Automatic Real-Time Facial Expression Recognition for Signed Language Translation

Jacob Richard Whitehill

A thesis submitted in partial fulfillment of the requirements for the degree of Magister Scientiae in the Department of Computer Science, University of the Western Cape.

May 2006

UNIVERSITY of the
WESTERN CAPE

http://etd.uwc.ac.za/

Keywords

Machine learning

Facial expression recognition

Sign language

Facial action units

Segmentation

Support vector machines

Boosting

Adaboost

Haar

Gabor

UNIVERSITY of the
WESTERN CAPE

http://etd.uwc.ac.za/

Abstract

Automatic Real-Time Facial Expression Recognition for Signed Language Translation

Jacob Richard Whitehill
M.Sc. thesis, Department of Computer Science, University of the Western Cape

We investigated two computer vision techniques designed to increase both the recognition accuracy and computational efficiency of automatic facial expression recognition. In particular, we compared a local segmentation of the face around the mouth, eyes, and brows to a global segmentation of the whole face. Our results indicated that, surprisingly, classifying features from the whole face yields greater accuracy despite the additional noise that the global data may contain. We attribute this in part to correlation effects within the Cohn-Kanade database. We also developed a system for detecting FACS action units based on Haar features and the Adaboost boosting algorithm. This method achieves equally high recognition accuracy for certain AUs but operates two orders of magnitude more quickly than the Gabor+SVM approach. Finally, we developed a software prototype of a real-time, automatic signed language recognition system using FACS as an intermediary framework.

22 May 2006

UNIVERSITY of the
WESTERN CAPE

http://etd.uwc.ac.za/

Declaration

I declare that *Automatic Real-time Facial Expression Recognition for Signed Language Translation* is my own work, that it has not been submitted for any degree or examination in any other university, and that all the sources I have used or quoted have been indicated and acknowledged by complete references.

Jacob Whitehill 22 May 2006

Signed:

http://etd.uwc.ac.za/

Foreword and Acknowledgment

Conducting this research at the University of the Western Cape (UWC) was a challenging and demanding experience, especially because of the limited material resources that UWC possesses and the small research staff that it hosts. It was exactly through overcoming these challenges, however, that I matured as an aspiring scientist while writing my MSc thesis. As my adviser so often reminds his students, this is *my* thesis, and any problems that arose during its completion were mine alone to solve. Learning to convert my moments of confusion into well-posed questions, and learning where to begin searching for answers to these questions, are lessons even more valuable than the considerable knowledge of automatic facial expression recognition I have amassed.

During this learning process I was aided by several people whom I would like to thank. First, Mr. David Petro of the Bastion Center for the Deaf in Cape Town generously volunteered his time and native knowledge of South African Sign Language. Without his help, the pilot study on SASL recognition in this thesis would not have been possible. The three examiners of this thesis provided useful feedback on improving the thesis presentation as well as several useful references on support vector machines (SVMs). Mr. Steve Kroon from the University of Stellenbosch kindly answered numerous questions on SVMs and statistics. Professor Marian Stewart Bartlett of the Machine Perception Laboratory (MPLab) at the University of California at San Diego gave me detailed and insightful feedback on my analysis of local versus global face analysis. To Dr. Gwen Littlewort, also of the MPLab, I express my particular gratitude for her generous, patient, encouraging, and helpful responses to my many email queries about Gabor filters, Adaboost, and FACS AU recognition. Finally, I thank my research adviser, Professor Christian W. Omlin, now at the University of the South Pacific in Fiji, for his faith in me as a researcher, his encouragement at times of frustration, his enthusiasm, and his high-level wisdom on this challenging research project.

This research was partially funded by the Telkom/Cisco Centre for Excellence for IP and Internet Computing at the University of the Western Cape.

http://etd.uwc.ac.za/

Contents

1 Introduction 2

1.1 Thesis Objectives . . . 3

1.2 Outline . . . 3

2 Facial Action Coding System 5

2.1 Purpose of FACS . . . 5

2.2 The Design of FACS . . . 6

2.2.1 AU Combinations . . . 6

2.2.2 AU Intensity . . . 7

2.3 Suitability of FACS for Sign Language Recognition . . . 7

2.4 Alternative Systems for Facial Expression Description . . . 7

2.5 Why Use FACS for SASL? . . . 8

2.6 Summary . . . 8

3 Literature Review 9

3.1 Comparing the Accuracy of FER Systems . . . 9

3.2 Local versus Global Segmentation . . . 10

3.3 Feature Extraction for FER: The Two Approaches . . . 11

3.4 Geometry-based Features . . . 11

3.4.1 Locations and Relative Distances . . . 12

3.4.2 Parameter Estimation . . . 13

3.4.3 Models of Face Musculature . . . 14

3.4.4 Dimensionality Reduction . . . 14

3.5 Appearance-based Features . . . 15

3.5.1 Optical Flow . . . 15

3.5.2 Pixel Intensity Values . . . 16

3.5.3 Dimensionality Reduction in Appearance-Based Systems . . . 17

UNIVERSITY of the WESTERN CAPE

http://etd.uwc.ac.za/

3.5.4 Gabor Filters . . . 18
3.5.5 Haar Wavelets . . . 21
3.6 Comparing the Two Approaches . . . 24
3.7 Combining Geometric and Appearance-based Features . . . 25
3.8 Conclusions . . . 26
3.9 Summary . . . 26

4 Support Vector Machines **27**
4.1 Premise . . . 27
4.2 Training Phase . . . 28
4.2.1 The Lagrangian Method and the Wolfe Dual Form . . . 29
4.2.2 Determining b . . . 31
4.3 Test Phase . . . 31
4.4 Linear Inseparability . . . 32
4.5 Non-linear Decision Surfaces . . . 33
4.5.1 Kernel Functions and Mercer's Condition . . . 35
4.6 Polychotomous Classification . . . 35
4.7 Summary . . . 36

5 Experimental Results **37**
5.1 Preliminary Parameters and Techniques . . . 37
5.1.1 Facial Expression Database . . . 37
5.1.2 Image Normalization . . . 38
5.1.3 AU Classification . . . 38
5.1.4 Metric of Accuracy . . . 38
5.1.5 Cross Validation . . . 39
5.2 Local versus Global Face Segmentation . . . 39
5.2.1 Feature Extraction . . . 39
5.2.2 Segmentations . . . 39
5.2.3 Results . . . 40
5.2.4 Discussion . . . 40
5.3 Haar Features and Adaboost for AU Recognition . . . 43
5.3.1 Feature Selection . . . 43
5.3.2 Face Region Segmentation . . . 44
5.3.3 Feature Extraction . . . 44

UNIVERSITY of the WESTERN CAPE
http://etd.uwc.ac.za/

5.3.4 Classification . . . 44
5.3.5 Results . . . 45
5.3.6 Theoretical Performance Analysis . . . 45
5.3.7 Empirical Performance Analysis . . . 47
5.4 Summary . . . 48

6 Real-Time SASL Video Analysis **49**
6.1 Uses of Facial Expressions in Signed Languages . . . 49
6.1.1 Lexical Functionality . . . 50
6.1.2 Adverbial Functionality . . . 50
6.1.3 Syntactic Functionality . . . 50
6.2 Expression Intensity . . . 51
6.3 Implications for Automatic Translation . . . 52
6.4 Recognizing Facial Expressions of SASL . . . 52
6.4.1 Test Case: A Simple Story . . . 54
6.5 Approach . . . 55
6.5.1 Method 1: Exact Matching . . . 57
6.5.2 Method 2: Cosine Similarity . . . 57
6.6 System Design . . . 57
6.7 Experiment . . . 58
6.8 Results . . . 59
6.9 Discussion . . . 59
6.10 Summary and Conclusions . . . 62

7 Conclusions and Directions for Further Research **64**
7.0.1 Facial Expression Recognition . . . 65
7.0.2 Automatic Signed Language Recognition . . . 65

A Mathematical Fundamentals and Computer Vision Algorithms **66**
A.1 Distance between a hyperplane H and the origin . . . 66
A.2 Time Complexity of 2-D FFT . . . 66
A.3 Principle Component Analysis . . . 67
A.4 Optic Flow Analysis . . . 68
A.5 Haar Wavelets . . . 69
A.5.1 One-dimensional Haar Wavelet Decomposition . . . 69
A.5.2 Two-dimensional Haar Wavelet Decomposition . . . 70

UNIVERSITY of the WESTERN CAPE
http://etd.uwc.ac.za/

B Representative ROC Curves **71**

B.1 Local Gabor+SVM 71

B.2 Global Gabor+SVM 73

B.3 Local Haar+Adaboost 75

UNIVERSITY of the
WESTERN CAPE

http://etd.uwc.ac.za/

Chapter 1

Introduction

In human-to-human dialogue, the articulation and perception of facial expressions form a communication channel that is supplementary to voice and that carries crucial information about the mental, emotional, and even physical states of the conversation partners. In their simplest form, facial expressions can indicate whether a person is happy or angry. More subtly, expressions can provide either conscious or subconscious feedback from listener to speaker to indicate understanding of, empathy for, or even skepticism toward what the speaker is saying. Recent research has shown that certain facial expressions may also reveal whether an interrogated subject is attempting to deceive her interviewer [Ekm01].

One of the lesser known uses of facial expression in human interaction is signed communication, i.e., "sign language." In signed languages, facial expressions are used to denote the basic emotions such as "happy" and "sad". Even more importantly, however, they also provide lexical, adverbial, and syntactic information. In some instances, a signer may use a facial expression to strengthen or emphasize an adverb which is also gestured through the hands. In others, the facial expression may serve to differentiate two nouns from each other. Any computer system designed to recognize a signed language must thus be able to recognize the facial expressions both accurately and efficiently.

Throughout the world, but especially in developing countries such as South Africa, deaf people face severely limited educational and occupational opportunities relative to a hearing person. The existence of a computer system that could automatically translate from a signed language to a spoken language and vice-versa would be of great benefit to the deaf community and could help to alleviate this inequality. In the South African Sign Language Project at the University at the Western Cape, of which this research is a part, we envision the development of a small, unobtrusive, hand-held computing device that will facilitate the translation between signed and spoken languages. This computer system will need to recognize both hand gestures and facial expressions simultaneously; it must then analyze these two channels linguistically to determine the intended meaning; and it will need to output the same content in the target language.

http://etd.uwc.ac.za/

All three stages must operate in real-time. In this thesis are interested in the facial expression recognition aspects of this translation device. We believe that the Facial Action Coding System (FACS, by Ekman and Friesen[EF78]), a well-known framework which objectively describes human facial expressions in terms of facial "action units", will serve as a useful intermediary representation for SASL expression recognition. In the section below, we describe our particular thesis goals.

1.1 Thesis Objectives

The goals of this thesis are two-fold:

- First, we wish to construct an automatic FACS action unit recognition system that supports the automated recognition and translation of South African Sign Language (SASL). Automatic FACS action unit recognition is useful in its own right and has numerous applications in psychological research and human-computer interaction.
- Second, using the action unit recognition system that we build, we will construct a software prototype for the recognition of facial expressions that occur frequently in SASL and evaluate this prototype on real SASL video.

Automatic facial expression recognition (FER) takes place during three phases: (1) image preprocessing, face localization and segmentation; (2) feature extraction; and (3) expression classification. This thesis investigates techniques across all three stages with the goal of increasing both accuracy and speed. In our first main experiment, we investigate the effect of local segmentation around facial features (e.g., mouth, eyes, and brows) on recognition accuracy. In our second experiment, we assess the suitability of using Haar features combined with the Adaboost boosting algorithm for FACS action unit recognition. We conduct both experiments using the Cohn-Kanade database [KCIT00] as our dataset, and using the area under the Receiver Operator Characteristics (ROC) curve, also know as the A' statistic, as the metric of accuracy. For statistical significance, we use matched-pairs, two-tailed t-tests across ten cross-validation folds.

1.2 Outline

The rest of this thesis is constructed as follows: in Chapter 2 we describe the Facial Action Coding System and motivate our decision to use this framework. In Chapter 3 we conduct a wide-ranging survey of historical and contemporary FER systems in order to discover which techniques and algorithms already exist. We place particular emphasis on the feature types that each surveyed FER system uses. Chapter 4 provides a derivation of the support vector machine (SVM) due to its importance in the FER literature. In Chapter 5 we assess whether local analysis of the face around particular features such as the mouth and

http://etd.uwc.ac.za/

eyes can improve recognition accuracy as well as increase run-time performance. We use support vector machines and Gabor features for this study. The results of this experiment underline the importance of establishing a large, publicly available facial expression database in which individual facial actions occur independently of others. Later in Chapter 5 we depart from the Gabor+SVM approach in order to test a new method of detecting FACS AUs: Haar wavelet-like features classified by an Adaboost strong classifier. Our results show that this new technique achieves the same recognition accuracy for certain AUs but operates two orders of magnitude more quickly than the Gabor+SVM method.

In Chapter 6 we use FACS as an intermediary expression coding framework and apply the FER system developed in Chapter 5 to our target application domain of SASL recognition. While the actual recognition results of this pilot study are unsatisfactory, we believe that the system architecture as well as the particular problems we encountered will be useful when designing future such systems. Finally, Chapter 7 suggests directions for future research.

With regards to the pilot project on signed language recognition we make one disclaimer: This thesis does *not* constitute *linguistic* research on South African Sign Language or signed communication in general. The purpose of this pilot application is to assess whether a simple object recognition architecture can support viable automatic signed language recognition, and to discover the most pressing problems that need to be solved in support of this goal. By implementing a software prototype of a SASL expression recognizer, we also provide future researchers of the South African Sign Language Project a firm starting point from which to conduct further research.

UNIVERSITY of the
WESTERN CAPE

http://etd.uwc.ac.za/

Chapter 2

Facial Action Coding System

In this thesis we use the Facial Action Coding System (FACS) [EF78] as an intermediary framework for recognizing the facial expressions of South African Sign Language (SASL). Two other research groups also use a FACS-based approach for their signed language recognition systems: the group of Professors Ronnie Wilbur and Aleix Martinez at Purdue University [Wil], and Ulrich Canzler [Can02] at the RWTH-Aachen. In order to motivate our own decision to use FACS, we must first describe the purpose and design of FACS and compare it to other representations that describe human facial expression. Later in this chapter we discuss the advantages and disadvantages of using FACS for our end-goal of automated SASL recognition.

2.1 Purpose of FACS

The primary goal of FACS was "to develop a comprehensive system which could distinguish all possible visually distinguishable facial movements" ([EFH02], p. 2). In contrast to other systems for facial expression coding, the development of FACS was governed by "the need to separate inference from description." In other words, the investigation of which emotion caused a particular facial expression should be determined independently from the description of the facial expression itself.

FACS is based on an eight-year, highly-detailed anatomical study of the muscles which control the face. It was designed to measure every *visible* movement of the face due to the contraction of facial muscles. In contrast to certain intrusive methods such as electromyography, in which wires must be connected to subjects' faces, FACS was designed for use on humans who are perhaps unaware of the fact they are being studied; coding of facial expression is therefore performed using only visual measurements. For this reason, FACS is not intended to measure muscle movements which result in no appearance change or whose effect on the face is too subtle for reliable human perception. FACS also does not register changes in facial appearance due to factors unrelated to muscles, e.g., blushing or sweating [EFH02].

UNIVERSITY of the WESTERN CAPE
http://etd.uwc.ac.za/

2.2 The Design of FACS

FACS' approach is to specify the minimal units of facial behavior. These units are known as *action units* (AUs). Some AUs have a one-to-one correspondence with a particular facial muscle. AU 13, for example, corresponds solely to the *caninus* muscle. Other AUs may be generated by any one of a set of face muscles whose effects on the face are indistinguishable from each other. In yet other cases, multiple AUs may be linked to the same muscle if different parts of that muscle can be activated independently. Both AUs 7 and 8, for example, pertain to *orbicularis oris* [EFH02].

Each AU is assigned a number to facilitate coding of faces. In the original FACS definition in 1978 [EF78], there were 44 AUs whose numbers ranged from 1 through 46 (numbers 3 and 40 are not used). The updated 2002 edition [EFH02], which incorporated movements of the eyeball and head, contains an additional 12 AUs numbered 51 and higher. In both editions, AUs 1 through 7 pertain to the upper-face actions whereas AUs numbered 8 through 46 relate to the lower face.

For each AU in FACS, the *FACS Manual* [EFH02] provides the following information:

- The muscular basis for the AU, both in words and in illustrations.
- A detailed description of facial appearance changes supplemented by photographs and film examples.
- Instructions on how to perform the AU on one's own face.
- Criteria to assess the intensity of the AU.

UNIVERSITY of the WESTERN CAPE

2.2.1 AU Combinations

As AUs represent the "atoms" of facial expressions, multiple AUs often occur simultaneously. Over 7000 such combinations have been observed [Ekm82]. Most such combinations are *additive*, meaning that the appearance of each AU in the combination is identical to its appearance when it occurs alone. Some combinations, however, are *distinctive* (sometimes also called *non-additive*) - in such cases, some evidence of each AU is present, but new appearance changes due to the joint presence of the AUs arise as well. In the *FACS Manual*, the distinctive AUs are described in the same detail as the individual AUs.

Further relationships among multiple AUs exist as well. For instance, in certain AU combinations, the *dominant* AU may completely mask the presence of another, *subordinate* action unit. For certain such combinations, special rules have been added to FACS so that the subordinate AU is not scored at all.[1] Another relationship among AUs is that of *substitutive* combinations. In these cases, one particular AU

[1] Most such rules were removed in 1992 after it had been determined that they they were mostly confusing.

http://etd.uwc.ac.za/

combination cannot be distinguished from another, and it is up to the FACS coder to decide which is more appropriate.

2.2.2 AU Intensity

In addition to determining which AUs are contained within the face, the intensity of each AU present must also be ascertained. Intensity is rated on a scale from A (least intense) through E (most intense). Criteria for each intensity level are given in the *FACS Manual* for each AU.

2.3 Suitability of FACS for Sign Language Recognition

In this project we chose FACS as our intermediary framework for facial expression recognition because of the level of detail it provides in describing expressions; because of its ability to code expression intensity; and because FACS is a standard in the psychology community. As we will describe in Chapter 6, we conducted a preliminary FACS analysis of 22 facial expressions that occur within SASL and determined that no pair of facial expressions contained exactly the same set of AUs. Although this study will have to be extended over more subjects and more expressions, it does support our belief that FACS is sufficiently detailed to enable sign language recognition.

2.4 Alternative Systems for Facial Expression Description

We are aware of only a few other systems designed to describe facial expressions in detail. One such system is the *Maximally Discriminative Facial Movement Coding System* (MAX), which was developed by C.E. Izard in 1979 [Iza79] and later updated in 1995. MAX was developed for psychological research on infants and small children, though with modification it can also be applied to persons of other age groups. Face analysis under MAX is performed using slow-motion video and proceeds in two stages. In the first stage, the face is divided into three regions: (1) the brows, forehead, and nasal root; (2) the eyes, nose, and cheeks; and (3) the lips and mouth. Each region is then analyzed independently for the occurrence of facial movements known as *appearance changes* (ACs). In the second stage, the ACs in each face region are classified either as one of eight distinct emotional states (interest, joy, surprise, sadness, anger, disgust, contempt, and fear), or as a complex expression comprising multiple simultaneous affects [Iza79]. Like FACS AUs, the MAX ACs are rooted anatomically in the muscles of the face. Unlike AUs, however, the set of ACs is not comprehensive of the full range of visually distinct human facial movement, nor does it distinguish among certain anatomically distinct movements (e.g., inner- and outer-brow movement) [OHN92]. MAX is therefore less appealing for signed language translation than FACS.

http://etd.uwc.ac.za/

Another approach is the Moving Pictures Expert Group Synthetic/Natural Hybrid Coding (MPEG-4 SNHC) [Mov] standard. MPEG-4 SNHC uses 68 *facial animation parameters* (FAPs) to describe movements of the face. The purpose of MPEG-4 SNHC, however, is to animate computer-generated graphics, not to recognize the expression on an actual human's face. Correspondingly, the set of FAPs is not comprehensive of all visible human face movement, nor do the individual FAPs correspond to the actual muscle groups of the human face. As with MAX, it is unlikely to be of use in sign language recognition.

2.5 Why Use FACS for SASL?

In this thesis we endeavor to build an automated system for the recognition of SASL facial expressions by first determining the set of AUs present in a particular face image, and then mapping these AUs to a particular SASL expression. While we have already explained the advantages of FACS over other expression recognition frameworks, we have not yet motivated why we need an intermediary framework at all.

Using an intermediary expression description framework does add an additional layer of complexity to a translation system that recognizes SASL expressions directly from the input images. However, the advantage of using a framework for expression description such as FACS is that linguistic research on SASL and machine learning research on expression recognition can be de-coupled. For example, if a new expression is discovered in SASL, it can be accommodated simply by adding an additional AU-to-expression mapping to the translation system. The AU recognition code, on the other hand, remains completely unchanged. In systems that are trained on individual SASL expression directly, on the other hand, a whole new set of training examples containing this newly-found expression must be collected, and a new classifier must be trained - this requires significant time and effort. We thus believe that the use of an intermediary framework, especially FACS, is a worthwhile component of our system design.

UNIVERSITY of the
WESTERN CAPE

2.6 Summary

We have described the purpose and basic architecture of FACS, including its set of action units and intensity ratings. We have explained some of the advantages of FACS over other expression coding systems for the task of signed language translation. Finally, we justified our use of an intermediary framework such as FACS in our SASL expression recognition system.

http://etd.uwc.ac.za/

Chapter 3

Literature Review

Automatic facial expression recognition (FER) is a sub-area of face analysis research that is based heavily on methods of computer vision, machine learning, and image processing. Many efforts either to create novel or to improve existing FER systems are thus inspired by advances in these related fields.

Before describing our own contributions to the field of automatic FER, we first review the existing literature on this subject. This survey includes the major algorithms that have significantly impacted the development of FER systems. We also describe more obscure algorithms of FER both for the sake of comprehensiveness, and to highlight the subtle benefits achieved by these techniques that may not be offered by more mainstream methods. In accordance with the experiments we perform in Chapter 5, we place particular emphasis in our survey on the role of feature type, and on the effect of local versus global face segmentation on classification performance.

UNIVERSITY of the WESTERN CAPE

3.1 Comparing the Accuracy of FER Systems

Objectively comparing the recognition accuracy of one FER system to another is problematic. Some systems recognize prototypical expressions, whereas others output sets of FACS AUs. The databases on which FER systems are tested vary widely in number of images; image quality and resolution; lighting conditions; and in ethnicity, age, and gender of subjects. Most databases include subjects directly facing the camera under artificial laboratory conditions; a few (e.g., [KQP03]) represent more natural data sets in which head posture can vary freely. Given such vastly different test datasets used in the literature, only very crude comparisons in accuracy between different FER systems are possible. However, for the sake of completeness, we do quote the reported accuracy of the systems we reviewed.

The most common metric of recognition accuracy used in the literature is the percentage of images classified correctly. An accuracy of 85% would thus mean that, in 85 out of 100 images (on average), the

http://etd.uwc.ac.za/

expression was predicted correctly, and in 15 images it was not. This metric is natural for characterizing a face as belonging to one of a fixed set of k emotions. For FACS AU recognition, however, this metric can be highly misleading: some expressions occur so rarely in certain datasets that a classifier could trivially always output 0 ("absent") for the expression and still score high accuracy. In such a system, even though the hit rate (% of positively labelled images classified correctly) would be low (0%), the percentage of images correctly classifier would still be high. A more sophisticated measure of recognition accuracy is the area under the ROC curve, also called the A' statistic, which takes into account both the true positive and false positive rates of a classifier. We use the A' metric in our own experimental work in Chapter 5. Most previous literature on FER presents results only as percent-correct, however, and in this literature review we are thus constrained to do the same.

3.2 Local versus Global Segmentation

The first issue we investigate, both in this survey and in Chapter 5, is whether analyzing a local subregion of the face around particular facial muscles can yield a higher recognition accuracy of certain FACS AUs than analyzing the face as a whole. Little research has been conducted on this issue for prototypical expressions, and no study, to our knowledge, has assessed the comparative performance for FACS AUs. Results for prototypical expressions are mixed:

Lisetti and Rumelhart developed neural networks to classify faces as either smiling or neutral [LR98]. They compared two networks: one which was trained and tested on the whole face, and one which was applied only to the lower half of the face (containing the mouth). For their application, local analysis of the lower face-half outperformed the global, whole-face analysis.

Padgett and Cottrell compared global to local face analysis for the recognition of six prototypical emotions. In particular, they compared principle component analysis (PCA) on the whole face (*eigenfaces*) to PCA on localized windows around the eyes and mouth (*eigenfeatures*). The projections onto the eigenvectors from each analysis were submitted to neural networks for expression classification. As in Lisetti and Rumelhart's study, the localized recognition clearly outperformed global recognition. Padgett and Cottrell attribute these results both to an increased signal-to-noise ratio and to quicker network generalization due to fewer input parameters [PC97].

However, Littlewort, et al [LFBM02] compared whole-face, upper-half, and lower-half face segmentations for the recognition of prototypical facial expressions. They classified Gabor responses (described later in this chapter) using support vector machines. In contrast to the other literature on this subject, their whole-face segmentation clearly outperformed the other two segmentation strategies by several percentage points [LFBM02].

UNIVERSITY of the WESTERN CAPE

http://etd.uwc.ac.za/

From the literature, there seems to be no definite answer as to which segmentation - local or global - yields higher accuracy. As we shall show in Chapter 5, the issue depends on the particular facial expression database on which the system is tested. It may also depend on the particular *feature type* that is used. In the rest of this chapter, we describe the many kinds of features that have been deployed for FER as well as the systems that deploy them.

3.3 Feature Extraction for FER: The Two Approaches

Research on automatic FER can largely be divided into two categories: *appearance-based* and *geometry-based* methods. The former uses color information about the image pixels of the face to infer the facial expression, whereas the latter analyzes the geometric relationship between certain key points (*fiducial points*) on the face when making its decision. We describe geometry-based methods in Section 3.4 and appearance-based methods in Section 3.5.

3.4 Geometry-based Features

Many modern FER systems use the geometric positions of certain key facial points as well as these points' relative positions to each other as the input feature vector. We refer to such FER systems as *geometry-based* systems. The key facial points whose positions are localized are known as *fiducial points* of the face. Typically, these face locations are located along the eyes, eyebrows, and mouth; however, some FER systems use dozens of fiducial points distributed over the entire face.

The motivation for employing a geometry-based method is that facial expressions affect the relative position and size of various facial features, and that, by measuring the movement of certain facial points, the underlying facial expression can be determined. In order for geometric methods to be effective, the locations of these fiducial points must be determined precisely; in real-time systems, they must also be found quickly. Various methods exist which can locate the face and its parts, including optic flow, elastic graph matching, and Active Appearance Models ([CET98]). Some FER systems (e.g., [TKC01]) require manual localization of the facial features for the first frame in a video sequence; thereafter, these points can be tracked automatically. Other approaches to fiducial point location do not actually *track* the points at all, but instead re-locate them in each frame of the video sequence.

The exact type of feature vector that is extracted in a geometry-based FER systems depends on: (1) which points on the face are tracked; (2) whether 2-D or 3-D locations are used; and (3) the method of converting a set of feature positions into the final feature vector. The first question (1) has no definitive best answer, but it is influenced by several factors, including (a) how precisely each chosen fiducial point can be tracked; and (b) how sensitive is the position of a particular fiducial point to the activation of the

UNIVERSITY of the WESTERN CAPE

http://etd.uwc.ac.za/

classified facial expression. The advantage of 3-D fiducial point tracking is that the resulting FER systems are arguably more robust to out-of-plane head rotation than are 2-D systems. The disadvantage is that these 3-D locations must usually be re-constructed from 2-dimensional camera data; the algorithms used to track fiducial points are thus more complex and slower. Only a few FER systems (e.g., [GTGB02] and [EP97]) use 3-D coordinates.

In terms of feature extraction, the most distinguishing factor in the design of geometry-based FER system is how the set of facial location vectors is converted into features. The simplest kind of feature vector in such systems contains either the relative positions of different facial landmarks (e.g., distance between left and right eyes) or the (x, y) displacements of the same feature points between frames in a video sequence. In the former case, relative positions are often normalized by the face size to improve generalization performance across different human subjects. In the following subsections we review geometry-based FER systems based on their method of converting raw position vectors into features.

3.4.1 Locations and Relative Distances

The simplest type of geometry-based feature vector is constructed from the locations and relative distances between feature points. One such system using this approach was developed by Sako and Smith [SS96]. It used color histograms to track the head and mouth, and template matching to track the eyes and brows. Their system computes the width and height of the mouth and face as well as the distance between the eyes and eyebrows as a feature vector. Using the nearest neighbor classifier, their FER system classifies the face as one of five prototypical facial expressions. It operates in real time and achieves 71% accuracy [SS96] on a test set containing only one test subject.

Wang, Iwai, and Yachida [WIY98] use labeled graph matching to track the positions of 12 fiducial points. The (x, y) displacements of the points between adjacent video frames are collected into a feature vector. Each of the three classified prototypical expressions is modeled as a collection of 12 B-spline curves (one for each fiducial point) describing the movements of the fiducial points through time. By tracking the (x, y) displacement of all fiducial points of the test subject in each video frame, the facial expression can be classified by selecting the collection of B-spline whose combined Euclidean distance from the test data is minimized. Their system also estimates the degree of facial expression. On a test database of 29 image sequences recorded from four test subjects, their system achieves 100%, 100%, and 83.7% accuracy, respectively, on the prototypical expressions happiness, surprise, and anger [WIY98].

Lien, et al [LKCL98] employ optical flow to track 3 fiducial points each around the left and right eyebrows. The x and y displacements of these six points are computed relative to the neutral video frame to form the feature vector. HMMs are then used to classify one of three possible AU-based expressions of the eyebrows. On a test database of 260 image sequences from 60 subjects, their system achieved 85% accuracy

http://etd.uwc.ac.za/

[LKCL98].

Cohn, et al [CZLK99] use optical flow to track 37 fiducial points in the upper and lower face, and then apply discriminate function analyzes to classify the x and y displacement of each fiducial point into FACS AUs. Their system achieves 91%, 88%, and 81% accuracy on the brow, eye, and mouth AUs, respectively [CZLK99].

Finally, the FER system of Bourel, et al [BCL02] measures the distances between facial landmarks for its feature extraction and compares them to the corresponding values in previous frames. Their approach transforms the distances into one of three possible states: Increasing, Decreasing, or Stationary. Using the k-nearest neighbors algorithm for expression classification, they show that their state-based approach is more robust to noisy data and partial occlusion of the face than non-discretized approaches. Overall accuracy is around 90% for 6 prototypical emotions [BCL02].[1]

3.4.2 Parameter Estimation

In several geometry-based FER systems, fiducial point locations and distances do not constitute the features directly, but rather are used first to estimate the parameters of some model. These parameters are then fed to a classifier for expression prediction. One such FER system was developed by Black and Yacoob [BY95]: their approach uses a perspective projection model to convert the location vectors of facial landmarks into model parameters of image motion. These low-level model parameters are then further transformed into mid-level "predicates" which describe the movement of facial muscles in such terms as "mouth rightward". Finally, these predicates are classified as a facial expression using a manually created rule-set. The onset of an "anger" expression, for example, is defined as a simultaneous "inward lowering of brows and mouth contraction." On a database of 70 image sequences from 40 subjects, their system achieves an average of 92% recognition accuracy on 7 prototypical expressions [BY95].

Tian, Kanade, and Cohn [TKC01] use multi-state models of the head and face (one state for each head pose) as well as optical flow to track the locations of the eyes, brows, and cheeks. These location vectors are converted into sets of 15 upper-face and 9 lower-face parameters based on the relative distance between certain points. For instance, one such parameter describes the height of the eye and combines distance information from three fiducial points on the face from both the current and the initial video frames. Using a neural network, their system classifies 7 upper-face AUs and 11 lower-face AUs with 95% and 96.7% accuracy, respectively [TKC01].

In Cohen, et al [CSC+03], fiducial points all over the face are tracked using template matching. The locations of these points are fit onto a 3-D mesh model and then transformed into a set of Bezier-volume control parameters. These parameters represent the magnitudes of pre-defined facial motions. The Bezier

[1] No numerical results were given in the paper; we estimated 90% based on their graph.

http://etd.uwc.ac.za/

parameters are then discretized into bins before being classified as a prototypical expression. Best results in this FER system are achieved using the Tree-Augmented Naive (TAN) Bayes classifier with an average recognition rate of 65.1% [CSC+03].

3.4.3 Models of Face Musculature

One particular form of geometric model with a clear biological justification is to use fiducial point movement to estimate activation of the underlying face muscles. Mase was, to our knowledge, the first researcher to propose such a scheme for FER ([Mas91]), but according to his paper he did not actually implement this strategy. Essa and Pentland [EP97] did implement a complete FER system using this approach. They use optical flow analysis to track the eyes, nose, and lips. Based on the coordinates of these landmarks, a 3-D mesh model of the face is fit to every video frame. The mesh consists of many adjacent triangular shell elements, which are parametrized by mass, stiffness, and damping matrices in order to model the material properties of human skin. On top of this skin model, an anatomically-based dynamic model of muscle movement is applied using an estimation and control framework. Expressions are predicted using template matching in two different ways: by classifying the predicted underlying facial muscle movements, and by classifying the optic flow vectors of each grid point directly. Both method achieve 98% accuracy on prototypical expressions over a database of 52 video sequences.

3.4.4 Dimensionality Reduction

The last kind of geometric feature vectors that we consider are those formed by applying a dimensionality reduction to the original fiducial point location vectors. Dimensionality reduction methods such as PCA are very common in machine learning applications. They are most useful when the dimension of the input vectors is very high, such as with appearance-based FER systems (described later in this chapter). However, these methods also find use in geometry-based approaches to FER; we describe some systems that use dimensionality reduction below.

One straightforward but useful modification to geometry-based feature extraction algorithms is to apply principle component analysis (PCA) prior to classification. PCA is a method of transforming the input vector so that most of the variance of the original data is captured in the dimension-reduced output vector. A derivation of PCA is given in Section A.3.

Two of the purely geometric-based FER systems in our survey use this approach. Kimura and Yachida [KY97] use a "potential net" model to track 899 (29x31) locations on the face. These points do not correspond directly to facial landmarks but instead are distributed in a grid pattern centered at the nose. The potential net models the deformation of the face as a set of forces applied to springs. Each grid point is connected to its four closest grid neighbors. By requiring that the total force within the potential net sum to

http://etd.uwc.ac.za/

zero, the motion of each fiducial point can be calculated. Kimura and Yachida's system uses a Karhunen-Loève expansion (a generalization of PCA) to reduce the dimensionality of the final feature vector. One model vector for each of 3 prototypical emotions is estimated in the low-dimensional space. For classification, the input vector of grid point motions is projected onto the axes that were computed from the K-L expansion. The distances of this projection from each of the expression models and from the origin are used to estimate the type and degree of expression, respectively. No numeric results were listed in the paper, but test results when classifying expression of novel human subjects were described as "unsatisfactory" in the paper [KY97].

Gokturk, Bouguet, Tomasi, and Girod [GTGB02] track 14 fiducial points on the face in three dimensions using a cost minimization-based monocular tracking system. Given the initial position vectors of the fiducial points for each subject, their system can subtract away the rigid motion of the head to compute the deformation of the face due solely to facial expression. Their system then applies PCA to the non-rigid face motion vectors to compute facial motion along the principle movement axes. The final feature vector includes not only the principle components themselves, but also their first temporal derivative. Support vector machines are then used to classify 5 prototypical expressions. Accuracy results of a database of 235 frames from two subjects were reported as 91% over the 5 expressions [GTGB02].

3.5 Appearance-based Features

The second main approach to automatic FER is the appearance-based approach. As stated earlier, these are methods that classify facial expressions based on the color of the face pixels. Appearance-based algorithms are wide-ranging and include optic flow, dimensionality reduction techniques such as PCA and ICA, and image filters. We describe each type of method and the associated FER systems below.

3.5.1 Optical Flow

One of the earliest developed appearance-based methods of FER was *optic flow analysis*. Optic flow analysis endeavors to track object movement within an image by analyzing the change in pixel intensity of each image location (x, y) over multiple frames in a time-ordered sequence. The output of an optic flow computation for a particular image is a vector (v_x, v_y) for each pixel in the input image; v_x and v_y represent the magnitudes of the image velocities in the x and y directions, respectively. The $\mathbf{v} = (v_x, v_y)$ vectors over multiple pixel locations can be combined into feature vectors and then classified as a particular facial expression. Feature vectors based on optic flow can consist of the image velocities of certain fiducial points or of flow fields computed over entire image patches. We give a short derivation of optic flow analysis in Section A.4.

http://etd.uwc.ac.za/

One of the first FER systems to employ optic flow was developed by Mase [Mas91]. Mase proposed two alternative approaches: top-down and bottom-up. The top-down method attempts to recognize facial expressions by first using optic flow to recognize the individual muscle activations which formed the expression. In the bottom-up approach, the facial expression is recognized directly from the optic flow fields over a grid of $M \times N$ small image rectangles. Mase's system implements the bottom-up method and calculates the mean and variance of the optic flow within each rectangle along both the horizontal and vertical directions. The feature vector is computed by selecting the c features which maximize the ratio of between-class to within-class distance in the training set. This vector is then processed by a k nearest neighbors classifier. For prototypical expressions, Mase's system achieves recognition rates of approximately 80% [Mas91].

Later research in FER using optic flow was conducted by Yacoob and Davis in [YD96]. Their approach resembles Mase's proposed top-down model in that it attempts to determine the underlying muscle movements of the face in order to determine the expression. Given rectangular windows surrounding the mouth and eyebrows of each face image, optic flow fields are calculated along eight principle directions. Each window is then partitioned using free-sliding dividers, and the optic flow along each principle direction is calculated within each window partition. The dividers are adjusted so that the strength of the flow fields as well as the fields' homogeneity within each window region are jointly maximized. Final feature vectors are calculated as the optic flow projections at the optimal divider settings, and these vectors are then processed by rule-based classifiers for expression classification, similar to [BY95]. Their system achieves a recognition accuracy of 86%. [2]

UNIVERSITY of the WESTERN CAPE

3.5.2 Pixel Intensity Values

Whereas optical flow was perhaps the first appearance-based technique applied to FER, the simplest type of feature in appearance-based FER systems is the color of an individual pixel. Most FER systems process gray-scale images, and thus the pixel color can be renamed pixel *intensity*. A set of pixel values extracted at certain key points or over a whole can region can then be fed to a classifier to determine the facial expression.

Very few FER systems classify raw pixel intensity values directly without at least employing some form of feature selection. Those systems that do use simple pixel values as feature type have exhibited low recognition accuracies compared to other systems. Littlewort, et al [LFBM02], in a comparative study of different FER techniques, classified six prototypical facial expressions using pixel intensity values and SVMs. Their system achieves only around 73% accuracy when pixels are extracted from the whole face. Despite the low accuracy that has been reported, pixel intensity features do offer one important benefit - they can be

[2] Accuracy was reported as a confusion matrix; we computed the percent correct ourselves.

http://etd.uwc.ac.za/

extracted simply and quickly.

3.5.3 Dimensionality Reduction in Appearance-Based Systems

In appearance-based facial expression recognition systems, the fundamental unit of information is the pixel value, and features may be extracted from a pixel set by means of cropping, scaling, and filtering. Even at low resolution, the number of pixels in a face image is on the order of hundreds. Moreover, many of the pixels in this vector may contain little information that is useful for classification. It is possible, for example, that pixels located in certain regions of the face may not change from one facial expression to another, thus rendering useless the corresponding coordinate of the feature vector. Another possibility is that one pixel value in the feature vector might be completely dependent on other (perhaps neighboring) pixels. In both cases, the feature vector contains redundant information, and classification performance might improve by removing the superfluous components. Standard techniques such as PCA and ICA are often applied for this task; we describe the associated appearance-based FER systems below.

Principle Component Analysis

One popular method of reducing the dimension of feature vectors is *principle component analysis* (PCA). When PCA is applied to a dataset of dimension n, each vector in that dataset is projected onto $p << n$ *principle components*. Because of the way the components were calculated, the resultant set of projections still retain most of T's original variance, but the dimension of the resulting dataset is much smaller. We give a derivation of PCA in Section A.3.

Several appearance-based FER systems use PCA prior to expression classification. Both Donato, et al [DBH+99] and Bartlett, et al [BDM+00] classify 6 upper- and 6 lower- face AUs using PCA and the nearest neighbor algorithm. The first 30 principle components of the difference images of the relevant half-face (upper or lower) are extracted and classified for AU content. The systems achieve 79.3% average accuracy on 12 AUs. Fasel and Lüttin [FL00] performed a similar experiment to classify 9 individual AUs and 16 AU combinations, but on a different test database. As in [DBH+99], their system achieves 79% accuracy on single AUs, and it delivers 74% accuracy when tested on both single AUs and combinations [FL00].

Finally, Bartlett, et al [BHES99] classify 6 upper- and 6 lower-face AUs by extracting the first 50 principle components of difference images. Using a two-layer neural network their system achieves recognition rates of 88.6%.

Independent Component Analysis

In PCA, the projections of T along the principle components are uncorrelated, but they are not necessarily statistically independent. Hence, certain higher-order image dependencies such as facial lines may remain

http://etd.uwc.ac.za/

across the data dimensions even after PCA is performed [DBH+99]. *Independent component analysis* (ICA) is a technique for removing such dependencies from the input data set. Under ICA, the set of generated basis vectors $\mathbf{e}_1, \ldots, \mathbf{e}_n$ are called independent components, and the projection of T onto each $\mathbf{e}_i$ is statistically independent of all the other projections. A derivation of ICA is available from Hyvarinen and Oja [HE00].

In contrast to PCA, the independent components of ICA are inherently unordered. Thus, when using ICA for dimension reduction of a feature set, a metric of ordering must be defined externally and then applied to the set of components. One possible metric is the class discriminability, defined as the ratio of the between-class to within-class variance of an independent component when applied to the training set. This approach has been used by [DBH+99].

For FER, ICA has proven to be highly effective, yielding recognition rates as high as with Gabor filters (see Section 3.5.4). In terms of execution time, ICA can outperform Gabor-based feature extraction by an order of magnitude [BDM+00]. In the literature, ICA has yet only been deployed in a few FER systems. In Bartlett, et al [BDM+00] and Donato, et al [DBH+99], an ICA representation achieves 96% accuracy when classifying 6 upper- and 6 lower-face AUs, thus tying for first place with Gabor filters among the techniques that were investigated. Fasel and Lüttin [FL00] used ICA and the nearest neighbor algorithm to classify 9 individual AUs and 16 AU combinations. Their system achieves 83% accuracy on single AUs and 74% accuracy when tested on both single AUs and their combinations.

3.5.4 Gabor Filters

Although ICA does deliver high recognition accuracy, it also suffers from the drawback of a long training time for the calculation of the independent components [Lit]. In general, dimensionality reduction techniques have given way to *image filtering* techniques in the FER literature. Filters are a means of enhancing the facial lines, skin bulges, and other appearance changes that facial expressions can induce.

One of the mostly commonly deployed and successful appearance-based methods for facial expression recognition is the Gabor decomposition. The *Gabor decomposition* of an image is computed by filtering the input image with a *Gabor filter*, which can be tuned to a particular frequency $\mathbf{k}_0 = (u, v)$ where $k = \|\mathbf{k}_0\|$ is the scalar frequency and $\varphi = \arctan(\frac{v}{u})$ is the orientation. Gabor filters accentuate the frequency components of the input image which lie close to k and φ in spatial frequency and orientation, respectively.

A Gabor filter can be represented in the space domain using complex exponential notation as:

$$F_{\mathbf{k}_0}(\mathbf{x}) = \frac{{\mathbf{k}_0}^2}{\sigma^2} \exp\left(-\frac{{\mathbf{k}_0}^2 \mathbf{x}^2}{2\sigma^2}\right)\left(\exp\left(i\mathbf{k}_0 \cdot \mathbf{x}\right) - \exp\left(-\frac{\sigma^2}{2}\right)\right)$$

where $\mathbf{x} = (x, y)$ is the image location and $\mathbf{k}_0$ is the peak response frequency [LVB+93]. An example of a Gabor filter is given in Figure 3.1, which shows the absolute value (left), real component (middle), and

http://etd.uwc.ac.za/

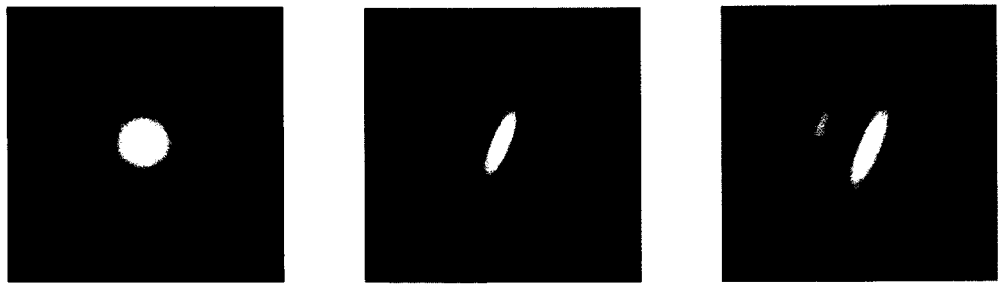


Figure 3.1: The left, middle, and right graphics above show the absolute value, and the real and imaginary components of a sample Gabor filter.

imaginary component (right) of the filter in the space domain. Notice how the filter is spatially local in all three cases. The real and imaginary components accentuate respectively the symmetric and asymmetric responses of the image to the filter's characteristic frequency and orientation. The filter can then be applied to an input image $\mathbf{I} \in \mathbb{R}^2$ using two-dimensional convolution. More commonly, however, the Gabor filter is computed in the frequency domain as:

$$G_{\mathbf{k}_0}(\mathbf{k}) = \exp\left(-\frac{\sigma^2\left(\mathbf{k}-\mathbf{k}_0\right)^2}{2{\mathbf{k}_0}^2}\right) - \exp\left(-\frac{\sigma^2\left({\mathbf{k}_0}^2+\mathbf{k}^2\right)}{2{\mathbf{k}_0}^2}\right)$$

where $\mathbf{k} = (u, v)$ represents the horizontal and vertical spatial frequency components of the input image (equation from [LVB+93]). The Fourier-transformed image is multiplied by G and the result is then inverse-transformed back into the space domain.

For FER, often a *filter bank* of multiple Gabor filters tuned to different characteristic frequencies and orientations is used for feature extraction. The combined response is called a *jet*. Filter banks typically span at least 6 different orientations and have frequencies spaced at half-octaves. Prior to classification, the extracted features are usually converted into real numbers by calculating the magnitude of the complex filter response.

Gabor filters can be used for feature extraction in two main ways: by extracting the Gabor responses at fiducial points on the face, or by extracting them over entire image regions. In the former case, the Gabor responses are best computed directly in the space domain by convolving each filter at the desired image locations. In the latter, it is usually faster to use the Fast Fourier Transform (FFT).

Some of the most successful appearance-based FER systems to-date employ Gabor filters for feature extraction. We discuss such systems below.

Gabor Responses at Fiducial Points

The first software systems to deploy the Gabor decomposition for FER calculated the Gabor responses only at specific locations on the face. Zhang, et al [ZLSA98], Zhang [Zha98], and Lyons and Akamatsu [LA98]

http://etd.uwc.ac.za/

were among the first to use such an approach. In their systems, a Gabor filter bank consisting of 3 spatial frequencies and 6 orientations is convolved with the input image at selected facial points derived from a facial mesh model. In [ZLSA98] and [Zha98], a multi-layer perceptron is trained to recognize prototypical expressions with accuracy near 90%. In [LA98], each face is allowed to express not just a single prototypical emotion, but instead multiple emotions at different intensities. Using the cosine similarity measure, Gabor responses are used to predict the intensity of each expression category. The intensities were correlated with those coded by human subjects, but no percent-correct statistics were reported. In later work [LPA00], Lyons, et al developed a FER system by classifying the Gabor responses along facial mesh points using linear discriminant analysis and the nearest neighbor classifier. The system achieved over 90% accuracy for prototypical expressions.

Point-sampled Gabor features have also been used to recognize FACS AUs. In Tian, et al [lTKC00], for example, a Gabor bank comprising 6 frequencies and 3 orientations is applied to the inner corner, outer corner, and middle of each eye. By processing the Gabor responses using a 3-layer neural network, their system achieves classification rates of 93% for AU 41, 70% for AU 42, and 81% for AU 43 [lTKC00]. In later work [lTKC02], Tian, et al created a similar system that samples the Gabor responses of 20 facial points of the eye, eyebrows, and forehead. They tested their method on a dataset in which subjects spanned a wider range of ethnicities and which contained more head movement than most other FER databases. Under these more challenging conditions, the Gabor-based system achieved an overall AU recognition rate on 8 AUs of only 32% [lTKC02].

Gabor Responses over Image Regions

The alternative to applying Gabor filters at specific points is to apply them instead to the whole face. Some of the highest recognition accuracies in the FER literature have been achieved using the Gabor decomposition over entire image regions for feature extraction. Bartlett, Donato, et al [DBH+99], [BDM+00] developed a recognition system using Gabor filters and the nearest neighbor classifier. Both implementations employ a filter bank of 5 frequencies and 8 spatial orientations. In order to reduce the dimensionality of the Gabor jets, the filtered images are sub-sampled by a factor of 16 prior to classification. This system achieves an overall classification rate of 96% on 6 upper- and 6 lower-face AUs [DBH+99].

In subsequent work, Bartlett, et al [MGB+03] developed a Gabor-based AU recognition system that is robust to natural, out-of-plane movements of the head. It employs both support vector machines and hidden Markov models for classification. When classifying the AU combination 1+2, it scores 90.6% accuracy, and on AU 4 it achieves 75.0% accuracy. Littlewort-Ford, et al [LFBM01] used Gabor filters on difference images of the face and support vector machines to classify AUs 6 and 12 in order to distinguish natural smiles from posed, "social" smiles. Using a linear SVM kernel to classify the Gabor-filtered images, 75% of

http://etd.uwc.ac.za/

smiles were classified correctly. Non-expert human subjects, on the other hand, achieved only 60% accuracy when scoring the same dataset [LFBM01].

Gabor Responses at Learned Locations

The final Gabor-based method of feature extraction that we consider combines the advantages of both of the previous approaches: a sparse set of Gabor responses from *learned* locations, frequencies, and orientations are selected from the whole face image, and the resultant feature vector is then classified. This strategy has been employed in two FER systems: Littlewort, et al [LBF+04] compare two methods of selected Gabor filter classification: in one, they classify Gabor responses selected by Adaboost [FS99] using support vector machines (AdaSVMs), and in the other, they classify the selected Gabor responses directly using Adaboost. Recognition rates when detecting 7 prototypical emotions were highest with AdaSVMs, up to 93.3% accuracy.

Finally, Bartlett, et al [BLF+06] use a similar method as in [LBF+04] for the classification of 20 AUs: they use Adaboost to classify Gabor responses extracted from automatically detected faces at 8 orientations and 9 frequencies. Percent-correct accuracy on a combined dataset from both the Cohn-Kanade and Ekman-Hager databases was 90.9%.

Configuring the Filter Bank

One consideration when using Gabor filter banks is the selection of peak frequencies and orientations of the individual filters. While most FER systems employ 8 spatial orientations spaced $\pi/8$ radians apart, there is no standard set of peak frequency values that has proven to be optimal. Little published research has explicitly investigated the ideal filter bank for face analysis. Fasel and Bartlett [FB02] investigated the optimum filter bank for the purpose of locating fiducial points of the face, and their results indicate that only one, very low-frequency value (4 iris widths per cycle) may be needed for optimal accuracy. However, Donato, et al [DBH+99] investigated the same question of optimum frequency values for the task of FER. Their results indicate that the *higher* frequencies were more important for classification. Optimum selection of frequencies thus likely depends on the specific application, and there is yet no consensus on the best choice of filter bank.

3.5.5 Haar Wavelets

Although Gabor feature-based systems have produced some of the highest recognition accuracies in FER, they also suffer from two drawbacks: the large size of the image representation, and the high computational expense involved in computing it. For a bank of 40 Gabor filters, for example, the combined Gabor responses over all image pixels consume 40 times as much memory as the single input image. In order

http://etd.uwc.ac.za/

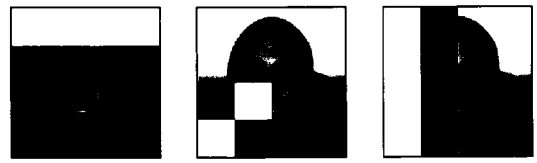


Figure 3.2: Examples of Haar wavelets in a true Haar decomposition superimposed onto a face image. Width, height, and (x, y) positions of all wavelets are aligned at powers of 2.

to apply a Gabor filter bank to an image, the input image must first be transformed into the frequency domain using an FFT. Then, for each filter G in the bank, the transformed image must be multiplied by G and then inverse-transformed back into the space domain. The total computational expense of the single Fourier transform and all the inverse transforms is substantial. Even when only selected Gabor responses are classified, the convolutions in the space domain incur some cost.

An alternative to Gabor filters which has already proven both effective and efficient in face analysis is the Haar filter, based approximately on the Haar wavelet decomposition. The two-dimensional Haar decomposition of a square image with n^2 pixels consists of n^2 wavelet coefficients, each of which corresponds to a distinct Haar wavelet. The first such wavelet is the mean pixel intensity value of the whole image; the rest of the wavelets are computed as the difference in mean intensity values of horizontally, vertically, or diagonally adjacent squares. Figure 3.2 shows three example Haar wavelets superimposed onto a face image. The Haar coefficient of a particular Haar wavelet is computed as the difference in average pixel value between the image pixels in the black and white regions. The two-dimensional Haar decomposition is exactly complete, i.e., the Haar decomposition of an image with n^2 pixels contains exactly n^2 coefficients. Each wavelet is constrained both in its (x, y) location and its width and height to be aligned on a power of 2. For object recognition systems, however, these constraints are sometimes relaxed in order to improve classification results.

In contrast to Gabor filters, Haar filters require no FFT for their extraction, and with the "integral image" technique demonstrated by Viola and Jones in their landmark face detection paper [VJ04], Haar features can be computed in only a few CPU instructions. In this thesis, we implement such a Haar feature-based system and evaluate its performance in Chapter 5. Section A.5 describes the Haar decomposition in greater detail. Here, we provide a brief review of object detection systems that deploy Haar wavelets for feature extraction.

Applications to Object Detection

One of the earliest applications of the Haar wavelet to object recognition was developed by Jacobs, et al [JFS95] for querying an image database. Theirs is the only object recognition system known to us that uses true Haar wavelets in the strict mathematical sense for feature extraction. In their application, the user

http://etd.uwc.ac.za/

recognition. We present our own study of this approach in Chapter 5 of this thesis.

3.6 Comparing the Two Approaches

Geometry- and appearance-based FER systems contrast starkly and are complementary. Geometry-based methods completely disregard all color information (except possibly to track the feature points). Their performance in classifying facial expressions depends on the particular set of facial points that the system designer chooses to track. Appearance-based methods, on the other hand, disregard the geometric relationships between different points on the face except to the extent that these relationships can be captured by frequency-tuned image filters. Given that these two paradigms of expression recognition differ so greatly, and given that both kinds of FER systems have achieved recognition accuracies above 90%, it is important to determine under which conditions each method delivers higher accuracy. Evaluating the comparative performance of these two approaches is difficult because different FER systems are tested on different datasets. A few research studies do exist, however, which compare the two strategies with respect to classification accuracy.

Zhang [Zha98] and Zhang, et al [ZLSA98] compare Gabor-based and geometry-based FER methods for prototypical expressions on an image database containing frontal faces. In their experiment, the Gabor decompositions are computed at 3 spatial frequencies and 6 orientations at 34 landmark points distributed over the face. In the geometry-based method, the feature vector consists of the positions of the same 34 fiducial points. For both approaches, a two-layer neural network is used as the classifier. Empirical results show that the appearance-based method delivers substantially higher recognition accuracy - typically around 20% - regardless of the number of hidden units [ZLSA98],[Zha98].

Tian, Kanade, and Cohn [lTKC02], however, dispute the higher recognition accuracy of the Gabor method claimed by Zhang. On an ethnically more heterogeneous database containing more head movement, they perform a similar experiment as Zhang, et al, except that AUs, not prototypical expressions, are classified. Their results show that, when classifying expressions with complex AU combinations, AU recognition accuracy fell dramatically to 32% with the Gabor method, whereas the geometry-based approach retained 87.6% accuracy. However, the comparison in [lTKC02] did not test the appearance-based approach with Gabor responses measured over the entire face - a method which has proven highly effective [DBH+99].

From the limited evidence available, it is difficult to predict which approach will ultimately prove superior. Cohn, et al [CKM+01] report that the face analysis group of CMU/Pittsburgh, which has used a geometry-based approach, and the group at UCSD, which uses only appearance-based features, are competing for higher recognition performance on the same real-world FACS AU recognition task. This study

http://etd.uwc.ac.za/

will hopefully help to differentiate the two approaches more clearly.

3.7 Combining Geometric and Appearance-based Features

As an alternative to choosing *either* appearance-based features *or* geometry-based features, FER systems can also be built that exploit both. Several systems already exist which take this approach: The system of Zhang, et al [Zha98],[ZLSA98], for example, uses a 3-layer neural network to classify a combined set of Gabor responses and raw facial point locations. The Gabor responses are sampled only at particular locations in the image. On an expression database containing approximately equal numbers of the 7 prototypical emotions [Zha98], their system achieves around 65% recognition accuracy. When classifying only Gabor features, their system achieves a much higher 90% accuracy. Surprisingly, the combined system - Gabor responses plus fiducial point locations - does no better than Gabor features alone (90%). This shows that combined-feature systems must be engineered carefully in order to reap the benefit of both feature types.

Tian, et al [lT04],[lTKC02] developed a similar system using a neural network to classify both Gabor and geometric features. In contrast to Zhang, et al [Zha98],[ZLSA98], however, their system converts the fiducial point locations into a set of 15 parameters describing the state (e.g., open/closed) of the lips, nasolabial furrows, and eyes. Moreover, the Gabor responses are calculated over the entire face, not just at particular points. The output of their classifier is a set of FACS action units. On their dataset, the combined approach (92.7% accuracy) demonstrates a clear advantage over either appearance-based (32%) or geometry-based features (87.6%) alone [lTKC02].

Cohn, et al [CKM+01] use manually constructed models to classify expressions of the eyes and brows. In particular, "brow-up","brow-down", and "non-brow motion" are classified using both appearance-based features quantifying the amount of edges detected in the forehead (for wrinkle detection) and geometry-based features measuring displacement of fiducial points along the eyebrows. Accuracy is reported as 57% across the three classified actions [CKM+01].

Datcu and Rothkrantz's system [DR04] classifies both prototypical expressions and AUs using a Bayesian belief network and a combined set of three feature types: (1) relative positions of fiducial points; (2) displacements of individual fiducial points through time; and (3) PCA projection coefficients of chin, forehead, and cheek regions. Unfortunately, although the system is described as "very promising", no accuracy statistics are reported in their paper.

Finally, Lanitis, et al [LTC95] use discriminate function analyzes to classify three types of features: (1) a geometric representation modeling the shape and pose of the face (Active Shape Models); (2) shape-invariant pixel intensity values computed by warping the face onto a standard model; and (3) pixel intensity values along specific lines normal to the edge of the face. All features are pre-processed using PCA prior to

http://etd.uwc.ac.za/

classification. The system achieves 74% accuracy when classifying prototypical expressions [LTC95].

3.8 Conclusions

In the preceding sections we have described numerous systems for automatic FER that utilize a diverse range of feature types, both appearance-based and geometry-based. One of the fundamental issues that concerns us is which of these two approaches is superior. Unfortunately, no study to date has conclusively answered this question, though the pending results of the study mentioned in [CKM+01] will be useful. Another important issue is how the strengths of both methods can effectively be combined in order to create a classifier superior to either individual method. Systems that combine the two approaches do exist (see Section 3.7), but they are not based on the most promising methods from each of the appearance- and geometry-based feature categories. One interesting study would be to create a combined feature vector of fiducial point locations as well as Adaboost-selected Gabor responses using support vector machines as the classifier. Given the high performance on FER tasks achieved by these machine learning tools individually, it would be instructive to investigate whether they could yield even higher performance in cooperation.

3.9 Summary

We have surveyed a broad-range of systems for automatic FER. In our survey we focused on two issues: whether local segmentations yield superior accuracy to global segmentations, and which category of feature vector - appearance-based or geometry-based - leads to higher accuracy. Finally, we compared the two approaches and suggested a possible choice of combining the strengths of both.

UNIVERSITY of the
WESTERN CAPE

http://etd.uwc.ac.za/

Chapter 4

Support Vector Machines

The development of the support vector machine (SVM) and kernel methods have garnered considerable attention in the machine learning literature in recent years. The basic principle of the SVM is simple: maximize the distance in the input space between the two classes of data points one wishes to classify. SVMs offer several advantages over other classifiers: For one, training time of the classifier does not suffer from a high dimensional feature vector. Given the high dimensionality of such feature types as the Gabor decomposition of an entire face, this advantage is significant. For another, the SVM offers both power and flexibility through use of the "kernel trick" - the default linear kernel can be replaced with a RBF, polynomial, sigmoidal, and many other kernels which may separate the data points more cleanly for the given problem domain. Because of these advantages, and because of the many successful deployments of the SVM in machine learning problems, both in FER and elsewhere, we provide a mathematical derivation of the support vector machine in the following sections. The interested reader may also wish to consult [Bur98] and [SS98].

UNIVERSITY of the WESTERN CAPE

4.1 Premise

Suppose $T = \{(\mathbf{x_1}, y_1), \ldots, (\mathbf{x}_l, y_l)\}$ is a set of l training data, where each $\mathbf{x}_i$ is a data point in $\mathbb{R}^d$ and each $y_i \in \{-1, 1\}$ is the corresponding classification label. Suppose also that the sets $T^+ = \{\mathbf{x}_i \mid y_i = 1\}$ from $T^- = \{\mathbf{x}_i \mid y_i = -1\}$ are linearly separable in $\mathbb{R}^d$ so that a hyperplane can be formed between them.

For any such separating hyperplane H, consider the subset of T^+ of points that lie closest to H. These points lie in a hyperplane H^+ which is parallel to H; denote the distance between H^+ and H as d^+. Similarly, the subset of T^- of points closest to H lie in a hyperplane H^-, which is parallel and distance d^- to H. The sum of d^+ and d^- equals the distance from H^+ to H^- and is known as the *margin* of H. Denote this margin as d.

http://etd.uwc.ac.za/

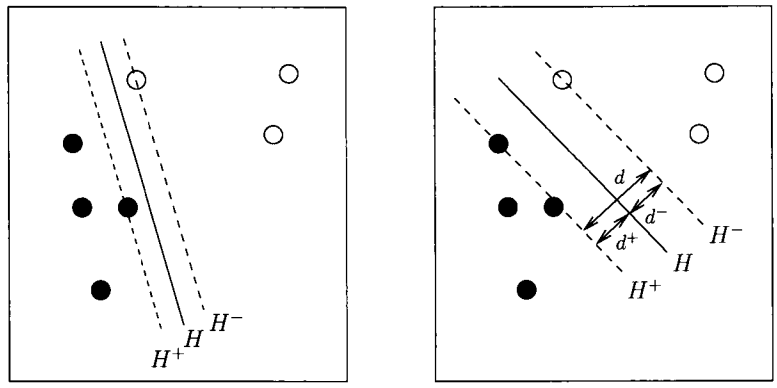


Figure 4.1: A hypothetical training set in $\mathbb{R}^2$ in which the solid points have positive label and the hollow points have negative label. Notice that, although the hyperplane H in each figure separates the two classes, only the hyperplane in (b) maximizes the margin d.

A *support vector machine* (SVM) is created by finding the unique separating hyperplane which maximizes the margin between T^+ and T^-. This optimal hyperplane lies halfway between H^+ and H^- so that the distance from any point in all of T to H is likewise maximized. Figure 4.1 illustrates a hypothetical data set and two separating hyperplanes; only the decision boundary in Figure 4.1(b) is optimal. The training points which lie on H^+ or H^- are called the support vectors of T.

4.2 Training Phase

In order to compute H, we must first describe it formally. The general equation for a hyperplane is $\mathbf{w}\cdot\mathbf{x}+b = 0$, where $\mathbf{w}$ is the normal vector and $b/\|\mathbf{w}\|$ is the perpendicular signed distance to the origin. The same plane can be described by an infinite number of equations by scaling $\mathbf{w}$ and b. For our purposes, we select a particular scale such that the equations for H^-, H, and H^+ are as follows (recall that, since all three planes are parallel, their normal vectors can be scaled to be equal):

$$H^- : \quad \mathbf{w}\cdot\mathbf{x}+b = -1 \tag{4.1}$$

$$H : \quad \mathbf{w}\cdot\mathbf{x}+b = 0 \tag{4.2}$$

$$H^+ : \quad \mathbf{w}\cdot\mathbf{x}+b = +1 \tag{4.3}$$

H^- and H^+ contain the negatively and positively labeled data points closest to H, respectively. Since all data points not in H^+ or H^- must lie even farther from H, we require that:

$$\mathbf{w}\cdot\mathbf{x}_i+b \geq +1 \quad \forall\mathbf{x}_i \in T^+ \tag{4.4}$$

$$\mathbf{w}\cdot\mathbf{x}_i+b \leq -1 \quad \forall\mathbf{x}_i \in T^- \tag{4.5}$$

http://etd.uwc.ac.za/

These two conditions can be unified by introducing the classification label y_i:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \quad \forall \mathbf{x}_i \in T \tag{4.6}$$

We must identify the hyperplane H with maximum margin. The margin d of H equals the distance between H^+ and H^-. The distance from H^+ to the origin is $\frac{1-b}{\|\mathbf{w}\|}$, and the distance from H^- to the origin is $\frac{-1-b}{\|\mathbf{w}\|}$. Therefore, the margin d equals:

$$\frac{1-b}{\|\mathbf{w}\|} - \frac{-1-b}{\|\mathbf{w}\|} = \frac{1-b+1+b}{\|\mathbf{w}\|} \tag{4.7}$$

$$= \frac{2}{\|\mathbf{w}\|} \tag{4.8}$$

The margin can thus be maximized by minimizing $\|\mathbf{w}\|$, or, equivalently, by minimizing $\frac{1}{2}\|\mathbf{w}\|^2$. The values for $\mathbf{w}$ and b must simultaneously fulfill the conditions $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1$ for every $\mathbf{x}_i \in T$. This is a constrained optimization problem, and we will use the Lagrangian method to solve it.

4.2.1 The Lagrangian Method and the Wolfe Dual Form

The Lagrangian method for solving constrained optimization problems includes three components: (1) the objective function $f(\mathbf{x})$ to be minimized (or $-f$ if f is to be maximized); (2) the constraint functions $c_1(\mathbf{x}), \ldots, c_n(\mathbf{x})$; and (3) the vector α of n Lagrange multipliers (one for each constraint function). The Lagrangian function is then assembled as:

$$L(\mathbf{x}, \alpha) = f(\mathbf{x}) - \sum_{i=1}^{n} c_i(\mathbf{x}) \tag{4.9}$$

The solutions to certain types of constrained optimization problems can be found by solving the *Wolfe dual problem*: instead of minimizing f subject to the constraints $c_1(\mathbf{x}), \ldots, c_n(\mathbf{x})$, one instead *maximizes* the Lagrangian subject to the constraint that L is minimized with respect to $\mathbf{x}$. Both the primal and dual problems find their solutions at the same point along the Lagrangian curve, namely the saddle point.

The Wolfe dual method is valid under the following conditions: (a) the optimization problem is a convex programming problem; (b) both the objective function and the constraint functions are differentiable; and (c) the constraints are linear[1]. Solutions to the Wolfe dual problem are then guaranteed to occur at global minima due to the convexity of f [Fle80].

Before applying Wolfe's dual to our problem, we first verify that it fulfills the stated assumptions. First, a *convex programming problem* consists of a convex objective function to be minimized over a convex set.

[1]In fact, the Wolfe dual also applies to convex programming problems with certain non-linear constraints, provided that these constraints meet a *regularity assumption* (see [Fle80]).

UNIVERSITY of the WESTERN CAPE
http://etd.uwc.ac.za/

In our problem, the objective function is $\frac{1}{2}\|\mathbf{w}\|^2$; since its second derivative is positive everywhere, it is a convex function. To verify that the feasible set of points satisfying the constraints is a convex set, we must first note that any single linear constraint defines a convex set. Since the intersection of multiple convex sets is likewise convex, and since multiple simultaneous linear constraints represent exactly such an intersection, our feasible set is convex. Finally, the conditions that all functions are differentiable, and that the constraints are linear, are clearly true. We may thus proceed.

The Lagrangian function of our optimization problem equals:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{l} \alpha_i \left[y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) - 1\right] \tag{4.10}$$

where α is the vector of the l Lagrange multipliers. Since the constraints we are dealing with are *inequality* constraints, each component of α must be non-negative at the solution. As stated above, we must minimize L with respect to $\mathbf{w}$ and b. This requires that the derivatives $\frac{\partial}{\partial \mathbf{w}} L$ and $\frac{\partial}{\partial b} L$ equal zero. The first such differentiation yields:

$$\frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}, b, \alpha) = \mathbf{w} - \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i = 0 \tag{4.11}$$

$$\implies \mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i \tag{4.12}$$

This equation reveals two facts about $\mathbf{w}$: First, since the $(\mathbf{x}_i, y_i)$ pairs are known, one need compute only the α_i to determine $\mathbf{w}$. Second, only those $\mathbf{x}_i$ for which $\alpha_i > 0$ affect the determination of the hyperplane. These data points lie on H^+ (or H^-) and are called the *support vectors* of the training set. All data which are not support vectors could, hypothetically, be removed from T without affecting the placement of H.

We can substitute $\sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i$ for $\mathbf{w}$ into the original Lagrangian to yield a simplified function W:

$$W(b, \alpha) = \frac{1}{2}\left(\sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i\right)^2 - \sum_{i=1}^{l} \alpha_i \left\{ y_i \left[\left(\sum_{j=1}^{l} \alpha_j y_j \mathbf{x}_j \right) \cdot \mathbf{x}_i + b \right] - 1 \right\} \tag{4.13}$$

$$W(b, \alpha) = \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j - \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j - \sum_{i=1}^{l} \alpha_i y_i b + \sum_{i=1}^{l} \alpha_i \tag{4.14}$$

$$W(b, \alpha) = \sum_{i=1}^{l} \alpha_i - b \sum_{i=1}^{l} \alpha_i y_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \tag{4.15}$$

We will use the second required differentiation (with respect to b) to simplify W further.

$$\frac{\partial}{\partial b} L(\mathbf{w}, b, \alpha) = \sum_{i=1}^{l} \alpha_i y_i = 0 \tag{4.16}$$

http://etd.uwc.ac.za/

Substituting 0 for $\sum_{i=1}^{l} \alpha_i y_i$ from Equation 4.16 we arrive at:

$$W(b, \alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \tag{4.17}$$

The simplified function W represents the Lagrangian minimized with respect to $\mathbf{w}$ and b. It must still be maximized with respect to the remaining variables, i.e., the Lagrange multipliers α. This represents a quadratic programming problem and can be computed efficiently using computer software. Once the values of α have been determined, we can then calculate $\mathbf{w}$ according to Equation 4.12.

4.2.2 Determining b

We must still determine b. To do so, recall that we first minimized L with respect to $\mathbf{w}$ and b. At such local minima, the Kuhn-Tucker necessary conditions for a local minimizer apply [Fle80]. These include, among others, the *complementarity* condition:

$$\alpha_i (y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1) = 0 \tag{4.18}$$

which means that either the constraint $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1$ must equal exactly 0 (an *active* constraint), or the associated α_i must equal zero. The points for which $\alpha_i \neq 0$ are, in fact, the support vectors. Once α and $\mathbf{w}$ have been calculated, they can be substituted into Eq. 4.18. By substituting any particular data point $\mathbf{x}_i$, the value of b can be retrieved. Usually, however, to enhance numerical stability in floating point computation, the average b over all i is used [Bur98].

The SVM has now been trained.

UNIVERSITY of the
WESTERN CAPE

4.3 Test Phase

Once the separating hyperplane has been identified, it can be used to classify a new data point $\mathbf{x}$ with an unknown classification label. Determining the associated y value requires merely testing on which side of H the point lies; this is evaluated:

$$y = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b) \tag{4.19}$$

The support vector machine is now a classifier.

http://etd.uwc.ac.za/

4.4 Linear Inseparability

For some training sets, it may be impossible to find a linear hyperplane which separates the points in T^+ from those in T^-. In such cases, there is always at least one data vector $\mathbf{x}_i$ for which Eq. 4.6 does not hold. The standard approach to handling this inseparability is the *soft-margin generalization* of the support vector machine. This approach introduces *slack variables* ξ_i which specify the amount by which Eq. 4.6 is violated. The new constraint functions then become:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i \tag{4.20}$$

for $\xi_i \geq 0$. The objective function is also augmented with an additional term (a function of ξ_i) to penalize errors:

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\left(\sum_{i=1}^{l} \xi_i\right)^k \tag{4.21}$$

Here, parameter C controls the amount by which errors are penalized (higher C results in larger penalty). For exponent $k = 1$ or $k = 2$, the optimization problem remains quadratic; k is usually set to 1 for pattern recognition problems.

The Lagrangian function of this new constrained optimization problem becomes:

$$L(\mathbf{w}, b, \xi, \alpha, \mu) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l} \xi_i - \sum_{i=1}^{l} \alpha_i \left[y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) - 1 - \xi_i\right] - \sum_{i=1}^{l} \mu_i \xi_i \tag{4.22}$$

where the new vector of Lagrange multipliers μ was introduced to ensure non-negativity of each ξ_i. Since each $\xi_i \geq 0$ is an *inequality* constraint, we require that $\mu_i \geq 0$.

The solution is found analogously to the linearly separable case - by minimizing with respect to the primal variables (including the new variables ξ_i) and maximizing with respect to the dual variables (including each μ_i). Minimization yields the following equations:

$$\frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}, b, \xi, \alpha, \mu) = \mathbf{w} - \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i = 0 \tag{4.23}$$

$$\implies \mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i \tag{4.24}$$

$$\frac{\partial}{\partial b} L(\mathbf{w}, b, \xi, \alpha, \mu) = \sum_{i=1}^{l} \alpha_i y_i = 0 \tag{4.25}$$

$$\frac{\partial}{\partial \xi_i} L(\mathbf{w}, b, \xi, \alpha, \mu) = C - \alpha_i - \mu_i = 0 \tag{4.26}$$

http://etd.uwc.ac.za/

Equations 4.24 and 4.25 are the same as for the separable case. The last equation $C - \alpha_i - \mu_i$ combined with the constraints $\alpha_i \geq 0$ and $\mu_i \geq 0$ yields the additional constraint that $\alpha_i \leq C$. All three inequalities must hold true at the solution. With the exception of the additional constraints $\alpha_i \leq C$ and $\mu_i \geq 0$, the solution to the optimization problem proceeds exactly as for the separable case.

Eq. 4.24 is substituted into Eq. 4.22, to arrive at the function W:

$$W(b,\xi,\alpha,\mu) = \frac{1}{2}\left(\sum_{i=1}^{l}\alpha_i y_i \mathbf{x}_i\right)^2 + C\sum_{i=1}^{l}\xi_i \quad - \\ \sum_{i=1}^{l}\alpha_i\left\{y_i\left[\left(\sum_{j=1}^{l}\alpha_j y_j \mathbf{x}_j\right)\cdot\mathbf{x}_i + b\right] - 1 - \xi_i\right\} - \sum_{i=1}^{l}\mu_i\xi_i \tag{4.27}$$

$$W(b,\xi,\alpha,\mu) = \frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j \mathbf{x}_i\cdot\mathbf{x}_j + C\sum_{i=1}^{l}\xi_i \quad - \\ \sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j \mathbf{x}_i\cdot\mathbf{x}_j - \sum_{i=1}^{l}\alpha_i y_i b + \sum_{i=1}^{l}\alpha_i + \sum_{i=1}^{l}\alpha_i\xi_i - \sum_{i=1}^{l}\mu_i\xi_i \tag{4.28}$$

$$W(b,\xi,\alpha,\mu) = \sum_{i=1}^{l}\alpha_i + \sum_{i=1}^{l}(C-\alpha_i-\mu_i)\xi_i - \frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j \mathbf{x}_i\cdot\mathbf{x}_j - b\sum_{i=1}^{l}\alpha_i y_i \tag{4.29}$$

We further substitute the expressions $\sum_{i=1}^{l}\alpha_i y_i = 0$ and $C - \alpha_i - \mu_i = 0$ to yield:

$$W(b,\xi,\alpha,\mu) = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j \mathbf{x}_i\cdot\mathbf{x}_j \tag{4.30}$$

Eq. 4.30 is a quadratic programming problem, and its solution can be computed efficiently as such.

4.5 Non-linear Decision Surfaces

Some data sets, while linearly inseparable in their natural feature space (we assumed $\mathbb{R}^d$), become separable after they are transformed into a space of higher dimension. The data in Figure 4.2, for example, are linearly inseparable in $\mathbb{R}^1$. When they are transformed into $\mathbb{R}^2$ under the map $\Phi(x) = (x, x^2)$, however, they become linearly separable; the corresponding optimal hyperplane is shown. This new-found separability in higher-dimensional (or even infinite-dimensional) spaces can be exploited due to particular properties of the SVM derivation. First, notice that the data points occur only in the form of inner products in the training phase:

$$W(b,\alpha) = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j \mathbf{x}_i\cdot\mathbf{x}_j \tag{4.31}$$

http://etd.uwc.ac.za/

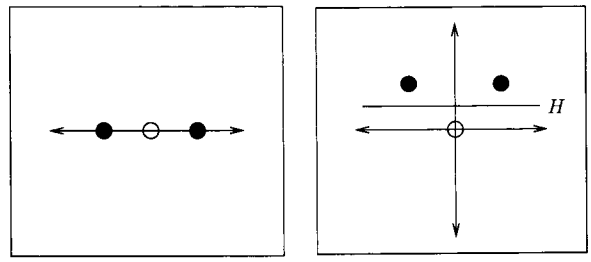


Figure 4.2: A hypothetical training set in $\mathbb{R}^1$ which is linearly inseparable (left). After it is mapped under $\Phi = (x, x^2)$ onto $\mathbb{R}^2$, however, the data is separable (right) with optimal hyperplane H.

In the test phase, a similar substitution for $\mathbf{x}_i$ and $\mathbf{x}_j$ can be made. First, however, we must substitute Eq. 4.12 for $\mathbf{w}$ (note that we have added the two subscripts j for uniformity of notation):

$$y_j = \text{sign}(\mathbf{w} \cdot \mathbf{x}_j + b) \tag{4.32}$$

$$= \text{sign}\left(\left(\sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i\right) \mathbf{x}_j + b\right) \tag{4.33}$$

$$= \text{sign}\left(\sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x}_j + b\right) \tag{4.34}$$

Now, since data only appear as inner products, we can replace all occurrences of the data vectors with a *kernel function* $K(\mathbf{x}, \mathbf{y})$. K first transforms each input vector under the map $\Phi : \mathbb{R}^d \rightarrow H$ and then returns the inner product in H. After substituting K for $\mathbf{x} \cdot \mathbf{y}$, Equations 4.31 and 4.34 become:

$$W(b, \alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{4.35}$$

and

$$y_j = \text{sign}\left(\sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_j) + b\right) \tag{4.36}$$

respectively.

In the example illustrated in Figure 4.2, the kernel function K equals:

$$K(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{y}) \tag{4.37}$$

$$= (x, x^2) \cdot (y, y^2) \tag{4.38}$$

$$= xy + x^2 y^2 \tag{4.39}$$

$$= \mathbf{x} \cdot \mathbf{y} + (\mathbf{x} \cdot \mathbf{y})^2 \tag{4.40}$$

UNIVERSITY of the WESTERN CAPE
http://etd.uwc.ac.za/

which describes a parabolic decision surface. Many kernel functions are possible - as long as K computes the inner product of $\mathbf{x}_i$ and $\mathbf{x}_j$ within *some* inner product space, it is irrelevant to the SVM derivation which *particular* space this is. Similarly, the transformation function Φ need not be known at all - only its *existence* need be certain. Usually, one starts by creating a kernel K as opposed to deciding on a particular transformation Φ [Bur98].

4.5.1 Kernel Functions and Mercer's Condition

The issue still remains of which kernel functions actually correspond to the inner product of two transformed input vectors. This question is answered by Mercer's theorem, which states that a function $K(\mathbf{x}, \mathbf{y})$ represents the inner product two vectors $\mathbf{x}$ and $\mathbf{y}$ in a Hilbert space if and only if the following condition holds true for any function g:

$$\int g(\mathbf{x})^2 d\mathbf{x} \quad \text{is finite} \implies \int K(\mathbf{x}, \mathbf{y}) g(\mathbf{x}) g(\mathbf{y}) d\mathbf{x} d\mathbf{y} \geq 0 \tag{4.41}$$

Note that this theorem helps to determine neither the transformation Φ nor the space H to which Φ maps its input. This theorem can be used, however, to prove the admissibility of certain kernels. The most common kernels in practice are:

- The Gaussian radial basis function (RBF) kernel: $K(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2/(2\sigma^2))$.
- The polynomial kernel: $K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^p$ for positive integers p.
- The sigmoid (hyperbolic tangent): $K(\mathbf{x}, \mathbf{y}) = \tanh(\kappa \mathbf{x} \cdot \mathbf{y} - \delta)$. Note that this last kernel fulfills Mercer's condition only for certain values of κ and δ [Bur98].

Alternatively, one can verify that a kernel K is admissible for SVM classification by showing that it is a dot-product kernel, or the kernel of a reproducing kernel Hilbert space. A particular class of kernel function that guarantees it is admissible for SVMs is *conditionally positive definite* functions, described in [SSM98].

Finally, it is important to note that the kernel trick does not render the soft-margin SVM generalization redundant - even when using a non-linear decision boundary, the data set will often be inseparable.

4.6 Polychotomous Classification

The SVM classifier introduced thus far can handle only 2-class (dichotomous) problems. A variety of techniques does exist, however, with with SVMs can be applied to multi-class problems. Although we make no attempt to survey all of them, we do describe two of the most common - the *one-versus-rest* (1-v-r) and *one-versus-one* (1-v-1) methods. In the following discussion we assume n classes.

UNIVERSITY of the WESTERN CAPE
http://etd.uwc.ac.za/

In 1-v-r, n classifiers are trained in total. Each SVM i separates points of class i from points of all other classes. When evaluating an unlabeled datum, the class i of the SVM with the highest output value (prior to calling the signum function) is taken as the point's class.

In 1-v-1, $\frac{n(n-1)}{2}$ classifiers are trained - one for each distinct pair (i, j) where $i \neq j$. During the test phase, a voting mechanism is used in which the unlabeled datum is assigned the class with the highest number of votes.

4.7 Summary

We have given a derivation for the support vector machine for both the linearly separable and the non-linearly separable, "soft-margin" case. We also described how the standard inner-product function can be replaced with a more powerful "kernel" function, provided that the kernel is Mercel admissible. Finally, we suggested how SVMs, which are inherently a binary classifier, can be used for polychotomous classification problems.

UNIVERSITY of the
WESTERN CAPE

http://etd.uwc.ac.za/

Chapter 5

Experimental Results

This chapter presents our original research contributions to the field of automated FACS AU detection. We investigate two issues: the effect of a local versus global segmentation on recognition accuracy, and the suitability of Haar features combined with the Adaboost boosting algorithm [FS99] for facial expression recognition. Before proceeding to describe the individual experiments, we first describe certain preliminary parameters and techniques that are common to all the experiments we conduct.

5.1 Preliminary Parameters and Techniques

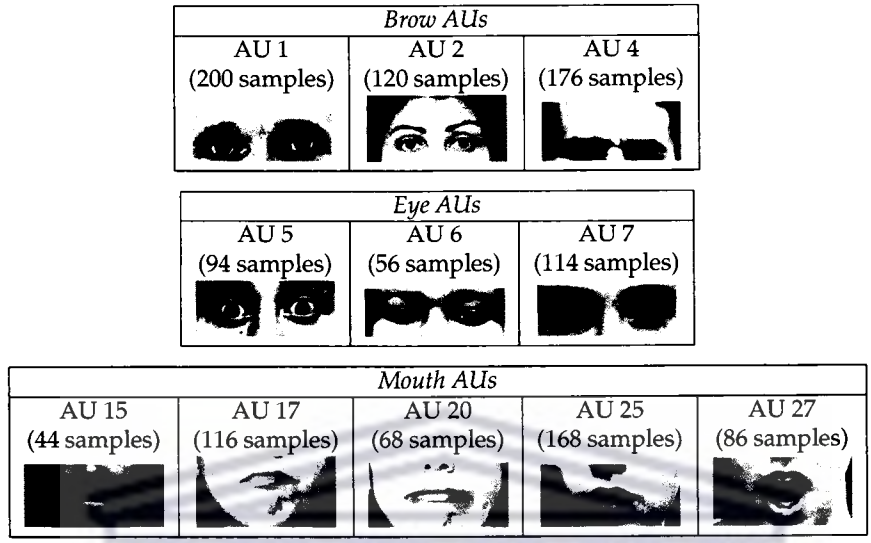
5.1.1 Facial Expression Database

For our experiments we use the Cohn-Kanade AU-Coded Facial Expression Database [KCIT00]. This database contains images of individual human subjects performing a variety of facial expressions. In the public version of this database, 97 different human subjects, ranging from ages to 18 to 30, performed six prototypical expressions: anger, disgust, fear, joy, sadness, and surprise. For each subject and expression, the database contains a sequence of face images beginning with the "neutral" expression (containing no AUs) and ending with the target expression. Certified FACS coders mapped each image sequence in the database to the set of AUs that were exhibited in that sequence. In all the experiments in this chapter, we trained and tested all classifiers on this data subset.

Our experiments required the positions of the eyes and mouth in each image. We used a subset of the Cohn-Kanade Database containing 580 images from 76 human subjects and located the eyes and mouth of each image manually. These locations were used to crop local windows around the eye, brow, and mouth regions.

From each image sequence of each subject, we used the first two images, which contained the "neutral" expression, and the last two images, in which the target expression was most pronounced. For each AU

UNIVERSITY of the WESTERN CAPE
http://etd.uwc.ac.za/



Pictures courtesy of Carnegie Mellon University Automated Face Analysis Group, `http://www-2.cs.cmu.edu/afs/cs/project/face/www/facs.htm`.

Figure 5.1: Classified AUs and Prevalence in Dataset

that we wished to classify, we randomly retrieved from the master database at least 40 images from image sequences containing that AU. Example images for each AU, along with the number of images in our dataset containing that AU, are shown in Figure 5.1.

5.1.2 Image Normalization

Prior to feature extraction and expression classification, each face (original size approximately 200-300 pixels wide) was rotated and scaled (using bilinear interpolation) such that the coordinates of the eyes and mouth were constant over all images. The face width was set to 64 pixels; the inter-ocular distance was set to 24 pixels; and the y-distance between the eyes and mouth was 26 pixels.

5.1.3 AU Classification

Each trained classifier detected the presence or absence of one AU, regardless of whether it occurred in combination. We did not attempt to account for non-additive AU combinations.

5.1.4 Metric of Accuracy

As we discussed in Chapter 3, the percent-correct statistic, despite its prevalence in the literature, is fundamentally flawed. For all our experiments we instead measured accuracy as the area under the Receiver Operator Characteristics (ROC) curve.

http://etd.uwc.ac.za/

5.1.5 Cross Validation

Ten-fold cross-validation was employed to test the generalization performance of each classifier. None of the validation folds contained the same human subject. We calculated mean accuracies (area under the ROC curve) over the ten test folds. When comparing recognition accuracy between two facial segmentations, we performed matched-pairs t-tests over all the folds in order to assess the statistical significance of any difference in mean performance.

5.2 Local versus Global Face Segmentation

The first issue we investigate in this chapter is the effect on AU recognition accuracy of a local versus a global segmentation. Local segmentation of facial images prior to expression classification can significantly reduce the computational cost of both the feature extraction and classification phases. Whether local face analysis improves classification accuracy is an open question: On the one hand, segmenting the image locally reduces the dimensionality of the feature vectors. This may help the classifiers to generalize better during the training phase given the relatively small training sets available for certain AUs. On the other hand, AUs can sometimes affect facial regions outside of their muscle origin. For example, AU 6 (cheek raise), though triggered by a muscle circling the eye, can also accentuate the nasolabial furrow around the mouth [EF78]. Local face analysis might suffer in this case due to the loss of relevant, global appearance information.

In this section we assess the relative performance of the local and global segmentation strategies in terms of AU recognition accuracy. We classify AUs using Gabor filters and linear SVMs - a prominent approach in the FER literature. The experimental setup is described below.

UNIVERSITY of the WESTERN CAPE

5.2.1 Feature Extraction

Gabor features be extracted from each image. Gabor filters were extracted in the following manner: Each segmented image was converted into a Gabor representation using a bank of 40 Gabor filters. Five spatial frequencies (spaced in half-octaves) and eight orientations (spaced at $\pi/8$) were used. Feature vectors were calculated as the complex magnitude of the Gabor jets, and vectors were then sub-sampled by a factor of 16 and normalized to unit length as in [DBH$^+$99].

5.2.2 Segmentations

For the local expression analysis, images were segmented by cropping square regions around the center of the eyes, brows, and mouth. The center of the brows was estimated by shifting the center of the eyes up by one-fourth the inter-eye width. In all cases, the width of each square was 24 pixels.

http://etd.uwc.ac.za/

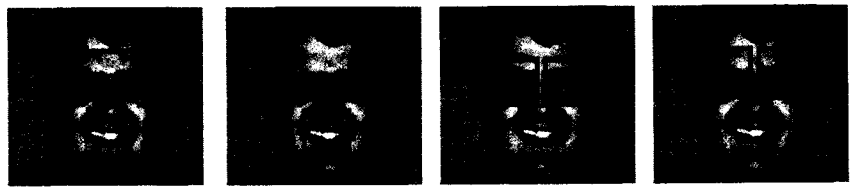


Figure 5.2: The global segmentation (left-most); and the local segmentations of the mouth, eye, and brow regions, respectively (right 3 images).

For global analysis, the face square region was cropped at a width of 64 pixels around (x_c, y_c), where x_c is the x-coordinate of the midpoint between the eyes, and y_c is the y-coordinate of the midpoint between the eyes and mouth. See Figure 5.2 for an illustration of image segmentation.

5.2.3 Results

Recognition accuracies for both classifiers are displayed in Table 5.1. Actual ROC curves for these classifiers are shown in Appendix B. The performance for each AU is reported for both the local and global segmentations; the particular local segmentation depended on the region in which the AU is centered. Whenever a statistically significant difference was identified (for 95% confidence, the p value of the t-test must be less than 0.05), the superior segmentation is listed. When no statistically significant difference was present, an = sign is listed. In some cases (e.g., AU 1), the mean accuracies between segmentations may differ by several percentage points and yet not be statistically significant.

To summarize the results, the local segmentation failed to achieve any consistent and statistically significant advantage over the global segmentation in terms of recognition accuracy. More surprising is that the global segmentation outperformed the local segmentation both for AU 6 in particular and on average.

5.2.4 Discussion

We view two factors as possibly responsible for the statistically indistinguishable, and sometimes even significantly superior performance of the global segmentation relative to the local strategy. The first is that certain AUs may affect regions of the face outside of the AUs' muscular origin (see Section 5.2), and therefore the global segmentation may profit from this non-local appearance information. The second is that, due to the high degree of AU correlation in the Cohn-Kanade database, one AU in one face region may be predictive of another AU elsewhere in the face.

Inter-AU Correlation

Some AUs are easier to detect than others, both by humans and, as witnessed by the results of Table 5.1, by computerized classification. Suppose now that AU i were more difficult to classify than AU j: If it

http://etd.uwc.ac.za/

Table 5.1: Cross-validation recognition accuracies (area under the ROC curve) for all AUs using support vector machines and Gabor features.

Local to Global Comparison

	Segmentation		
AU #	Local	Global	Best
Brow AUs			
1	89.97	96.43	=
2	94.58	95.17	=
4	93.20	97.04	=
Eye AUs			
5	98.48	95.47	=
6	89.71	96.12	Global
7	98.53	98.64	=
Mouth AUs			
15	97.95	97.56	=
17	93.29	95.90	=
20	97.29	96.49	=
25	98.92	98.52	=
27	99.54	99.83	=
Avg	**95.59**	**97.01**	

were known that AU i were perfectly correlated with another AU j ($\rho_{ij} = 1$), then a classifier for AU i could attempt to classify instead AU j, and then output the same result for AU i. Note that the *global* segmentation could benefit from this correlation even if AUs i and j occur in different parts of the face. A *local* segmentation strategy, on the other hand, would be unable to observe AU j's appearance changes on the face (since they would lie outside AU i's local segmentation) and thus would not profit from this correlation.

This hypothesis is supported by the matrix of inter-AU correlations over our data subset given in Table 5.2. Correlation coefficients over the entire Cohn-Kanade database are similar. We considered the correlation between AUs i and j to be high if $|\rho_{ij}| \geq 0.60$; the corresponding entries are shown in bold. Notice how AUs in one region of the face may be highly correlated with AUs in a different region. In particular, AU 1 is highly correlated with AU 25, and AU 2 is highly correlated with both AU 25 and AU 27.

In order to test the effect of inter-AU correlation on recognition performance, we performed the following experiment: To every feature vector of both the global and local segmentations, we appended the classification label $\text{au}_i \in \{0, 1\}$ of every AU *except* the one to be classified. For instance, for a classifier for AU 1, we augmented the standard Gabor feature vector $\mathcal{F}_n$ of each classified image n to be:

$$\mathcal{F}'_n = \mathcal{F}_n \,.\, (\text{au}_2, \text{au}_4, \text{au}_5, \text{au}_6, \text{au}_7, \text{au}_9, \text{au}_{10}, \ldots)$$

where the dot . represents vector concatenation, and au_i is the actual classification label for AU i in image n. Each feature vector was thus given perfect knowledge of the presence or absence of *every* other AU (not

UNIVERSITY of the WESTERN CAPE
http://etd.uwc.ac.za/

Table 5.2: Inter-AU correlation matrix. Entries ρ_{ij} where $|\rho_{ij}| \geq 0.60$ (other than self-correlation) are marked in bold.

	Brow AUs			Eye AUs			Mouth AUs				
AU #	1	2	4	5	6	7	15	17	20	25	27
1	1.00	**0.69**	0.26	0.59	-0.08	-0.07	0.39	0.15	0.38	**0.69**	0.58
2	**0.69**	1.00	-0.18	**0.76**	-0.13	-0.23	0.02	-0.08	0.01	**0.65**	**0.83**
4	0.26	-0.18	1.00	-0.13	0.46	**0.73**	0.26	**0.61**	0.45	0.17	-0.23
5	0.59	**0.76**	-0.13	1.00	-0.08	-0.15	-0.09	-0.15	0.05	**0.65**	**0.76**
6	-0.08	-0.13	0.46	-0.08	1.00	0.53	-0.09	0.26	0.18	0.11	-0.13
7	-0.07	-0.23	**0.73**	-0.15	0.53	1.00	-0.08	0.43	0.31	0.14	-0.21
15	0.39	0.02	0.26	-0.09	-0.09	-0.08	1.00	0.54	-0.10	-0.15	-0.08
17	0.15	-0.08	**0.61**	-0.15	0.26	0.43	0.54	1.00	-0.12	-0.26	-0.21
20	0.38	0.01	0.45	0.05	0.18	0.31	-0.10	-0.12	1.00	0.54	-0.12
25	**0.69**	**0.65**	0.17	**0.65**	0.11	0.14	-0.15	-0.26	0.54	1.00	**0.65**
27	0.58	**0.83**	-0.23	**0.76**	-0.13	-0.21	-0.08	-0.21	-0.12	**0.65**	1.00

Table 5.3: Recognition accuracies (area under the ROC curve) with SVMs and Gabor features for the local and global segmentations, using both the standard and augmented feature vectors (with inter-AU correlation information).

	Feature Vector			
AU #	Standard Local	Augmented Local	Standard Global	Augmented Global
1	89.97	95.13	96.43	96.82
2	94.58	97.40	95.17	95.51
4	93.20	95.09	97.04	97.47
6	89.71	90.73	96.12	96.09
7	98.53	98.86	98.64	98.83
17	93.29	97.56	95.90	96.24
27	99.54	98.59	99.83	99.83

just the 11 AUs we classified). If the correlation effect was truly responsible for the global segmentation strategy's superior classification performance, then there should be no statistically significant difference in recognition accuracies of the local and global strategies when using the modified feature vectors.

We modified the feature vectors for AUs 1, 2, 4, 6, 17, and 27 - all the AUs for which the global segmentation had shown superior performance when SVMs were used. Classification results are displayed in Table 5.3.

The local segmentations for AUs 1, 2, 4, 6, and 17 all benefited by at least 1% accuracy from the appended correlation information. The corresponding global segmentations, on the other hand, did not improve substantially despite the added correlation data. These results suggest that the correlation information was already present in the global segmentation but not in the local segmentation. It also shows how the performance of an AU classifier can be "improved" by supplying information about other, related AUs. The problem with this "improvement" is that, if the same classifier is applied to a different database with

http://etd.uwc.ac.za/

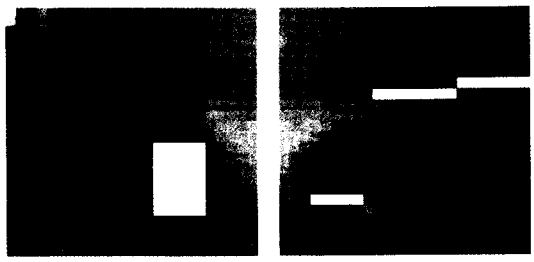


Figure 5.3: Examples of Haar features (selected by Adaboost for AU 1) used for AU classification in our system.

different AU correlations, the accuracy may fall drastically.

Given the strong correlations within the Cohn-Kanade dataset, a conclusive answer to our original question - whether a local segmentation yields higher accuracy - may not yet be attainable. To investigate this issue effectively, one first needs a larger expression database in which AUs occur singly, or at least in which the correlations between them are weaker.

5.3 Haar Features and Adaboost for AU Recognition

The second significant research contribution of this thesis to the FER literature is a study of the effectiveness of using Haar features and Adaboost for FACS AU recognition. Recent computer vision research has demonstrated that the Haar wavelet is a powerful image feature for object recognition. In this study we use the same kinds of Haar-like features deployed in the Viola-Jones face detector [VJ04]. Examples of these features are shown in Figure 5.3.

Because the number of such features in a face image is large, we use Adaboost both to select a subset of these features and to perform the actual classification. We compare this Haar+Adaboost approach to the popular Gabor+SVM method. Part of our source code for this experiment was based on the code of [WRM04]. The next sections describe this comparative experiment in greater detail.

5.3.1 Feature Selection

The set of Haar features used by Viola and Jones is many times over-complete. While this allows very fine-grained inspection of an image, it also increases the training time and can reduce generalization performance. For these reasons, the Viola-Jones approach uses the Adaboost boosting algorithm as a means of feature selection by constructing a weak classifier out of each Haar feature. Specifically, a threshold-based binary classifier is created from each Haar feature so that the weighted training error is minimized. During each round of boosting, the single best weak classifier for that round is chosen (corresponding to a particular Haar feature). The final result of boosting is a strong classifier whose output is computed as a

http://etd.uwc.ac.za/

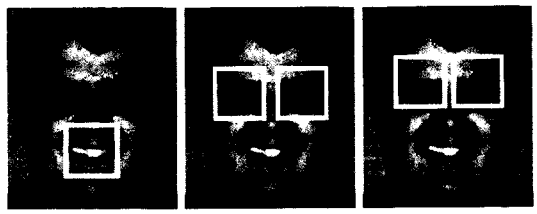


Figure 5.4: The local face regions of the mouth (left), eye (middle), and brow (right) regions from which features were selected for each AU classifier.

thresholded linear combination of the weak classifiers. The Viola-Jones face detector has demonstrated that this classification method is both fast and effective for object recognition.

5.3.2 Face Region Segmentation

In order to reduce the length of time necessary for the lengthy Adaboost-based feature selection process, we designed our system to recognize AUs from local subregions of the face instead of the whole face window. Performing this segmentation greatly reduces the size of the set of all possible features from which a few can be selected. Local subregions were set to squares of width 24 pixels around the mouth, each eye, and each brow. Figure 5.4 shows the face regions that were cropped from each image.

5.3.3 Feature Extraction

The Viola-Jones "integral image" method (see [VJ04] for details) was used to extract features from images. For each AU, we used Adaboost to select 500 Haar features for classification. Features for classifying mouth AUs were selected only from the corresponding mouth region. Features for the eye AUs were extracted both from the left and the right eye regions; a similar approach was taken for the brow AU classifiers. Figure 5.3 shows examples of Haar features that were actually chosen for AU recognition during the feature selection process.

5.3.4 Classification

Each feature in the set of 500 Haar features for each AU was fed to the corresponding weak classifier, which outputs a label in $\{-1, 1\}$. The Adaboost-based strong classifier then outputs the final classification label for that AU based on whether the weighted sum of the weak classifiers' outputs exceeds the strong classifier's threshold. See Freund and Schapire [FS99] for details.

http://etd.uwc.ac.za/

Table 5.4: Recognition accuracy (area under the ROC curve) for the Gabor+SVM method and the Haar+Adaboost method. The Haar+Adaboost approach performed well for the eye and brow AUs but poorly for the mouth AUs.

Haar+Adaboost (H+A) versus Gabor+SVMs (G+S)

	Method		
AU #	Gabor+SVM	Haar+Adaboost	Best
Brow AUs			
1	89.97	89.72	=
2	94.58	97.67	H+A
4	93.20	90.34	G+S
Eye AUs			
5	98.48	98.10	=
6	89.71	92.91	=
7	98.53	96.11	G+S
Mouth AUs			
15	97.95	53.62	G+S
17	93.29	60.51	G+S
20	97.29	81.04	G+S
25	98.92	66.53	G+S
27	99.54	82.81	G+S

5.3.5 Results

Accuracy statistics measured as area under the ROC curve are given in Table 5.4. Actual ROC curves are presented in Appendix B. As shown in the table, the Haar+Adaboost method achieved comparable accuracy to the Gabor+SVM method for AUs of the eye and brow regions. Interestingly, it performed very poorly for AUs of the mouth. We view two factors are possibly responsible for this performance difference: First, it is possible that the Haar+Adaboost combination is only effective when many training examples are available. For example, only 44 training examples were available for AU 15, which is the AU on which the Haar+Adaboost method performed the worst (53.62%). The small number of training samples would not, however, explain why the classifier AU 17, with only 68 examples, performed relatively well. The second possible explanation for the poor performance on the mouth region is that mouth AUs exhibit greater variability in the location of skin bulges and wrinkles than do the upper-face AUs [Bar]. It is possible that the Gabor filters, since their Gaussian component implicitly performs smoothing, are less sensitive to this variation.

5.3.6 Theoretical Performance Analysis

Besides comparing the Gabor+SVM and Haar+Adaboost methods in terms of accuracy, we also compare them in terms of speed. We perform a theoretical analysis of run-time performance in this section and an empirical one in the next. We consider both feature extraction and classification.

UNIVERSITY of the WESTERN CAPE
http://etd.uwc.ac.za/

Feature Extraction

The main advantage of Haar+Adaboost over Gabor+SVMs is speed. The steps involved in extracting Gabor features from a face image are shown below. Algorithmic complexity is measured as a function of the number of image pixels (N). Note that FFT stands for Fast Fourier Transform.

1. Transform the image using the FFT: $O(N \log N)$
2. For each filter
   (a) Multiply the transformed image by the pre-computed filter: $O(N)$
   (b) Inverse-transform the result using the Inverse FFT: $O(N \log N)$

The number of filters P (in our system, 40) is a constant that does not depend on N. Thus, the computational complexity of this algorithm is $O(N \log N)$.

The extraction of Haar features, on the other, is far less expensive. The necessary steps are as follows:

1. Calculate the integral image: $O(N)$
2. For each of M features
   (a) Extract each feature from the integral image: $O(1)$

The number of extracted features M (in our system, 500) is a constant that does not depend on N. Hence, the total time complexity for Haar feature extraction is $O(N)$, which is considerably less than $O(N \log N)$.

An additional performance advantage offered by the Haar method is that adding an additional feature to the extracted set increases the running time only by a constant number of CPU instructions. Adding another filter to a filter bank, on the other hand, requires an additional $O(N \log N)$ machine instructions.

Classification

We compare the algorithmic complexity of classification in terms of the number of extracted features M. Classification with the Haar+Adaboost method consists of the following algorithm:

1. Set T to 0
2. For each i of M features
   (a) Determine if feature i exceeds threshold i
   (b) If yes, then add α_i to T.
3. Return 1 if T is at least $\frac{1}{2}\sum_i \alpha_i$ (the total threshold); return 0 otherwise.

http://etd.uwc.ac.za/

For M features, the algorithm is thus $O(M)$.

Classification with a linear SVM is similar in algorithmic complexity to Adaboost. For a linear SVM, the separating hyperplane can be calculated offline based on the support vectors and the corresponding Lagrange multipliers; classification then requires only one inner product of the test point with the hyperplane. For M features (and thus M vector components), classification requires $O(M)$ operations, which is equivalent to the Adaboost method. It should be noted, however, that the popular `libsvm` library [CL01], which we used in this thesis, implements linear kernels in $O(Q*M)$ time, where Q is the number of support vectors. [1] Thus, in our software implementation, the Haar+Adaboost method performs much more quickly than even the linear SVM.

Classification with a non-linear kernel SVM is generally slower. For higher-dimensional kernels, a test point must be multiplied with each of Q support vectors, resulting in Q inner products and thus $O(Q*M)$ operations. The cost of computing each inner product is also higher because of the kernel function itself, which may be computationally expensive. Certain methods such as [DeC02] do exist, however, which may serve to partially reduce the computational cost of SVM classification.

5.3.7 Empirical Performance Analysis

Feature Extraction

In addition to the theoretical analysis of the two feature types, we also performed an empirical study by extracting features from sample input images. For the FFT implementation we used the popular library *FFTW* (the Fastest Fourier Transform in the West) [FJ05]. For basic image manipulation, we employed the simple and efficient *TiP* library (Tools for Image Processing) [GGJ].

We performed experiments for two different image sizes: 24x24 and 64x64. The smaller window size is suitable for classifying facial expression from individual local regions of the face (e.g., mouth); the larger window size is appropriate when analyzing the face as a whole. For Haar feature extraction, 500 selected features were computed. For Gabor features, we applied a standard filter bank of 5 frequencies and 8 orientations and extracted Gabor responses at all points in each filtered image. The execution times were measured on a Pentium 4 1.8 GHz machine and averaged over 1000 rounds of extraction; results are shown in Table 5.5. The results show that, for 24x24 images, Haar feature extraction is approximately 80 times faster than Gabor feature extraction. For 64x64 images, the Haar features can be extracted nearly 160 times more quickly.

[1] The software implementation is simpler if *all* kernels - including the linear kernel - are implemented as the sum of Q inner products.

http://etd.uwc.ac.za/

Full Gabor versus Selected Haar Extraction Times

Feature Type	Resolution	Extraction Time
Haar	24x24	0.11msec
	64x64	0.31msec
Gabor	24x24	8.8msec
	64x64	49.3msec

Table 5.5: Execution times of feature extraction for Gabor features versus selected Haar features.

Adaboost versus SVM Classification Times

Classifier	Classification Time
Adaboost	0.02msec
SVM (Linear)	21.17msec
SVM (RBF)	93.97msec

Table 5.6: Execution times of classification for an Adaboost strong classifier versus a linear SVM.

Classification

Using the same parameters as in Section 5.3.7, we compared empirically the running times of the boosted classifier of the Haar+Adaboost method with the SVM of the Gabor+SVM method. We used the `libsvm` library [CL01] for the SVM implementation. Execution times are shown in Table 5.6. As illustrated by the running times, the Adaboost strong classifier is 3 orders of magnitude faster than the SVM.

5.4 Summary

This chapter has investigated two important issues in the field of automatic FER: First, we compared local to global segmentation of facial images in terms of accuracy when recognizing FACS AUs. As a follow-up, we also studied the effect of inter-AU correlation within facial images on the recognition accuracies of various AUs. Our results show that this correlation effect can impact recognition rates significantly. Such correlation effects may be of little consequence when recognizing prototypical expressions, in which high AU correlation is natural. They are of considerable importance, however, when analyzing single AUs, as recognition rates will appear misleadingly high. We would thus like to underline the importance of establishing a large, publicly available AU database with singly-occurring AUs to facilitate future research.

Second, we compared the popular Gabor+SVM method of AU recognition to the previously untested Haar+Adaboost approach. Accuracy with the Haar+Adaboost approach was high for the eye and brow AUs, but low for the mouth AUs. We discussed probable causes for these findings. Finally, we performed a performance comparison of these two methods. Experimental results show that Haar+Adaboost operates several orders of magnitude more quickly.

http://etd.uwc.ac.za/

Chapter 6

Real-Time SASL Video Analysis

In this section we apply the AU recognition system we developed in Chapter 5 to the real-world problem of recognizing from video some of the expressions that occur in South African Sign Language. This task is extremely challenging for contemporary FER systems because of the significant out-of-plane rotation that occurs in natural human conversation. The fact that most publicly available AU training data are taken from posed *prototypical* human expressions in strictly controlled laboratory environments instead of from natural human behavior makes the challenge even more difficult. Nevertheless, we hope that, by analyzing the performance of our system on this task, we may gain insight into how FER systems can be improved to facilitate automated signed language recognition.

In any effort to design an automated system designed to recognize the facial expressions of a signed language, it is important to understand how these expressions are used linguistically. Facial expressions, along with movements of the head and upper torso, constitute the set of *non-manual* communication channels involved in signed languages. In the following subsections we discuss the roles that non-manual actions of signed languages can play and show example expressions from our target language: South African Sign Language (SASL). Because linguistic research on SASL is so limited, however, we will illustrate certain linguistic concepts common to signed languages with examples from American Sign Language (ASL).

6.1 Uses of Facial Expressions in Signed Languages

Just as in spoken languages, facial expressions can be used in signed languages to convey the affective state of the speaker. In ASL, for example, the emotional states "sad", and "smile" are signified by producing the corresponding prototypical expression in the face (p. 371, [RMB90]). Unlike in spoken languages, facial expressions in signed languages also provide crucial lexical, adverbial, and syntactic functionality that extends far beyond the affective expressions mentioned above. We elaborate on and discuss the importance

http://etd.uwc.ac.za/

of each category of facial expression usage in the sections below.

6.1.1 Lexical Functionality

Some signs are either obligatorily or optionally accompanied by non-manual actions. In contrast to the non-manual actions with a syntactic function, lexical non-manual actions are articulated only for the duration of the accompanying manual gesture - they do not extend over neighboring parts of the sentence. The ASL sign for "give in", for example, is accompanied in the face by dropping the jaw for the duration of the hand gesture (p. 16, [Lid80]). Another example in ASL is the sign for "not yet," which requires that the tongue protrude slightly. Without the accompanying facial action, the sign would instead mean "late" (p. 40, [NKM+99]).

6.1.2 Adverbial Functionality

Non-manual actions can also serve an adverbial role in signed languages. Such actions are not required for the articulation of a particular sign, but they may modify the intended meaning. In ASL, for example, the sentence "the boy is writing a letter" can be changed to "the boy is writing a letter carelessly" by thrusting the tongue during the manual sign for "write" (p. 371, [RMB90]). As with lexical facial expressions, adverbial expressions are executed only for the duration of the single sign that it modifies (p. 43, [NKM+99]).

6.1.3 Syntactic Functionality

In addition to their role in articulating single signs and adverbs, non-manual components of signed language also provide crucial syntactic functionality. Several categories of such syntactic use of facial expressions exist, including *topics*, *relative clauses*, *conditionals*, *negations*, and *questions*. We briefly describe each category below.

Negations

One simple but important syntactic service that facial expressions provide is the negation of clauses. In ASL, for example, although a manual gesture for "not" also exists, the non-manual action - consisting of furrowed eyebrows and a shaking head - is obligatory (p. 45, [NKM+99]).

Conditionals

A *conditional* is an *if-then* structure describing one state or event that is conditional on another. An example of a conditional is, "If you insult George, then Jane will be angry." In ASL, conditionals are signified by obligatory non-manual features including facial expression, eye movement, and head orientation. Without

UNIVERSITY of the WESTERN CAPE
http://etd.uwc.ac.za/

these accompanying non-manuals, the example sentence above would reduce to two simple propositions: "You insulted George, and Jane got angry" (p. 372, [RMB90]).

Relative Clauses

A relative clause is a "dependent clause introduced by a relative pronoun" [Her00]. For instance, in the sentence, "the person who bought the mop is frugal", the relative clause "who bought the mop" serves to specify which person is frugal. In ASL, a relative clause is signified by raising the eyebrows, tilting the head backward, and raising the upper lip during the manual articulation of the clause (p. 22, [Lid80]).

Questions

In ASL, non-manual signs are used both for *yes-no* and *wh*-questions. As expected, yes-no questions are those which ask the listener for "yes" or "no" response. In ASL, yes-no questions require that the eyebrows be raised and that both the head and body be projected forward (p. 168, [Lid80]). *Wh*-questions correspond to interrogatory pronouns such as "who" and "what", i.e., questions whose English counterparts begin with the letters "wh". *Wh*-questions must be accompanied by "furrowed brows, squinted eyes, and a slight side-to-side head shake" (p. 111, [NKM+99]). These facial expressions are used for the same purpose in SASL.

Topics

A *topic* is an element of "old information about which some comment will be made" (p. 22, [Lid80]), and they are used extensively in signed languages. An example of a topicalized sentence translated from ASL into English is: "Chris - Jessie likes him." In this sentence, "Chris" is the topic; the fact that Jessie likes him is the appended comment. In English, the sentence would have to read "As for Chris, Jessie likes him" in order to preserve grammaticality. In ASL, however, this introduction is implicit. ASL utilizes both "moved" and the more complex "base-generated" topics. Each type of topic is denoted by its own set of eye, eyebrow, and head movements (p. 50, [NKM+99]).

6.2 Expression Intensity

So far in our discussion of the linguistic use of facial expressions in signed languages we have not mentioned expression *intensity*. The intensity carries important information that may benefit automatic linguistic analysis. Whereas the non-manual components of *lexical* and *adverbial* signs may appear with uniform intensity and for short duration, *syntactic* facial expressions typically reach an apex intensity and then gradually diminish. The point of highest intensity corresponds to the "node of origin" (p. 45, [NKM+99]) with

http://etd.uwc.ac.za/

which the non-manual is associated. In ASL, for example, a sentence is negated both by using a hand gesture and by simultaneously articulating a negative facial expression. The node of origin in this case is the negative hand gesture, and the corresponding facial expression reaches its apex at that same moment (p. 45, [NKM+99]). While the pilot project of this chapter attempts only to recognize expressions as present/absent, future systems will need to estimate the expression intensity as well.

6.3 Implications for Automatic Translation

Based on the usage of facial expressions in signed languages as described above, we can highlight two main results that may influence the design of a signed language recognition system:

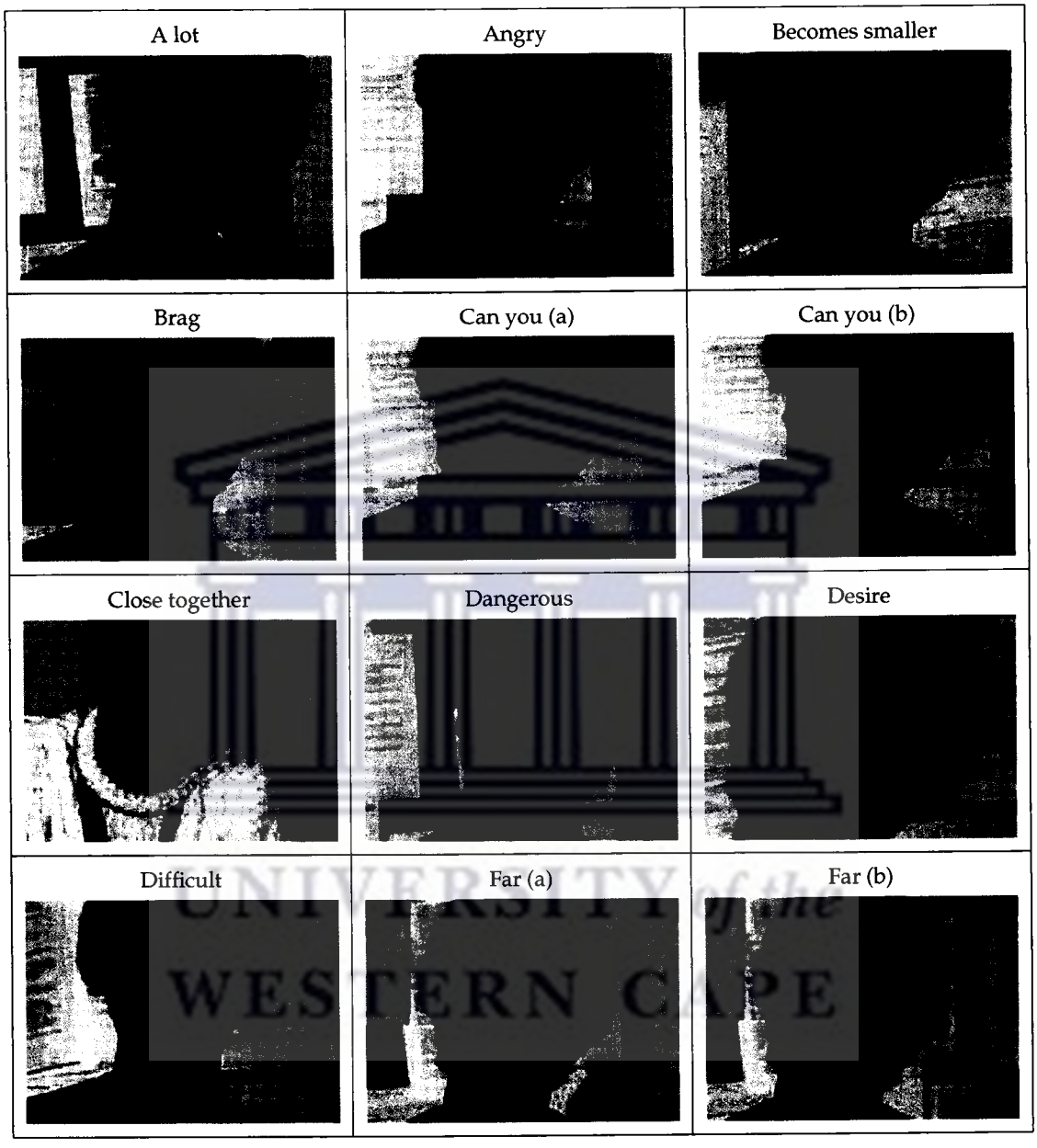
- Facial expressions that perform a lexical or adverbial function take place over a short duration. It is thus conceivable that recognizing only the apex of expression intensity would be sufficient to enable effective linguistic analysis.
- Syntactic facial expressions are articulated over a longer time span. Some method of smoothing of the predicted expression intensity may thus be appropriate in order to estimate accurately the onset and offset of syntactic facial expressions.

6.4 Recognizing Facial Expressions of SASL

For the pilot study of SASL recognition in this thesis, we employed the assistance of SASL speaker David Petro from the Cape Town Bastion Center for the Deaf. Petro is deaf and, though knowledgeable in English, communicates primarily in SASL. He is also a SASL instructor at the Bastion Center and thus knowledgeable in SASL grammar and usage.

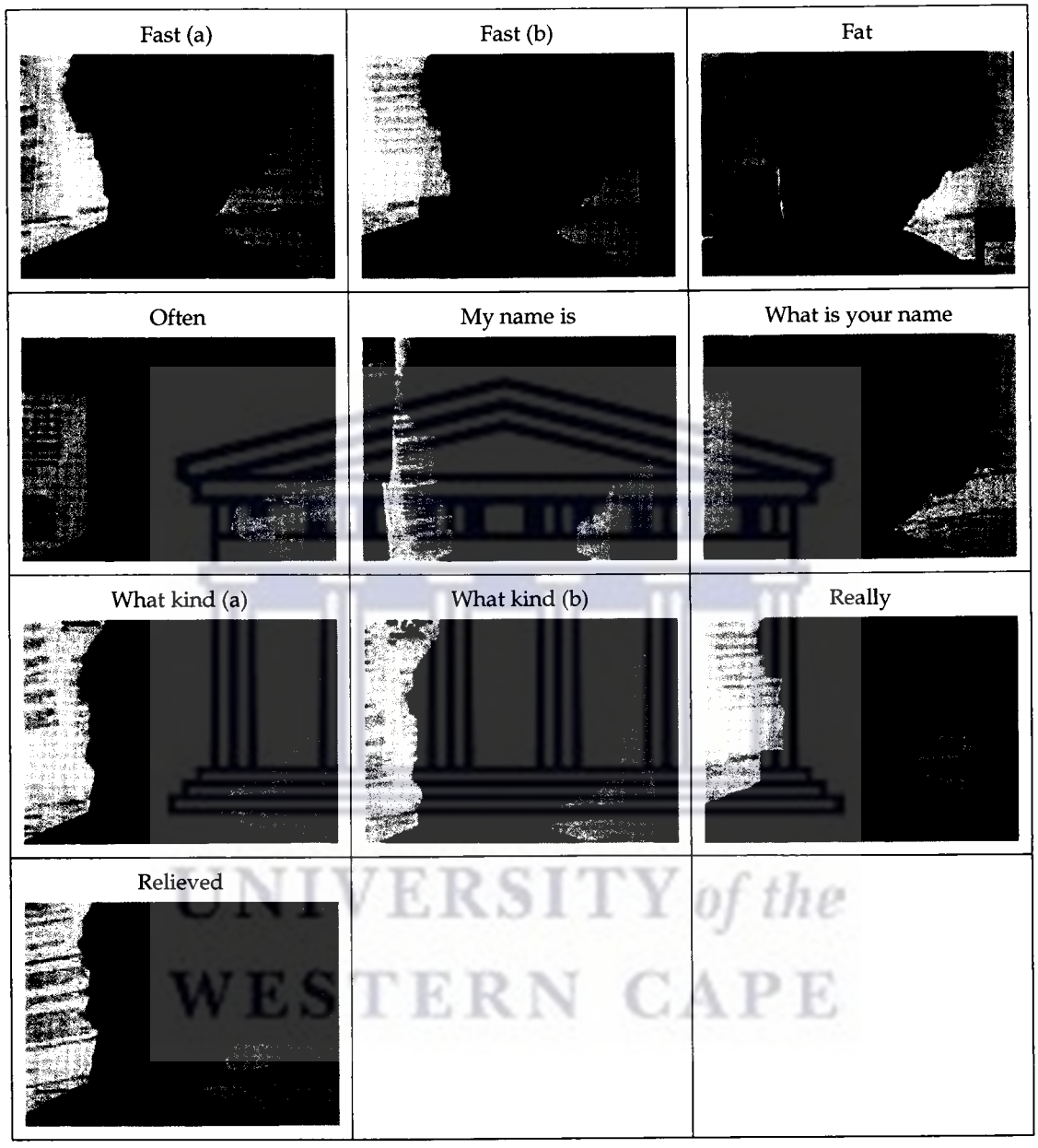
Together with Mr. Petro, we identified 18 nouns, adjectives, adverbs, and phrases from SASL which occur commonly in conversation and which require facial expressions for their articulation. Three of these SASL expressions - *fast*, *far*, and *what kind* - contain two parts (a) and (b) which must be performed by the speaker in succession. We asked Mr. Petro to perform each of these 18 expressions in front of a digital camera; the photographs are displayed in the table below. In the sections thereafter, we describe our FACS-based approach to recognizing these SASL expressions automatically.

http://etd.uwc.ac.za/



UNIVERSITY of the
WESTERN CAPE

http://etd.uwc.ac.za/



6.4.1 Test Case: A Simple Story

Given this set of facial expressions, we composed a simple story which two deaf SASL signers from the Bastion Center, David Petro and Carmen Fredericks, then narrated in front of a video camera. The story was written not to achieve literary greatness, but rather to elicit most of the SASL expressions that were photographed and analyzed of David Petro. The signers were requested to keep their faces in clear and unobstructed view of the camera whenever possible, and to sign the facial expressions clearly and deliberately. However, each participant exercised some freedom in narrating the text; hence, not all expressions in our story were actually signed, nor do the expressions appear in exactly the same order as in the English

UNIVERSITY of the WESTERN CAPE
http://etd.uwc.ac.za/

Table 6.1: Full AU decomposition for SASL expressions. Each number represents an AU; each letter following the AU number specifies the intensity (A through E); and an L or R preceding the AU number specifies an asymmetric action on the left or right side of the face, respectively.

SASL Expression	AU Decomposition
a lot	1E 2E L4A 5E 18D 34B 55C
angry	1A 2A 23C 25B 38A 52B 54A
becomes smaller	1B 2A L4B 18C 55B
brag	1B L2C R2A 4A 7B 20D 25A 53B 58D
can you (a)	1D 2D 4C 5C 25B 26E 55C
can you (b)	1D 2D 4D 5B 34D 55D
close together	1B 2B 4D 24D 38A 55C 57B
dangerous	1D 2D 4B 5C 7A 16C 22E 25D 55C
desire	1C 4C L6A 16B 25C 53C 55C
difficult	4D L6B 7D 20D 58C
far (a)	4D L6B 10B 25B 32C 38B 53D 55B
far (b)	1A 4C 25B 26D 53D 55B
fast (a)	1C 2C 4A 5A 23B 38B 55B
fast (b)	1C 2C L4B 5A 18B 25B 26B 55C 57B
fat	1C 2C 4B 5A 34D 39B 55C
my name is	1C 2B 4B 5C 7A 52B 53B 55C
often	1A 4D 7D 24C S26A 55D
really	1C 2C 4B 5A 10B 17B 25B 32B 55C
relieved	25A 53B 55C
what is your name	4D 7E 55D 57C
what kind (a)	4B 7C 10B 25A 53C 55D
what kind (b)	4B 25B 26C 53E 55D

text. The story text appears below, with the key SASL expressions in italics:

> Hello, *my name is* _____. I want to tell you a story about my day. This morning I woke up late. I was scared that my boss would be *angry* if I came late to work. My house is *far* from my office, so I had to drive *fast* to save time. I *often* drive fast to work, but today it was *dangerous* because the roads were wet.
>
> I stopped at a traffic light. Beside me was a very *fat* woman whom I had long *desire*d. Our cars were *close together*, and I shouted to her, "*what is your name*?" She said, "Priscilla." She then started to *brag* about how *fast* her car was. I asked her, "*what kind* of car is it?" She said it was a Porsche.
>
> We decided to race to the next traffic light. I drove as *fast* as I could, but the rain made it *difficult* to see. Her car was *fast*er, and she won the race. I felt very embarrassed. But at least I was not late to work - I arrived two minutes early. I was very *relieved*.

6.5 Approach

In this thesis, we endeavor to recognize the SASL expressions in the narrative above using FACS as an intermediary framework. As the first step towards this goal, the photographs of Petro were FACS-coded by expert FACS consultant Dr. Erika Rosenberg for both the presence and degree of the exhibited AUs. The full AU decomposition of each expression, including intensity values, is listed in Table 6.1. Each AU which appeared asymmetrically in only the left or right half of the face is preceded with "L" or "R", respectively.

http://etd.uwc.ac.za/

SASL Expression	FACS Action Units																	
	1	2	4	5	6	7	10	15	16	17	20	23	24	25	26	27	38	39
a lot	✓	✓	✓	✓														
angry	✓	✓										✓		✓			✓	
becomes smaller	✓	✓	✓															
brag	✓	✓	✓			✓					✓			✓				
can you (a)	✓	✓	✓	✓										✓	✓			
can you (b)	✓	✓	✓	✓														
close together	✓	✓	✓										✓				✓	
dangerous	✓	✓	✓	✓		✓			✓					✓				
desire	✓		✓		✓				✓					✓				
difficult			✓		✓	✓					✓							
far (a)			✓		✓		✓							✓			✓	
far (b)	✓		✓											✓	✓			
fast (a)	✓	✓	✓	✓								✓					✓	
fast (b)	✓	✓	✓	✓										✓	✓			
fat	✓	✓	✓	✓														✓
my name is	✓	✓	✓	✓		✓												
often	✓		✓			✓							✓		✓			
really	✓	✓	✓	✓			✓			✓				✓				
relieved														✓				
what is your name			✓			✓												
what kind (a)			✓			✓	✓							✓				
what kind (b)			✓											✓	✓			

Table 6.2: AU mappings for each of the sample SASL expressions. Note that the expressions may also contain other AUs not shown in this table - we list only those AUs for which we trained a classifier.

Table 6.2 contains similar information for each expression. In contrast to Table 6.1, however, this table decomposes the expressions only in terms of the 18 AUs for which sufficient training examples existed in our AU training set. This table confirms that our set of AU classifiers is rich enough to differentiate each of the selected facial expressions of SASL even when expression intensity is not considered. In both the approaches to recognizing SASL expressions that we describe below, we represent each SASL expression as a vector

$$\mathbf{x} = (au_1, au_2, au_4, au_5, au_6, au_7, au_{10}, au_{15}, au_{16}, au_{17}, au_{20}, au_{23}, au_{24}, au_{25}, au_{26}, au_{27}, au_{38}, au_{39})$$

where each $au_i \in \{0,1\}$. Thus, each SASL expression vector $\mathbf{x} \in \{0,1\}^{18}$ stores the set of AUs it comprises, as described in Table 6.2. The SASL expression for "fat", for example, is represented by $\mathbf{x}_{fat} = (1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1)$. Given the expression vector $\mathbf{x}$ for each SASL expression, and given the AU detection vector $\mathbf{v}$ containing the AUs present in a particular input image, we attempted to recognize SASL expressions from the frames in a video sequence.

Recognizing each SASL expression can thus be decomposed into first detecting AUs, and then mapping the exhibited AUs to the SASL expression that triggered them. In the sections below, we describe two implementations of this technique: one in which the set of AUs must match the SASL expression exactly (Exact Matching), and one in which the best-possible match (assessed using Cosine Similarity) is used for

http://etd.uwc.ac.za/

SASL expression prediction.

6.5.1 Method 1: Exact Matching

In our first approach to SASL expression recognition, we consider an expression to be present if and only if the vector $\mathbf{v}$ of AUs present in the input image exactly matches the expression vector $\mathbf{x}_i$ for SASL expression i. This is a very strict matching condition, and it means that an expression consisting of AUs $\{1, 2, 4, 24, 38\}$ ("close together") will not be recognized if the face image contains only AUs $\{2, 4, 24, 38\}$, nor will it be recognized if the image contains $\{1, 2, 4, 5, 24, 38\}$ (AU 5 is superfluous).

6.5.2 Method 2: Cosine Similarity

In our second approach to FACS-based SASL expression recognition, we considered a SASL expression $\mathbf{x}_i$ to be present if $\cos(\angle(\mathbf{x}_i, \mathbf{v})) = \frac{\mathbf{x}_i \cdot \mathbf{v}}{\|\mathbf{x}_i\|\|\mathbf{v}\|} \geq \tau_i$, where $\mathbf{v}$ is the vector of AUs detected in the input image and τ_i is the expression-specific recognition threshold (determined empirically). This approach allows an expression to be recognized even if, say, one or two AUs of a particular SASL expression are absent from the face, or if a few extra AUs not in $\mathbf{x}_i$ are contained in the face image. Since $\cos(\angle(\mathbf{x}_i, \mathbf{v})) = 1 \iff \mathbf{x}_i = \mathbf{v}$, the Exact Matching method of the previous subsection emerges as a special case of the Cosine Similarity method.

The threshold τ_i would need to be determined based on a training set of SASL video data. Since our training data were so limited, however, we employed a modified version of this algorithm in which each video frame was mapped to the SASL expression $\mathbf{x}_i$ for which the cosine similarity metric was highest. This means that one SASL expression will be detected for every frame in the video sequence - a situation that is admittedly improbable - but it also increases the chance that our expression recognizer will output the correct expression for the frames that do contain a SASL expression.

6.6 System Design

Given the two methods of SASL expression mapping of the previous section, we now describe our SASL recognition pipeline from start to finish. We used two alternative methods from the previous chapter for AU recognition: Gabor+SVMs and Haar+Adaboost. The former has the advantage of higher overall accuracy over all AUs, whereas the latter is advantageous in its speed. Note that, on our computer system, only the Haar+Adaboost method was sufficiently fast to enable real-time performance.

- **Input**: Video frame containing face.
- **Desired output**: Predicted SASL expression.

http://etd.uwc.ac.za/

Table 6.3:
Apex Frames of Petro's SASL Expressions

Expression	Time (s)	In-plane Rotation	Out-of-plane Rotation	Occlusion
My name is	0.2			✓
Angry	18.7	✓		
Fast (a)	22.0		✓	
Fast (b)	22.1		✓	
Close together	30.4	✓		
Fat	34.8		✓	✓
Desire	40.4	✓		
What is your name	44.1	✓	✓	
What kind (a)	54.1		✓	✓
What kind (b)	54.4			
Far (a)	69.3		✓	✓
Far (b)	69.5		✓	✓
Difficult [to see]	75.7		✓	✓
Relieved	87.1	✓	✓	

- **Procedure**:

  1. *Face detection*: To determine the location of the face within the input image, we used the *Machine Perception Toolbox* `MPISearch` program [FDH+]. This program operates at frame rate and outputs the coordinates of the found face box.
  2. *Face normalization*: Given the location and size of the face (if found), the face is normalized to a standard size.
  3. *AU recognition*: Using either the Gabor+SVM or Haar+Adaboost method, estimate the AUs contained within the input face.
  4. *SASL Expression mapping*: Using either the Exact Matching or Cosine Similarity method, predict the SASL expression of the current frame.

Using the Haar+Adaboost procedure, this pipeline is fully automatic and operates in real time.

6.7 Experiment

Given the two video narratives (by both Petro and Fredericks) of the SASL story listed previously, we measured the accuracy of the SASL expression recognition system described in the previous section. Together with David Petro, we marked the video frames containing the apex of each SASL expression in both videos. The expressions and the times when they occurred are listed in Tables 6.3 and 6.4. Note that, although each SASL narrator was presented exactly with the story we listed above, they exercised some freedom in telling it; hence, not all expressions occurred in the same order for both signers, and some expressions were not

http://etd.uwc.ac.za/

Table 6.4:
Apex Frames of Fredericks' SASL Expressions

Expression	Time (s)	In-plane Rotation	Out-of-plane Rotation	Occlusion
My name is	2.4	✓		✓
Angry	17.0	✓	✓	✓
Far (a)	21.4	✓	✓	
Far (b)	21.9	✓	✓	
Often	26.2	✓	✓	✓
What is your name	37.8	✓	✓	✓
What kind (a)	42.7	✓	✓	✓
What kind (b)	42.8	✓	✓	
Difficult [to see]	51.8		✓	✓
Fast (a)	53.7	✓		
Fast (b)	54.2	✓		
Relieved	62.4	✓	✓	

articulated at all.

Given each apex expression, occurring at some time t, we fed each video frame that occurred within the time window $[t - 0.2s, t + 0.2s]$ to our SASL expression recognizer. Given that the frame rate was 25 fps (every 0.04 s), this amounts to 9 frames per apex.

6.8 Results

The predicted SASL expressions for each signer (Petro and Fredericks) and for each of the Gabor+SVM and Haar+Adaboost methods are displayed in Tables 6.5, 6.6, 6.7, and 6.8 along with the exact cosine similarity values. The predicted expressions using the Cosine Similarity method can be read directly from the table; expressions were recognized with the Exact Matching method only when the cosine similarity value was 1.00, which occurred rarely. Note that, in some video frames, no face was detected at all, and hence that video frame does not appear in the table. Frames whose cosine similarity value was 0.00 contained no AUs, and for these frames the associated SASL expression was chosen arbitrarily. Unfortunately, both the Gabor+SVM and Haar+Adaboost methods, combined either with Cosine Similarity or Exact Matching, demonstrated only very modest results: only the "relieved" expression could be recognized from the video input.

6.9 Discussion

Given the small size of our SASL video test set, it is difficult to draw statistically significant conclusions. It does appear, however, that the Gabor+SVM method performed slightly better than the Haar+Adaboost approach: for both the Petro and Fredericks videos, the former AU recognition approach recognized the

http://etd.uwc.ac.za/

Table 6.5: Recognition Using Gabor+SVM and Cosine Similarity Matching: Petro Video

Time	Expression	Cos	Time	Expression	Cos	Time	Expression	Cos
0.00	relieved	1.000	30.24	far b	0.447	54.28	what is your name	0.707
0.04	relieved	1.000	30.28	far b	0.500	54.32	relieved	0.577
0.08	relieved	1.000	30.32	far b	0.447	54.36	becomes smaller	0.471
0.12	a lot	0.000	30.36	relieved	0.577	54.40	relieved	0.500
0.16	relieved	1.000	30.40	far b	0.500	54.44	far b	0.447
0.20	relieved	0.707	30.44	far b	0.500	54.48	a lot	0.000
0.24	relieved	0.707	30.48	far b	0.500	54.52	a lot	0.000
0.28	relieved	1.000	30.52	far b	0.500	54.56	a lot	0.000
0.32	relieved	0.707	30.56	far b	0.500	54.60	becomes smaller	0.577
0.36	relieved	0.577	30.60	far b	0.500	69.08	a lot	0.433
18.52	becomes smaller	0.577	40.20	relieved	0.707	69.16	a lot	0.433
18.56	becomes smaller	0.577	40.24	relieved	0.707	69.24	a lot	0.433
18.60	relieved	0.707	40.28	relieved	0.707	69.28	relieved	0.447
18.64	relieved	0.577	40.32	far b	0.447	69.60	a lot	0.500
18.68	relieved	0.577	40.36	what is your name	0.408	75.48	what is your name	0.447
18.72	relieved	0.577	40.40	becomes smaller	0.500	75.72	a lot	0.408
18.76	relieved	0.577	40.44	becomes smaller	0.447	75.80	what is your name	0.707
18.80	relieved	0.707	40.48	what is your name	0.447	75.84	what is your name	0.577
18.84	relieved	1.000	40.52	what is your name	0.447	75.88	becomes smaller	0.408
18.88	relieved	0.577	40.56	far b	0.577	80.08	becomes smaller	0.408
18.92	a lot	0.447	40.60	far b	0.500	80.24	a lot	0.000
21.80	a lot	0.354	43.92	a lot	0.000	80.44	a lot	0.378
21.84	relieved	0.707	44.00	relieved	0.577	86.88	relieved	0.707
21.88	a lot	0.354	44.04	relieved	0.500	86.92	**relieved**	0.577
21.96	what is your name	0.447	44.08	becomes smaller	0.408	86.96	**relieved**	0.577
22.00	becomes smaller	0.577	44.12	far b	0.408	87.00	**relieved**	0.500
22.08	a lot	0.378	44.16	far b	0.447	87.08	**relieved**	0.577
22.12	what is your name	0.447	44.20	far b	0.500	87.12	**relieved**	0.707
22.16	a lot	0.000	44.24	far b	0.500	87.16	**relieved**	0.707
22.20	what is your name	0.500	44.28	relieved	0.500	87.20	**relieved**	0.577
22.24	what is your name	0.577	54.20	what is your name	0.707	87.24	**relieved**	0.577
22.28	what is your name	0.707	54.24	what is your name	0.707	87.28	**relieved**	0.707
30.20	relieved	0.577						

http://etd.uwc.ac.za/

Table 6.6: Recognition Using Haar+Adaboost and Cosine Similarity Matching: Petro Video

Time	Expression	Cos	Time	Expression	Cos	Time	Expression	Cos
0.00	really	0.267	30.24	what is your name	0.707	54.28	relieved	0.500
0.04	relieved	0.577	30.28	relieved	0.447	54.32	relieved	0.500
0.08	what is your name	0.408	30.32	a lot	0.387	54.36	really	0.378
0.12	relieved	0.577	30.36	a lot	0.354	54.40	difficult	0.500
0.16	relieved	0.707	30.40	really	0.378	54.44	becomes smaller	0.289
0.20	relieved	0.500	30.44	a lot	0.447	54.48	often	0.354
0.24	relieved	0.408	30.48	what is your name	0.500	54.52	what is your name	0.354
0.28	relieved	0.577	30.52	becomes smaller	0.408	54.56	angry	0.316
0.32	really	0.267	30.56	a lot	0.408	54.60	becomes smaller	0.365
0.36	difficult	0.408	30.60	what is your name	0.500	69.08	relieved	0.577
18.52	far (b)	0.500	40.20	relieved	0.500	69.16	relieved	0.447
18.56	far (b)	0.577	40.24	difficult	0.354	69.24	fat	0.316
18.60	far (b)	0.577	40.28	far (b)	0.447	69.28	relieved	0.500
18.64	far (b)	0.500	40.32	becomes smaller	0.447	69.60	a lot	0.000
18.68	relieved	0.707	40.36	becomes smaller	0.408	75.48	what is your name	0.500
18.72	far (b)	0.577	40.40	relieved	0.447	75.72	relieved	0.577
18.76	relieved	0.577	40.44	relieved	0.500	75.80	a lot	0.408
18.80	what is your name	0.408	40.48	becomes smaller	0.408	75.84	what is your name	0.447
18.84	what is your name	0.408	40.52	far (b)	0.378	75.88	a lot	0.000
18.88	becomes smaller	0.378	40.56	difficult	0.408	80.08	a lot	0.354
18.92	a lot	0.408	40.60	becomes smaller	0.365	80.24	a lot	0.000
21.80	my name is	0.316	43.92	relieved	0.408	80.44	far (b)	0.378
21.84	my name is	0.346	44.00	far (b)	0.447	86.88	relieved	0.577
21.88	my name is	0.365	44.04	relieved	0.408	86.92	**relieved**	0.707
21.96	what is your name	0.353	44.08	relieved	0.408	86.96	**relieved**	0.500
22.00	what is your name	0.408	44.12	becomes smaller	0.577	87.00	becomes smaller	0.577
22.08	a lot	0.000	44.16	becomes smaller	0.577	87.08	**relieved**	0.447
22.12	often	0.316	44.20	becomes smaller	0.408	87.12	**relieved**	1.000
22.16	my name is	0.387	44.24	difficult	0.500	87.16	**relieved**	0.707
22.20	fat	0.316	44.28	becomes smaller	0.577	87.20	a lot	0.500
22.24	becomes smaller	0.333	54.20	a lot	0.000	87.24	really	0.378
22.28	what is your name	0.500	54.24	becomes smaller	0.333	87.28	becomes smaller	0.577
30.20	what is your name	0.500						

UNIVERSITY of the
WESTERN CAPE

Table 6.7: Recognition Using Gabor+SVM and Cosine Similarity Matching: Fredericks Video

Time	Expression	Cos	Time	Expression	Cos	Time	Expression	Cos
2.28	a lot	0.000	51.92	becomes smaller	0.577	54.28	becomes smaller	0.471
2.36	becomes smaller	0.471	53.52	becomes smaller	0.500	54.32	becomes smaller	0.577
2.40	becomes smaller	0.577	53.60	becomes smaller	0.408	54.36	becomes smaller	0.471
2.44	becomes smaller	0.577	53.64	becomes smaller	0.447	54.40	becomes smaller	0.471
2.52	what is your name	0.707	53.68	becomes smaller	0.447	62.20	a lot	0.000
2.56	a lot	0.000	53.72	becomes smaller	0.408	62.24	a lot	0.000
2.60	a lot	0.000	53.80	angry	0.365	62.28	**relieved**	1.000
16.80	what is your name	0.707	53.84	angry	0.365	62.32	a lot	0.000
21.48	difficult	0.500	53.88	a lot	0.000	62.36	a lot	0.000
21.76	difficult	0.500	53.92	relieved	0.447	62.40	**relieved**	0.707
51.60	a lot	0.000	54.04	becomes smaller	0.408	62.44	becomes smaller	0.577
51.64	what is your name	0.707	54.08	becomes smaller	0.408	62.48	a lot	0.000
51.68	far b	0.500	54.12	angry	0.365	62.52	a lot	0.000
51.72	far b	0.577	54.16	becomes smaller	0.577	62.56	far b	0.577
51.76	becomes smaller	0.447	54.20	becomes smaller	0.577	62.60	**relieved**	0.707
51.88	becomes smaller	0.577	54.24	relieved	0.707			

http://etd.uwc.ac.za/

Table 6.8: Recognition Using Haar+Adaboost and Cosine Similarity Matching: Fredericks Video

Time	Expression	Cos	Time	Expression	Cos	Time	Expression	Cos
2.28	a lot	0.000	51.92	often	0.387	54.28	becomes smaller	0.577
2.36	my name is	0.346	53.52	relieved	0.577	54.32	becomes smaller	0.408
2.40	really	0.309	53.60	relieved	0.500	54.36	difficult	0.354
2.44	a lot	0.408	53.64	a lot	0.354	54.40	difficult	0.500
2.52	becomes smaller	0.408	53.68	difficult	0.250	62.20	difficult	0.500
2.56	really	0.267	53.72	relieved	0.707	62.24	difficult	0.500
2.60	what is your name	0.500	53.80	relieved	0.500	62.28	difficult	0.500
16.80	relieved	0.577	53.84	far b	0.500	62.32	difficult	0.408
21.48	a lot	0.000	53.88	often	0.408	62.36	really	0.378
21.76	a lot	0.000	53.92	becomes smaller	0.408	62.40	becomes smaller	0.471
51.60	a lot	0.289	54.04	relieved	0.577	62.44	a lot	0.354
51.64	what is your name	0.447	54.08	a lot	0.000	62.48	often	0.354
51.68	becomes smaller	0.408	54.12	becomes smaller	0.408	62.52	a lot	0.354
51.72	a lot	0.316	54.16	difficult	0.354	62.56	a lot	0.000
51.76	relieved	0.500	54.20	relieved	0.500	62.60	really	0.309
51.88	becomes smaller	0.333	54.24	relieved	0.500			

"relieved" expression more consistently over the corresponding time window. This is consistent with our findings in Chapter 5.

We believe that the primary difficulty for our system in recognizing the expressions was the variability in head pose in the video. The signers were requested to look directly into the camera and to keep the face clear as much as possible. Nonetheless, the video frames contain considerable in-plane and out-of-plane head rotation as well as partial occlusion of the face by the hands, which makes both face detection and facial expression analysis more difficult. Tables 6.3 and 6.4 show the presence or absence of rotation and occlusion of the face (as assessed by a human coder) for each frame. Notice how most of the frames contained out-of-plane rotation of the face.

Another possible explanation for the low accuracy of our system is variability in the AU decomposition of the SASL expressions. If SASL expressions vary significantly in the AUs they comprise, either across different signers or across different occurrences for the same signer, then a simple AU-to-SASL mapping may not be possible, and natural language processing may be necessary in order to recognize a particular expression confidently. Only further research into SASL facial expressions can answer this question.

6.10 Summary and Conclusions

We have constructed an automatic, real-time SASL expression recognition system that uses FACS as an intermediary representation. We presented two approaches to mapping AUs to SASL expressions: an Exact Matching method, and a Cosine Similarity method. We tested both approaches, using the Haar+Adaboost and Gabor+SVM AU classifiers from the previous chapter, on two videos containing a SASL narrative. Only one SASL expression ("relieved") was recognized correctly from the video. We attribute these results

http://etd.uwc.ac.za/

to significant in-plane rotation, out-of-plane rotation, and occlusion of the face.

As demonstrated in Tables 6.4 and 6.3, natural signed communication is replete with 2-D and 3-D rotation of the head and partial occlusion of the face. FER systems for real-world applications must thus be robust to these conditions in order to be useful. In support of this goal, a publicly accessible facial expression database containing a variety of head poses would be extremely useful. As pointed out in Chapter 5, these databases should ideally contain singly-occurring AUs so that correlation effects do not adversely affect the training of the classifier.

UNIVERSITY of the
WESTERN CAPE

http://etd.uwc.ac.za/

Chapter 7

Conclusions and Directions for Further Research

This thesis has made several important contributions to the field of automatic facial expression recognition. First, we examined the issue of whether local face segmentation yields higher AU recognition accuracy than whole-face analysis. We found that global analysis yields superior recognition rates on our dataset and showed that this phenomenon is at least partially due to the strong correlation between AUs in the Cohn-Kanade database. This result underlines the importance of establishing a publicly available dataset in which AUs either occur individually or with low correlation.

Second, we have developed a new approach to FACS AU recognition based on Haar features and the Adaboost classification method. Our system achieves equally high recognition accuracy as the Gabor+SVM approach but operates two orders of magnitude more quickly.

Finally, we have proposed a plausible architecture for using FACS as an intermediary framework for recognizing the facial expressions of SASL. While our system is not yet mature for effective SASL recognition, conducting this pilot study has proven that SASL expression recognition using FACS is fundamentally possible. It also underlines the fact that considerable in-plane, out-of-plane, and occlusion of the face occurs even in a laboratory environment, and that AU classifiers for real-world applications must be robust to handling these conditions.

Conducting this research has revealed a number of new research questions, both on facial expression recognition itself and on using FER systems to recognize signed languages. We discuss open questions in both fields separately.

UNIVERSITY of the WESTERN CAPE
http://etd.uwc.ac.za/

7.0.1 Facial Expression Recognition

One of the most fundamental issues in automatic FER is the best type of feature to use for classification. Many types of features exist, and some of these - e.g., Gabor, Haar, pixel intensities, geometric relationships between fiducial points, etc. - have been applied to automatic expression recognition. Many more exist, however, and have not yet been evaluated for expression analysis. In particular, edge orientation histograms have been shown to outperform Viola-Jones Haar features (Levi and Weiss [LW04]) when the training set is small. Scale-invariant "SIFT" features (developed by David Lowe [Low04]), which are reportedly invariant to changes in scale, translation, and rotation, may also be useful in the domain of FER.

As illustrated by our pilot study of SASL recognition, real-world expressions occur with considerable 3-D rotation and occlusion of the faces. One important open issue is whether the expressions within these faces should best be recognized using pose-specific expression detectors, or instead by a single detector that is robust to strong changes in pose. As one particular implementation of the latter strategy, 3-D face tracking could be employed to rotate the detected face back to a canonical, frontal view, and expression recognition could proceed from there [Mar].

Finally, regardless of which kinds of image features are used, an important question is the kind of classifier used for expression recognition. Support vector machines have demonstrated good performance over all the AUs we tested, as has Adaboost for particular AUs. Other boosting techniques, such as Logitboost and Gentle Adaboost [FHT98], also exist, however, and may also prove effective for expression recognition.

7.0.2 Automatic Signed Language Recognition

From the limited data we collected, we consider it likely that the FACS framework is sufficiently discriminative to enable SASL expressions to be distinguished and recognized by the AU sets that they comprise. However, it remains to be investigated whether SASL signs are consistent in their AU decomposition across different signers, and whether they are even consistent across different instances from the same person.

Despite the difficulties we encountered in our pilot study of SASL recognition, we hope that our software prototype will provide a firm ground from which progeny of our project can progress. Researching and writing this thesis has been enormously educational for this researcher; we hope that future members of the SASL Project at the University of the Western Cape are equally rewarded.

http://etd.uwc.ac.za/

Appendix A

Mathematical Fundamentals and Computer Vision Algorithms

A.1 Distance between a hyperplane H and the origin

Let H be described as $\mathbf{w} \cdot \mathbf{x} + c = 0$, where $\mathbf{w}$ is normal to H, and c is the bias. The shortest vector $\mathbf{x}^*$ from the origin to H must be normal to H and thus parallel to $\mathbf{w}$. Since $\mathbf{x}^*$ lies in H, it must satisfy

$$\mathbf{w} \cdot \mathbf{x}^* + c = 0 \tag{A.1}$$

Since $\mathbf{w}$ and $\mathbf{x}^*$ are parallel, their inner-product equals $\|\mathbf{w}\|\|\mathbf{x}^*\|$, and thus

$$\|\mathbf{w}\|\|\mathbf{x}^*\| + c = 0 \tag{A.2}$$

$$\|\mathbf{x}^*\| = \frac{-c}{\|\mathbf{w}\|} \tag{A.3}$$

A.2 Time Complexity of 2-D FFT

It is generally known that the time complexity of a 1-D Fast Fourier Transform (FFT) is $O(N \log N)$, where N is the number of discrete points to be transformed. A 2-D Fourier transform can be computed by first transforming each column of the image, and then transforming each row of the result.

Suppose that a square image contains M rows and M columns for a total of $M^2 = N$ pixels. Then the number of CPU instructions required to transform all the columns is $O(M * M \log M)$. The rows of the resulting image must then also be transformed, which requires $O(M * M \log M)$ more operations. In total,

UNIVERSITY of the WESTERN CAPE
http://etd.uwc.ac.za/

the FFT of the 2-D image requires:

$$O(M * M \log M + M * M \log M) = O(2 * M * M \log M) \quad (A.4)$$

$$= O(M^2 \log M^2) \quad (A.5)$$

operations. We thus conclude that, for a square image with N pixels, the 2-D FFT takes $O(N \log N)$ operations.

A.3 Principle Component Analysis

Let $T = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ be a set of training data such that each $\mathbf{x} \in \mathbb{R}^n$. The mean μ of T is assumed to be zero; if $\mu \neq 0$, then each $\mathbf{x} \in T$ is first reduced by μ. Principle component analysis of T consists of finding a new sequence of n basis vectors $\mathbf{e}_1, \ldots, \mathbf{e}_n$, called the *principle components*; Each principle component $\mathbf{e}_j$ is calculated to give the jth largest variance when the vectors in T are projected onto it. Since T was assumed to have zero mean, the variance resulting from each basis vector is determined by:

$$\text{var}_j = \frac{1}{N-1} \sum_{i=1}^{N} [\mathbf{e}_j^T (\mathbf{x}_i - \mu)][\mathbf{e}_j^T (\mathbf{x}_i - \mu)]^T \quad (A.6)$$

$$= \frac{1}{N-1} \sum_{i=1}^{N} \mathbf{e}_j^T (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T \mathbf{e}_j \quad (A.7)$$

$$= \mathbf{e}_j^T \left(\frac{1}{N-1} \sum_{i=1}^{N} (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T \right) \mathbf{e}_j \quad (A.8)$$

$$\implies \text{var}_j = \mathbf{e}_j^T \Sigma \mathbf{e}_j \quad (A.9)$$

where Σ is the covariance matrix of T. When computing the first principle component $\mathbf{e}_1$, the variance should be maximized. Maximizing var_j is equivalent to maximizing the inner product of $\mathbf{e}_1^T$ and $(\Sigma \mathbf{e}_1)$, which is greatest when $\mathbf{e}_1^T$ is parallel to the eigenvector of Σ with the largest associated eigenvalue λ_1. Computing $\mathbf{e}_2$ is then achieved by choosing $\mathbf{e}_j$ to be parallel to the eigenvector with second-greatest associated eigenvalue λ_2, and so on.

After determining the principle components, PCA can then be used for dimensionality reduction by projecting each $\mathbf{x} \in T$ onto the first $p << n$ principle components, resulting in a smaller p-dimensional feature vector. Because of the way the components were calculated, the resultant set of projections still retain most of T's original variance.

http://etd.uwc.ac.za/

A.4 Optic Flow Analysis

In order to compute optic flow, image intensity is modeled as a function of not only x and y, but also of time t. Suppose that a pixel moves from location (x, y) at time t to location $(x + \Delta x, y + \Delta y)$ at time $t + \Delta t$. Then the intensity values at these two locations and times will be equal, i.e.:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \tag{A.10}$$

The right-hand-side of Equation A.10 can be approximated to first order by means of a Taylor series:

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t + \ldots \tag{A.11}$$

where the ellipsis stands for small higher-order terms which are assumed to be small enough to ignore. Combining Equations A.10 and A.11 we arrive at:

$$I(x, y, t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t = I(x, y, t) \tag{A.12}$$

$$\implies \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t = 0 \tag{A.13}$$

In order to convert from *displacement* of pixel location (x, y) into *velocity*, we divide both sides of the last equation by Δt:

$$\frac{\partial I}{\partial x}\frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y}\frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t}\frac{\Delta t}{\Delta t} = 0 \tag{A.14}$$

$$\implies \frac{\partial I}{\partial x}v_x + \frac{\partial I}{\partial y}v_y + \frac{\partial I}{\partial t} = 0 \tag{A.15}$$

$$\tag{A.16}$$

where $v_x = \frac{\Delta x}{\Delta t}$ and $v_y = \frac{\Delta y}{\Delta t}$.

The partial derivatives of I with respect to x, y, and t represent the spatial and temporal image gradients; they can be computed using derivative filters over the image sequence. After computing these values, there still remain two unknowns for only one equation, and thus the system is under-determined. In order to solve for v_x and v_y, additional constraints must be provided. Commonly used algorithms for providing such constraints and for completing the optic flow calculation are the Lucas-Kanade method, which assumes that flow is constant within small local windows about each pixel, and the iterative Horn-Schunck approach, in which "smoothness" of an energy function is enforced.

UNIVERSITY of the WESTERN CAPE
http://etd.uwc.ac.za/

A.5 Haar Wavelets

A.5.1 One-dimensional Haar Wavelet Decomposition

The one-dimensional Haar wavelet decomposition of an n-element input array is computed recursively using a two-step process of averaging and differencing. In order to emphasize the main concepts of the algorithm, we ignore the normalization constants that must be considered in the actual transform.

In the *averaging* stage, the input array is reduced in length by half by averaging the value of every pair of neighboring values. For instance, the input array $[3, 1, 4, 6, 9, 3]$ is converted to $[2, 5, 6]$. Clearly, information has been lost by this averaging step. In order to recover the lost information, $\frac{n}{2}$ *detail coefficients* are appended to the output array during the *differencing* stage. Each detail coefficient d is the amount by which the first element in the averaged pair exceeds that pair's average. For example, for the first pair $(3, 1)$, whose average is 2, the first element 3 exceeds the average by 1; hence, the detail coefficient for the first pair of numbers is 1. For the second pair $(4, 6)$, the average is 5. Since the first number 4 exceeds 5 by -1 (because $4 - (-1) = 5$), the detail coefficient is -1.

After appending the $\frac{n}{2}$ detail coefficients to the array of averaged pairs, the array once again has length n. The two stages of averaging and differencing are then repeated on the first half of the array. At the next level of recursion, the first quarter of the array will be averaged, and so on. The recursion is complete after $\log_2 n$ levels when only one pair of numbers is averaged.

We illustrate the entire transform on a generic array of length 4, whose elements are $[a_1, a_2, a_3, a_4]$. The transform proceeds as follows (each line represents one averaging and differencing step):

$$[a_1, a_2, a_3, a_4] \tag{A.17}$$

$$\left[\frac{a_1 + a_2}{2}, \frac{a_3 + a_4}{2}, a_1 - \frac{a_1 + a_2}{2}, a_3 - \frac{a_3 + a_4}{2}\right] \tag{A.18}$$

$$\left[\frac{a_1 + a_2 + a_3 + a_4}{4}, \frac{a_1 + a_2}{2} - \frac{a_1 + a_2 + a_3 + a_4}{4}, a_1 - \frac{a_1 + a_2}{2}, a3 - \frac{a_3 + a_4}{2}\right] \tag{A.19}$$

Combining fractions and factoring out the denominator, we can simplify the final array:

$$\left[\frac{a_1 + a_2 + a_3 + a_4}{4}, \frac{a_1 + a_2 - (a_3 + a_4)}{4}, \frac{a_1 - a_2}{2}, \frac{a_3 - a_4}{2}\right] \tag{A.20}$$

The first element of the output array equals the overall average of the input array. More important for purposes of image classification, however, are the detail coefficients: The detail coefficients express the difference between neighboring array values, or between sums of neighboring sets of array values. For instance, the second element of the output array equals the difference in value between the first and second pairs of array values. The last two elements of the output array equal the difference between the first

http://etd.uwc.ac.za/

and second, and third and fourth input array elements, respectively. In the realms of object recognition and detection, when the two-dimensional Haar decomposition is applied to the input image, this property becomes extremely useful in its effectiveness for detecting edges and other differences in pixel intensity.

In practice, the magnitudes of many of the detail coefficients are typically very small, and they can be ignored with little reconstruction error [SDS94]. In this sense, the Haar decomposition naturally lends itself to feature selection because some of the wavelet coefficients have a greater impact on the image's appearance than others.

A.5.2 Two-dimensional Haar Wavelet Decomposition

There are two methods of generalizing the one-dimensional Haar decomposition to the two dimensional case. In the *standard decomposition*, the transform is first applied to each row of the input matrix. After transforming all rows, the transform is then applied to each column.

In the *non-standard decomposition*, the transform is alternately applied to rows and columns at each recursive level of the transform. More precisely, one averaging and one differencing stage is first applied to each row of the input matrix. Then, one averaging and one differencing stage is applied to each column of the matrix. The transform then proceeds again on the rows at the next recursive level.

UNIVERSITY of the
WESTERN CAPE

http://etd.uwc.ac.za/

Appendix B

Representative ROC Curves

In this appendix we show a representative set of of 33 Receiver Operator Characteristics (ROC) curves from the experiments we performed in Chapter 5. For each of the 11 AUs that we classified, and for each of the 3 AU recognition algorithms we studied - local Gabor+SVM, global Gabor+SVM, and local Haar+Adaboost - we present the ROC curve and the Area Under the Curve (AUC) of one validation fold (Fold #1).

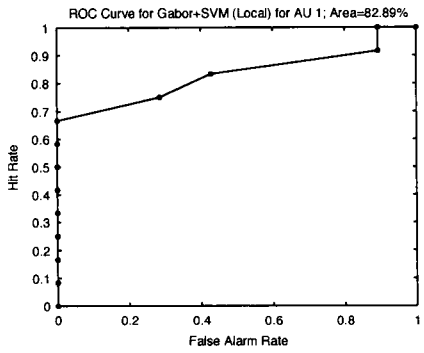
In some of the curves displayed below, the classifier was able to separate the positive and negative data points completely, with no errors. In such cases, the Area under the Curve is 100%, and no "curve" appears inside the graph window at all - only a set of dots corresponding to different classifier threshold values appears on the x and y axes.

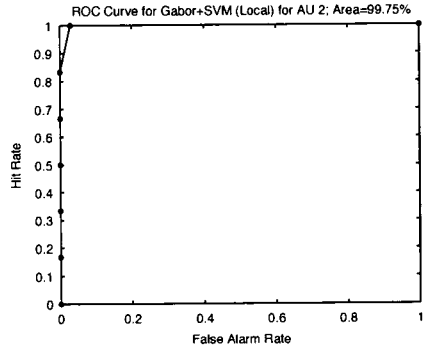
Note that the AUC values reported in Chapter 5 were averaged over all 10 cross-validation folds, and that the AUC values listed for the individual ROC curves in this appendix can stray from this average considerably. We thus strongly advise against comparing classifiers based on their performance of only a single cross-validation fold.

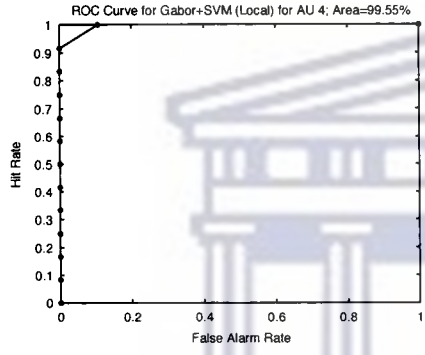
B.1 Local Gabor+SVM

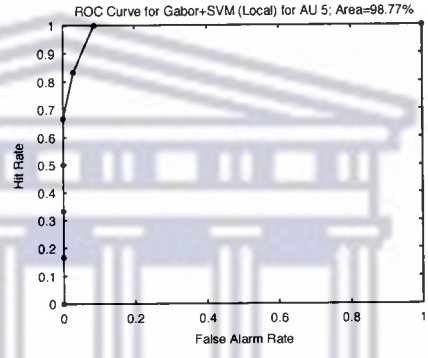
ROC curves and Area under the Curve values for the local Gabor+SVM classifier. Curves are shown for Validation Fold #1 only.

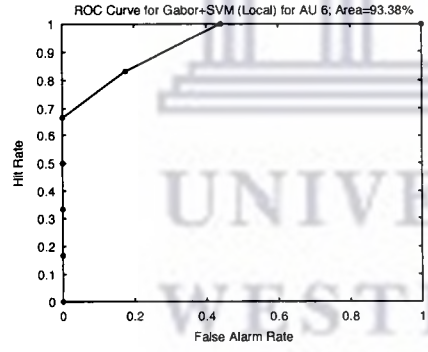
http://etd.uwc.ac.za/

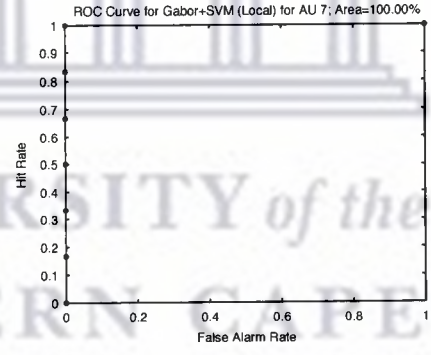






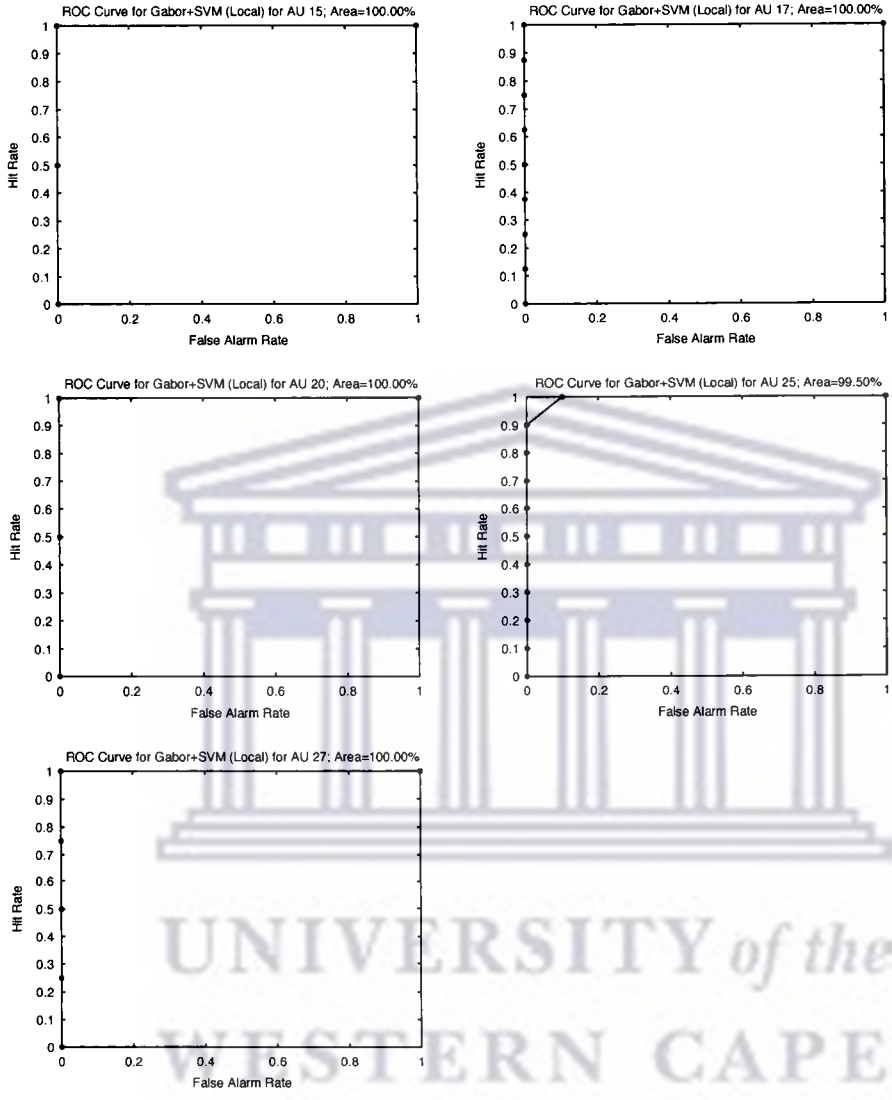






UNIVERSITY of the
WESTERN CAPE

http://etd.uwc.ac.za/

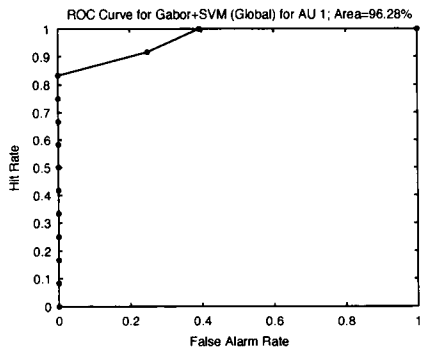


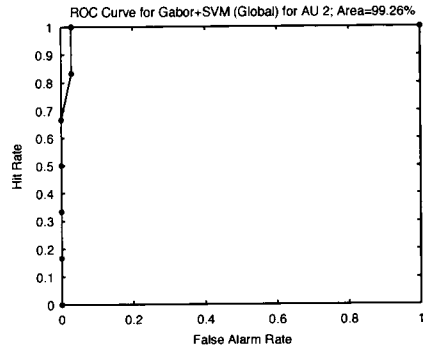
UNIVERSITY of the WESTERN CAPE

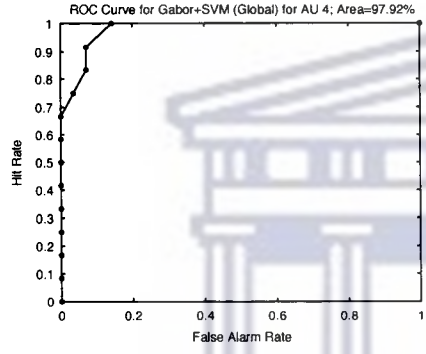
B.2 Global Gabor+SVM

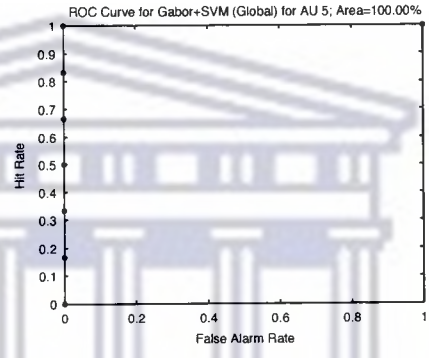
ROC curves and Area under the Curve values for the global Gabor+SVM classifier. Curves are shown for Validation Fold #1 only.

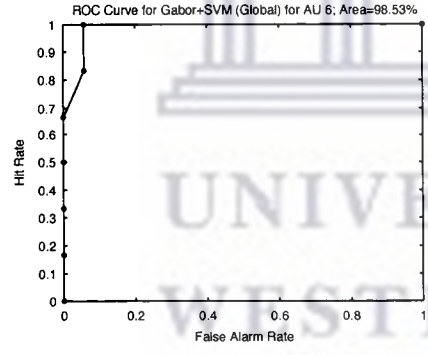
http://etd.uwc.ac.za/

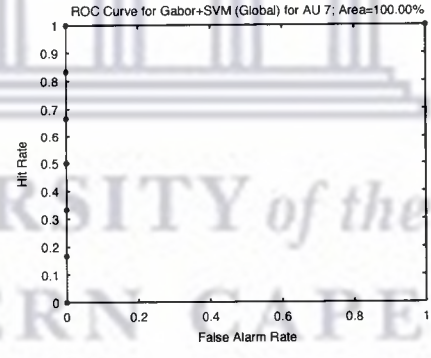






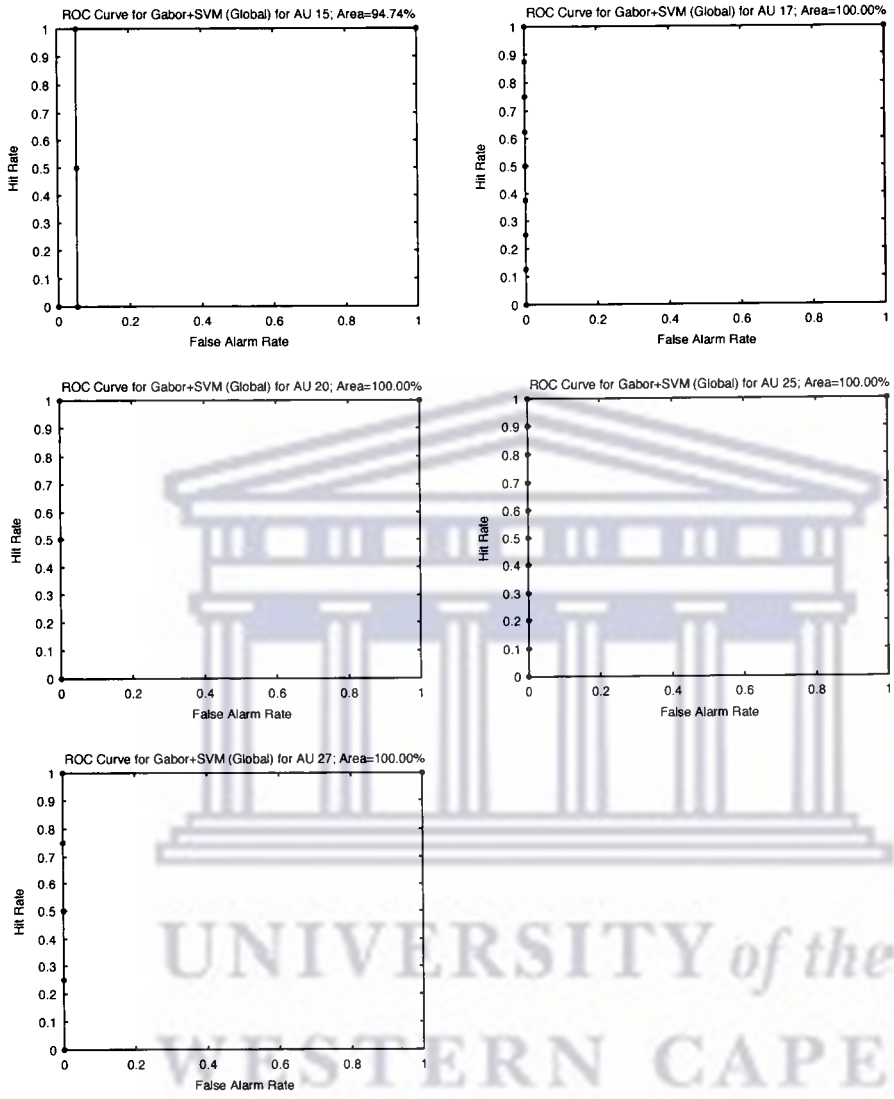






UNIVERSITY of the
WESTERN CAPE

http://etd.uwc.ac.za/

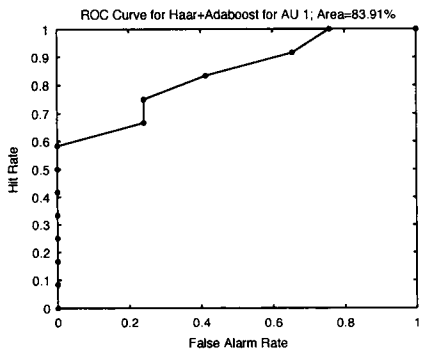


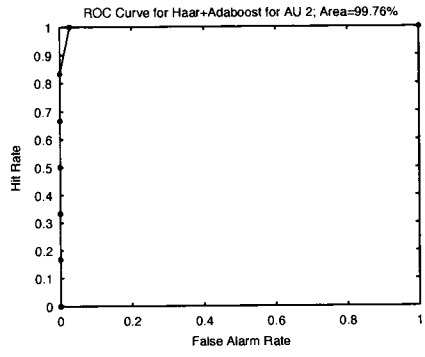
UNIVERSITY of the WESTERN CAPE

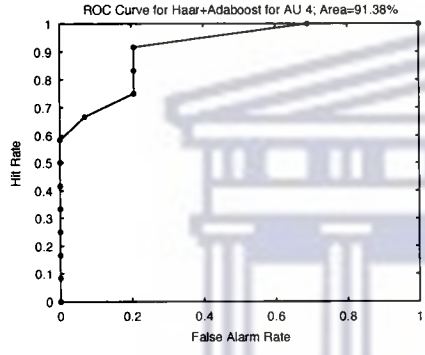
B.3 Local Haar+Adaboost

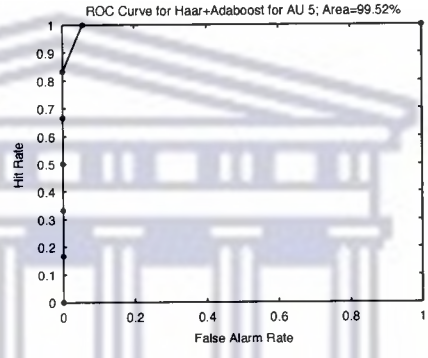
ROC curves and Area under the Curve values for the local Haar+Adaboost classifier. Curves are shown for Validation Fold #1 only.

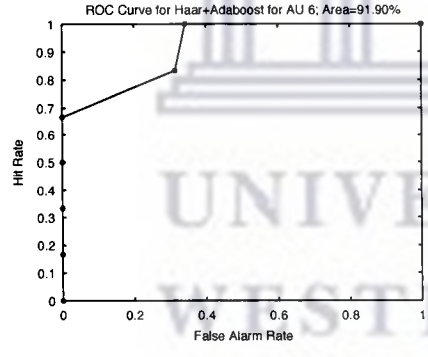
http://etd.uwc.ac.za/

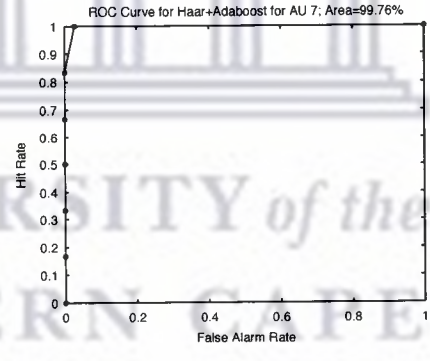






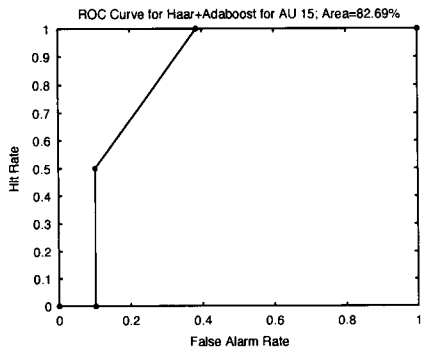


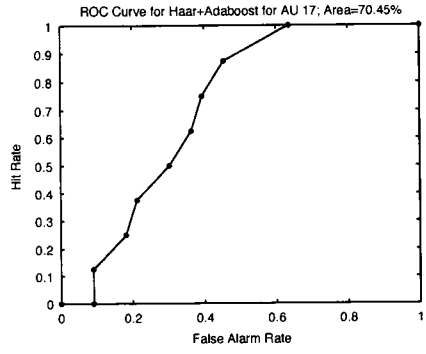


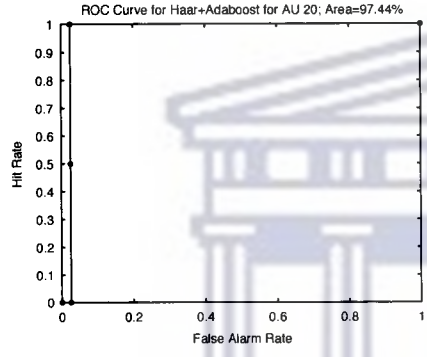


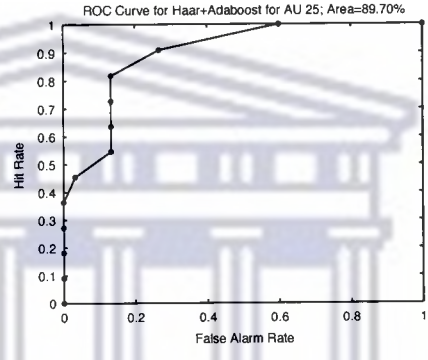
UNIVERSITY of the
WESTERN CAPE

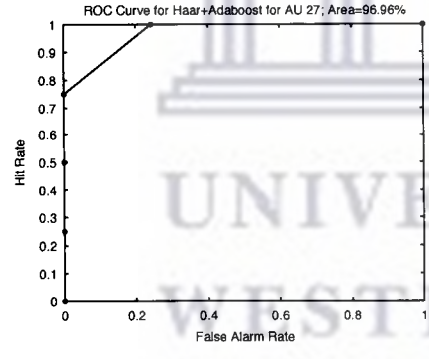
http://etd.uwc.ac.za/











UNIVERSITY of the
WESTERN CAPE

http://etd.uwc.ac.za/

Bibliography

[Bar] Dr. Marian Bartlett. Personal communication.

[BCL02] F. Bourel, C.C. Chibelushi, and A.A. Low. Robust facial expression recognition using a state-based model of spatially-localised facial dynamics. In *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, 2002.

[BDM+00] M. Bartlett, G. Donato, J. Movellan, J. Hager, P. Ekman, and T. Sejnowski. Image representations for facial expression coding. In S.A. Solla, T.K. Leen, and K.-R. Muller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 2000.

[BHES99] Marian Stewart Bartlett, Joseph C. Hager, Pault Ekman, and Terrence J. Sejnowski. Measuring facial expressions by computer image analysis. *Psychophysiology*, 36(2):253–263, 1999.

[BLF+06] Marian Stewart Bartlett, Gwen Littlewort, Mark Frank, Claudia Lainscsek, Ian Fasel, and Javier Movellan. Fully automatic facial action recognition in spontaneous behavior. In *Proceedings of the Seventh IEEE International Conference on Automatic Face and Gesture Recognition*, 2006.

[Bur98] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[BY95] Michael J. Black and Yaser Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. In *Proceedings of the Fifth International Conference on Computer Vision*, pages 374–381, 1995.

[Can02] Ulrich Canzler. LTI bi-annual report. Technical report, RWTH Aachen, 2002.

[CET98] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *Lecture Notes in Computer Science*, 1407:484–, 1998.

[CKM+01] J. Cohn, T. Kanade, T. Moriyama, Z. Ambadar, J. Xiao, J. Gao, and H. Imamura. A comparative study of alternative facs coding algorithms. Technical Report CMU-RI-TR-02-06, Robotics Institute, Carnegie Mellon University, 2001.

http://etd.uwc.ac.za/

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

[CSC+03] Ira Cohen, Nicu Sebe, Larry Chen, Ashutosh Garg, and Thomas S. Huang. Facial expression recognition from video sequences: Temporal and static modelling. *Computer Vision and Image Understanding: Special Issue on Face Recognition*, 2003.

[CZLK99] Jeffrey Cohn, Adena Zlochower, Jenn-Jier James Lien, and Takeo Kanade. Automated face analysis by feature point tracking has high concurrent validity with manual facs coding. *Psychophysiology*, 36:35 – 43, 1999.

[DBH+99] Gianluca Donato, Marian Stewart Bartlett, Joseph C. Hager, Paul Ekman, and Terrence J. Sejnowski. Classifying facial actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):974–989, 1999.

[DeC02] Dennis DeCoste. Anytime interval-valued outputs for kernel machines: Fast support vector machine classification via distance geometry. In *International Conference on Machine Learning*, 2002.

[DR04] D. Datcu and L.J.M. Rothkrantz. Automatic recognition of facial expressions using bayesian belief networks. In *Proceedings of IEEE Systems, Man and Cybernetics*, 2004.

[EF78] P. Ekman and W. Friesen. *The Facial Action Coding System: A Technique For The Measurement of Facial Movement*. Consulting Psychologists Press, Inc., San Francisco, CA, 1978.

[EFH02] Paul Ekman, Wallace V. Friesen, and Joseph C. Hager. *Facial Action Coding System (FACS)*. A Human Face, Salt Lake City, 2002.

[Ekm82] P. Ekman. Methods for measuring facial action. In K.R. Scherer and P. Ekman, editors, *Handbook Of Methods in Nonverbal Behavior Research*, pages 45–90. Cambridge University Press, Cambridge, 1982.

[Ekm01] Paul Ekman. *Telling Lies: Clues to Deceit in the Marketplace, Politics, and Marriage*. W. W. Norton & Company, 2001.

[EP97] Irfan A. Essa and Alex P. Pentland. Coding, analysis, interpretation, and recognition of facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):757–763, 1997.

[FB02] Ian R. Fasel and Marian S. Bartlett. A comparison of gabor filter methods for automatic detection of facial landmarks. In *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, 2002.

UNIVERSITY of the WESTERN CAPE

http://etd.uwc.ac.za/

[FDH+] Ian Fasel, Ryan Dahl, John Hershey, Bret Fortenberry, Josh Susskind, and Javier R. Movellan. Machine perception toolbox MPISearch.

[FHT98] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. Technical report, Stanford University and University of Toronto, July 1998.

[FJ05] Matteo Frigo and Steven G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE Special Issue on "Program Generation, Optimization, and Platform Adaptation"*, 93(2):216–231, 2005.

[FL00] Beat Fasel and Jürgen Lüttin. Recognition of asymmetric facial action unit activities and intensities. In *Proceedings of International Conference on Pattern Recognition*, Barcelona, Spain, 2000.

[Fle80] R. Fletcher. *Practical methods of optimization*, volume 2. John Wiley & Sons, Inc., Chichester, 1980.

[FS99] Y. Freund and R. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 1999.

[GGJ] Simona Grigorescu, Cosmin Grigorescu, and Andrei Jalba. Tools for Image Processing (TiP) Library. No longer available; originally at `http://www.cs.rug.nl/~cosmin/tip/TiP-0.0.1.tar.gz`.

[GTGB02] Salih Burak Göktürk, Carlo Tomasi, Bernd Girod, and Jean-Yves Bouguet. Model-based face tracking for view-independent facial expression recognition. In *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, 2002.

[HE00] A. Hyvarinen and E.Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4-5):411–430, 2000.

[Her00] American Heritage. *American Heritage Dictionary of the English Language*. Houghton Mifflin Company, fourth edition, 2000.

[IEG06] Ramana Isukapalli, Ahmed Elgammal, and Russell Greiner. Learning to identify facial expression during detection using markov decision process. In *Proceedings of the Seventh IEEE International Conference on Automatic Face and Gesture Recognition*, 2006.

[Iza79] C.E. Izard. *The Maximally Discriminative Facial Movement Coding System (MAX)*. University of Delaware, Instructional Resource Center, Newark, 1979.

[JFS95] C. Jacobs, A. Finkelstein, and D. Salesin. Fast multiresolution image querying. In *Proceedings of SIGGRAPH 95*, New York, 1995.

http://etd.uwc.ac.za/

[KCIT00] Takeo Kanade, Jeffrey Cohn, and Ying li Tian. Comprehensive database for facial expression analysis. In *Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition (FG'00)*, pages 46 – 53, March 2000.

[KQP03] Ashish Kapoor, Yuan Qi, and Rosalind W. Picard. Fully automatic upper facial action recognition. In *Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, 2003.

[KY97] Satoshi Kimura and Masahiko Yachida. Facial expression recognition and its degree estimation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 1997.

[LA98] Michael Lyons and Shigeru Akamatsu. Coding facial expressions with gabor wavelets. In *Proceedings of the Third International Conference on Face & Gesture Recognition*, pages 200–205, Nara, Japan, 1998.

[LBF+04] G.C. Littlewort, M.S. Bartlett, I.R. Fasel, J. Chenu, T. Kanda, H. Ishiguro, and J.R. Movellan. Towards social robots: Automatic evaluation of human-robot interaction by face detection and expression classification. In *Advances in Neural Information Processing Systems*, volume 16, 2004.

[LFBM01] Gwen Littlewort-Ford, Marian Stewart Bartlett, and Javier R. Movellan. Are your eyes smiling? Detecting genuine smiles with support vector machines and gabor wavelets. In *Proceedings of the 8th Annual Joint Symposium on Neural Computation*, 2001.

[LFBM02] Gwen Littlewort, Ian Fasel, Marian Stewart Bartlett, and Javier R. Movellan. Fully automatic coding of basic expressions from video. INC MPLab Tech Report 3, University of California, San Diego, La Jolla, CA, 2002.

[Lid80] Scott K. Liddell. *American Sign Language Syntax*. Mouton, The Hague, 1980.

[Lit] Dr. Gwen Littlewort. Personal communication.

[LKCL98] J.J. Lien, T. Kanade, J. Cohn, and C. Li. Automated facial expression recognition based on facs action units. In *Third IEEE International Conference on Automatic Face and Gesture Recognition*, 1998.

[Low04] David G. Lowe. Distinctive image features from scale-invariant keypoint. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[LPA00] M.J. Lyons, J. Budynek A. Plante, and S. Akamatsu. Classifying facial attributes using a 2-d gabor wavelet representation and discriminant analysis. In *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 202–207, Grenoble, France, April 2000.

UNIVERSITY of the WESTERN CAPE

http://etd.uwc.ac.za/

[LR98] C. Lisetti and D. Rumelhart. Facial expression recognition using a neural network. In *Proceedings of the 11 th International FLAIRS Conference*, 1998.

[lT04] Ying li Tian. Evaluation of face resolution for expression analysis. In *Proceedings of CVPR Workshop on Face Processing in Video*, Washington, DC, 2004.

[LTC95] A. Lanitis, C.J. Taylor, and T.F. Cootes. A unified approach to coding and interpreting face images. In *Proceedings of the Fifth International Conference on Computer Vision*, 1995.

[lTKC00] Ying li Tian, Takeo Kanade, and Jeffrey F. Cohn. Eye-state action unit detection by gabor wavelets. In *ICMI*, pages 143–150, 2000.

[lTKC02] Ying li Tian, Takeo Kanade, and Jeffrey F. Cohn. Evaluation of gabor-wavelet-based facial action unit recognition in image sequences of increasing complexity. In *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, 2002.

[LVB+93] Martin Lades, Jan C. Vorbrüggen, Joachim Buhmann, Jürg Lange, Christoph v.d. Malsburg, Rolf P. Würtz, and Wolfgang Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42(3):300–311, 1993.

[LW04] Kobi Levi and Yair Weiss. Learning object detection from a small number of examples: The importance of good features. In *International Conference on Computer Vision and Patern Recognition (CVPR)*, 2004.

[Mar] Dr. Tim K. Marks. Personal communication.

[Mas91] Kenji Mase. Recognition of facial expression from optical flow. *Institute of Electronics, Information, and Communication Engineers Transactions*, E74(10):3474–3483, 1991.

[MGB+03] Bartlett MS, Littlewort G, Braathen B, Sejnowski TJ, and Movellan JR. A prototype for automatic recognition of spontaneous facial actions. In *Advances in Neural Information Processing Systems*, volume 15, 2003.

[Mov] Moving Pictures Expert Group. *MPEG-4 Synthetic/Natural Hybrid Coding (SNHC)*.

[NKM+99] Carol Jan Neidle, Judy Kegl, Dawn MacLaughlin, Benjamin Bahan, and Robert G. Lee. *The Syntax of American Sign Language: Functional Categories and Hierarchical Structure*. MIT Press, Cambridge, MA, 1999.

[OHN92] H. Oster, D. Hegley, and L. Nagel. Adult judgments and fine-grained analysis of infant facial expressions: Testing the validity of a priori coding formulas. *Developmental Psychology*, 28:1115–1131, 1992.

http://etd.uwc.ac.za/

[PC97] C. Padgett and G. Cottrell. Representing face images for emotion classification. In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, 1997.

[POP98] Constantine P. Papageorgiou, Michael Oren, and Tomaso Poggio. A general framework for object detection. In *Proceedings of the Sixth International Conference on Computer Vision*, 1998.

[RMB90] Judy Snitzer Reilly, Marina Macintire, and Ursula Bellugi. The acquisition of conditionals in american sign language: Grammaticized facial expressions. *Applied Psycholinguistics*, 11:369–392, 1990.

[SDS94] Eric J. Stollnitz, Tony D. DeRose, and David H. Salesin. Wavelets for computer graphics: A primer. *IEEE Computer Graphics and Applications*, 15(3,4):75–85,76–84, 1994.

[SS96] Hiroshi Sako and Anthony V.W. Smith. Real-time facial expression recognition based on features' positions and dimensions. In *Proceedings of International Conference on Pattern Recognition*, 1996.

[SS98] Alex. J. Smola and Bernard Schoelkopf. A tutorial on support vector regression. In *NeuroColt2 Technical Report Series*, volume 30, 1998.

[SSM98] Alex J. Smola, Bernhard Schoelkopf, and Klaus-Robert Mueller. The connection between regularization operators and support vector kernels. *Neural Networks*, 11(4):637–649, 1998.

[TKC01] Y. Tian, T. Kanade, and J. Cohn. Recognizing action units for facial expression analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):97–115, 2001.

[VJ04] Paul Viola and Michael Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[WAWH04] Yubo Wang, Haizhou Ai, Bo Wu, and Chang Huang. Real time facial expression recognition with adaboost. In *Proceedings of International Conference on Pattern Recognition*, 2004.

[Wil] Dr. Ronnie Wilbur. Personal communication. 2005.

[WIY98] M. Wang, Y. Iwai, and M. Yachida. Expression recognition from time-sequential facial images by use of expression change model. In *Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition*, 1998.

[WRM04] Jianxin Wu, James M. Rehg, and Matthew D. Mullin. Learning a rare event detection cascade by direct feature selection. In *Advances in Neural Information Processing Systems*, volume 16, 2004.

UNIVERSITY of the WESTERN CAPE

http://etd.uwc.ac.za/

[YD96] Yasser Yacoob and Larry S. Davis. Recognizing human facial expressions from long image sequences using optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(6):636–642, June 1996.

[Zha98] Zhengyou Zhang. Feature-based facial expression recognition: Experiments with a multi-layer perceptron. Technical Report 3354, INRIA Sophia Antipolis, February 1998.

[ZLSA98] Zhengyou Zhang, Michael Lyons, Michael Schuster, and Shigeru Akamatsu. Comparison between geometry-based and gabor-wavelets-based facial expression recognition using multi-layer perceptron. In *Proceedings of the Third International Conference on Face & Gesture Recognition*, Nara, Japan, 1998.

UNIVERSITY of the
WESTERN CAPE

http://etd.uwc.ac.za/