

Facial Expression Recognition with Support Vector Machines

Martin Michael Schulze

Thesis presented in partial fulfilment
of the requirements for the degree of
Diplom-Informatiker (FH)
at the University of Applied Sciences Konstanz, Germany

Supervisor:

Prof. Christian W. Omlin (University of the Western Cape)

Prof. Dr. Wilhelm Erben (University of Applied Sciences Konstanz)

Co-supervisors:

Dr. Konrad Scheffler (University of the Western Cape)

December 2002

Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

Signature:

Date:

Facial Expression Recognition with Support Vector Machines

Martin Michael Schulze

December 20, 2002

Abstract

This thesis investigates methods for the recognition of facial expressions using support vector machines. Rather than trying to recognize specific emotional expressions such as joy, anger, surprise and fear, we choose to recognize facial actions in the face such as raised eyebrow, mouth open and frowns. These facial actions are described in the Facial Action Coding System (FACS) and are essential facial components, which can be combined to form facial expressions. We perform independent recognition of 6 upper and 10 lower action units in the face, which may occur either individually or in combination. Based on a feature extraction from grey-level values, the system is expected to recognize under real-time conditions. Results are presented with different image resolutions, SVM kernels and variations of low-level features.

Acknowledgments

I wish to thank Prof. Christian Omlin for his support which made it possible for me to visit the University of the Western Cape. His organizational help and his input for this topic were of great value. Also I wish to thank Dr. Konrad Scheffler for his continuous input and technical guidance which improved my understanding of the subject considerably and made it possible to refine important aspects of this thesis. Furthermore, I like to thank the students and staff in the Department of Computer Science for their support.

Contents

Abstract	i
Acknowledgments	ii
1 Introduction	1
1.1 Introduction	1
1.2 Applications Scenarios	2
1.2.1 Improved Human Computer Interaction	2
1.2.2 Virtual Environments	3
1.2.3 Mobile Communication	3
1.2.4 Signed Language Translation	4
1.3 Facial Action Coding System	4
1.4 Related Fields	5
1.4.1 Face Recognition	5
1.4.2 Face Detection	6
1.5 Problem Statement	7
1.6 Technical Objectives	7
1.7 Methodology	8

1.8	Accomplishments	8
1.9	Thesis Outline	8
2	Literature Review	9
2.1	Introduction	9
2.2	Processing Stages in Facial Expression Recognition	10
2.3	Facial Feature Location	10
2.3.1	Permanent Facial Features	10
2.3.2	Transient Facial Features	11
2.3.3	Whole Face	11
2.4	Facial Feature Extraction	12
2.4.1	Spatial Deformation Approaches	12
2.4.2	Motion Based Approaches	13
2.5	Facial Feature Representation	14
2.5.1	Appearance-Based Models	14
2.5.2	Statistical Transformations	15
2.6	Facial Action Classification	16
2.6.1	Selected Facial Action Recognition Systems	17
2.6.2	Comparison of Selected Facial Action Recognition Systems	18
3	Support Vector Machines	20
3.1	Introduction	20
3.2	Binary Classification	21
3.2.1	The Learning Task	21
3.2.2	Finding the Optimal Hyperplane	22

3.2.3	Classification	23
3.3	Non Linear SVMs	23
3.3.1	Kernel Substitution	24
3.3.2	Suitable Kernels	24
3.3.3	Training and Classification	24
3.4	Non Separable Data	25
3.5	Multi-Class Classification Methods	25
3.6	Weighted SVMs	26
3.7	Summary	27
4	Facial Expression Recognition	28
4.1	Introduction	28
4.1.1	Classification Method	28
4.2	System Based on Low-Level Features	29
4.2.1	Facial Expression Database	29
4.2.2	Feature Extraction	31
4.2.3	Evaluation Criteria	32
4.2.4	Experiment Setup	33
4.2.5	Influence of the SVM-Kernel	33
4.2.6	Influence of Different Features	35
4.2.7	Influence of Different Numbers of Blocks	36
4.3	Evaluation and Results	38
5	Conclusions and Directions for Future Research	39
5.1	Conclusion	39

5.2	Future Work	40
-----	-----------------------	----

List of Tables

4.1	<i>Average recognition rates for AUs using different kernels</i>	34
4.2	<i>Average recognition rates for AUs using different features</i>	35
4.3	<i>Average recognition rates for AUs using numbers of blocks</i>	36

List of Figures

3.1	<i>Separating hyperplanes. (left) a random one, (right) one that maximizes the margin of separability</i>	22
4.1	<i>FACS AUs to be recognized</i>	30
4.2	<i>20x20 (left) and 30x30 (right) bitmap representations</i>	32
4.3	<i>AU recognition rate using different kernels</i>	34
4.4	<i>AU recognition rates using different features</i>	37
4.5	<i>AU recognition rates using different numbers of blocks</i>	37

Chapter 1

Introduction

1.1 Introduction

Human communication is to a large extent nonverbal. Body gestures consisting of hand signs, pose and facial expressions play an important role in communication as they support the meaning of speech and regulate social relationships. Facial expressions as the most expressive part in communication perform a number of different functions: Besides informing us about the affective and cognitive state of a person, they are used as social and conversational cues and carry semantics as well. A system that can recognize facial expressions in real-time will have applications in numerous fields such as computer vision, computer animation and psychology; It will be important to support digital visual communication and allow computers to interact more naturally with people.

Facial expression recognition is a hard problem. The face provides large amounts of information about psychological, physiological and cognitive processes in a person's mind. This information can be thought as signals or observations being emitted from an underlying hidden state of the person. These facial signals are very complicated and we need a system to classify those observations. The Facial Action Coding System (FACS) developed by Ekman and Friesen [12] is a method to measuring and describing facial activity. The description defines facial appearances in terms of facial muscle movements. FACS is the most common method to describe facial movement at the muscle level. It has been used for years in behavioral sciences; and is increasingly used as a basis for facial expression recognition systems.

While a lot of research has been directed towards systems recognizing 'prototypic' facial expressions such as fear, anger or disgust, few approaches exist which recognize facial expressions in terms of facial actions such as 'eye open' or 'mouth stretched'. State of the art systems recognize facial actions using high-level feature extraction and traditional learning techniques such as neural networks for classification. Not much work has been done on facial action recognition using support vector machines. Support Vector Machines are a fairly new technique introduced by Vapnik [26]. They are known to outperform traditional learning techniques even with simple feature extraction and can classify in real-time. Therefore, it is worth to investigate the performance of Support Vector Machines on facial action recognition using low-level feature extraction and the FACS method.

1.2 Applications Scenarios

Facial expressions are perhaps the most powerful 'channel' of nonverbal communication. We 'encode' messages in our own facial expressions, and we simultaneously 'decode' the faces of the people around us. In even the most simple interaction, our attention naturally focuses on the face, seeking to read some of the vital information we know is 'written' there. We constantly monitor the face because it provides vital clues to a variety of possibilities: attraction, whether a person likes or dislikes us, the complexity of emotions, identity, age, humor, and a person's regional and even national background.

Facial expression recognition is of great value to supplement human communication because of its importance for nonverbal communication. An automated system finds applications in Internet or cell-phone communication and in translation systems. The combination of recognition systems with the understanding of how facial expressions correspond to emotions and social situations may influence how humans interact with machines.

1.2.1 Improved Human Computer Interaction

A lot of research is being done to improve computers' social and emotional intelligence. A machine that could understand various social and emotional situations improves the interaction between humans and machines. Such a system can be used to assist humans

in situations that require people to make decisions based on a number of social and emotional variables. It could be a learning companion which is a teacher that helps scholars through their learning journey. Recognition of facial expressions is needed in a learning companion and is in general essential for the realization of natural and more intuitive human computer interaction.

1.2.2 Virtual Environments

Video conferencing, tele-presence and tele-teaching are video coding applications that have gained considerable interest in the past years. The transmission of image sequences imposes high demands of network bandwidth. Common Internet connections such as ISDN cannot handle the amount of data arising when transmitting video uncompressed. Compression techniques such as MPEG achieve high compression factors up to 1:100. The use of a facial expression recognition system which encodes the face within the image could further improve compression. Simultaneously, the scene content can be manipulated interactively. The idea of facial expression coding into a parametric description can be used to create a virtual 3-D world with actors generated synthetically. Their shape of the face and their facial expression are described by facial action parameters. The 2-D image sequence can finally be reconstructed by rendering the 3D objects onto a 2-D image using computer graphics techniques.

1.2.3 Mobile Communication

Face image compression will be of great importance to mobile communication. Current cell phone standards such as GSM do not allow the transmission of high quality visual content in real-time due to the lack of bandwidth. Current and upcoming cell phones are already and will be equipped with cameras and a common programming environment like the Java Micro Edition. A facial expression recognition system can encode faces with a few parameters. The parameters can be transmitted using GSM with low bit rates which enables visual communication with existing cell phones.

1.2.4 Signed Language Translation

Signed languages such as South African (SASL), British (BSL) or German (GSL) sign language are fully developed languages. They are used by the deaf community, a significant number of users amongst people. The languages are expressed by manual parameters and non-manual parameters. The former are expressed by hand gestures and body poses. The latter consist movements of the head, shoulders and trunk and facial expressions. A system recognizing non-manual parameters can therefore contribute significantly to an automated signed language translation system and can enhance quality of life of the Deaf community. A generic signed language translation systems for everyday use does not yet exist. However, developments are in progress for SASL [19] and GSL [4] for example. It seems that FACS is suited to recognize non manual parameters as it fully describes facial expressions.

1.3 Facial Action Coding System

Currently, most computer-based facial expression recognition systems classify expressions into a few broad categories. Those systems describe emotional states such as joy, anger, happy- and sadness. In the field of facial expression analysis, they are called 'prototypic' facial expressions since they model well known emotional categories. Many approaches in facial expression analysis use 'prototypic' facial expressions as a basis for recognition. A major disadvantage with this approaches is that they are not able to recognize all possible facial expressions. Prototypic facial expressions are used very seldom in non verbal communication. The majority of facial expressions in non verbal communication encompasses transitions between different states of facial expressions and movements of one or more facial features. The resulting variations in the arrangement and possible positions of facial features in the face contain vital information about facial expressions. So rather than using a high-level description such as emotional categories, a more precise description is needed to model the dynamic and subtle nature of the face. The Facial Action Coding System (FACS) developed by Ekman [12] is considered as the standard for model facial expressions. Compared to other coding systems such as Izard Maximally Discriminative Affect Coding System (MAX), which describes only a limited set of facial expressions and the MPEG-4-SNHC, which is a standard for the synthesis and animation of faces in movies, FACS is the most comprehensive choice for

the purpose of facial expression recognition in still images. FACS uses so called action units (AUs) to describe facial motion. AUs are derived from the physical face model and describe changes in the face caused by one muscle or a combination of muscles. The activity of face muscles can cause more changes in appearance. For example, the 'frontalis' muscle responsible for the eyebrows lifts the inner and outer eyebrow independently. Therefore, activity of a muscle is separated into several individual 'actions', resulting into two action units (AU1+2) in the case of raised eyebrows. FACS defines 48 AUs for the description of movements in the face in the eyebrows, mouth corners, eyelids, etc. Each AU describes the appearance of muscle movements in certain regions of the face. Each AU is designed to be independent from each other. A facial expression appearance can therefore be decomposed into combinations of single AUs. The ability to analyze facial expressions by decomposing them into smaller parts of visible changes allows to divide the problem of facial expression recognition into smaller subproblems. Researchers have therefore the possibility to solve each subproblem independently from each other.

1.4 Related Fields

Face detection and recognition are related to facial expression recognition in their methods used for feature extraction and classification. The face as a biometric plays an important role in modern day security procedures as it is a unique feature among people. Face detection is an independent field of research and is used for facial expression and face recognition systems. Those systems require face detection as a initial step to supplement their operation in a real-world environment.

1.4.1 Face Recognition

Face recognition verifies the identity of a person by a digital snapshot of the face. Facial features such as the position and size of the eyes, nose and mouth and the overall shape of the face contribute to the verification of a face. They are apt to uniquely identify a person among others. This feature is used in modern security systems, where the identity claimed by a person has to be verified. Such systems are becoming feasible and are already used at large public places such as sport stadiums to assist security personnel to detect the presence of known hooligans and troublemakers.

The face of an average person undergoes several changes over time due to changing hairstyle, weight fluctuations, hair growth, wearing of glasses, etc.. These changes pose challenges to face recognition systems. In order to recognize faces regardless of their possible changes, facial features are chosen, which do not change over time. These are permanent facial features such as the position of the eyes, nose and mouth as well as their spatial arrangement. The methods used for facial feature extraction and recognition are similar to those of facial expression recognition. One primary difference is the feature extraction. Face recognition uses features which are unique to a person's identity whereas facial expression recognition chooses features independent of individual characteristics.

1.4.2 Face Detection

Given a single image, the goal of face detection is to identify all image regions containing a face, regardless of its three dimensional position, orientation and lighting conditions. Additionally, robust face detection needs to deal with occlusions such as objects in the foreground of the face and on the face such as glasses, beards, hats, etc. There are various methods for detecting faces in an image. Computational cheap approaches use skin color for detection of the face or perform background removal based on color differences of the face and its background. More sophisticated approaches work with motion estimates of moving faces or combinations of motion, background and color. Approaches such as neural networks or support vector machines also find an application in this field [23], [20]. Training is performed on a large set of different faces from an image database. Learning machines detect a face by scanning the image for face-like patterns at all possible scales and classifying them to determine the appropriate class, i.e. face or non-face. The choice of face detection method depends on the envisaged application. Machine learning based approaches tend to be very robust due to their training in 'real world' environments. Algorithmic approaches tend to be very fast, as they can be specifically adjusted to the needs. Face detection in facial expression recognition has to be performed accurately and independently of the facial expression. Several facial expressions change the appearance of the face considerably like a wide opened mouth does and therefore the face would be detected with different results. Invariant detection of faces performing different facial expressions places additional requirements on the method of face detection.

1.5 Problem Statement

We investigate the recognition of facial expressions with FACS Action Units (AUs) using support vector machines based on the grey-level feature extraction. Facial expression recognition using action units has already been successfully done using high-level feature extraction and classifiers such as hidden Markov models and neural networks [16], [8]; Little research has been done on AU recognition using low level feature extraction but We know from related fields such as face verification and face detection that these tasks have been accomplished with very promising results. Both tasks utilize low level feature extraction; thus, it seems viable to use these methods since they are closely related to facial expression recognition.

1.6 Technical Objectives

In order to implement and experiment with a recognition system, feature extraction must be implemented and training must be performed on the features. Features extracted from images are presented as grey-level values. To investigate, which features are optimal, experiments on different features must be performed. Features which are promising and easy to compute are mean grey values and their derivatives variance and standard deviation. We have to find empirically, which features and combinations give the best recognition results. Training is performed on the features to recognize action units. We must be able to recognize action units either singly or in multiple combination as their presence or absence in a face informs us about the facial expression. The contents of the image database, i.e. the number of persons, the number of expressions by each individual and the overall amount of distinct expressions, determines the number of AUs which can be recognized. We want to recognize a basic subset of action units which consists of eyebrow-, eye- and mouth patterns according to the facial expression database we use.

1.7 Methodology

We first need to implement feature extraction. The faces, and their facial expressions, are given from plain grey-level images. The features in an image are presented as grey-level values. Not all pixels in an image contain a face. The background surrounding the face is not needed for recognition. To get rid of the background, a face has to be located within the image. Once the boundaries of a face in the entire image is known, only the pixel values in the boundaries are used for feature extraction. The remaining face region is still large in an standard sized image, i.e of dimension 640 x 400 pixels. This results in a large data space which becomes storage intensive and difficult to handle for a classifier. To avoid huge data dimensions, we have to subsample the image to a suitable resolution. The optimal resolution has to be found empirically. The second step is to train SVMs on the extracted features to recognize action units. We assume that facial expressions used for training are FACS coded manually by human observers. Additionally, we want to investigate different variations of features based on grey-level values and compare their classification performance.

1.8 Accomplishments

We implemented the facial expression recognition system under the Linux operating system. We used Imlib, a fast image processing library, to implement feature extraction. For classification we used LIBSVM, a library for support vector machines [5].

1.9 Thesis Outline

This thesis is organized as follows: In Chapter 2, we review the progress up to now within the field of AU recognition. We proceed to give a short introduction of support vector machines and their extensions relevant for our experiments in Chapter 3. This prepares the reader for Chapter 4 where we describe our AU recognition system and present experimental results. We conclude this thesis with a summary of our work and possible directions for future research.

Chapter 2

Literature Review

2.1 Introduction

Most of the computer vision systems for recognizing facial expressions attempt to classify expressions into a few broad emotional categories of emotion, such as happiness, sadness, anger and surprise [9], [18], [7]. Emotional or mental states in general are not sufficient to describe all possible facial expressions. If facial expression recognition is done only by emotional categories, much information is lost: variations within facial expressions, subtle changes in states of facial features, e.g. eyebrows, different intensities, conversational signals.

The set of expressions which can be described with FACS are almost complete. Therefore, facial action recognition based on FACS is different from emotion recognition in the methodology because it has to deal with the dynamic and subtle nature of the face. The literature review presented is focused on facial action recognition. Methods used for feature extraction and classification relate with methods used in computer vision in general. They also have their place in face recognition and emotional expression recognition.

We describe the relevant processing stages a facial expression system can be decomposed in and discuss various techniques used at each stage. Then, we will describe prominent facial expression recognition systems.

2.2 Processing Stages in Facial Expression Recognition

Various methods and techniques for solving complex task of facial expression recognition have been proposed. A generic facial expression recognition system can be decomposed in the following procession stages: Firstly, facial features are located in the face to focus on details in the face. Next, facial motion or deformation of facial features are extracted. The extracted features are often modeled prior to recognition with different representations. Finally, classification takes place to classify the expressions by action units.

2.3 Facial Feature Location

The information about facial expressions is contained in various regions of the face. Depending on the feature extraction method used, regions of the face are located and further features are extracted. Facial feature extraction from regions of interest improves recognition results significantly. Facial feature location takes place prior to feature extraction; it preselects certain regions of the face and leaves out uninteresting regions.

2.3.1 Permanent Facial Features

Relevant face regions are regions containing permanent facial features which form the facial expressions. These are normally the eyes, nose, eyebrows and the mouth but can also be permanent facial furrows which come with age. The deformation of those give important information about the expression. Systems which track facial features in moving pictures rely on such permanent features. [15] tracks eyes by detecting pupils using the red eye effect with an external camera. Bitmap regions are defined around the detected pupils with 140 x 80 pixel each for the eyes and 170 x 80 pixel for the eyebrows. The tracking algorithm works reliably even for head movements but fails for very fast movements. [25] proposed a method based on saccadic search to find the eyes and center of the mouth region. The approach is closely related to the human saccadic system. The system finds facial features by a sequence of large 'jumps', also known as saccades. Jumps are performed to regions in the face of interest with increased accuracy towards a facial feature. Support vector machines are separately trained on both eyes

and the center of the mouth and rate locations in the neighborhood. The next saccade is then performed on the best rated location. The saccadic search converges quickly, on average within 5 jumps. This fact makes the saccadic system feasible for real-time facial feature detection.

2.3.2 Transient Facial Features

In addition to permanent facial features that move and change their shape, facial movement also causes facial wrinkles, known as transient facial features. Transient facial features such as wrinkles that appear around the eyes, forehead and mouth occur often in combination with facial expressions displayed with high intensities. They are crucial for detection of certain AUs. Furrowed brows, for example, cause vertical wrinkles between them and are coded in FACS (AU4). Furthermore, raising of the cheeks (AU6) can be detected by crow's-feet wrinkles around the outside corners of the eyes. The presence or absence of facial furrows can be determined by the detection of frequent regularities formed by furrows in the skin texture. [16] detects wrinkles in the nasolabial region, upper nose and the outer corners of the eye. These areas are defined using the detected locations of permanent facial features. To quantify the amount and orientation of furrows, edge detection is used. [17] detects wrinkles and furrows using high gradient filters in different regions such as nasolabial region, lips, cheeks and forehead. The filters are sized and oriented differently to detect furrows in different orientation and smaller sizes such as in the chin region.

2.3.3 Whole Face

A face, separated from its background, can be used completely for feature extraction with no previous facial feature detection involved. It is useful to recognize facial expressions holistically. Holistic descriptions of the face preserve all the information in the face; they preserve regions which would be lost using only regions obtained from positions of detected facial features. There are different approaches used applied to the whole face. [1] uses difference images to find changes in the face. [17] uses dense optical flow and [10] uses Gabor wavelets to extract both motion and spatial features. With FACS coding, faces may be analyzed by treating upper and lower face parts separately. The lower face part encompasses the region from the nostrils downwards to the chin

whereas the upper face region goes upwards from the nostrils. Upper and lower face action units move relatively independent when forming facial expressions and thus be analyzed independent.

2.4 Facial Feature Extraction

Facial feature extraction methods differ according to the type of features used. Modern approaches can be categorized in motion extraction and spatial deformation extraction. Facial motion from an image sequence can be obtained from a sequence of images. It plays an important role in AU recognition in video sequences. Facial deformation extraction, on the other hand, obtains information from the texture of the face and permanent facial features. All information is obtained from a still image. Feature extraction may take place either in a holistically, using the whole face for processing, or locally, by focusing on permanent and transient facial features. In following we want to discuss important spatial deformation and motion based approaches.

2.4.1 Spatial Deformation Approaches

High Spatial Gradients

Most static facial expression recognition methods focus on high spatial gradients which are good indicators for facial actions. Features used in the frequency domain are obtained via high-pass or Gabor wavelet filters. High-pass filters enhance transient facial furrows and wrinkles in the face as well as edges from permanent facial features. Edges obtained from high-pass filters give information about shape of facial features but they are very vulnerable to varying contrast and illumination conditions. Therefore they are of limited use. Gabor wavelets are an improvement over high-pass filters because they remove most of the variability in images that occur to variation in lighting and contrast. They represent line endings and edge borders over several scales with different orientations. This property is of great value as the the shapes of facial features represent important features.

Gabor wavelets seem to perform well for AU recognition [1]. The extraction of Gabor components from entire images is computational expensive and not all the information is needed. Instead, extracting Gabor features from particular face locations provides

better results. [16] combined Gabor features with parametric description of facial features. Gabor wavelets were extracted from 20 locations automatically defined by geometric features for different spatial frequencies and orientations, resulting in 800 Gabor features. The results are improved from 88% using parametric descriptions only to 93% with Gabor features added; this is a modest gain when one keeps in mind the additional significant cost of the Gabor wavelet transformation.

2.4.2 Motion Based Approaches

Optical Flow

In contrast to spatial deformation-based approaches, motion-based feature extraction methods measure displacements of facial regions and facial features either over several frames or a frame and a reference frame. Most recent research has focused on optical flow. These approaches assume that facial motion can be tracked based on pixel motion in the face. Specifically, the velocity and direction of pixels moving in the entire face or locally within windows placed on certain face regions are computed. The computation results in a set of flow vectors pointing to the new position of pixel clusters compared to the previous frame. The mean similarity of flow vectors provides an estimate of muscle activity in the face. Basically, there are two optical flow methods: dense optical flow and feature point tracking. The main difference between these methods is the area they measure on. Dense optical flow estimates motion of all image pixels of the entire face whereas feature point tracking measures motion only on selected points in the face in order to track them.

[8] tracks feature points around the contours of the brows, eyes, nose and mouth. The feature points are marked manually in the first frame. Each feature point is automatically tracked using optical flow. [1] uses dense estimation of image motion to analyze facial expression changes. To compute the motion vectors, three algorithms are tried which are based on spatial gradients, correlation-based extraction of local velocity information and its refinement. A drawback of using dense optical flow is that it is relatively slow as it computes flow for every image pixel. Feature point tracking only extracts selected feature point locations whereas ignoring other facial regions. This results in a significant performance improvement which is an advantage over dense flow. A drawback of the feature point approach is that the placement of feature points often

has to be done manually and accurately which is time consuming.

Difference Images

Another motion based approach uses difference images which are created by subtracting a given test face from its reference neutral frame. Deformation of facial features and skin texture become visible in the resulting difference image. The approach is motivated from human experience. Human expert FACS coders can code some AU combinations more reliably, if a reference neutral frame is available. This suggests that computer implementations also take advantage of neutral reference frames. [10], [13] successfully applied difference images in combination with various statistical representation methods such as principle and independent component analysis. Difference images have the advantage of focusing on facial expressions and leave out individual characteristics of the face. A disadvantage is that neutral face frames have to be available for each person and precise alignment of test faces onto the neutral frame is necessary. Otherwise, the resulting difference image would not only reflect facial motion, but also disturbing pose information.

2.5 Facial Feature Representation

Facial features are modeled prior to recognition in many approaches. Holistic or local features are transformed to more meaningful, higher-level features. Transformation into higher-level features opens the possibility to reduce disturbing image noise and to control the amount of features needed and thus the computational expense. Facial features may be modeled manually with 2D or 3D facial shape models or automatically using unsupervised data transformations such as PCA or IDA.

2.5.1 Appearance-Based Models

Appearance-based models represent faces by using either 2-D or 3-D descriptions of the face. The descriptions are obtained by projecting the face in image onto an appropriate model. The model describes shapes and states of transient and permanent facial features such as eye openings, mouth width, etc. [16] models facial expressions using multistate face components. The model describes both permanent, e.g. lips, eyes,

brows, and cheeks, and transient components, e.g. furrows. The lip state is described by three states: open, closed and tightly closed. The eyes are described by two states, namely open and closed. The brow and cheek have a one state model. Transient features are included with present or absent states. The states are modeled with templates which are matched onto the face image. The templates must be aligned, partly by manual intervention. [15] defines an eyebrow and eye template for recognizing upper face AUs. The templates describe a detailed shape of the eyes and eyebrows by sequences of connected points. The points are extracted from rectangular regions around the pupils using example based learning. [11] estimate 3-D motion from 2-D image sequences. Head and shoulders are modeled by a detailed triangular mesh. To align the face onto the model, texture and depth information from a laser scanner are obtained. The set of facial expressions which can be displayed are mapped to the facial action parameters (FAPs) defined by the MPEG4 standard.

Common to both 2-D and 3-D models is the trade-off between accuracy of the parameters and number of parameters used to describe the facial expressions. A problem with 3-D models is the complex mapping procedure on the physical model which is very time consuming. Furthermore, transient facial features such as wrinkles are hard to represent with 3-D face models. Appearance based models in general tend to be very accurate and report the best AU recognition results [16]. One drawback is the need for manual intervention for the construction of the shape models. Shape models need to be placed precisely around permanent facial features.

2.5.2 Statistical Transformations

In contrast to appearance based approaches where prior knowledge of the face is necessary, pure data-driven methods can be applied without prior knowledge and find good parameters by themselves. Unsupervised methods such as principle component analysis (PCA) and independent component analysis (ICA), expose statistical properties of the input data to discover relevant features. PCA and ICA are often applied for computer vision problems in general and also specifically also to facial expression recognition.

Various face processing fields, such as face recognition, age estimation, gender classification and facial expression recognition have successfully applied PCA. PCA is considered as an standard approach for face processing to improve recognition results. PCA can

be used on plain image data as well as on higher-level features, where it is uncertain what the relevant features are or if parts of features can be left out. Relevant features have a high covariance with other features. PCA provides a way to label features accordingly to their relevance using eigenvalues. Features with low eigenvalue are insignificant in the data and can be omitted without significant loss of information. If the most significant features are kept for representation of the input data, a considerable dimensionality reduction is obtained. Faces transformed onto lower dimensions of principal components are called 'eigenfaces'. An individual's face can be reconstructed using a linear combination of eigenvalues.

A method related to PCA is Independent Component Analysis (ICA). Representations such as eigenfaces are based on second-order dependencies, the covariance of features (pixels), but are insensitive to high-order dependencies such as edges, elements of shape and curvature. In facial expression analysis, much information may be contained in high-order relationships in the face. ICA discovers high order dependencies by transforming the features in such that features components become statistically independent and not only uncorrelated like in PCA. This property makes ICA superior to PCA in recognition of identity. Both PCA and ICA have been proven to be suitable for facial action recognition [10], [13].

2.6 Facial Action Classification

There are few systems which recognize facial actions reliably and in sufficient detail. It is also important to distinguish between single AU and AU combination classification. The former one is easier to accomplish because a classifier can be trained on each AU separately. Classifying AU combinations is much harder due to the large number of possible combinations. Each AU cannot be treated separately. AUs can appear differently when occurring alone or in combination. For example, when AU4 occurs alone, the brows are drawn together and lowered. In AU1+4, the brows are drawn together but raised due to the appearance of AU1. In 'non additive' patterns, the combinations of AUs change the appearance of constituent AUs. This side-effect increases difficulty of AU recognition furthermore.

2.6.1 Selected Facial Action Recognition Systems

[16] recognizes 6 upper and 10 lower face AUs with an accuracy of 96%. The system uses appearance based models to train neural network classifiers separately on upper and lower face regions. AUs are classified by multiple outputs of the network, where each output node indicates the absence or presence of an AU. The performance was evaluated on faces performing up to combinations of three AUs. Misclassifications occurred due to confusion between similar AUs. The confusion was attributed to strong correlation in the eyebrow movements (AU1,2), similarities of movement in the eyelid (AU6,7) and movement of jaw in AU6, which was not detected. The system was extended to recognize further AUs, with more possible AUs combinations. The additional Gabor features did not significantly improve recognition results. Support vector machines and nearest neighbor classifiers archived more than 75% recognition rate for 26 AUs using regional appearance patterns.

[15] proposed a framework for analyzing facial actions and head gestures in real-time. It recognizes 7 upper FACS action units including neutral expression with support vector machines. A support vector machine is trained and tested to classify a single AU. A parametric description of eyes and eyebrows is extracted to represent the features. Recognition rates of 68% and 61% were achieved for individual AUs and combinations of AUs, respectively. Some AUs such as AU6 have been missed completely or are in the region of 50%. The results could not compete with those of comparable approaches. One reason for that was the usage of a natural dataset with occlusions and head movements for evaluation.

[10], [2] proposed a hybrid system that integrates three feature extraction methods: holistic PCA with difference images, local feature measurements for wrinkles and holistic dense optical flow. These three methods were tested on 6 upper and lower AUs giving recognition rates of 88%, 57% and 85%, respectively. The hybrid system achieved an accuracy of 91% using a single neural network. The results provide evidence for the importance of combining different features to improve recognition results.

[13] recognize asymmetric AUs and AU intensities. They used left and right halves of the face independently for recognition. In each half, 2 upper and 7 lower AU activities and 5 intensity levels per AU were recognized. For feature extraction, difference color images used. Experiments were of 83% with PCA and ICA in combination with a nearest neighborhood classifier. Best results for single AU occurrences were achieved using ICA. The highest recognition rate for AU combinations was 74% using ICA or

PCA. Intensity level measurement was not as accurate and the results remained below 50% using 5 intensity levels. Allowing AUs intensities to be ± 1 level away from ground truth resulted in 65% accuracy for AU combinations using ICA.

[8] recognizes 15 AUs and AU combinations. The system uses feature tracking on 37 points in the face which are manually marked in the first frame. Classification was done via separate discriminate function analysis conducted on the tracked points. The average recognition rate was 91%, 88%, and 81% for action units in the brow, eye, and mouth regions, respectively. Disagreements have been reported such as the distinction between AU1+4 and AU1 for the eyebrows and AU25 and AU26 for parted lips.

2.6.2 Comparison of Selected Facial Action Recognition Systems

It is difficult the performance of facial action recognition systems found in the literature due to different implementations taken. First of all, there are systems that are designed for recognition in image sequences and others are based on still images. Secondly, not all systems recognize the same set of AUs. Recognizing fewer AUs always leads to better recognition results as complexity decreases. Thirdly, almost all systems were tested on different databases providing different sets of AUs in different amount of samples. The database also limits the AUs which must be recognized. Furthermore, the test subjects vary according to gender, age, origin and are displayed in different light, pose and scale. Using a natural database changes recognition results considerable [15]. For a comparison, all systems would have to be tested on the same database and code the same set of AUs. [8] and [16] recognize the most AUs whereas [10], [2],[13] focus on smaller sets of AUs. [13] is one of the first approaches to handle AU intensity explicitly. It must be mentioned here that approaches using neural networks or SVMs have the ability to detect AU intensity levels inherently in the classification output intensity. All systems detect AU combinations and it can be seen that performance decreases significant when evaluating on combinations [15] and [13]. Others are not explicitly stating separate recognition rates. They are tested with a mixture of single AU and AU combinations. Testing on many combinations is important because it shows a system ability to detect 'real-world' facial expressions. [16] did an extensive evaluation on AU combinations and is the only one which measures performance based on correctly recognized facial expression samples. Others are testing recognition performance of single AUs and state the accuracy of AUs separately. AU recognition systems can be further distinguished according to their performance. Most complex systems

based on dense optical flow extraction hardly achieve real-time performance without employing special purpose hardware. Furthermore, most systems described in this section still need manual intervention. These are disadvantages because computer-based, non-verbal communication applications demand autonomous recognition in real-time. [15] shows that recognition is possible in real-time even though it recognizes only upper face AUs.

Chapter 3

Support Vector Machines

3.1 Introduction

The theory of support vector machines (SVMs) was introduced by Vapnik in 1979 [26]. Since then, it has found its way into many research areas, most notably image processing and speech recognition. The field of face processing, a sub-discipline of image processing, has successfully applied SVMs.

One reason for that is the efficient and fast implementations of SVMs such as sequential minimal optimization (SMO) [21]. This is a very important aspect of face processing because the best features are not known in advance; it is possible to accomplish the recognition task without extensive preprocessing and hardware requirements. Even with simple preprocessing and low-level feature extraction, face recognition, face detection and emotional expression recognition have been shown to be successful using SVMs.

As the work described in this thesis uses SVM extensively, we have included a chapter on the topic. It should be noted, however, that this chapter will not attempt to replace the excellent introductory work on SVMs found in papers such as the tutorial from Burges [3].

3.2 Binary Classification

SVMs are inherently binary classifiers. From the perspective of statistical learning theory, the motivation for considering binary classifier SVMs comes from theoretical bounds on the generalization error. Firstly, the error bound is minimized by maximizing the *margin*, i.e. the minimal distance between the hyperplane separating two classes and the data points closest to the hyperplane. Secondly, the upper bound on the generalization error does not depend on the dimension of the space.

3.2.1 The Learning Task

In the following sections SVMs are formulated as maximum margin classifiers. For simplification we assume that the input data is linearly separable. Under this assumption, two classes can be separated by finding a linear hyperplane between them.

Let us consider a binary classification task with input points $x_i (i = 1, \dots, l)$ having corresponding class labels $y_i = \pm 1$ and let the decision function be:

$$f(x) = \text{sign}(w \cdot x + b) \quad (1)$$

where $w \cdot x$ denotes the inner product and b the bias of the decision function.

A point x lying directly on the hyperplane satisfies the condition

$$w \cdot x + b = 0 \quad (2)$$

The label y_i of a data point x_i can be determined by evaluating the left side of equation 2. The sign of the result tells us about the class-label.

Points lying on the right or left side of the hyperplane must satisfy following conditions:

$$x_i \cdot w + b > 0 \quad (3)$$

$$x_i \cdot w + b < 0 \quad (4)$$

Both equations can be implicitly formulated as:

$$y_i (x_i \cdot w + b \geq 0) \quad (5)$$

Classification is correct if Equation 5 holds for all input points. It is clear that we therefore have to optimize the parameter w and b . The number of possible combinations of

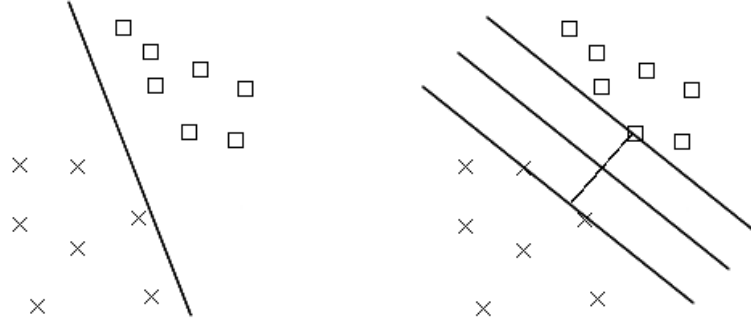


Figure 3.1: *Separating hyperplanes. (left) a random one, (right) one that maximizes the margin of separability*

weights w and bias b are large and may be not optimal. There is only one optimal separating hyperplane. An optimal hyperplane is one that maximizes the margin between two sets. The margin between two sets is given by the distances d_+ and d_- from the closest point of a set to the separating hyperplane. Figure 3.1 shows two separating lines, from which the right hyperplane is optimal.

3.2.2 Finding the Optimal Hyperplane

It is obvious that the margin ($d_+ + d_-$) reaches its maximum for $d_+ = d_-$. Let us denote H_- is a hyperplane which satisfies $x_i \cdot w + b = -1$ and H_+ is a hyperplane which satisfies $x_i \cdot w + b = 1$. Hence, with $d_- = d_+ = \frac{1}{\|w\|}$, the margin becomes $\frac{2}{\|w\|}$. Thus, the hyperplane that optimally separates the data is the one that minimizes $\|w\|^2$, subject to constraints in Equation 5.

We introduce Lagrangian multipliers α , $i = 1, \dots, l$, one for each of the inequality constraints of Equation 5 and achieve the following Lagrangian,

$$L_P \equiv \frac{1}{2} \|w\|^2 - \sum_{i=1}^l \alpha_i y_i (x_i \cdot w + b) + \sum_{i=1}^l \alpha_i \quad (6)$$

We must minimize L_P with respect to w and b .

This requires that the derivatives of L_P with respect to all the α_i vanish. Taking the

derivatives with respect to b and w gives:

$$\sum_{i=1}^l \alpha_i y_i = 0 \quad (7)$$

$$w = \sum_{i=1}^l \alpha_i y_i x_i \quad (8)$$

and re-substituting them back into Equation 6 gives:

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \quad (9)$$

Training is therefore accomplished by maximizing L_D with respect to constraints (7) and (8). In the solution, each Lagrange multiplier α_i is associated with a training point. Training points having $\alpha_i > 0$ are called 'support vectors', hence the name support vector machine. Support vectors lie closest to the separating hyperplane and are therefore critical elements in the dataset. Changing them results in a different hyperplane. All other points, with $\alpha_i = 0$ do not influence the shape of the decision boundary.

3.2.3 Classification

Once we have found a solution to the optimization problem, the SVM can attempt to classify unseen instances. An instance x can be classified by determining the side of the decision boundary it falls, i.e. we compute:

$$f(x) = \text{sign}(w \cdot x + b) = \text{sign} \left(\sum_{i=1}^l \alpha_i y_i (x_i x) + b \right) \quad (10)$$

3.3 Non Linear SVMs

We assumed the input data is to be linearly separable. Real world data, however is generally not linear separable. To address this issue, SVMs separate non linear data using a trick. The trick is to extend linear SVMs to nonlinear SVMs by mapping the input data nonlinearly into a higher dimensional space called *feature space*. In sufficient high dimensions, the data becomes linearly separable. With this mapping in mind, the SVM can solve the optimization problem in the feature space as it would do in the input space and find an optimal separating hyperplane. Once the optimal

hyperplane is found, it is mapped back into the input space resulting in a non-linear decision surface.

3.3.1 Kernel Substitution

For the objective function in Equation 9, we notice that the data points x_i only appear in the form of an inner product. Thus, the mapping of features into higher dimensional space is achieved through a substitution of the inner product:

$$x_i \cdot x_j \rightarrow \phi(x_i) \cdot \phi(x_j) \quad (11)$$

The mapping $\phi(x)$ and its dot product in Equation 11 must not be computed as it is computational expensive and storage intensive. Instead, the mapping is implicitly defined by a kernel:

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) \quad (12)$$

3.3.2 Suitable Kernels

Common kernels used for SVMs are:

$$K(x_i, x_j) = (x_i \cdot x_j + 1)^d \quad (13)$$

$$K(x_i, x_j) = e^{\frac{\|x_i - x_j\|^2}{2\sigma^2}} \quad (14)$$

$$K(x_i, x_j) = \tanh(\beta x_i \cdot x_j + b) \quad (15)$$

Equation (13) results in a polynomial classifier with degree d . Eq. (14) gives Gaussian radial basis function classifiers. Eq. (15) emulates two layer sigmoid neural networks.

3.3.3 Training and Classification

The optimization problem for different kernel. The learning task therefore involves maximization of the objective function:

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (16)$$

subject to constraints (7) and (8).

Classification remains the same, except the inner product becomes a kernel evaluation:

$$f(x) = \text{sign}(w \cdot x + b) = \text{sign} \left(\sum_{i=1}^l \alpha_i y_i K(x_i, x) + b \right) \quad (17)$$

3.4 Non Separable Data

Most real world data sets contain noise and cannot be separated by an optimal hyperplane without leading to poor generalization. If there are input points (x_i, y_i) within the margin of separation, optimization of Equation 5 is not possible. These input points are points which either fall on the correct side of the decision surface, but within the region of separation, or fall on the wrong side of the decision surface. To take violating points into account, the definition of the optimal hyperplane has to be extended. Therefore, we introduce a slack variable ξ for each training sample and require that

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i \quad (18)$$

be satisfied. This extension allows the old constraint (5) to be violated, but in a way the violation is penalized. The new optimization problem thus becomes maximization of the margin between two sets and the minimization of misclassifications caused by points lying in the range of $0 \leq \xi$. In training, however, one wants to regularize those two aspects of optimization according to the type of problem one wants to solve. Regularization is done by weighting the error of misclassifications with a value C . A high C value causes a high penalty assigned to classification errors leading to better separation. A low C value causes the margin to be soft with lots of permitted errors and causes the separation to be fuzzy. The C value has to be found empirically in training because it is not known in advance what the training data looks like and how general or specific separation must be done.

3.5 Multi-Class Classification Methods

Since an SVM is an inherently binary classifier, several schemes have been applied to enable multi-class classification. There are two approaches for construction of multi-class SVMs: The first consists of a combination of several binary classifiers, and the

second which considers all data in the optimization. The number of binary SVMs constructed to solve multi-class classification is proportional to the number of classes; this multi-class SVMs are computationally more expensive to train than binary SVMs.

One-Versus-All

The one-versus-all approach combines $k - 1$ out of k classes into a single class and trains it against the remaining class. To train all classes, the procedure is repeated for each single class in k ; thus training results in k SVMs.

One-Versus-One

The one-versus-one approach trains $k(k - 1)/2$ classifiers. Each classifier is trained on data from two classes. To select the most probable candidate out of k classes, a voting system is used: if x is classified into the m th class, then a vote for the n th class is increased by one, otherwise the n th class is increased by one. The class with the most number of votes wins. If two classes have the same number of votes, the class with the lower index is selected.

3.6 Weighted SVMs

For many recognition problems and the problem addressed in this thesis the numbers of data points in different classes are unbalanced. Classes represented with a small amount of training samples causes the SVM to misclassify them completely in some cases. Therefore, it is necessary to weight a training sample for a low representative class higher than for a less representative. One way is to penalize misclassifications for each class differently. Hence, some researchers (e.g. Osuna [20]) have proposed to use different penalty parameters in the SVM formulation. Since C is an error penalty attached to the data vectors, two C 's can be used to implement two different error penalties. Positive errors can be weighted by C_+ and negative errors by C_- . [6] did experiments with different error penalties for a two class radar detection problem. The positive class were the target images such as vehicles, the negative those without vehicles. The class set size ratio was 1:20 for positive to negative class, respectively. They found empirically that, with a weighting ratio equal to the ratio of positive to

negative training samples gave a good compromise between probability of detection and probability of false alarm. The time to determine the best weighting ratio was significantly less than it would take researching for better features.

3.7 Summary

Support vector machines are a statistical learning paradigm well-suited for use with low-level features. The adjustable parameters are intuitive and allow to use related classification methods such as neural networks via a the kernel trick. Classification methods emulated via the kernel archive better performance due to the optimal separating hyperplane the SVM seeks to find. It provides a better solution for the problem with a high probable global optimum. This chapter provided a short introduction on support vector machines to understand the methodology described in the next chapter. Support vector machines are well documented and several implementations exist for download from the Internet.

Chapter 4

Facial Expression Recognition

4.1 Introduction

4.1.1 Classification Method

Our aim to recognize combinations of AUs leads to a multi-class problem. However, unlike the well-studied binary SVM classifiers, construction of a multi-class SVM is still believed to be an unresolved problem [22]. An appropriate method for multi-class classification depends on the underlying input representation and the decision how the different classes should be recognized. There are two issues we have to consider in order to achieve good recognition results. In multi-class recognition, machine learning approaches such as SVMs, Bayesian classifier and radial basis function networks presuppose the statistical independence of different classes. As they are based on separation of classes via a decision surface, learning is accomplished best if they form distinct clusters in the feature space. In FACS, however, this is not the case. AUs are dependent on each other in their appearance as they naturally correlate with other AUs. For example, lifting the inner eyebrow causes the outer eyebrow to raise a little and vice versa. Such appearances are often misclassified by humans as well. Furthermore AU recognition requires one to assign multiple AUs to a facial expression. The need to assign multiple classes to an instance arises naturally in bio-informatics or text classification where a single instance can belong to multiple target classes, i.e. multiple functions for a gene. Existing SVM multi-class classification approaches are not natively supporting

multi-label classification. A common solution is the decomposition of multi-label classification into many binary classification tasks. We are adopting the binary approach for its simplicity. It opens the possibility to train an AU separately and avoid confusion with other dependent AUs. To distinguish between the occurrence of an AU and its neutral state, all images containing the AU are used as positive class and remaining images, not containing the AU as negative class. To improve the reliability of detection and reduce the probability of misclassification of an AU, all remaining AUs are used for negative examples.

We constructed separate SVMs for each AU. During testing, an AU is recognized in a facial expression if its corresponding SVM detects it in an image. In order to detect all AUs in an image, each SVM is evaluated. Facial expressions are detected by integrating all SVM outputs.

4.2 System Based on Low-Level Features

4.2.1 Facial Expression Database

We performed experiments using the Cohn-Kanade Facial Expression Database (CMU database) [14]. It provides image sequences of facial expressions performed by adults of different genders, age and country of origin. Each image sequence starts with a neutral expression and ends with the performed facial expression. The last image in the sequence is manually AU coded by certified FACS coders and provides action units either singly or in combination. Our version of the database contains 97 persons each performing 3 to 10 facial expressions. Altogether, the database contains 480 image sequences with 20 pictures per sequence on average. We used the first and the last two images in the sequence as a neutral expression with no AU associated and the final FACS coded facial expression, respectively. Very minor changes in the expression at the beginning and the end of the image sequences allow us to double the amount of available still images.

The number of action units which we can do experiments on, depends on the number of samples the database provides for each AU. Our database provides 33 AUs of which 10 AUs correspond to facial actions in the upper face region and 23 to the lower face region. Some AUs are present in very few images such as lip AUs (AU10,13,16,18,22,28), cheek

dimpler (AU14) and AUs describing movements in the eye region (AU41-46). They could not be used for experimentation. Preliminary tests showed the SVM could not perform well on AUs present in a low number of samples (below 100). We achieved a high misclassification rate for the detection of the presence for these AUs. We only used AUs which occurred in more 120 images. We attempt to recognize 6 upper and 10 lower AUs in this study. Figure 4.1 depicts the upper and lower face AUs respectively.

















AU1	AU2	AU4	AU5
			
Inner portion of the brow is raised	Outer portion of the brow is raised	Brows lowered and drawn together	Upper eyelids are raised
AU6	AU7	AU9	AU12
			
Cheeks are raised	Lower eyelids are raised	Nose is wrinkled	Lip corners are pulled upwards
AU15	AU17	AU20	AU23
			
The corners of the lip are pulled down	The chin boss is pushed upwards	Lips are stretched	Lips are tightened
AU24	AU25	AU26	AU27
			
Lips are pressed together	Lips are relaxed and parted	Lips are relaxed and parted: jaw is lowered	Mouth is stretched and open

Figure 4.1: *FACS AUs to be recognized*

4.2.2 Feature Extraction

Before the extraction of luminance values from the image, the image has to be preprocessed. The preprocessing consists of the following stages:

- Face localization
- Grey-level conversion
- Block segmentation

Usually, in real-world recognition systems, the face has to be detected and located within the entire image. Such systems are designed to work in environments where the content and quality of the image is not known in advance. Interferences in the acquisition of images via video or different illumination and contrast conditions make expensive and comprehensive preprocessing necessary. At this stage we do not concern ourselves with face detection and robustness of the system for the experiments. We thus can assume that the presence of a face in an image is always assured and lightning conditions remain constant. To simplify face localization faces are located manually; we identified the face regions of the images by hand by defining a rectangular bounding box around the face for each image in the database. We sized the bounding boxes from the middle of the forehead to the bottom of the chin; the left and right borders are located at the intersection of ears and cheekbone.

From the defined region, the RGB values of each pixel were converted into grey-level values. The formula used to get the luminance of a pair (R,G,B) is:

$$L = 0.34 * R + 0.50 * G + 0.16 * B$$

The dimension of the face is relatively large (300 x 400 pixels and higher) and with it the dimension of feature vectors. SVMs tend to overfit for high dimensional feature vectors if the number of training images in the database is comparably small. In order to avoid overfitting the data, we needed to reduce the size of the input vectors. Thus, we divided the located image region of the face into a constant number of blocks. We chose block sizes which divide the image into 15x15, 20x20 and 30x30 blocks. From each block, the average and variance of luminance values from all pixels in the block were extracted. Figure 4.2 depicts bitmap representations of facial expressions for 15x15 and 20x20 blocks after cropping the images along the bounding boxes.



Figure 4.2: *20x20 (left) and 30x30 (right) bitmap representations*

4.2.3 Evaluation Criteria

In our investigation, we trained and tested each AU separately on the database. The recognition rate of an AU contributes to the overall recognition rate. We computed the recognition results for an AU based on the 1 vs. all classification method where an AU is compared against all other AUs including the neutral state (AU0). Let T_p be the number of positive samples (AU) and T_n the number of negative samples (non AU) in a two class problem. C_p and C_n are the number of detected and misclassified AUs and F_n the number of false alarms. The recognition rate R for an AU and false alarm rate F is computed by Equation 19 and 20, respectively.

$$R = \frac{C_p + C_n}{T_p + T_n} \quad (19)$$

$$F = \frac{F_n}{T_p + T_n} \quad (20)$$

The average overall recognition rate R_{over} for AUs is computed by the average of recognition rates for all 16 AUs and is shown in Equation 21:

$$R_{over} = \frac{1}{16} \left(\sum_{i=1}^{16} R_{AU_i} \right) \quad (21)$$

The overall average recognition rate R_{over} can be misleading, because of low prior probabilities for AU activations. This means that a high overall recognition rate can be due to a low false alarm rate and does not necessarily imply a high rate of correct detection. For better clarification we also present the recognition results for positive and negative class separately.

It is difficult to compute a facial expression recognition rate based on the recognition rates of single AUs. AUs occur with different prior probabilities and are correlated. An

overall recognition rate cannot be computed simply by the product of all AU recognition rates. Different facial expressions occur with varying frequency, e.g. a smile might occur more often than a fright in empirical studies. The probabilities of facial expressions and thus those of AUs can be described via a facial expression grammar, however, this is beyond the scope of this thesis.

4.2.4 Experiment Setup

We performed 10-fold cross validation. In order to create distinct datasets for cross validation, none of the individuals in a folder appear in any of the remaining folders. We formed a unique cross validation set for each AU to be recognized. The sizes of the different folds are only approximately equal as the database contains different numbers of facial expressions for different individuals. Since the number of positive examples in the cross validation set is smaller than the number of negative examples, we chose the weights for parameter C , $C-$ and $C+$, according to the ratio of negative and positive examples for each AU class.

4.2.5 Influence of the SVM-Kernel

We performed training on the following kernels:

- Linear
- Polynomial of degree 3
- Radial basis function (RBF)

The performance of a kernel is dependent on the error tolerance parameter C . For a two class problem, the kernel generalizes optimally, when the error tolerance parameter C is set to the optimal value. A large C value means low error tolerance while smaller C means high error tolerance. C has to be determined empirically and cannot be known in advance. AU classification can thus be further optimized by finding optimal C values for each SVM. It is computational expensive to determine optimal C parameter for all SVMs on a specific feature set. For this study, we set parameter C to its default value. Testing on a small subset of AUs showed best results in a range of 0.1 to 10 for C which

is in the range of the default value. The default value performs well on several AUs and has minor less performance to the best values we found by a manual search. The kernel parameters are kept unchanged.

Figure 4.3 shows the AU based recognition performance in percentage for each AU. Each column of the bar-chart shows the accuracy of the above three kernels. Each SVM is trained on mean values of 20x20 block segmentation resulting in 400 features for a training sample.

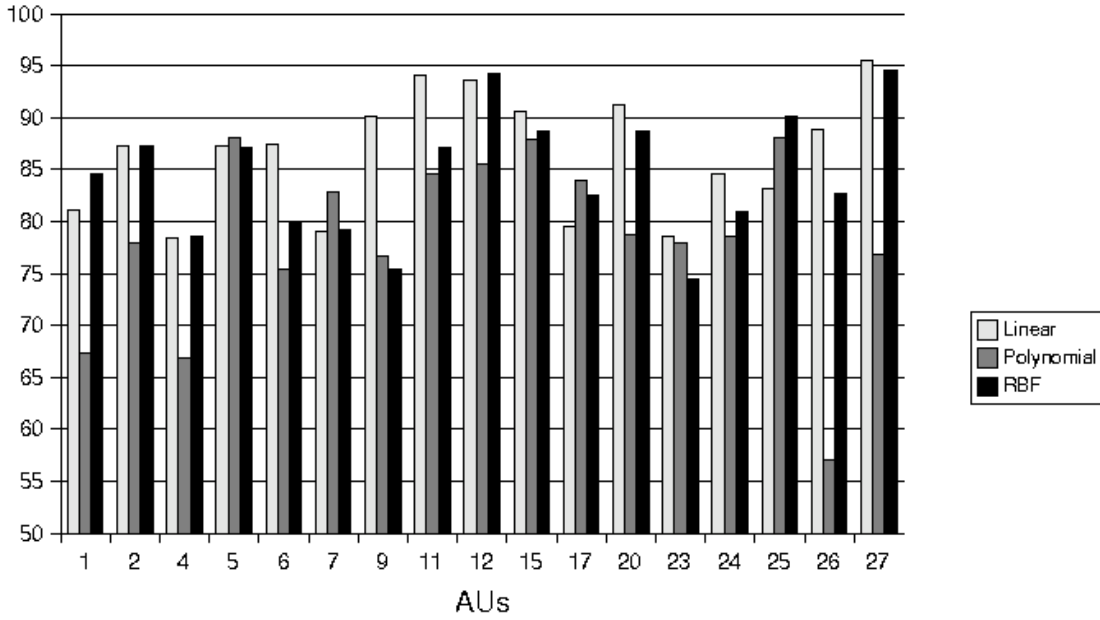


Figure 4.3: *AU recognition rate using different kernels*

The overall average AU recognition performance and the overall positive and negative average class performances are depicted in Figure 4.3. It turned out that the linear ker-

Kernels	ϕ overall	ϕ positive	ϕ negative
Linear	87	67	87
Polynomial	78	66	80
RBF	84	78	85

Table 4.1: *Average recognition rates for AUs using different kernels*

nel performed best in minimizing the total number of errors. The RBF kernel achieved

the best positive recognition rate with 78% and a slightly decreased AU rejection rate. The polynomial kernel has problems on some AUs, i.e. AU25. The total average recognition performance of linear and RBF kernels are in close bounds for most AUs with a covariance of 7%, where the RBF kernel is more stable and outperforms the linear kernel at some AUs (1,12,25). The RBF kernel gives a better balanced recognition performance for the detection and rejection of an AU.

The performance differences of the three kernels suggest to use a kernel for an AU which showed best performance instead of a kernel which performed best on average. The average recognition performance will be improved by choosing the best kernel for each AU.

4.2.6 Influence of Different Features

We compare the following pixel based features:

- mean values
- variance values
- mean + variance values

We fixed an RBF kernel and 20x20 blocks. Figure 4.4 depicts the results of those experiments.

We can observe in Figure 4.2 that variance values give best average overall recognition performance. The average recognition rates for the positive and negative classes are listed separately on the right columns.

Features	ϕ overall	ϕ positive	ϕ negative
Mean	84	78	85
Variance	91	39	95
Mean + Variance	90	50	93

Table 4.2: *Average recognition rates for AUs using different features*

The good average accuracy of variance and mean + variance result from a high recognition rate on the negative class and low recognition rate on the positive class. Due

to the unbalanced class probabilities, using variance minimizes the total number of errors better than mean alone. But a positive AU detection rate of 39% and 50% for adding variance features respectively, is poor. Mean values alone give a better equal error rate. The poor positive recognition rate for variance features shows that variance values cannot be used as the only features. Only in combination with mean values, the results get better. Mean and variance together do not perform better than mean values alone because of the double amount of features used. Our result shows the problem of 'overfitting' if we simply add all 400 variance features. More training samples are needed in order to get better results for mean + variance features.

4.2.7 Influence of Different Numbers of Blocks

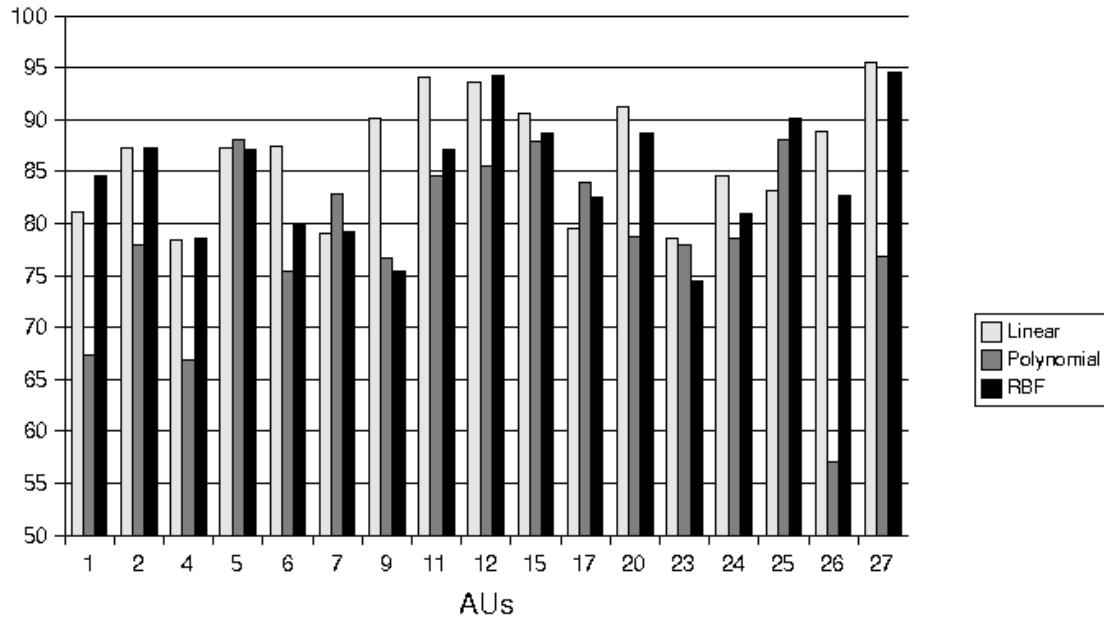
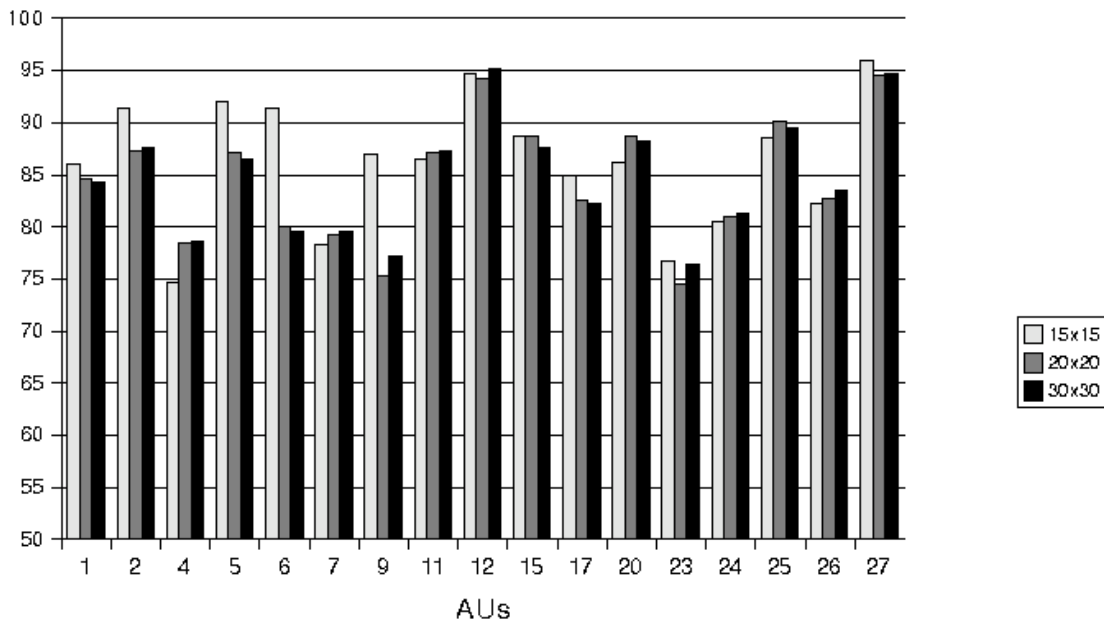
We experimented with block sizes, segmenting the face image into 15x15, 20x20 and 30x30 blocks resulting in 225, 400 and 900 features, respectively. We used mean values and the RBF kernel to obtain the results as they gave the best performance. Figure 4.5 depicts the results for all AUs using the 3 different block segmentations.

We can observe in figure 4.3 that using 15x15 blocks give best average recognition performance. The average recognition rates for the positive and negative classes are listed separately on the right columns.

Blocks	ϕ overall	ϕ positive	ϕ negative
15x15	86	77	87
20x20	84	78	85
30x30	85	78	86

Table 4.3: *Average recognition rates for AUs using numbers of blocks*

Figure 4.5 shows that the AU recognition rates for 15x15 blocks outperform those for 20x20 and 30x30 on some AUs (5,6,9,27). Results do not improve significantly using 20x20 and 30x30 blocks. The average recognition rate of 15x15 blocks shows that 225 features already contain enough information to recognize AUs. Adding more detail by reducing the block sizes increases average positive recognition rates.

Figure 4.4: *AU recognition rates using different features*Figure 4.5: *AU recognition rates using different numbers of blocks*

4.3 Evaluation and Results

Our experiments show best overall recognition performance for variance features, a linear kernel and image resolution of 30x30 blocks. But the results for the linear kernel and variance features show a lack of performance for detection of an AU. The results using an RBF kernel and mean values provide better performance for AU detection.

Due to uneven prior probabilities in training and classification of an AU, the linear kernel and variance values provide low false alarm rates. These are not optimal for AU recognition. The distribution of AU occurrences is sparse and facial expressions occur seldom with a high amount of AU activations. Facial expressions in our version of the CMU database have on average 1 to 3 AUs activated. A suboptimal classifier can achieve good performance by guessing permanently the negative, not activated AU class. As can be seen from the tables, this is approximately the case for variance features, linear and polynomial kernel. They have low false alarm and high misclassification rates.

We believe that a balanced positive and negative class accuracy is more expressive to state the ability of the system to recognize AUs. A better AU detection rate provides evidence that the SVM could separate both classes and generalize well to unseen positive and negative instances.

Therefore, adding variances or using variance features alone is not optimal to represent faces. Mean grey-level values provide better generalization and we assume they are sufficient to represent AU holistically.

Changing the image resolution by reducing or widening the block size has little effect on the recognition performance. All three resolutions have minor performance differences. We expect that, by reducing the image resolution further, the accuracy decreases proportional. 30x30 pixels seem to perform best and we expect no further improvements with bigger resolutions.

Chapter 5

Conclusions and Directions for Future Research

5.1 Conclusion

The results in this thesis demonstrate that facial activity can be recognized with low-level features. Rather than trying to recognize specific prototypical emotional facial expressions like joy, anger, surprise and fear, this system recognizes a set of 6 upper and 12 lower action units as defined in the Facial Action Coding System (FACS) [12].

Our system achieved 86% overall average accuracy with an AU detection rate of 77% using SVMs. We used a 15x15 pixel image resolution and grey-level mean values as features. The RBF kernel provides balanced generalization on AU and non AU occurrences.

The overall performance is comparable to recognition systems introduced in Section 2.6.1. The work of [16] shows that using high level feature extraction is vital to increase performance to a level above 95% and further improvements must be done to reach that accuracy. SVMs demonstrate promising performance which enables our system to compare with related recognition systems using improved feature extraction techniques (PCA, ICA). We believe that SVMs in conjunction with low-level feature extraction can be used for real-time recognition.

5.2 Future Work

Our AU recognition system is currently based on simple low-level feature extraction. We want to investigate different feature representation methods based on low-level features to improve recognition results. Preliminary experiments with PCA on the eyebrows (AU1+2) showed that performance increases of 5-7% are possible. We thus want to further investigate PCA to represent the low-level features. Promising PCA variations are kernel-PCA and kernel-LDA. Both methods perform PCA in the high dimensional feature space related to some input space by a non-linear SVM kernel. Experiments are reported in [24] show better results on kernel-PCA than PCA in the input space.

Finally, we want to investigate real-time high-level feature extraction that provides the necessary robustness and accuracy needed to operate in a natural scenario. Saccadic Search [25] is a promising method to locate facial features in real-time. Tracking and feature-extraction based on the localization results have to be investigated. Details of transient and permanent facial features enables us to improve detection of subtle changes in the face. But still, holistic methods based on grey or color values are good at representing prevalent facial expressions. We believe that low-level features and higher-level feature extraction combined provide better results than using one alone.

Bibliography

- [1] M. Bartlett, *Face Image Analysis by Unsupervised Learning and Redundancy Reduction*. Phd thesis, University of California, San Diego, 1998. Available as citeseer.nj.nec.com/bartlett98face.html.
- [2] M. Bartlett, J. Hager, P. Ekman, and T. Sejnowski, "Measuring facial expressions by computer image analysis," *Psychophysiology*, 36:253–263, 1999. Available as citeseer.nj.nec.com/bartlett99measuring.html.
- [3] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998. Also available as citeseer.nj.nec.com/burges98tutorial.html.
- [4] U. Canzler, "Gesture and mimic interpretation for sign language recognition," in *Book of Extended Abstracts pp. IC1*, (Prag), Proceedings of the 4th International Student Conference on Electrical Engineering POSTER 2000, May 25. Also available as <http://www.techinfo.rwth-aachen.de/cgi-bin/veroeffentl-db.pl?SLR>.
- [5] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [6] H. Chew, R. Bogner, and C. Lim, "Target detection in radar imagery using support vector machines with training size biasing," 2000. Available as www.kernel-machines.org/papers/upload_11483_ICARCV2000-4.ps.
- [7] I. Cohen, A. Garg, and T. Huang, "Emotion recognition from facial expressions using multilevel hmm," In *Neural Information Processing Systems*, 2000. Also available as citeseer.nj.nec.com/cohen00emotion.html.

- [8] J. Cohn, A. Zlochow, J. Lien, and T. Kanade, "Automated face analysis by feature point tracking has high concurrent validity with manual facs coding," 1999. Also available as citeseer.nj.nec.com/cohn99automated.html.
- [9] M. D. Department, "Emotional expression recognition using support vector machines." Available as citeseer.nj.nec.com/448808.html.
- [10] G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski, "Classifying facial actions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 974–989, 1999. Also available as citeseer.nj.nec.com/donato99classifying.html.
- [11] P. Eisert and B. Girod, "Analyzing facial expressions for virtual conferencing," *IEEE Computer Graphics and Applications*, vol. 18, no. 5, pp. 70–78, /1998. Also available as citeseer.nj.nec.com/eisert98analyzing.html.
- [12] P. Ekman and W. Friesen, *Manual for the Facial Action Coding System*. Consulting Psychologists Press, 1977.
- [13] B. Fasel and J. Luettin, "Recognition of asymmetric facial action unit activities and intensities," pp. 1100–1103, Proceedings of ICPR 2000, Barcelone, Spain, December 1999. Also available as citeseer.nj.nec.com/fasel99recognition.html.
- [14] T. Kanade, J. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis.," Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG'00), 2000. Also available as http://vasc.ri.cmu.edu/idb/html/face/facial_expression.
- [15] A. Kapoor, "Automatic facial action analysis," Master's thesis, School of Architecture and Planning, MIT, June 2002. Available as citeseer.nj.nec.com/kapoor02automatic.html.
- [16] Y. li Tian, T. Kanade, and J. F. Cohn, "Recognizing action units for facial expression analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 97–115, 2001. Also available as citeseer.nj.nec.com/article/tian01recognizing.html.

- [17] J. Lien, T. Kanade, J. Cohn, and C. Li, "Detection, tracking and classification of actions units in facial expressions." Available as citeseer.nj.nec.com/lien99detection.html.
- [18] C. L. Lisetti and D. E. Rumelhart, "Facial expression recognition using a neural network," 11th International Flairs Conference, AAAI Press, 1998. Also available as citeseer.nj.nec.com/lisetti98facial.html.
- [19] C. W. Omlin and M. M. Glaser, "Real-time sign language translation," Current research programme available at <http://www.cs.uwc.ac.za/~comlin/sasl.html>.
- [20] E. Osuna, R. Freund, and F. Girosi, "Support vector machines: Training and applications," Tech. Rep. AIM-1602, 1997. Also available as citeseer.nj.nec.com/osuna97support.html.
- [21] J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," tech. rep., rep. 98-14, Microsoft Research, Redmond, April 1998. Available as <http://www.research.microsoft.com/~jplatt/smo.html>.
- [22] J. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin dags for multiclass classification," *Advances in Neural Information Processing Systems*, 12 ed., S.A. Solla, T.K. Leen and K.-R. Muller, MIT Press, 2000. Also available as citeseer.nj.nec.com/platt00large.html.
- [23] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23–38, 1998. Also available as citeseer.nj.nec.com/rowley96neural.html.
- [24] B. Schoelkopf, A. Smola, and K.-R. Mueller, "Kernel principal component analysis," in *Advances in Kernel Methods - Support Vector Learning*, pp. 327–352, Cambridge, MA: MIT Press, 1999.
- [25] F. Smeraldi, N. Capdevielle, and J. Bigun, "Facial features detection by saccadic exploration of the Gabor decomposition and Support Vector Machines," in *Proceedings of the 11th Scandinavian Conference on Image Analysis - SCIA 99, Kangerlussuaq, Greenland*, vol. I, pp. 39–44, 1999. Also available as citeseer.nj.nec.com/article/smeraldi99facial.html.
- [26] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer, N.Y., 1995.