faw06]:

$$precision = \frac{truepositives}{truepositives + falsepositives} \tag{6.1}$$

$$recall = truepositiverate = \frac{truepositives}{totalpositives} \tag{6.2}$$

$$specificity = \frac{truenegatives}{truenegatives + falsepositives} \tag{6.3}$$

$$falsepositiverate = 1 - specificity \tag{6.4}$$

		recognized class						
		happy	surprised	sad	fearful	angry	disgusted	neutral
true class	happy	23	0	0	0	0	0	0
	surprised	0	4	5	0	2	0	12
	sad	3	0	11	1	4	1	3
	fearful	2	1	8	1	5	2	4
	angry	2	0	2	3	12	1	3
	disgusted	5	0	0	0	1	16	1
	neutral	0	0	6	5	8	1	3

Figure 6.7: Confusion matrix of 60% images trained and 40% images tested

$$accuracy = \frac{truepositives + truenegatives}{totalpositives + totalnegatives} \tag{6.5}$$

The true positives represent the images recognized correctly. If an emotion is classified wrongly, the value on the correspondent field in the confusion matrix is incremented.

In the evaluated partition shown in Figure 6.6, precision for the happy emotion is computed as follows:

$$precision_{\texttt{happy}} = \frac{21}{21+0+0+0+0+0+0} = 1.0 \tag{6.6}$$

Value 21 shows that 21 of 23 happy facial expressions are recognized correctly. The values $0+0+0+0+0+0$ are the false positives. These images would have been recognized as happy, but actually belong to other emotions.

Recall for the happy emotion is computed as follows:

$$recall_{\texttt{happy}} = \frac{21}{21+0+0+0+0+2+0} = 0.913 \tag{6.7}$$

The values $21+0+0+0+0+2+0$ are the total positives. This number of images is truly happy, even if they are recognized differently.

Specificity for the happy class is computed as follows:

$$specificity_{\texttt{happy}} = \frac{tn}{tn+0+0+0+0+0+0} = 1.0 \tag{6.8}$$

With $tn = 18+2+1+1+1+1+1+1+1+4+8+17+2+7+7+1+13+1+6+4+1+2+21+1+3+1+5+7$ defined as the true negatives. True negative images are correctly recognized as not happy. The values $0+0+0+0+0+0$ present the false positives, as described above.

The false positive rate of happy is computed as:

$$falsepositiverate_{\texttt{happy}} = 1.0 - 1.0 = 0.0 \tag{6.9}$$

By means of these calculations, specific graphs are able to visualize the results of different training sets to compare with each other.

Receiver Operating Characteristics Graph

Receiver operating characteristics (ROC) graphs visualize and compare classifiers on the basis of their performance, as described by Fawcett et al. [Faw06]. The x-axis represents the false positive rate. The y-axis displays the recall, also called the true positive rate. Thereby, benefits and costs are compared. One point in the graph stands for one classifier output. Fawcett et al. mentioned important points in a ROC graph: point $(0, 0)$ shows that this classification achieves no false positives as well as no true positives. There are no positive results. Point $(1, 1)$ illustrates a classifier which always provides a positive result, but causes a high false positive rate as well. A classifier on point $(0, 1)$ is a perfect classifier. Classifiers on the diagonal line $y = x$ achieve the same results like random guessing. Classifiers on the lower left side provide only few correct results, but few incorrect as well. Classifiers which lie below the diagonal line achieve results inferior to random guessing.

Figure 6.8 shows a ROC graph for distance parameters on partition number 5 with 60% images trained and 40% images tested. Seven points illustrate the results of the seven emotions. Figure 6.9 visualizes this partition with angle parameters. Appendix D presents ROC graphs of distance and angle parameters for each evaluated partition. In both Figures 6.8 and 6.9, happy is recognized nearly perfect. Angles cause more false positives of happy than distances. Distance parameters achieve less true positives, but no false positives. With distances, also disgusted and surprised achieve results comparable to state-of-the-art approaches. Fearful is recognized correctly more often with distances, but sad achieves better results with angles.

Figures 6.11 and 6.10 combines results of distance and angle parameters each over all partitions evaluated. Distances provide three emotions in the upper left side of the ROC graph. Three emotion classes are located in the middle and one in the lower left side. In contrast, angles provide only one emotion class in the upper left side, four classes in the middle and two in the lower left side.

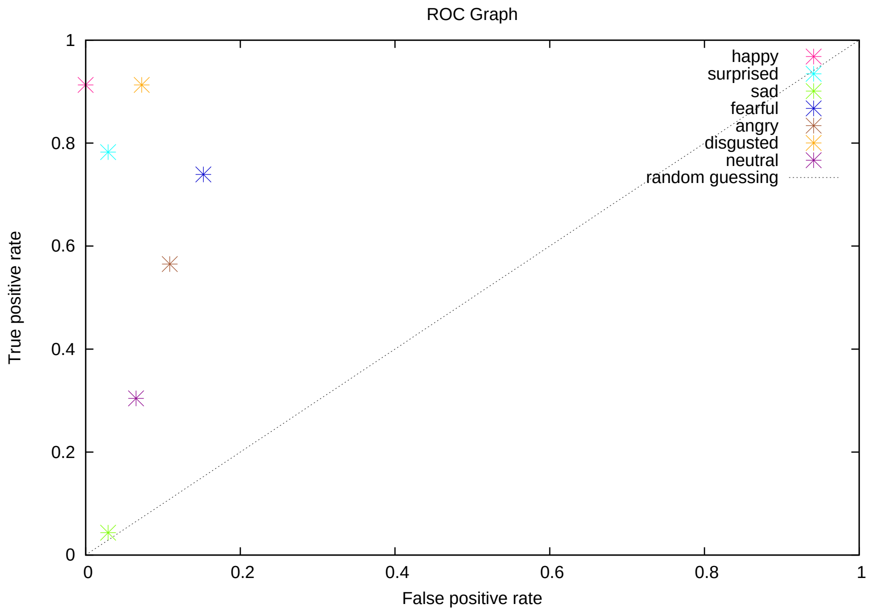


Figure 6.8: ROC graph for distance parameters with 60% images trained and 40% images tested

Figure 6.12 summarizes the distribution of all emotions for distances and angles in one graph. Blue points represent results gained by distance parameters. Red points visualize results of angle parameters. The points generated by distance parameters are located on the left side whereas the points of angle parameters are oriented to the lower right side.

These ROC graphs show more stable results generated by means of distance parameters for facial feature representation. The next Subsection analyzes the accuracy which provides more precise information.

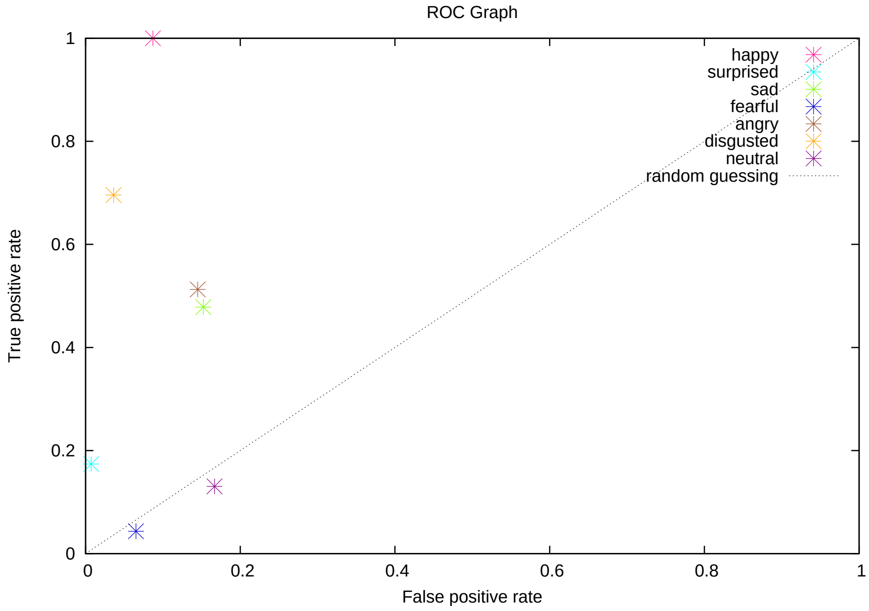


Figure 6.9: ROC graph for angle parameters with 60% images trained and 40% images tested

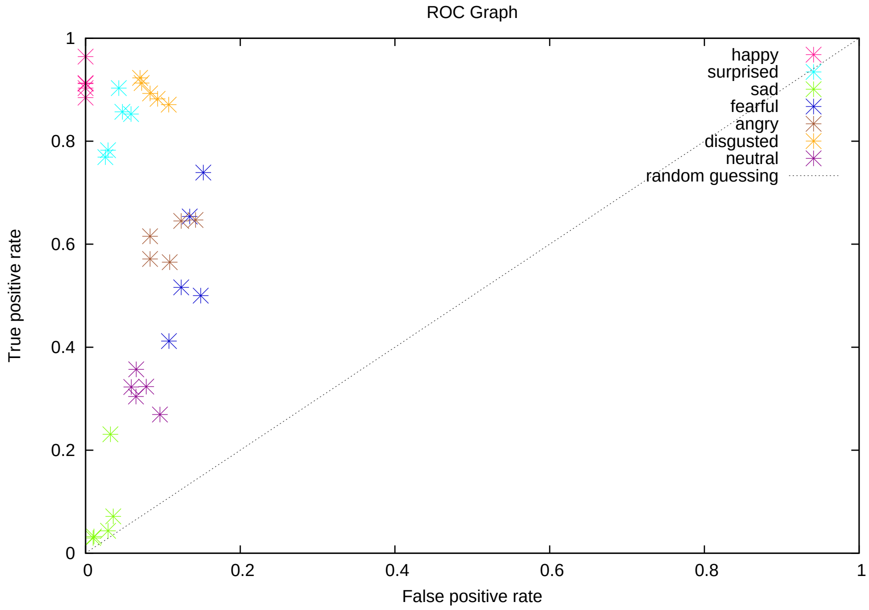


Figure 6.10: ROC graph for distance parameters over all partitions evaluated

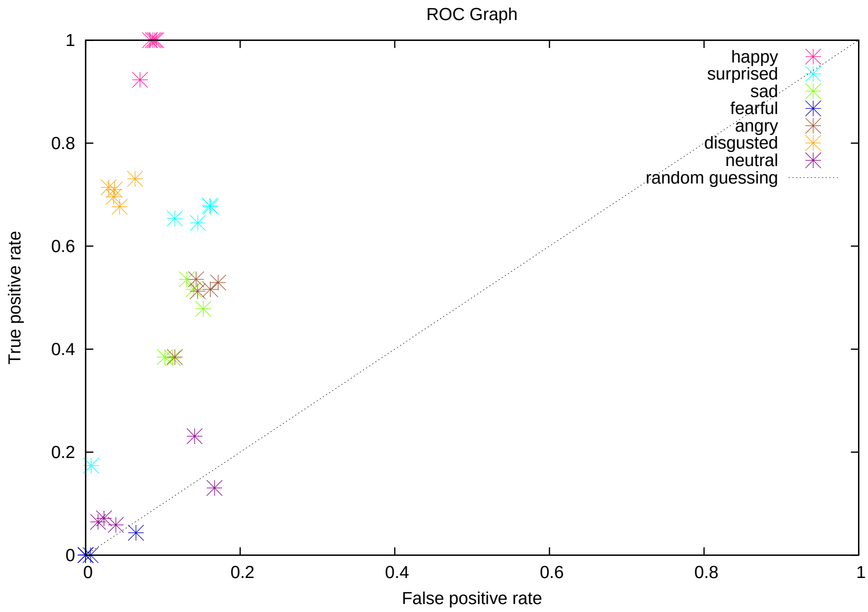


Figure 6.11: ROC graph for angle parameters over all partitions evaluated

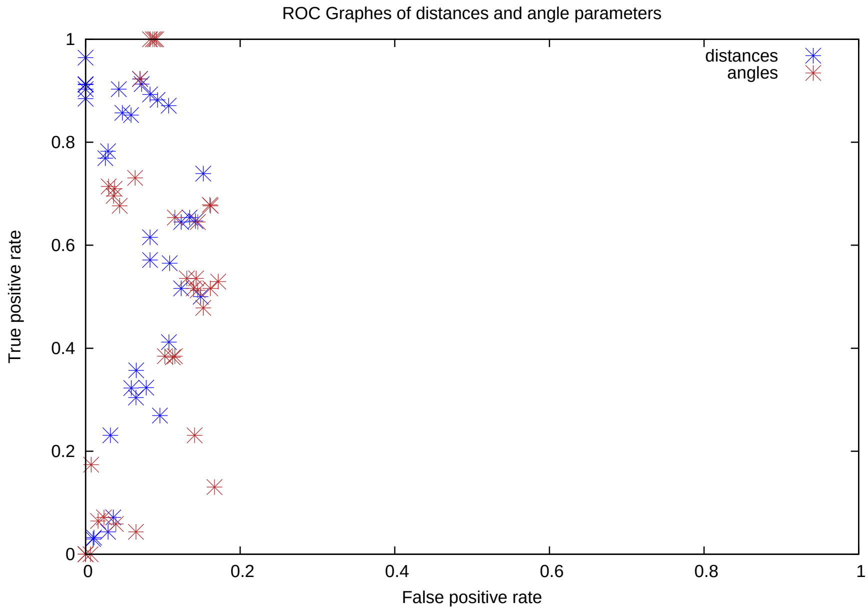


Figure 6.12: ROC graph for distance and angle parameters over all partitions evaluated

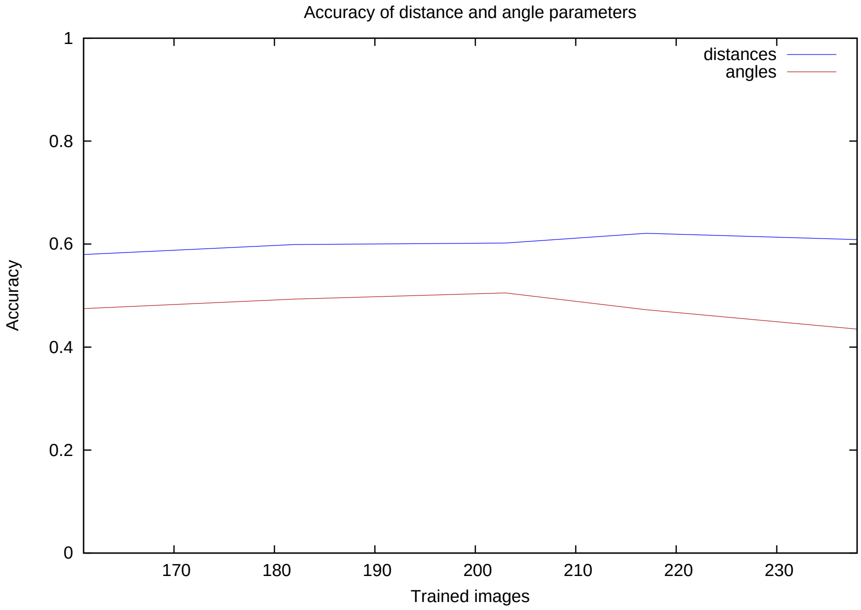


Figure 6.13: Accuracy of distance and angle parameters over all partitions evaluated

Accuracy

The accuracy of the entire AFERS is computed as follows:

$$accuracy = \frac{truepositives}{totalnumberofimagestested} \tag{6.10}$$

The images recognized correctly are divided by the number of all testing images. The true positives are computed as the sum of values in the major diagonal of the corresponding confusion matrix.

Minimal accuracy achieved by distance parameters is 57.98% on partition number 1. Maximal accuracy is gained on partition number 4 with 62.09%. Angle parameters achieve a minimal accuracy of 43.48% on the last partition evaluated. Maximal accuracy shows partition number 3 with 50.51%. Overall, AFERS based on distance parameters achieves higher accuracy. Figure 6.13 makes this result obvious. Accuracy computation starts with 161 images trained in partition number 1 and ends with 238 images trained in partition number 5.

6.2.5 Conclusion of Experiments

Experiments show that face detection based on the OpenCV face detector provides stable results for actual requirements. Feature point detection sometimes is falsified due to erroneous facial component segmentation. In some cases, the lower lip contour is discarded and the lowermost point of the mouth is not detected correctly. Especially this corrupts accuracy, because the mouth carries the most information of all facial features. Considering symmetric parameters of the face separately improves accuracy. Overall, the AFERS recognizes the neutral expression worst. The system using distance parameters achieves worst results recognizing a sad emotion. In every partition evaluated, fearful is mostly mistaken for sad. Partition number 4 gains the best recall of sad. The experiments show that the AFERS based on distance parameters provides more accurate recognition results than using angle parameters for feature extraction.

In the next Section, the system based on distance parameters is compared with the results of related FERS.

6.3 Results

Experiments show maximal accuracy achieved by distance parameters is 62.09% on partition number 4. In this evaluation set, 55% of 399 images are used for training and 45% for testing. Figure 6.14 presents the ROC graph of this system. Figure 6.15 visualizes the precision for each emotion class. Figure 6.16 shows the confusion matrix of this evaluation set.

Although the emotion recognition system based on distance parameters for facial feature representation achieves better results than an angle-based approach, the sad and neutral expressions are not sufficient separable from other emotions. The happy expression is recognized nearly perfect due to large distances on mouth width. Every other emotion is built up of small mouth width. Recognition of a disgusted face achieves stable results as well, but it involves a higher false positive rate.

The overview Section of Chapter 3 summarizes the results of related systems. It turns out that every feature-based system needs a reference face for comparison. Feature-based systems achieve recognition results of 84.13% on average. Image-based systems average 84.35%. Fully automatic systems achieve 83.57% on average. Comparing the presented system with related work is difficult. Every system utilizes miscellaneous approaches for face detection, facial data extraction and emotion recognition. Several systems only recognize AUs and other concentrate merely on emotion outputs. Considering accuracy only, the introduced system achieves less recognition results than state-of-the-art FERS. Merely the system of Srivastava et al. [Sri12] provided less accuracy. Considering the true positive rate of happy, surprised and disgusted, the presented system is able to keep up with state-of-the-art FERS.

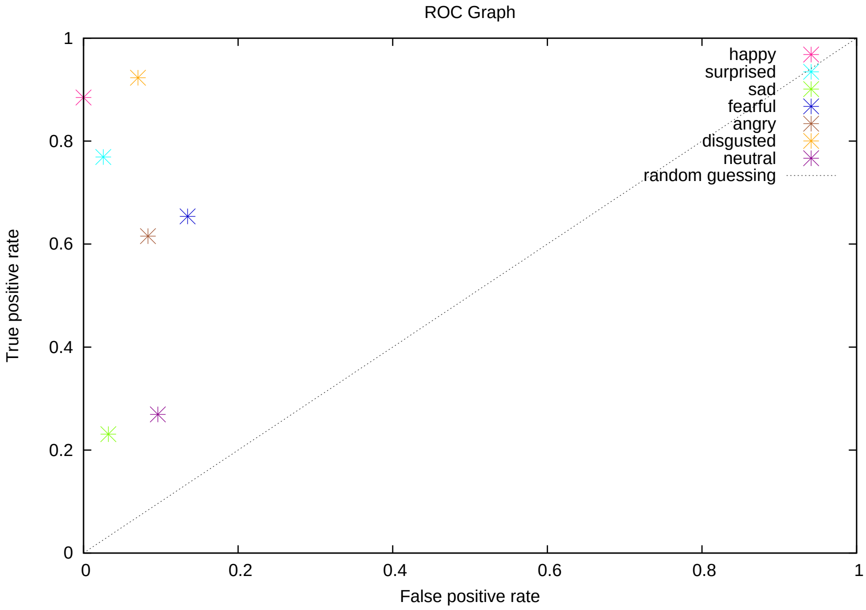


Figure 6.14: ROC graph of distance parameters over all emotions with 55% images trained and 45% images tested

6.4 Summary

The experiments and results described in this Chapter give more information about the effectiveness of the AFERS presented. For a comparison with related systems, best recognition results are evaluated. The evaluation set of 55% images trained and 45% images tested provides highest accuracy. Additionally, the system based on distance parameters recognizes best. Maximal accuracy achieved is 62.09% on average over all emotions.

The next Chapter outlines this thesis and the introduced system. It ranks positive features and assesses features to optimize in future work.

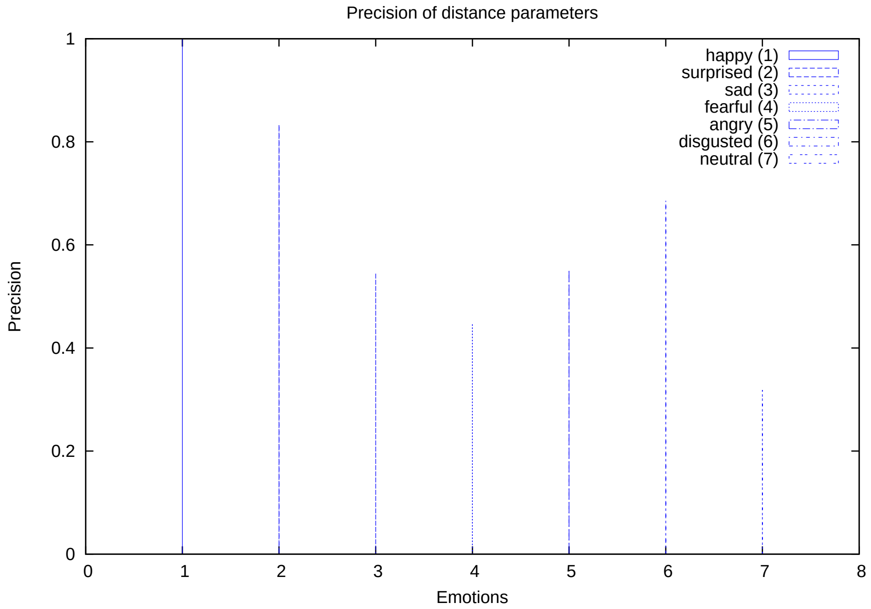
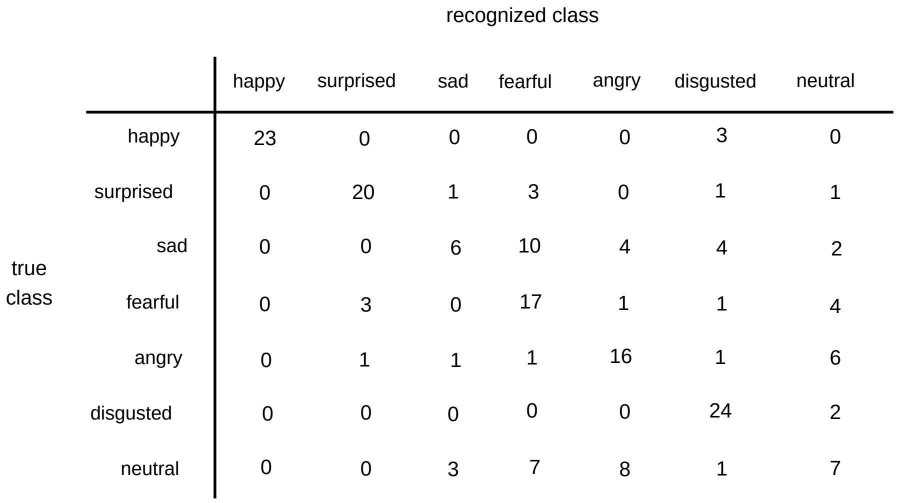


Figure 6.15: Precision of distance parameters over all emotions with 55% images trained and 45% images tested



		recognized class						
		happy	surprised	sad	fearful	angry	disgusted	neutral
true class	happy	23	0	0	0	0	3	0
	surprised	0	20	1	3	0	1	1
	sad	0	0	6	10	4	4	2
	fearful	0	3	0	17	1	1	4
	angry	0	1	1	1	16	1	6
	disgusted	0	0	0	0	0	24	2
	neutral	0	0	3	7	8	1	7

Figure 6.16: Confusion matrix of 55% images trained and 45% images tested

Chapter 7

Summary and Conclusion

The last Chapter summarizes this thesis and draws a conclusion of the approach of an AFERS. The thesis closes with an outlook on future work.

7.1 Summary

This thesis examines a fully automatic emotion recognition system based on visual features of frontal faces. The system works image-based and employs feature-based feature extraction. A novel approach is developed to detect permanent and transient features with a model-based method dividing the face in components. The system works without any reference image. It extracts two types of facial parameters: distances and angles. Two multi-class SVMs classify facial parameters. The results are codes of 16 AUs of the Facial Action Coding System, single or in combination. These codes are mapped to a facial emotion by means of a third SVM. The system covers the six basic emotions (happy, surprised, sad, fearful, angry, and disgusted) plus the neutral facial expression. It is implemented in C++ and provided with an interface to ROS.

Experiments on the Radboud Faces Database show that the AFERS based on distance parameters achieves better results than utilizing angle parameters. Accuracy is 62.09% with 55% of 399 images trained and 45% images tested.

However, the presented system provides less accuracy than state-of-the-art approaches. Related work discovered that facial expression recognition is harder on images which provide only static 2D information. Srivastava et al. [Sri12] utilized a median of 21 feature points detected in ten frames. Therefore, detection errors could be avoided. Pantic et al. [PP05] developed a sequence-based recognition system considering temporal dynamics of emotions. Hence, they were able to respect more information given in the onset, apex, and offset. On the basis of this approach, they recognized 27 different AUs and their combination. Therefore, their approach achieved better results than the system introduced in this thesis. Sebe et al. [SLS+07] assessed that most information about human emotions is received from video sequences. Pantic et al. [PP06] pointed out that several movements of facial parts are hard to detect in 2D frontal face images. Hence, they decided to work on video sequences with profile faces.

By means of these conclusions, it turns out that an image-based FERS lasts for recognizing the six basic emotions and the neutral face. If further information should be gained, three- or four-dimensional input is more efficient.

No feature-based related work can be found operating without a reference image. Merely appearance-based approaches as described in [WO06, LFBM02, BLL+04] either did not need a reference face or did not mention it explicitly. Both systems of Littlewort et al. [LFBM02] and Bartlett et al. [BLL+04] were sequence-based and worked on videos with the neutral emotion as the first frame. Like the presented system, Whitehill et al. [WO06] received static images as input. In contrast to the system introduced in this thesis, Whitehill et al. utilized Haar-like features plus AdaBoost as an appearance-based approach for feature extraction. Hence, no feature points had to be detected which is more error-prone than considering the face as a whole. Additionally, they recognized AUs without interpreting the emotion. Their system achieved an accuracy of 92.4%, but worked not fully automatically. ROIs were selected manually. On the contrary, the system presented in this thesis detects facial components automatically. This approach is more complex, but requires no manual interaction.

Facing the context that this system works without any learning phase or reference image and only with the actual expression, the system shows respectable results. This thesis proves that emotion recognition is possible considering only the actual input image. Knowing the observed person is not mandatory.

7.2 Conclusion

Bettadapura et al. [Bet12] listed features a good FERS has to offer:

- fully automatic
- sequences and images
- real-time
- spontaneous expressions
- all AUs and expressions
- different lighting

- occlusions
- no preprocessing
- person independent
- different cultures, skin colors, and age
- different resolutions
- frontal, profile, and rotated images

The introduced system is fully automatic and receives static images as input. Sequences are not covered. It recognizes facial expressions with a fast computation performance under one second, but not in real-time. Only posed expressions of 16 AUs and hence seven emotions are covered. Different lighting conditions can be handled by means of image normalization. Certain occlusions can be avoided, for example facial hair is not considered as facial components. There is no preprocessing before the system starts. The system does not need a training phase or a reference image of a person. Preprocessing of the input image is done by normalization of scale and lighting. Every person can be tested at every time because the presented AFERS is independent of specific characteristics. Furthermore, it is trained with Caucasian female, male, and Moroccan male. Hence, it works on different cultures and skin colors. Every input image is scaled to a specified resolution. If input images are smaller, facial information could disappear. Currently, the system only treats frontal faces. It should be extended in future work.

Taken together, this system serves as a notable foundation for future work where accuracy can be further enhanced.

With the proposed emotion recognition system, our daily life can be improved. Autonomous systems like robots obtain more human skills. They can help us in different situations by recognizing our mental state. Assisting systems become more specific to humans. By means of the introduced system, machines understand what humans mean with certain behavior and the other way around. Systems can recognize the emotion and interact adequately to human.

7.3 Future Work

In future work, it would be profitable to recognize spontaneous expressions. Posed expressions are easier to recognize, but if a FERS should be used in real life, it has to adapt to the people and not the other way around.

More AUs and combinations could be trained to recognize a wide range of facial expressions.

In addition, the system should receive static images as well as sequences as input. Thus, the recognition system can be used for more conditions.

Border cases like occlusions by glasses, sunglasses, facial hair, etc. should be handled. They could be trained or feature points could be detected more robustly in order not to lose information by occlusions. It should be tested if the presented system already recognizes emotions correctly, even if the eyes are occluded. Since the system favors the output of ROI2, occlusion errors of ROI1 should be avoided.

In future work, in-plane and out-of-plane rotations should be considered. Up to a specific angle, rotated faces are detected with the OpenCV face detector. Profile faces are not supported. If the face is rotated, it can be discovered by the OpenCV findHomography function [1]. Afterwards, it can be handled with the OpenCV geometric image transformation [2].

With more time, the infants of the database could be trained and tested as well. If the system does not work on infants as well, feature detection should be modified.

To compare recognition results with related FERS, this system should be tested on other databases.

Furthermore, the system could be tested with more different cultures. In case of need, these different cultures can be trained before tested.

To improve accuracy, certain approaches can be evaluated. First of all, more facial data could be extracted. The experiments of this thesis show that less parameters degrade recognition results. It should be discovered if more parameters improve accuracy. Furthermore, the system could utilize a rule-based approach for classifying facial component states to emotions. Therefore, the SVMs of ROI1 and ROI2 do not recognize AUs but states. States of the mouth could be opened, closed, and expanded. Tian et al. [TKC01] developed a related approach and achieved an accuracy of 96%. The approach should be extended to work automatically and without a reference face of a person.

In future work, the developed system could be connected with other communication systems, for example speech or gesture. With this, all received information can be adjusted. Hence, autonomous systems gain the ability to react even more appropriate to their counterparts.

[1] http://docs.opencv.org/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html#findhomography

[2] http://docs.opencv.org/modules/imgproc/doc/geometric_transformations.html#getPerspectiveTransform

Appendix A

Action Units of the Facial Action Coding System

This Appendix refers to the Section “Facial Action Coding System” in Chapter 2. It presents a full Table of 42 AUs assembled by Cohn et al. [CAE07] and the robotics institute of the Carnegie Mellon University [1].

A.1 Action Units

[1] http://www.cs.cmu.edu/~face/facs.htm

Table A.1: AUs of FACS

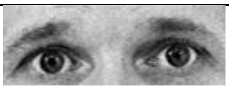
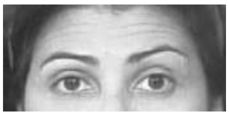
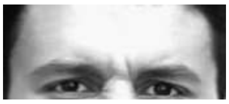
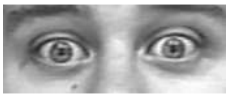
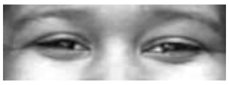
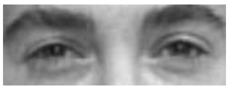
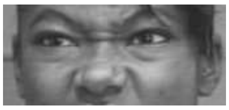
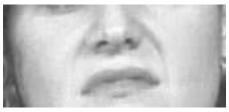
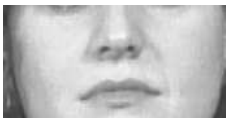
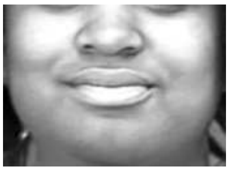
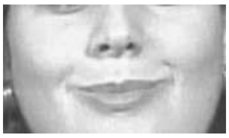
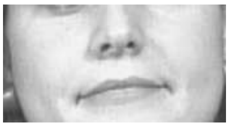
AU	Example Image	Description	Related Muscle
1		Inner brow raiser	Frontalis, pars medialis
2		Outer brow raiser	Frontalis, pars lateralis
4		Brow lowerer	Corrugator supercilii, Depressor supercilii
5		Upper lid raiser	Levator palpebrae superioris
6		Cheek raiser	Orbicularis oculi, pars orbitalis
7		Lid tightener	Orbicularis oculi, pars palpebralis
9		Nose wrinkler	Levator labii superioris alaquae nasi
10		Upper lip raiser	Levator labii superioris
11		Nasolabial deepener	Zygomaticus minor
12		Lip corner puller	Zygomaticus major
13		Cheek puffer	Levator anguli oris (a.k.a. Caninus)
14		Dimpler	Buccinator

Table A.2: AUs of FACS

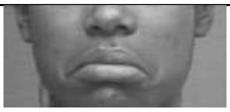
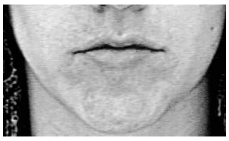
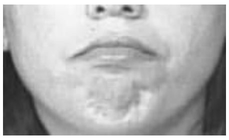
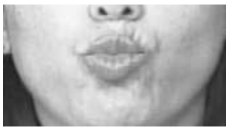
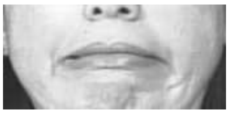
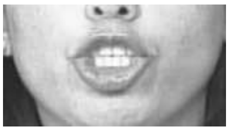
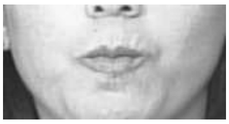
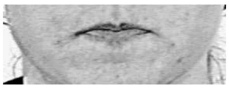
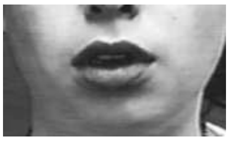
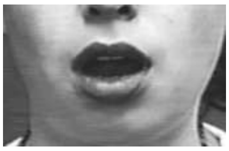
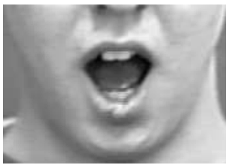
AU	Example Image	Description	Related Muscle
15		Lip corner depressor	Depressor anguli oris (a.k.a. Triangularis)
16		Lower lip depressor	Depressor labii inferioris
17		Chin raiser	Mentalis
18		Lip puckerer	Incisivii labii superioris and Incisivii labii inferioris
20		Lip stretcher	Risorius with platysma
22		Lip funneler	Orbicularis oris
23		Lip tightener	Orbicularis oris
24		Lip pressor	Orbicularis oris
25		Lips part	Depressor labii inferioris or relaxation of Mentalis, or Orbicularis oris
26		Jaw drop	Masseter, relaxed Temporalis and internal Pterygoid
27		Mouth stretch	Pterygoids, Digastric

Table A.3: AUs of FACS

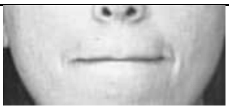
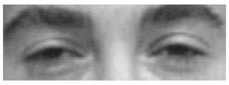
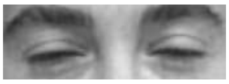
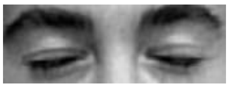
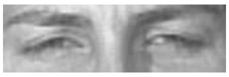
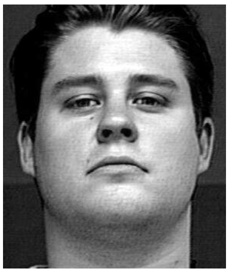
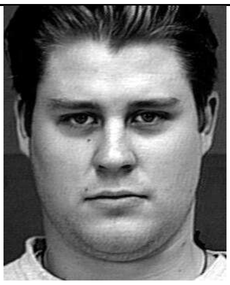
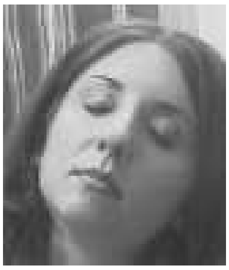
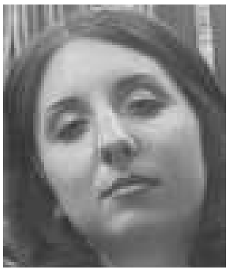
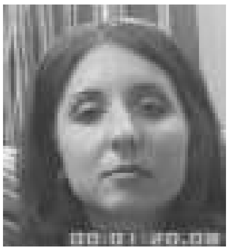
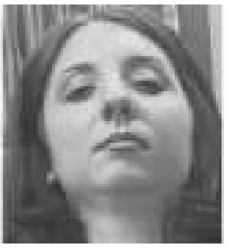
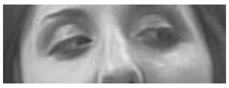
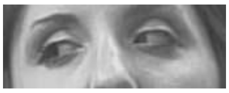
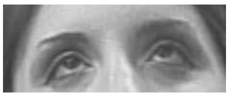
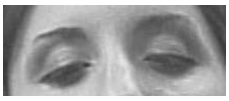
AU	Example Image	Description	Related Muscle
28		Lip suck	Orbicularis oris
41		Lid droop	Relaxation of Levator palpebrae superioris
42		Slit	Orbicularis oculi
43		Eyes closed	Relaxation of Levator palpebrae superioris; Orbicularis oculi, pars palpebralis
44		Squint	Orbicularis oculi, pars palpebralis
45		Blink	Relaxation of Levator palpebrae superioris; Orbicularis oculi, pars palpebralis
46		Wink	Relaxation of Levator palpebrae superioris; Orbicularis oculi, pars palpebralis
51		Head turn left	
52		Head turn right	
53		Head up	

Table A.4: AUs of FACS

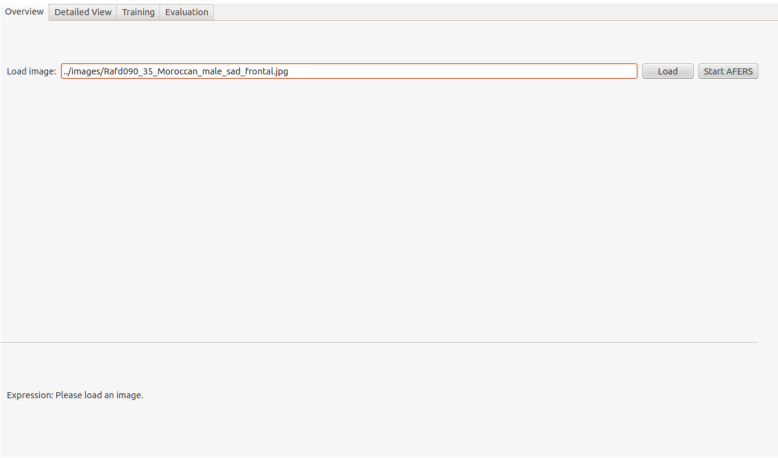
AU	Example Image	Description	Related Muscle
54		Head down	
55		Head tilt left	
56		Head tilt right	
57		Head forward	
58		Head back	
61		Eyes turn left	
62		Eyes turn right	
63		Eyes up	
64		Eyes down	

Appendix B

Screenshots of the Graphical User Interface

This Appendix refers to the Section “Graphical User Interface” in Chapter 5.

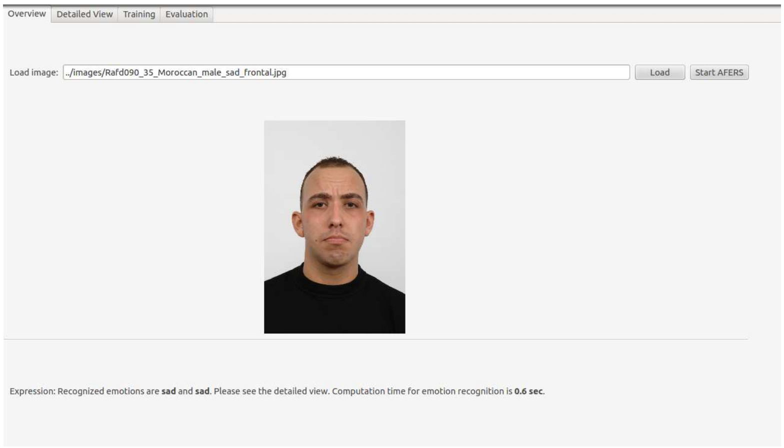
B.1 Screenshots



Load image: ../images/Rafd090_35_Moroccan_male_sad_frontal.jpg

Expression: Please load an image.

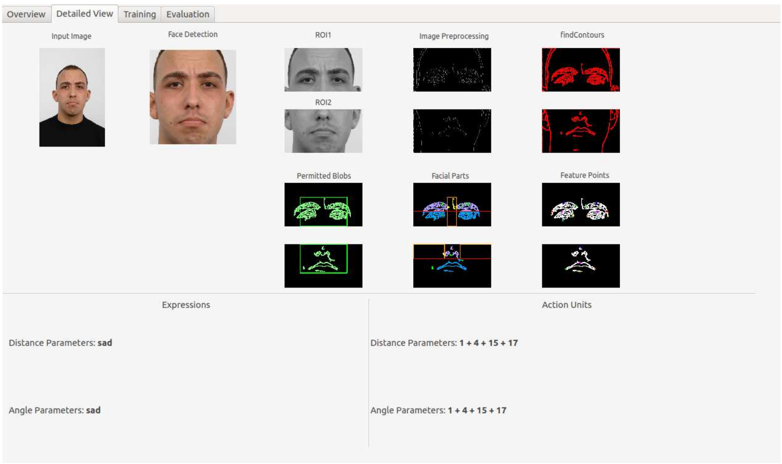
Figure B.1: First view of the user interface (overview)



Load image: ../images/Rafd090_35_Moroccan_male_sad_frontal.jpg

Expression: Recognized emotions are **sad** and **sad**. Please see the detailed view. Computation time for emotion recognition is **0.6 sec**.

Figure B.2: First view of the user interface (overview) with emotion recognition result



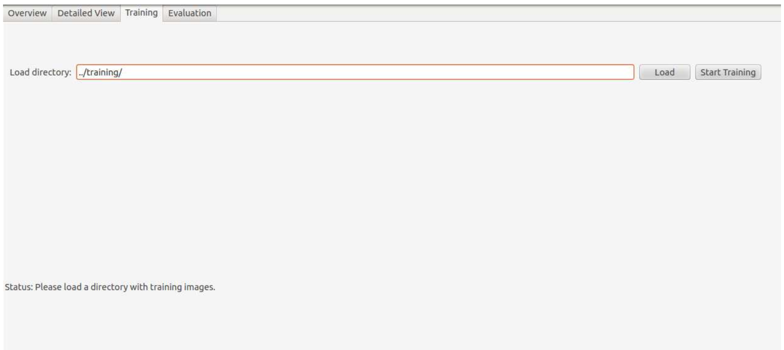
Distance Parameters: **sad**

Angle Parameters: **sad**

Distance Parameters: **1 + 4 + 15 + 17**

Angle Parameters: **1 + 4 + 15 + 17**

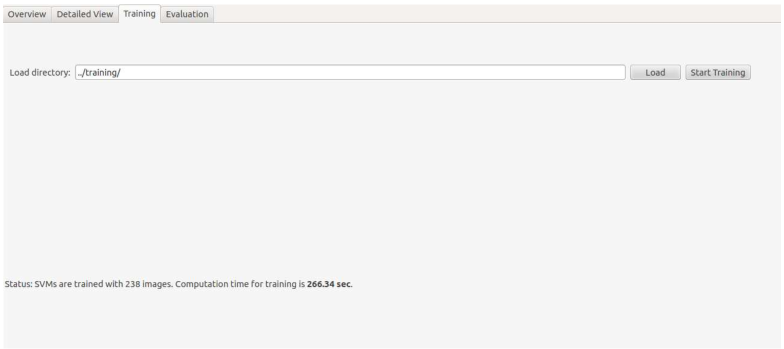
Figure B.3: Second view of the user interface (detailed view)



Load directory: ../training/

Status: Please load a directory with training images.

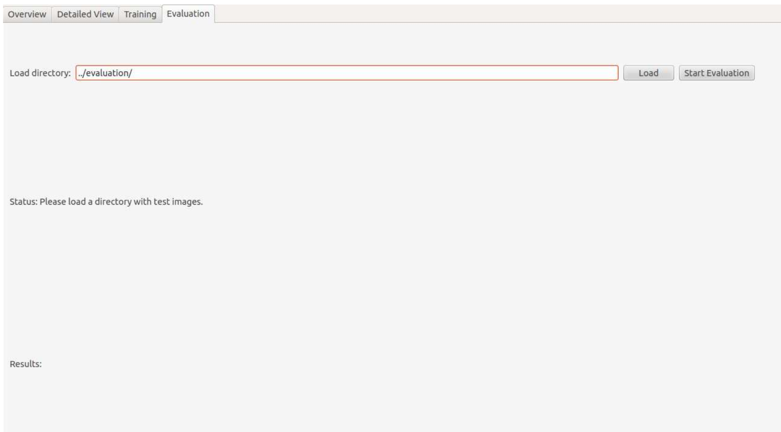
Figure B.4: Third view of the user interface (training mode)



Load directory: ../training/

Status: SVMs are trained with 238 images. Computation time for training is **266.34 sec**.

Figure B.5: Third view of the user interface (training mode) with images trained

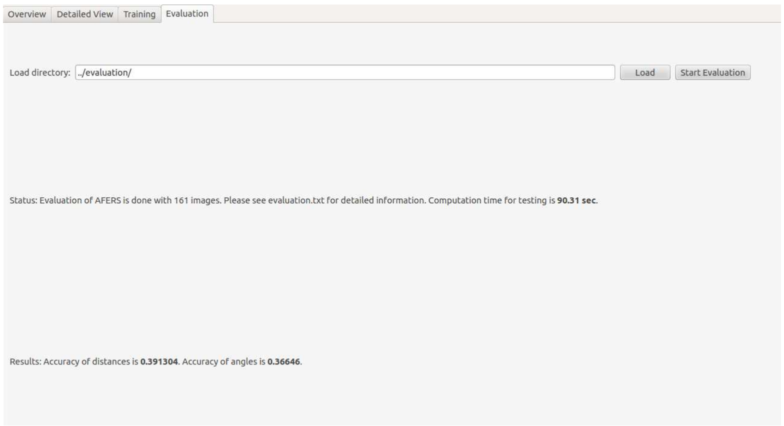


Load directory: ../evaluation/

Status: Please load a directory with test images.

Results:

Figure B.6: Fourth view of the user interface (evaluation mode)



Load directory: ../evaluation/

Status: Evaluation of AFERS is done with 161 images. Please see evaluation.txt for detailed information. Computation time for testing is **90.31 sec**.

Results: Accuracy of distances is **0.391304**. Accuracy of angles is **0.36646**.

Figure B.7: Fourth view of the user interface (evaluation mode) with evaluation result

Appendix C

Emotions of the Radboud Faces Database

This Appendix refers to the Section “Database” of Chapter 6. It presents several images of the seven emotions used in this thesis. All images are assembled from the RaFD [LDB+10].

C.1 Seven Emotions of Caucasian Female of the RaFD

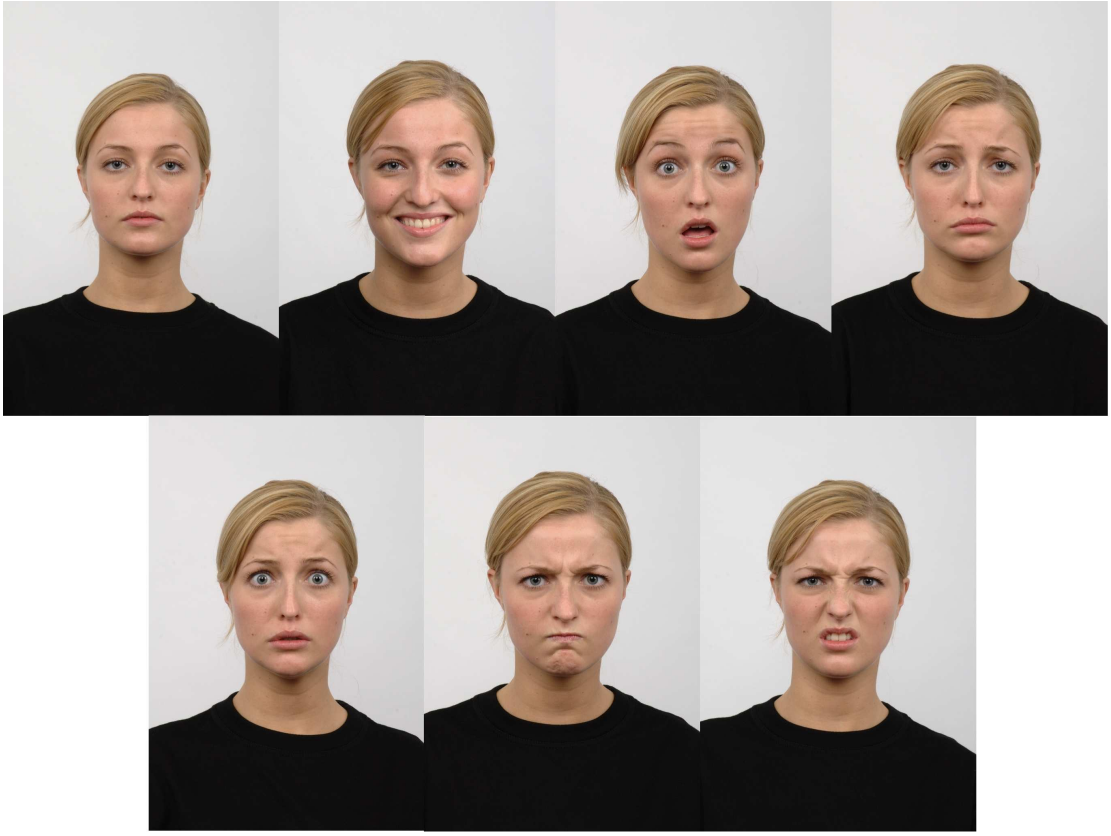


Figure C.1: Caucasian female expressing neutral, happy, surprised, sad, fearful, angry and disgusted

C.2 Seven Emotions of Caucasian Male of the RaFD

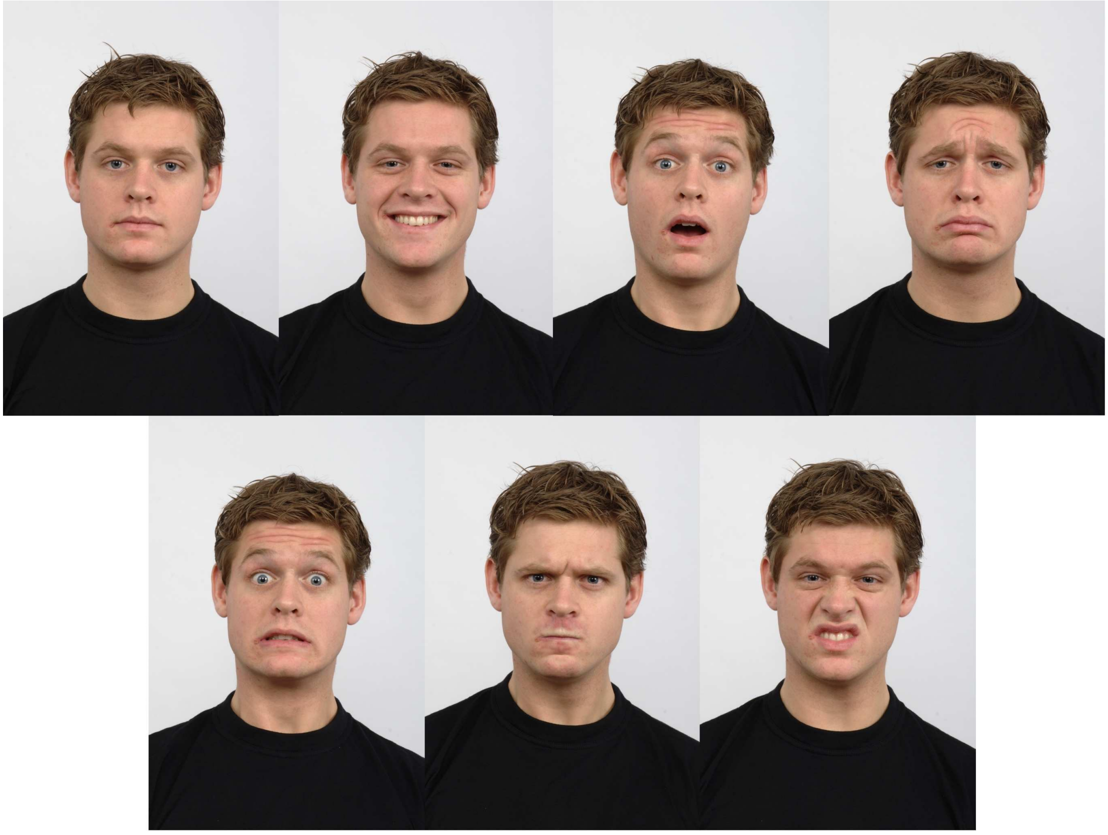


Figure C.2: Caucasian male expressing neutral, happy, surprised, sad, fearful, angry and disgusted

C.3 Seven Emotions of Moroccan Male of the RaFD

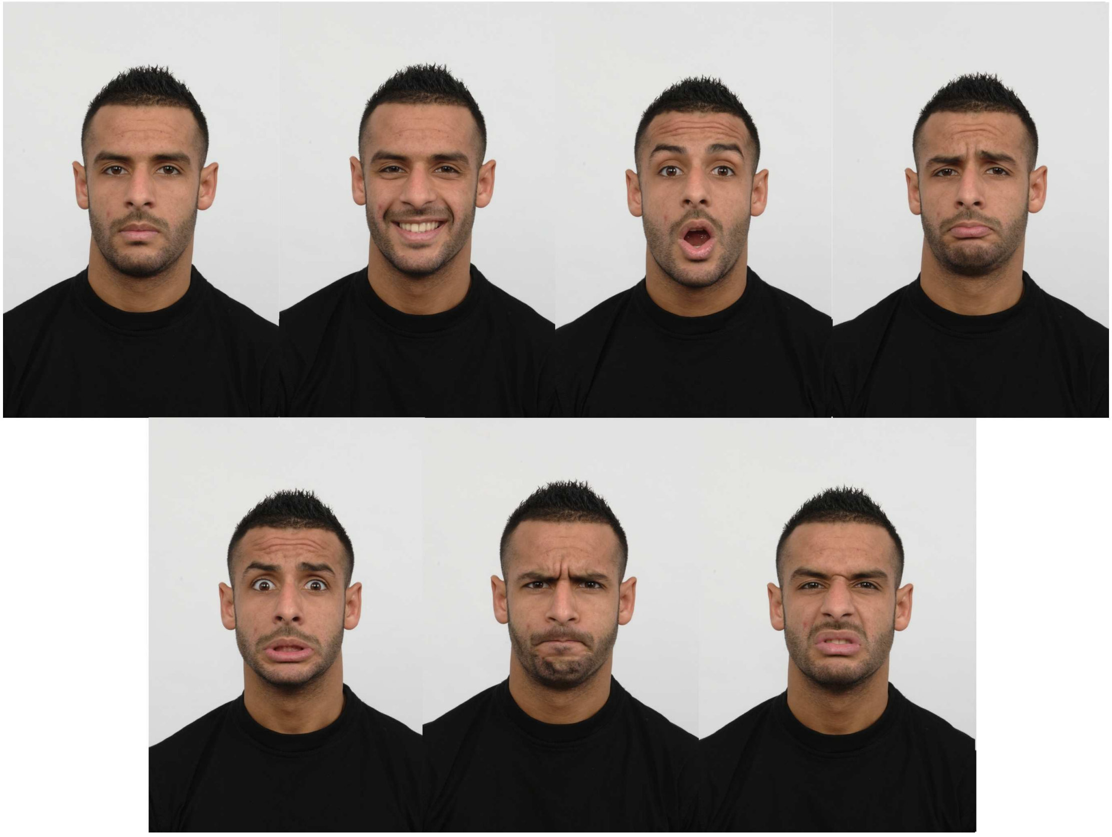


Figure C.3: Moroccan male expressing neutral, happy, surprised, sad, fearful, angry and disgusted

Appendix D

Receiver Operating Characteristics Graphs

This Appendix refers to the Sections "Confusion Matrix" and "Receiver Operating Characteristics Graph" of Chapter 6.

D.1 ROC Graphs of Distance Parameters

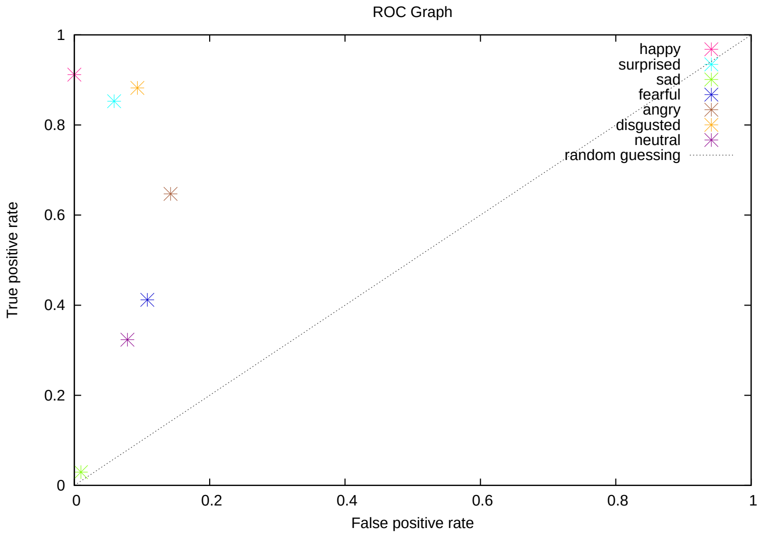


Figure D.1: True positive rate and false positive rate of 40% images trained and 60% images tested

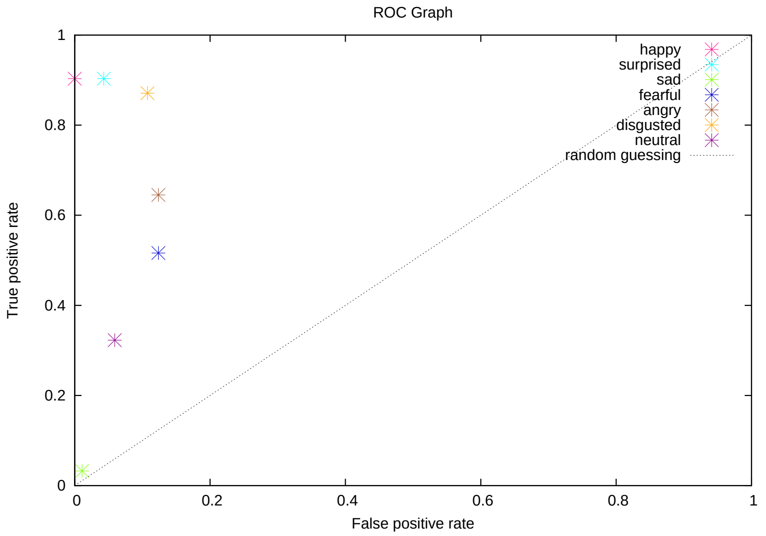


Figure D.2: True positive rate and false positive rate of 45% images trained and 55% images tested

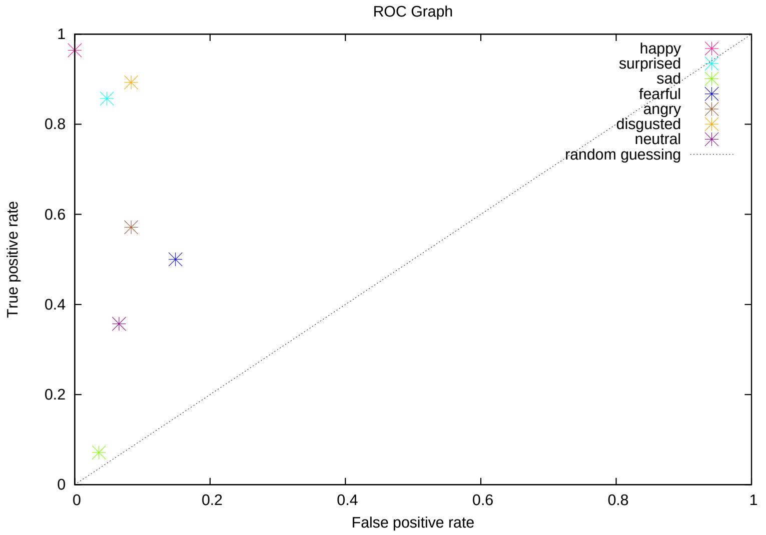


Figure D.3: True positive rate and false positive rate of 50% images trained and 50% images tested

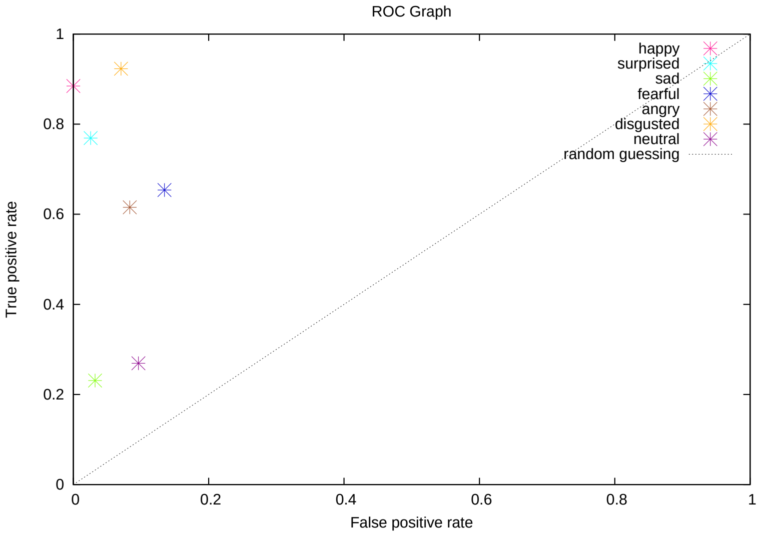


Figure D.4: True positive rate and false positive rate of 55% images trained and 45% images tested

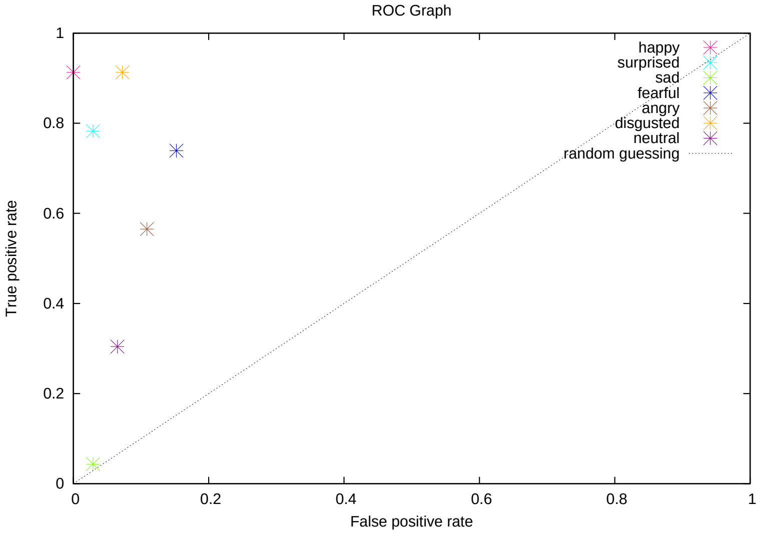


Figure D.5: True positive rate and false positive rate of 60% images trained and 40% images tested

D.2 ROC Graphs of Angle Parameters

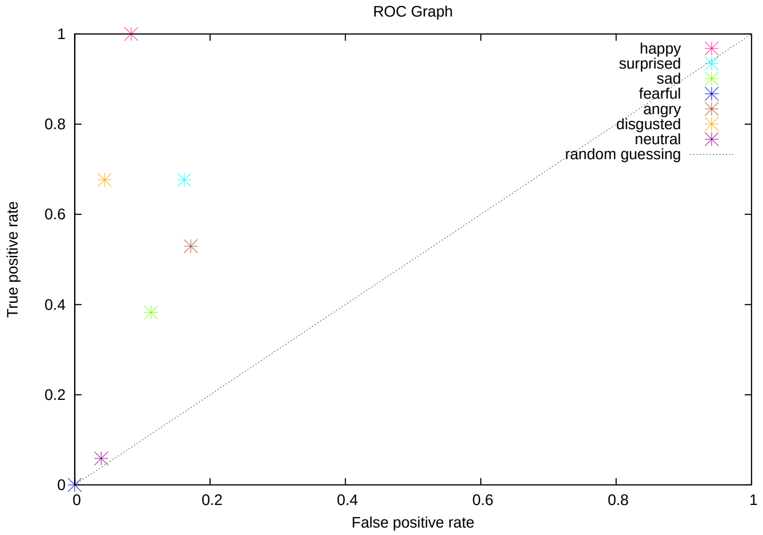


Figure D.6: True positive rate and false positive rate of 40% images trained and 60% images tested

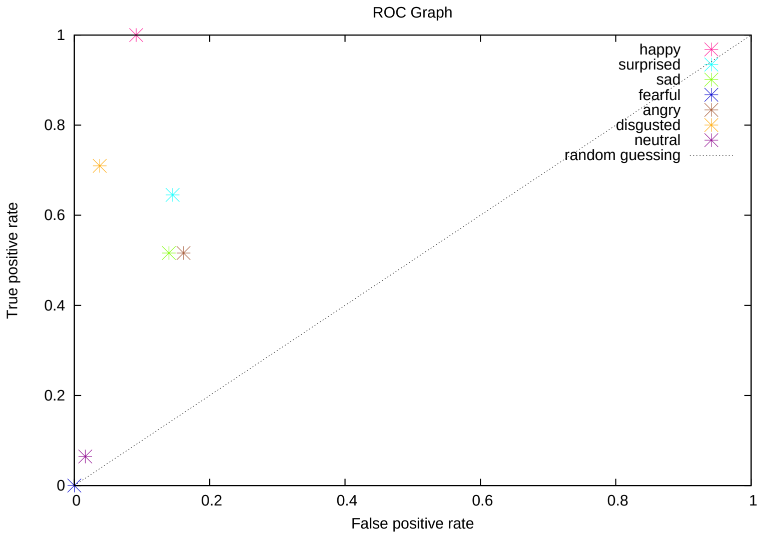


Figure D.7: True positive rate and false positive rate of 45% images trained and 55% images tested

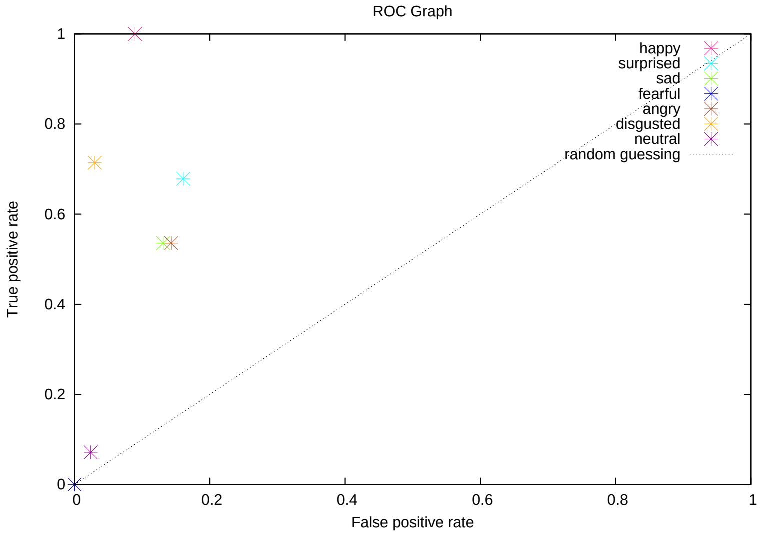


Figure D.8: True positive rate and false positive rate of 50% images trained and 50% images tested

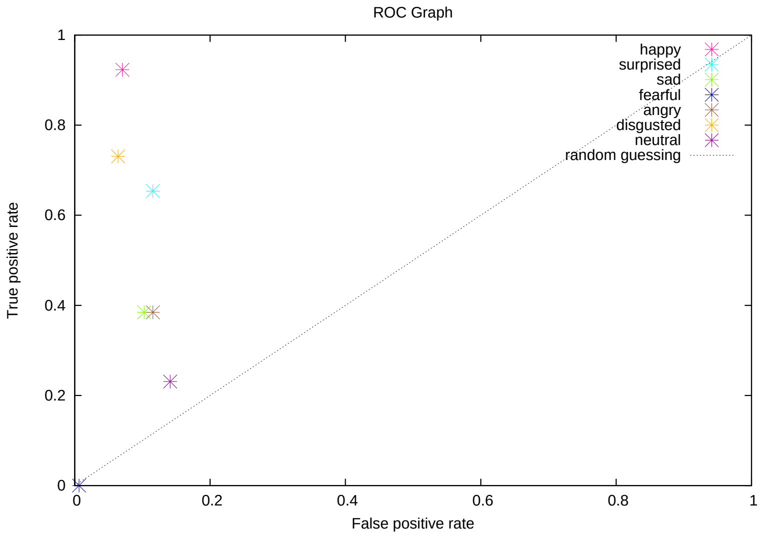


Figure D.9: True positive rate and false positive rate of 55% images trained and 45% images tested

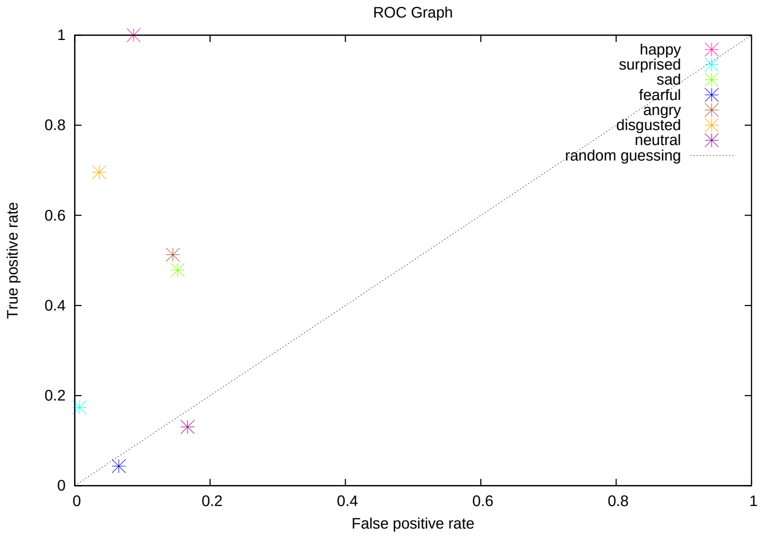


Figure D.10: True positive rate and false positive rate of 60% images trained and 40% images tested

D.3 Confusion Matrices of Distance and Angle Parameters

		recognized class						
		happy	surprised	sad	fearful	angry	disgusted	neutral
true class	happy	31	0	0	0	0	3	0
	surprised	0	29	1	3	1	0	0
	sad	0	2	1	11	10	4	6
	fearful	0	6	0	14	5	6	3
	angry	0	1	1	1	22	4	5
	disgusted	0	1	0	0	1	30	2
	neutral	0	2	0	7	12	2	11

Figure D.11: Confusion matrix of distance parameters with 40% images trained and 60% images tested

		recognized class						
		happy	surprised	sad	fearful	angry	disgusted	neutral
true class	happy	34	0	0	0	0	0	0
	surprised	0	23	7	0	4	0	0
	sad	3	4	13	0	9	2	3
	fearful	2	13	7	0	8	4	0
	angry	4	4	3	0	18	1	4
	disgusted	8	1	0	0	1	13	1
	neutral	0	11	6	0	13	2	2

Figure D.12: Confusion matrix of angle parameters with 40% images trained and 60% images tested

		recognized class						
		happy	surprised	sad	fearful	angry	disgusted	neutral
true class	happy	28	0	0	0	0	3	0
	surprised	0	28	1	2	0	0	0
	sad	0	1	1	11	9	5	4
	fearful	0	4	0	16	4	6	1
	angry	0	1	1	1	20	4	4
	disgusted	0	1	0	0	1	27	2
	neutral	0	1	0	9	9	2	10

Figure D.13: Confusion matrix of distance parameters with 45% images trained and 55% images tested

		recognized class						
		happy	surprised	sad	fearful	angry	disgusted	neutral
true class	happy	31	0	0	0	0	0	0
	surprised	0	20	7	0	4	0	0
	sad	4	2	16	0	7	1	1
	fearful	3	9	8	0	8	3	0
	angry	4	4	4	0	16	1	2
	disgusted	6	2	0	0	1	22	0
	neutral	0	10	7	0	10	2	2

Figure D.14: Confusion matrix of angle parameters with 45% images trained and 55% images tested

		recognized class						
		happy	surprised	sad	fearful	angry	disgusted	neutral
true class	happy	27	0	0	0	0	1	0
	surprised	0	24	1	3	0	0	0
	sad	0	1	2	11	5	4	5
	fearful	0	5	1	14	2	5	1
	angry	0	1	1	3	16	3	4
	disgusted	0	1	0	0	1	25	1
	neutral	0	0	3	8	6	1	10

Figure D.15: Confusion matrix of distance parameters with 50% images trained and 50% images tested

		recognized class						
		happy	surprised	sad	fearful	angry	disgusted	neutral
true class	happy	28	0	0	0	0	0	0
	surprised	0	19	6	0	3	0	0
	sad	3	1	15	0	6	1	2
	fearful	3	10	8	0	5	2	0
	angry	3	5	2	0	15	1	2
	disgusted	6	1	0	0	1	20	0
	neutral	0	10	6	0	9	1	2

Figure D.16: Confusion matrix of angle parameters with 50% images trained and 50% images tested

recognized class

		happy	surprised	sad	fearful	angry	disgusted	neutral
true class	happy	23	0	0	0	0	3	0
	surprised	0	20	1	3	0	1	1
	sad	0	0	6	10	4	4	2
	fearful	0	3	0	17	1	1	4
	angry	0	1	1	1	16	1	6
	disgusted	0	0	0	0	0	24	2
	neutral	0	0	3	7	8	1	7

Figure D.17: Confusion matrix of distance parameters with 55% images trained and 45% images tested

recognized class

		happy	surprised	sad	fearful	angry	disgusted	neutral
true class	happy	24	0	0	0	0	2	0
	surprised	0	17	3	0	2	0	4
	sad	2	2	10	0	5	2	5
	fearful	1	8	6	0	5	4	2
	angry	3	1	1	0	10	1	10
	disgusted	5	0	0	0	1	19	1
	neutral	0	7	6	1	5	1	6

Figure D.18: Confusion matrix of angle parameters with 55% images trained and 45% images tested

		recognized class						
		happy	surprised	sad	fearful	angry	disgusted	neutral
true class	happy	21	0	0	0	0	2	0
	surprised	0	18	1	4	0	0	0
	sad	0	0	1	8	7	4	3
	fearful	0	2	1	17	1	1	1
	angry	0	0	1	2	13	2	5
	disgusted	0	1	0	0	1	21	0
	neutral	0	1	1	7	6	1	7

Figure D.19: Confusion matrix of distance parameters with 60% images trained and 40% images tested

		recognized class						
		happy	surprised	sad	fearful	angry	disgusted	neutral
true class	happy	23	0	0	0	0	0	0
	surprised	0	4	5	0	2	0	12
	sad	3	0	11	1	4	1	3
	fearful	2	1	8	1	5	2	4
	angry	2	0	2	3	12	1	3
	disgusted	5	0	0	0	1	16	1
	neutral	0	0	6	5	8	1	3

Figure D.20: Confusion matrix of angle parameters with 60% images trained and 40% images tested

Appendix E

Installation and User Guide

This Appendix presents an installation and user guide for the system introduced in this thesis.

E.1 Installation Guide

The ROS release “fuerte” has to be installed to run the system. There are installation instructions [1] for certain operating systems. ROS supports several Ubuntu platforms.

The *emrec* package of the CD has to be copied to the ROS workspace. The images of the database can be copied to the directories mentioned in the next Section.

If there are any OpenCV errors while starting the AFERS, OpenCV 2.4.3 has to be installed separately.

E.2 User Guide

The AFERS starts with a launchfile. Therefore type *roslaunch emrec start.launch* in the terminal and the GUI opens.

For training mode, please select the “Training” tab. Insert the directory of training images in the input field. All training images should contain the correct emotion in their names. As default directory, “../training/” is displayed. This directory is located in the *emrec* package. An example of training images can be found on the CD in “Sonstiges/Datenbank RaFD/55 Training - 45 Testing/Training” where 55% of all 399 images are stored. If the directory is chosen, click on the “Load” button and then “Start Training”. The terminal presents all images

[1] `http://www.ros.org/wiki/fuerte/Installation`

trained. After training, the GUI shows the number of training images and computation time. Please note that minimum 200 images can be trained with the OpenCV train_auto function. For less training images, uncomment and comment the specific lines in the code of the files *AUTraining.cpp* and *EmotionTraining.cpp*.

For evaluation mode, please click on the "Testing" tab. Insert the directory of testing images in the input field. All testing images should contain the correct emotion in their names. As default directory, "../evaluation/" is displayed. This directory is also located in the *emrec* package. An example of testing images can be found on the CD in "Sonstiges/Datenbank RaFD/55 Training - 45 Testing/Evaluation" where 45% of all 399 images are stored. If the directory is chosen, click on the "Load" button and then "Start Evaluation". The terminal presents all images tested. After evaluation, the GUI shows the number of tested images and computation time. Additionally, accuracy of distance and angle parameters is shown. Please note that the SVMs have to be trained before evaluation can be done.

In order to test single images, please select the "Overview" tab. Insert the path of the image in the input field. As an example, "../images/Rafd090_01_Caucasian _female_happy_frontal.jpg" is displayed. The "images" directory is also located in the *emrec* package. If the image is chosen, click on the "Load" button. The input image is shown in the GUI. Click on "Start AFERS" to start the emotion recognition. Please note that the SVMs have to be trained before tested. When recognition is done, the GUI presents the emotions classified by distance and angle parameters, plus computation time. Select the "Detailed View" tab in order to see the results of several recognition steps. Additionally, the recognized AU-codes are presented. The terminal shows more detailed information about face detection, wrinkle detection, and resulting outputs. In addition, the system stores the output images of all steps in the "output" directory of the *emrec* package.

In order to use the system presented with other systems based on ROS, proceed as follows. Advertise the path of the input image with the topic "emrec_path". Publish a path, for example "../images/Rafd090_01_Caucasian_female_neutral_frontal.jpg". Select the "Overview" tab in the GUI. Click on the "Load" button. If a message containing a path of an image is received, the information "Image path received via ROS message" appears at the bottom of the GUI. The input image is shown in the middle. Click on "Start AFERS" to start the emotion recognition and to see all information in the "Detailed View" tab and in the terminal. Subscribe to the topic "emrec" in order to receive messages of the ROS node *emrec*. When emotion recognition finishes, the AFERS publishes messages like "neutral". Merely the recognized emotion is sent. Hence, systems subscribing the messages of the *emrec* node, are able to work directly with the received information. Emotion recognition only publishes results gained by distance parameters.

Bibliography

[AP99] Abrantes, G.A. ; Pereira, F.: MPEG-4 facial animation technology: Survey, implementation, and results. In: *Circuits and Systems for Video Technology, IEEE Transactions on* 9 (1999), Nr. 2, S. 290–305 xiii, 19

[Bet12] Bettadapura, V.: Face expression recognition and analysis: The state of the art. In: *arXiv preprint arXiv:1203.6722* (2012) 2, 3, 4, 5, 6, 25, 67, 84, 96

[BLL+04] Bartlett, M.S. ; Littlewort, G. ; Lainscsek, C. ; Fasel, I. ; Movellan, J.: Machine learning methods for fully automatic recognition of facial expressions and facial actions. In: *Systems, Man and Cybernetics, 2004 IEEE International Conference on* Bd. 1 IEEE, 2004, S. 592–597 7, 13, 20, 25, 29, 96

[Bur98] Burges, Christopher J.: A tutorial on support vector machines for pattern recognition. In: *Data mining and knowledge discovery* 2 (1998), Nr. 2, S. 121–167 34

[CAE07] Cohn, J.F. ; Ambadar, Z. ; Ekman, P.: Observer-based measurement of facial expression with the Facial Action Coding System. In: *The handbook of emotion elicitation and assessment* (2007), S. 203–221 6, 99

[Can83] Canny, J. F.: Finding edges and lines in images / M.I.T. Artificial Intell. Lab. Cambridge, 1983 (720). – Forschungsbericht 20

[CH06] Cerezo, E. ; Hupont, I.: Emotional facial expression classification for multimodal user interfaces. In: *Articulated Motion and Deformable Objects* (2006), S. 405–413 xiii, xiv, 16, 18, 21, 22, 27, 29, 53

[CL01] Chang, Chih-Chung ; Lin, Chih-Jen: *LIBSVM: a library for support vector machines*, 2001. – Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm 67

[CV95] CORTES, Corinna ; VAPNIK, Vladimir: Support-Vector Networks. In: *Machine Learning* 20 (1995), Nr. 3, 273-297. `citeseer.ist.psu.edu/cortes95supportvector.html` xiii, 34, 35, 36

[DBH$^+$99] DONATO, G. ; BARTLETT, M.S. ; HAGER, J.C. ; EKMAN, P. ; SEJNOWSKI, T.J.: Classifying facial actions. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 21 (1999), Nr. 10, S. 974–989 67

[DK05] DUAN, K.B. ; KEERTHI, S.: Which is the best multiclass SVM method? An empirical study. In: *Multiple Classifier Systems* (2005), S. 732–760 35

[EF77] EKMAN, P. ; FRIESEN, W.V.: Facial action coding system. (1977) 5

[EFH02] EKMAN, P. ; FRIESEN, W.V. ; HAGER, J.C.: Facial action coding system. In: *A Human Face* (2002) 5

[EWKK07] ESAU, N. ; WETZEL, E. ; KLEINJOHANN, L. ; KLEINJOHANN, B.: Real-time facial expression recognition using a fuzzy emotion model. In: *Fuzzy Systems Conference, 2007. FUZZ-IEEE 2007. IEEE International* IEEE, 2007, S. 1–6 xiii, 11, 12, 13, 14, 16, 17, 20, 21, 24, 26, 27, 29, 62, 64

[Faw06] FAWCETT, T.: An introduction to ROC analysis. In: *Pattern Recognition Letters* 27 (2006), Nr. 8, S. 861–874 84, 85, 87

[FL03] FASEL, B. ; LUETTIN, J.: Automatic facial expression analysis: a survey. In: *Pattern Recognition* 36 (2003), Nr. 1, S. 259–275 2, 3, 4, 6, 7, 9, 10, 13, 14, 20, 64

[FS95] FREUND, Y. ; SCHAPIRE, R.: A Decision-Theoretic Generalization of Online Learning and an Application to Boosting. In: *Computational Learning Theory, Second European Conference, EuroCOLT '95, Barcelona, Spain, March 13-15, 1995, Proceedings.* Barcelona, Spain, 1995, S. 23–37 40, 41

[FS$^+$96] FREUND, Y. ; SCHAPIRE, R. u. a.: Experiments with a new boosting algorithm. In: *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-* MORGAN KAUFMANN PUBLISHERS, INC., 1996, S. 148–156 41

[FSA99] FREUND, Y. ; SCHAPIRE, R. ; ABE, N.: A short introduction to boosting. In: *Journal-Japanese Society For Artificial Intelligence* 14 (1999), Nr. 771-780, S. 1612 41

[HCL08] Hsu, C.W. ; Chang, C.C. ; Lin, C.J.: A practical guide to support vector classification / National Taiwan University. 2008. – Forschungsbericht 35, 36, 37, 82

[HSK05] Huang, L.L. ; Shimizu, A. ; Kobatake, H.: Robust face detection using Gabor filter features. In: *Pattern Recognition Letters* 26 (2005), Nr. 11, S. 1641–1649 12, 13

[KCT00] Kanade, T. ; Cohn, J.F. ; Tian, Ying-Li: Comprehensive database for facial expression analysis. In: *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on* IEEE, 2000, S. 46–53 1, 6, 7, 73, 74

[KF09] Koller, D. ; Friedman, N.: *Probabilistic graphical models: principles and techniques.* The MIT Press, 2009 26

[KP07] Kotsia, I. ; Pitas, I.: Facial expression recognition in image sequences using geometric deformation features and support vector machines. In: *Image Processing, IEEE Transactions on* 16 (2007), Nr. 1, S. 172–187 20, 25, 29

[LAKG98] Lyons, M. ; Akamatsu, S. ; Kamachi, M. ; Gyoba, J.: Coding facial expressions with gabor wavelets. In: *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on* IEEE, 1998, S. 200–205 73, 74

[LCK$^+$10] Lucey, P. ; Cohn, J.F. ; Kanade, T. ; Saragih, J. ; Ambadar, Z. ; Matthews, I.: The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression. In: *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on* IEEE, 2010, S. 94–101 73, 74

[LDB$^+$10] Langner, O. ; Dotsch, R. ; Bijlstra, G. ; Wigboldus, D.H.J. ; Hawk, S.T. ; Knippenberg, A. van: Presentation and validation of the Radboud Faces Database. In: *Cognition and Emotion* 24 (2010), Nr. 8, S. 1377–1388 xiv, 26, 67, 69, 70, 73, 74, 75, 76, 109

[LFBM02] Littlewort, G. ; Fasel, I. ; Bartlett, M.S. ; Movellan, J.: Fully automatic coding of basic expressions from video. In: *INC MPLab TR* (2002), S. 53–56 12, 20, 25, 27, 29, 96

[LM02] Lienhart, Rainer ; Maydt, J.: An extended set of haar-like features for rapid object detection. In: *Image Processing. 2002. Proceedings. 2002 International Conference on* Bd. 1 IEEE, 2002, S. I–900 46

[OFG97a] Osuna, E. ; Freund, R. ; Girosi, F.: Training Support Vector Machines: An Application to Face Detection. In: *IEEE Conf. Computer Vision and Pattern Recognition* (1997) 12

[OFG97b] Osuna, E. ; Freund, Robert ; Girosi, F.: Support Vector Machines: Training and Applications. Version: 1997. `citeseer.ist.psu.edu/osuna97support.html`. 1997 (AIM-1602). – Forschungsbericht 37

[PF02] Pandzic, I.S. ; Forchheimer, R.: The origins of the MPEG-4 facial animation standard. In: *MPEG-4 Facial Animation* (2002), S. 1 15, 52

[PP05] Pantic, M. ; Patras, I.: Detecting facial actions and their temporal segments in nearly frontal-view face image sequences. In: *Systems, Man and Cybernetics, 2005 IEEE International Conference on* Bd. 4 IEEE, 2005, S. 3358–3363 26, 27, 29, 73, 95

[PP06] Pantic, M. ; Patras, I.: Dynamics of facial expression: Recognition of facial actions and their temporal segments from face profile image sequences. In: *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 36 (2006), Nr. 2, S. 433–449 26, 27, 29, 96

[PR04] Pantic, M. ; Rothkrantz, L.J.M.: Facial action recognition for facial expression analysis from static face images. In: *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 34 (2004), Nr. 3, S. 1449–1461 11, 14, 26, 27, 29

[PVRM05] Pantic, M. ; Valstar, M. ; Rademaker, R. ; Maat, L.: Web-based database for facial expression analysis. In: *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on* IEEE, 2005, S. 5–pp 73, 74

[RBK98] Rowley, Henry A. ; Baluja, Shumeet ; Kanade, Takeo: Neural Network-Based Face Detection. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 20 (1998), Nr. 1, S. 23–38 12, 14

[RCL$^+$09] Ryan, A. ; Cohn, J.F. ; Lucey, S. ; Saragih, J. ; Lucey, P. ; Torre, F. De l. ; Rossi, A.: Automated facial expression recognition system. In: *Security Technology, 2009. 43rd Annual 2009 International Carnahan Conference on* IEEE, 2009, S. 172–177 43

[SA85] Suzuki, Satoshi ; Abe, Keiichi: Topological structural analysis of digitized binary images by border following. In: *Computer vision, graphics, and image processing* 30 (1985), 4, Nr. 1, S. 32–46 54

[Sch03] Schapire, R.: The boosting approach to machine learning: An overview. In: *LECTURE NOTES IN STATISTICS-NEW YORK-SPRINGER VERLAG-* (2003), S. 149–172 40, 41, 42

[SF68] Sobel, Irwin ; Feldmann, G.: *A 3x3 Isotropic Gradient Operator for Image Processing*. 1968. – Unpublished, but often cited. Presented at a talk at the Stanford Artificial Project. 20

[SLS+07] Sebe, N. ; Lew, M.S. ; Sun, Y. ; Cohen, I. ; Gevers, T. ; Huang, T.S.: Authentic facial expression analysis. In: *Image and Vision Computing* 25 (2007), Nr. 12, S. 1856–1863 25, 96

[Sri12] Srivastava, S.: Real time facial expression recognition using a novel method. In: *arXiv preprint arXiv:1206.3559* (2012) 11, 12, 15, 20, 27, 29, 92, 95

[SZPY12] Sandbach, G. ; Zafeiriou, S. ; Pantic, M. ; Yin, L.: Static and dynamic 3D facial expression recognition: A comprehensive survey. In: *Image and Vision Computing* (2012) 27

[TK09] Theodoridis, Sergios ; Koutroumbas, Konstantinos: *Pattern Recognition*. 4. Academic Press, http://www.elsevier.com, 2009 25

[TKC01] Tian, Y.L. ; Kanade, T. ; Cohn, J.F.: Recognizing action units for facial expression analysis. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 23 (2001), Nr. 2, S. 97–115 xiii, 3, 7, 12, 15, 20, 24, 25, 27, 29, 62, 67, 98

[TKC05] Tian, Y.L. ; Kanade, T. ; Cohn, J.F.: Facial expression analysis. In: *Handbook of face recognition* (2005), S. 247–275 1, 2, 3, 4, 9, 10, 13, 14, 20, 62, 76

[VJ01a] Viola, P. ; Jones, M. J.: Robust Real-Time Face Detection. In: *International Conference on Computer Vision* Bd. 2 IEEE, 2001. – ISBN 0–7695–1143–0, S. 747 31, 34

[VJ01b] Viola, Paul ; Jones, Michael: Rapid Object Detection using a Boosted Cascade of Simple Features. In: *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on* 1 (2001), S. 511. ISBN 1063–6919 xiii, 31, 32, 33, 34, 46

[VKA$^+$11] VELUSAMY, S. ; KANNAN, H. ; ANAND, B. ; SHARMA, A. ; NAVATHE, B.: A method to infer emotions from facial Action Units. In: *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on* IEEE, 2011, S. 2028–2031 13, 20, 24, 25, 26, 29

[Wag12] WAGNER, Philipp: Machine Learning with OpenCV2. (2012) 82

[WO06] WHITEHILL, J. ; OMLIN, C.W.: Haar features for FACS AU recognition. In: *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on* IEEE, 2006, S. 5–pp 14, 20, 27, 29, 96

[WW$^+$98] WESTON, J. ; WATKINS, C. u. a.: Multi-class support vector machines / Citeseer. 1998. – Forschungsbericht 35

[YKA02] YANG, Ming-Hsuan ; KRIEGMAN, David J. ; AHUJA, N: Detecting Faces in Images: A Survey. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (2002), 1, Nr. 1 10, 12

[ZJZY08] ZHANG, Y. ; JI, Qiang ; ZHU, Zhiwei ; YI, B.: Dynamic facial expression analysis and synthesis with MPEG-4 facial animation parameters. In: *Circuits and Systems for Video Technology, IEEE Transactions on* 18 (2008), Nr. 10, S. 1383–1396 6, 13, 15, 21, 52

[ZPRH09] ZENG, Z. ; PANTIC, M. ; ROISMAN, G.I. ; HUANG, T.S.: A survey of affect recognition methods: Audio, visual, and spontaneous expressions. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31 (2009), Nr. 1, S. 39–58 1

[ZZ10] ZHANG, C. ; ZHANG, Zhengyou: A survey of recent advances in face detection. In: *Microsoft Research, June* (2010) 10