Karlsruhe Institute of Technology

Faculty of Informatics
Institute for Anthropomatics
Chair Prof. Dr.-Ing. R. Stiefelhagen
Facial Image Processing and Analysis Group

Facial Expression Classification on Web Images

STUDY THESIS OF

Matthias Richter

At the faculty of Computer Science
Institute for Anthropomatics

ADVISORS

Dr.-Ing. Hazım Kemal Ekenel
Dipl.-Inform. Tobias Gehrig

JANUARY 2012

Facial Image Processing and Analysis Group
Institute for Anthropomatics
Karlsruhe Institute of Technology
Title: Facial Expression Classification on Web Images
Author: Matthias Richter

Matthias Richter
Durlacher Allee 44
76131 Karlsruhe
matthias@vrld.org

Statement of authorship

I hereby declare that this thesis is my own original work which I created without illegitimate help by others, that I have not used any other sources or resources than the ones indicated and that due acknowledgement is given where reference is made to the work of others

Karlsruhe, 31. January 2012

.....
(Matthias Richter)

Abstract

Recognition and interpretation of facial expressions is a vital task in human-to-human communication. Accessing this channel of communication would open up a wide range of possibilities in human-computer-interaction, health-care, education, the entertainment industry and other areas. An effective expression recognition system depends on three high-level parts: Effective machine learning algorithms, robust facial representation and an encompassing ground truth to train the classifiers. The latter is addressed by developing a new image database compiled from manually labelled web images. The database contains a large number of male and female subjects of different age groups and ethnicities performing seven basic expressions with varying head pose and under uncontrolled lighting conditions. Three facial descriptors based on the *discrete cosine transform* (DCT), *local binary patterns* (LBP) and *Gabor filters* are formulated in terms of regions around key points. Automatic key point selection using boosting is compared to a common block-based feature extraction method. In extensive experiments the web image database is utilized to compare *AdaBoost* and *support vector machine* (SVM) classifiers using the different facial representations. DCT and LBP features produce best results with a combination of per-expression selected key points and SVM classifiers, whereas the Gabor filter based representation yields optimal performance when the regions are placed on a regular grid. It is furthermore observed that, contrary to intuition, selection of many key points might deteriorate performance rather then improve it.

Contents

List of Figures

List of Tables

List of Abbreviations

AU	Action Unit
CERT	Computer Expression Recognition Toolbox
DCT	Discrete Cosine Transform
FACS	Facial Action Coding System
FERA	Facial Expression Recognition and Analysis
GEMEP	Geneva Multimodal Emotion Portrayals
LBP	Local Binary Pattern
LDA	Linear Discriminant Analysis
MCT	Modified Census Transform
NN	Nearest Neighbor
PCA	Principal Component Analysis
RBF	Radial Basis Function
ROC	Receiver Operating Characteristic
SVM	Support Vector Machine

1. Introduction

1.1 Motivation

Humans undertake great efforts to interpret and understand faces of other humans and do so robustly under the most difficult conditions. We are able to reliably recognize a vast amount of known persons even when their face is only partially visible or has undergone changes due to aging or makeup. But faces are more than just a complicated identity card. They are the medium of one of the most important non-verbal communication channels: facial expression.

At a low level, facial expressions subconsciously mirror internal affective states, which often are too complicated to verbalize. While simple emotions like anger or fear are fairly easy to describe, the mix of joy, sadness, and surprise when seeing a close relative after a long period of separation is hard to put into words.

Higher, more conscious levels enable facial expressions to carry subtext in speech and to emphasise certain words. Casual jokes are frequently accompanied with a smirk, the ambiguity in ironic statements can often only be resolved by examining the speaker's face and mimes tell entire stories without using words. Social interaction is depending so much on this channel that a person who loses the ability to interpret facial expressions may be severely impeded in performing every day tasks such as grocery shopping and participation in communal events.

A method to tap into this channel would give rise to numerous applications. Portable computer systems could aid the visually impaired or people on the autistic spectrum to decipher facial expressions. Nurses could be automatically alerted when hospitalized patients show signs of intense pain (Fig. 1.1, see [LBL07]) but are unable to attract attention to themselves.

Service robots could decide to leave a scene if they are not equipped to handle the situation. Car computers could react to the driver's emotional state and limit information flow to the driver accordingly, similar to K.I.T.T. from the 1982 science fiction TV series *Knight Rider*.

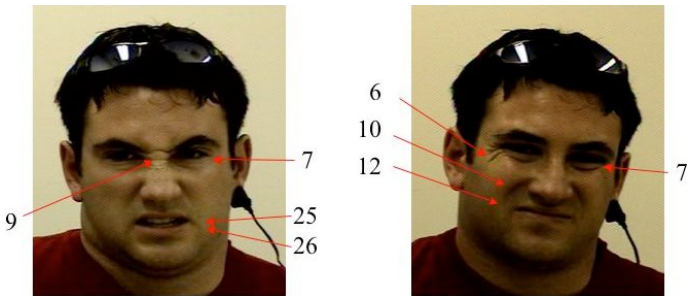


Figure 1.1: Real and faked expression of pain. Image source: [LBL07]

Figure 1.2: The Einstein Project, MPLab at UCSD. Image source: `http://mplab.ucsd.edu/wordpress/?page_id=70`

In classrooms, facial expression analysis systems would direct the attention of a tutor to pupils who are not being able to cope with a task or are working below their capabilities. Littlewort et al. [LBSR11] already developed a system to monitor children's facial expressions while they are solving puzzles of varying difficulty levels. A similar method could be a useful tool for market researchers to evaluate an advertisement's effectiveness. Already *Affectiva* offers a system to measure emotional responses using only a webcam as sensor [Pic11].

The entertainment industry would without a doubt find a number of ways to use facial expression analysis in their products. Dolls that show facial expression to mirror their internal state (Fig. 1.2, [BLFM03]), video game characters that synthesize facial expressions (see Fig. 1.3) and artificial poker-players that are able to call bluffs by analyzing the other player's faces instead of relying on statistics are just some of the applications that come to mind.

1.2 Related Work

1.2.1 Expression and Emotion

Human facial expression has attracted a great deal of attention from the scientific community not only in the field of computer science. As early as 1872, Darwin proposed in his book *Expression of the Emotions in Man and Animals* that emotions have evolved by natural selection and that therefore universal emotions must exist [Dar09]. But it was perhaps the work of Ekman that influenced research on human expression the most. Ekman

Figure 1.3: The *Motion Scan* technology developed for Rockstar Games' *L.A. Noire* transfers facial expressions of real actors to video game characters.

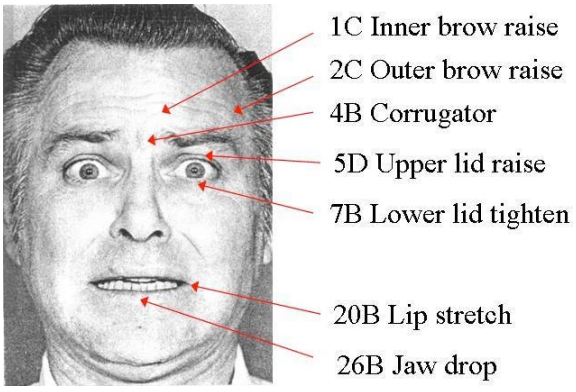


Figure 1.4: A facial expression decomposed using FACS.

conclusively argued for the existence of basic emotions – anger, fear, disgust, joy, sadness and surprise – that show the same facial expressions across different cultures [Ekm99]. In collaboration with Friesen, Ekman developed the *Facial Action Coding System* (FACS) [EF78]. Using FACS, a facial expression is deconstructed into several anatomic *Action Units* (AU), which correspond to different muscle groups. While each action unit has no interpretation other than the contraction or relaxation of a facial muscle, the combination of several AUs can be used to precisely describe an expression. For example, a sad expression is composed of the AUs 1 (inner brow raise), 4 (brow lower) and 15 (lip corner depress).

While AU detection offers a rich and granular method to recognize and analyze facial expressions, the system needs to be trained using FACS-labelled data which is often unavailable. As a consequence, classification systems often focus on Ekman's canonical expressions, sometimes augmented by the neutral, expressionless face. While such an approach is relatively straightforward, it is unable to detect mixed or blended expressions. For comprehensive studies of the challenges involved in expression analysis as well as successful approaches, see for example [PR00], [CDCT$^+$01] and [FL03].

1.2.2 Expression Classification

While there are numerous approaches to facial expression classification, it is the work of Bartlett et al. and Shan et al. that influenced this study the most. Their studies relevant to this work will be briefly discussed in the following paragraphs.

In 1999, Bartlett proposed in cooperation with Ekman, Hager and Sejnowski to automatically analyze facial expressions in image sequences using FACS [BHES99]. Pursuing this line of thought – i.e. utilizing FACS in sequences of images – eventually lead to the development of the *Computer Expression Recognition Toolbox* (CERT) [LWW+11].

CERT is a fully automated real-time facial expression recognition system, able to detect 19 facial actions as well as 6 canonical facial expressions. It achieves real-time performance on a customary laptop computer. Faces are detected using a Viola-Jones face detector and rectified using an affine warp to minimize the L2-distance between 10 canonical facial feature points and the corresponding positions on the input face which are determined using a combination of GentleBoost and linear regression. The preprocessed images are convolved using a Gabor-filter bank of 8 orientations and 9 scales. The magnitudes of the complex filter outputs are concatenated to build the feature descriptor, which is fed into linear support vector machine (SVM) classifiers, where each SVM corresponds to one AU. The distance of the feature vector to the SVM hyperplane encodes the AU intensities for each frame of the video or video stream. Surprisingly, the continuous output is relatively smooth and contains no sudden jumps in AU intensity. In an extension, the intensities are used to detect Ekman's six basic expressions and the neutral face.

Earlier approaches focus on detection of the canonical expressions instead of action units [BLFM03, BLF+05]. Again, faces are represented by convolving the aligned images with Gabor-filters and concatenating the magnitudes of Gabor filter responses. To classify a sample, seven SVMs with linear, polynomial and radial basis function (RBF) kernels perform a one-vs-all binary decision task, where each SVM corresponds to one expression. The SVM output is transformed into a probability distribution using a softmax competition. Using this scheme, linear and unit-width Gaussian kernel SVMs show best performance. In a second experiment, AdaBoost is used to select a subset of all available filter responses. The resulting classifier committee performs comparable to the linear SVM classifier and is significantly faster, since fewer filter responses have to be computed. In a combined approach, features selected by AdaBoost are used as reduced representation for the SVM classifiers. This *AdaSVM* classifier outperforms the naive SVM approach while retaining the speedup of the AdaBoost classifier.

The Gabor feature descriptor used in this thesis (Section 3.1.4) is similar to the one used in [BLF+05], yet differs in the way features are selected. While Barlett et al. select a number individual filter responses, the selection method described in Section 3.4 picks blocks of filter responses that are derived from the same source pixel.

Because of the computational complexity of Gabor filters, Shan et al. employ local binary patterns (LBP) as basis for their experiments [SGM09]. Facial images are extracted from a sequence of images based on automatically detected eye locations and transformed using the extended LBP operator considering 8 samples on a circle of radius 2 to the center pixel. The LBP image is divided into equally sized regions, on which histogram of uniform LBP features are extracted. The concatenation of all histograms is used as feature descriptor

in several machine learning techniques. Template matching is employed as first method. Based on a number of training samples, a prototypical template histogram is assigned to each class. A testing sample is classified by choosing the class of the nearest neighboring template based on the weighted χ^2 distance, $\chi^2(\mathbf{S}, \mathbf{M}) = \sum_{i,j} w_j \frac{(S_{ij}-M_{ij})^2}{S_{ij}+M_{ij}}$, where the region-weights w_j were chosen empirically. A second method uses linear, polynomial and RBF kernel SVMs. For each expression one SVM is trained in a one-vs-all fashion. The class of a testing sample is determined by considering the largest SVM output. This approach yields significantly better results than template matching. Comparison of SVM classifiers based on Gabor features akin to Bartlett's approach shows that LBP features perform marginally better while consuming significantly less time and memory. In another approach the features are projected into a lower dimensional space using principal component analysis (PCA). Classification can be achieved using a nearest neighbor (NN) classifier in a six-dimensional subspace that was found using multiclass linear discriminant analysis (LDA) on the PCA space. A second classification scheme uses linear SVMs on PCA transformed features. This method performs better than LDA+NN, but significantly worse than the SVM classifier described above. Finally a linear programming technique is used to find hyperplanes that separate features of one expression against one other expression (one-vs-one classifiers) with minimal accumulated average misclassification error. The binary classifiers are combined using a voting scheme to produce the final classification result. This technique proves to have a recognition performance comparable to one-vs-all linear SVM classifiers.

Experiments on low-resolution facial images which are typically found in video surveillance and smart rooms indicate a possible performance gain when using arbitrarily positioned windows of varying size instead of the fixed regions to extract LBP histograms. AdaBoost with histogram-based template matching as weak classification is used to select the most discriminative among all possible windows. The resulting committee shows a performance gain of 5.9% compared to template matching using fixed regions. Consequently, SVM classification using the boosted LBP features outperforms both non-boosted SVM classification and boosted template matching. The LDA classifier, too, benefits from boosting, but is still inferior to SVM based classification.

Both the research of Bartlett et al. [BLF+05] and Shan et al. [SGM09] show the potential benefit of AdaBoost based feature selection. This result will be confirmed in this study.

1.3 Overview

The remainder of this thesis is organized as following: Chapter 2 addresses the problem of suitable training and evaluation databases and establishes a novel dataset focusing on images collected from the internet. Chapter 3 prepares the theoretical background for this thesis. It introduces three feature extraction methods based on the discrete cosine transform, local binary patterns and Gabor filters and gives a short introduction on support vector machines and the AdaBoost machine learning algorithm. Building on that, a boosting based key point selection method is developed. In Chapter 4 the different methods discussed in Chapter 3 are put to the test using a modular evaluation framework. Chapter 5 closes by discussing the results and offering further research opportunities.

2. The Web Image Database

Two frequently used databases utilized for facial expression analysis are the Cohn-Kanade [KCT00] and GEMEP-FERA [BS10] datasets. The Cohn-Kanade corpus consists of videos of 100 university students, most of which are women. The subjects have an age in the range of 18 to 30 years and belong to different ethnicities. Instructed by a researcher, the students perform a series of 23 different facial expressions, while each time starting from a neutral state. The GEMEP-FERA dataset is a subset of the Geneva Multimodal Emotion Portrayals (GEMEP) dataset created for the Facial Expression Recognition and Analysis (FERA) challenge. It features ten actors displaying a range of different expressions while being engaged in a meaningless dialog. In contrast to the Cohn-Kanade dataset, each image sequence contains more than one expression.

However, both databases suffer from artificial circumstances. The image resolution, lighting conditions and head pose are chosen relatively generously towards recognition systems. A classifier trained on these datasets may not perform as well as presumed in uncontrolled situations, such as video surveillance, TV broadcasts and web or magazine images.

To account for this issue, a different dataset is needed. Key requirements are:

1. Coverage of the canonical expressions with varying intensities and the neutral face.
2. Inclusion of a great number of male and female subjects.
3. Inclusion of different age groups and ethnicities.

Such a database could be built using web images, since the Internet grants access to an enormous amount of images. *Facebook* allows its users to upload photos and tag their friends on them. Amateur photographers employ *Flickr* to show their work and members of the community may assign arbitrary tags to uploaded photos. Services like *stock.xchng* offer a platform to share and trade stock photos. Personal web space has become cheap and enables users to present images in a personalized way. The problem reduces to how to select candidate images from these sources. Fortunately, Google images provides a vast, search- and filterable web image index that can be used to compile a list of initial images.

2.1 Data Acquisition

Google's image search tool is accessible by issuing a HTTP GET request [BLFF96] to a special URL that encodes the search options. It must take the following form:

```
http://images.google.com/images?q=<query>&start=<result-number>&<options>
```

`<query>` contains the URL-encoded search term, `<result-number>` defines a starting index – Google delivers the query results broken down into chunks of 20 matches per request – and `<options>` contains further options to limit the search. For example, `tbs=itp:face` limits the results to contain mostly images of faces.

The search term is built by combining two words describing the target expression and the subject. Ten words were used to describe the gender and age of the subject – "Baby", "Boy", "Child", "Elderly", "Girl", "Grandfather", "Grandmother", "Man", "Person" and "Woman" – as well as the generic search term "Face".

For each of the seven target expressions, four to eight adjectives describing different intensities of an expression were manually selected using the wordnet database [Fel98] by searching for words related to the expression. Table 2.1 lists the describing adjectives.

Each combination of subject and expression description was used to fetch 200 results. Along with the image URL, Google images returns other metadata: the image dimensions and file size, the URL of the containing document, the text in the document that matched the query and an URL to a downscaled version of the image. Under normal conditions, i.e. when using a web-browser, the search results are presented as HTML formatted document. However, when transmitting an empty `User-Agent` header field, the result is encoded using JSON [Cro06], which is much easier to parse than HTML. All available metadata was recorded to a file. Before fetching the images, duplicate URLs were removed to filter out obvious duplicate images. This resulted in the download of more than 80000 pictures.

The initial image database contained a large number of exact or near duplicates and suffered from other defects shown in Figure 2.1. While the first results of a given query

Expression	**Describing adjectives**
Anger	aggravated, angry, raging, smoldering
Disgust	disgusted, displeased, fed, frowning, nauseated, repelled, scowling
Fear	afraid, alarmed, dreaded, dreadful, fearful, fearsome, frightened
Joy	cheering, euphoric, felicitous, happy, joyful, laughing, smiling
Neutral	indifferent, inert, neutral, uncharged
Sadness	depressed, distressed, melancholic, mourning, sad, sorrowful, weeping
Surprise	amazed, astonished, astounded, dumbfounded, flabbergasted, startled, stunned, surprised

Table 2.1: Words used to describe the target expression of the search query.

(a) False positives ("Anger"). (b) Obstructing watermarks. (c) Partial face.

Figure 2.1: Defects of the initial image database.

generally contained the desired expression, the number false positives increased drastically when more items were taken into account (Fig. 2.1(a). In addition, a large number of images are watermarked, which is problematic when the watermark covers the face so that the expression is not visible, for example in Figure 2.1(b). As depicted in Figure 2.1(c), some images show only partial faces and are unusable. Other less frequent defects included altered pictures, drawings, animal faces and corrupted, unreadable files.

2.2 Filtering and Labelling

The process of discarding these defective images and labelling the remaining faces was broken down into four steps.

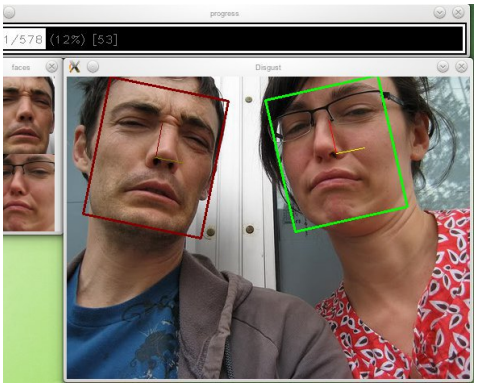
The first step was to decide whether a given image belongs to the target expression or not. A simple user interface shown in Figure 2.2(a) aided this process. It displayed the image in question, the target expression alongside with the matched string and the current labelling progress. Pressing of a key either accepted or rejected the current image or revoked the last decision. With this simple user interface it was possible to classify up to three images per second.

In the second filtering step rotated bounding boxes were used to label the faces in preparation of the third filtering step (Fig. 2.2(b)). At the same time, false positives that were not removed in the first step were discarded.

In the next step the enhanced discrete cosine transform (DCT) feature described in Section 3.1.2 was computed for each face using these bounding boxes. A list of possible duplicates was compiled according to the L1-distance of the current feature to the features of every other image (Fig. 2.2(c)). Images to discard had to be manually selected from that list.

In the fourth and final step the eye locations were semi-automatically labeled using a modified census transform (MCT) detector [FE04]. To account for erroneous detections, manual labelling was prompted when the annotated bounding box rotation angle deviated too much from the rotation angle computed using the supposed eye positions, i.e. when $\mathbf{v}^T\mathbf{u} > \tau\|\mathbf{v}\|\|\mathbf{u}\|$, where $\mathbf{v}$ denotes the vector from left to right eye, $\mathbf{u}$ is the vector of the manually labelled bounding-box pointing upward, and τ is a user-defined threshold (Fig. 2.2(d)).

(a) Binary decision task.



(b) Marking with rotated bounding boxes.

(c) Semi-automatic duplicate removal.

(d) Semi-automatic eye tagging.

Figure 2.2: User interfaces for the different labelling steps.

2.3 Resulting Database

The filtering reduced the more than 80000 images to a database of 4761 tagged faces with marked eye positions. However, the expression do not feature an equal amount of entries. Table 2.2 shows that nearly half of the faces show a joyful expression, while ”Fear” features the least amount of samples. This imbalance has to be taken into account when training a classifier on this dataset. Sample images are shown in Figure 2.3.

Expression	**Number of samples**
Anger	648
Disgust	368
Fear	288
Joy	2185
Neutral	388
Sadness	327
Surprise	557
Sum	4761

Table 2.2: Number of tagged faces by expression.

(a) Anger (b) Disgust

(c) Fear (d) Joy

(e) Sadness (f) Surprise

(g) Neutral

Figure 2.3: Sample images from the final database.

The resulting database shows all the desired properties: It contains samples of seven basic expressions performed by a large number of subjects of different gender, age and ethnic groups. The photographs were taken under varying lighting conditions and depict their subject with non-standardized head poses. However, because a lot of the pictures are stock photos, the expressions are mostly artificial and not spontaneous. In some cases the faces are partially occluded by watermarks, which may be considered a drawback. At the same time, this restriction can be viewed as an advantage, e.g. when the goal is to build an expression classifier for web and magazine images, in which such occlusions are natural.

Apart from the raw images with marked eye positions, the database contains other meta information (see Section 2.1) that could be used to build a multi-modal classifier that uses computer vision techniques as well as linguistic models. However, such a system is not in the scope of this work.

3. Theoretical Background

The process of image classification – regardless of the classification subject and categories – can be divided into two sub-problems: Feature description and machine learning. A Feature descriptor is a reduced representation of an image that ideally retains only the information needed to discriminate different classes. The machine learning algorithm finds criteria to separate these classes by previously examining a number of samples. An excellent in-depth description of several popular machine learning algorithms can be found in [Bis06a]. This chapter will describe the theoretical aspects necessary to understand the classification framework used in this thesis.

3.1 Feature Description

A good feature descriptor should describe a given sample using the most expressive information available. In the context of facial image processing, this often means to describe a face in terms of spatial relationships. In the following, three different feature extraction strategies based on the discrete cosine transform, local binary patterns and Gabor filters are presented. The performance of each method is evaluated in Section 4.4.

3.1.1 Preprocessing

Regardless of the extraction strategy, all input images undergo the same pre-processing steps to reduce the variability in the feature descriptors and thus increase the classification

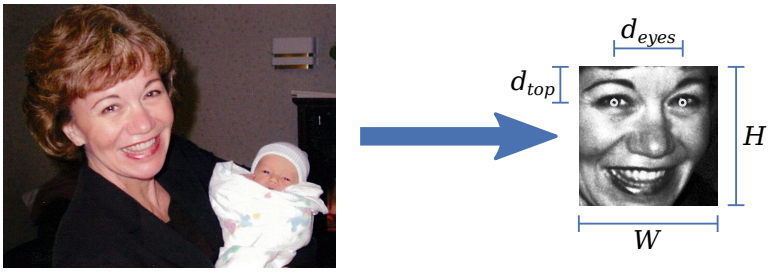


Figure 3.1: Image processing in preparation of feature extraction.

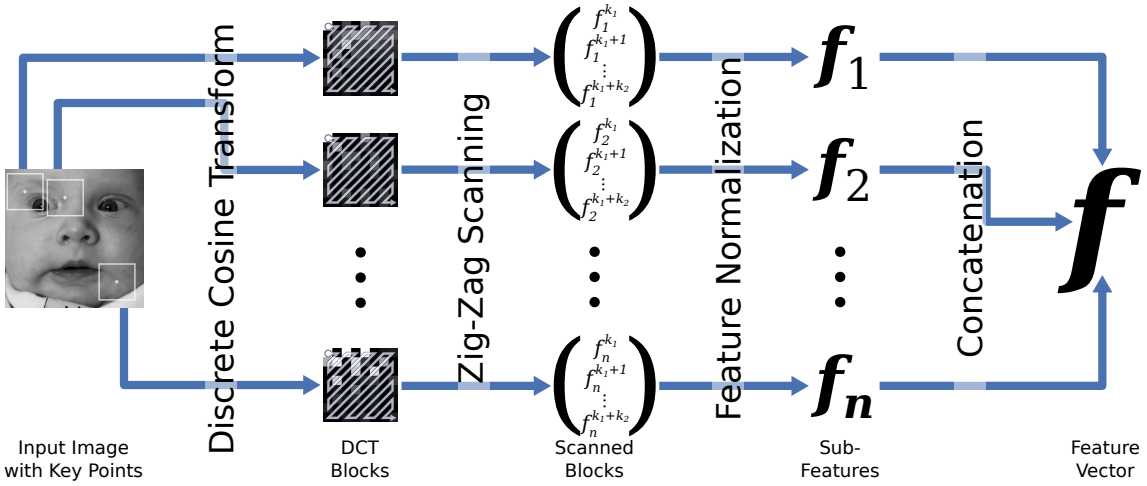


Figure 3.2: Feature extraction pipeline of the Enhanced DCT descriptor.

performance of the machine learning algorithm.

Since all of the following methods focus only on textural and spatial information, the first step is to remove color information. The image is then transformed using an affine warp so that eye centers have a defined distance to each other. The transformation scales and rotates, but does not shear or deform the face in any other way. A rectangular area covering the face is extracted based on the eye positions. In the last step, the image contrast is adjusted using histogram equalization to reduce the impact of different lighting conditions on the classification result.

3.1.2 Enhanced DCT Feature

The discrete cosine transform is the basis of the first feature extraction method. The key point based formulation can be interpreted as a generalized formulation of an earlier approach by Ekenel [Eke09].

The DCT is a special real-valued case of the discrete Fourier transform, which describes a signal as the sum of complex waves with different frequency and amplitude. The two dimensional DCT of an $m \times m$ image block X is given by

$$Y(u,v) = \alpha(u)\alpha(v) \sum_{y=0}^{m-1} \sum_{x=0}^{m-1} X(x,y) \cos\left[\frac{(2x+1)u\pi}{2m}\right] \cos\left[\frac{(2y+1)u\pi}{2m}\right], \tag{3.1}$$

where m is even, $u, v \in [0, m-1]$ and

$$\alpha(w) = \begin{cases} \sqrt{\frac{1}{m}} & \text{if } w = 0, \\ \sqrt{\frac{2}{m}} & \text{otherwise.} \end{cases} \tag{3.2}$$

Transforming the whole image at once to obtain a feature descriptor would dismiss a large amount of spatial information. To preserve those relationships, multiple transformations

of regions around certain key points are considered. The regions $\mathcal{R}_i$ are defined by a center point (x_i, y_i) and a radius r and measure $2r \times 2r$ pixels, i.e.

$$\mathcal{R}_i = [x_i - r, x_i + (r-1)] \times [y_i - r, y_i + (r-1)]. \tag{3.3}$$

The DCT coefficients for each region $\mathcal{R}_i$ are computed using eq. (3.1) and recorded into a list by applying a zig-zag scan. The lower and higher frequency components are discarded by dropping the first k_1 and collecting the next k_2 coefficients. The remaining coefficients f_j^i are recorded into the feature block $\mathbf{f}_i = (f_i^{k_1+1}, \ldots, f_i^{k_1+k_2})^T$.

In a first and optional normalization step the coefficients f_i^j are normalized by their standard deviation $\sigma(f^j)$, i.e.

$$f_i^j \leftarrow \frac{f_i^j}{\sigma(f^j)}. \tag{3.4}$$

This step balances the individual coefficients' impact on the classification. The $\sigma(f^j)$ were taken from [Eke09] and not learned from the dataset described in Chapter 2 due to time constraints.

In a second normalization step the coefficient block is scaled to unit length, to balance each block's impact on the classification result:

$$\mathbf{f}_i \leftarrow \frac{1}{\|\mathbf{f}_i\|}\mathbf{f}_i. \tag{3.5}$$

Finally, the feature vectors from each block $\mathbf{f}_i$ are concatenated to form the $n \cdot k_2$ dimensional feature vector $\mathbf{f}$, where k_2 is the number of coefficients extracted from each block and n is the number of considered regions. Note that it is not specified how the key points are obtained. One approach is to algorithmically find points that maximize the content of discriminative information. One such method is described in Section 3.4.

Another approach places the key points on a regular grid so that the whole image is covered by non-overlapping regions. Doing so will result in the same local appearance based feature descriptor employed by Ekenel [Eke09]. Figure 3.2 .llustrates the feature extraction procedure.

3.1.3 LBP Feature

Local Binary Patterns were originally introduced by Ojala et al. as a device for texture analysis [OPH96], but it has been shown that they can successfully be applied in the context of facial image processing [AHP04, AHP06, SGM09].

The basic LBP operator assigns a label to each pixel by thresholding a 3×3 neighborhood around the input pixel and interpreting the results as an eight bit number. Figure 3.3 illustrates this idea. More formally, with $(x_i, y_i) \in \mathcal{N}(x, y)$ denoting the pixel neighborhood of (x, y) and

$$S[p] = \begin{cases} 1 & \textit{if } p \textit{ is } \text{true} \\ 0 & \textit{otherwise}, \end{cases} \tag{3.6}$$

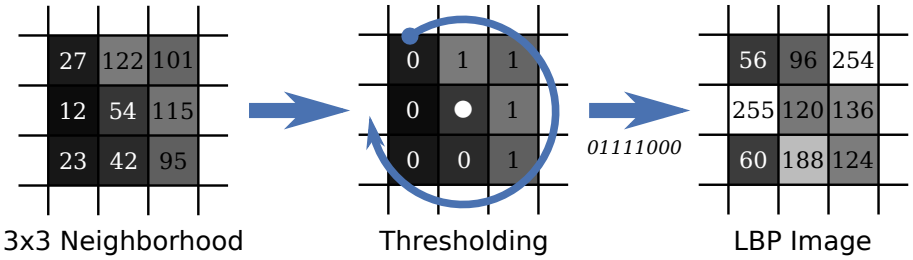


Figure 3.3: The basic LBP operator.

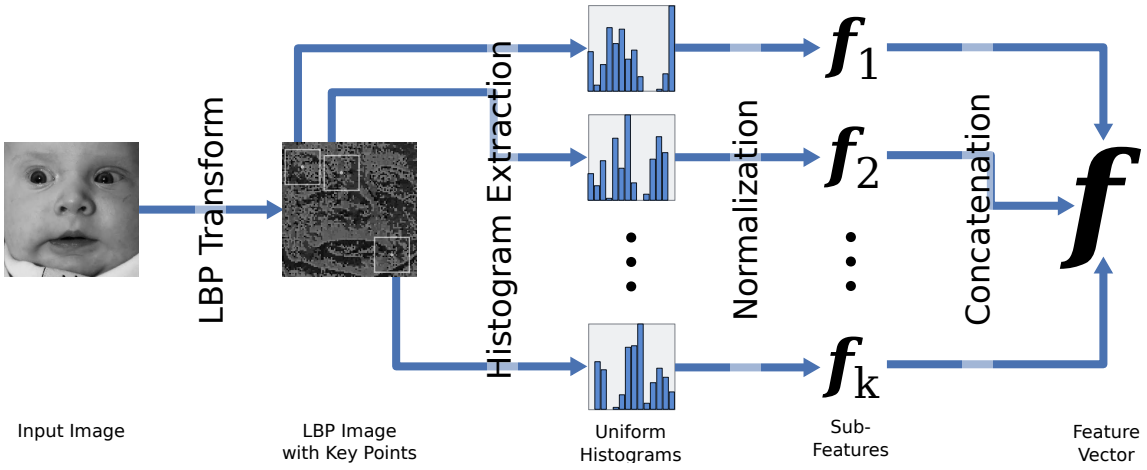


Figure 3.4: Feature extraction pipeline of the Local Binary Pattern descriptor.

the LBP labels are computed as:

$$I_{LBP}(x,y) = \sum_{i=0}^{7} S[I(x,y) \leq I(x_i,y_i)]2^i. \tag{3.7}$$

The labels can be interpreted as being 256 different textons that code small scale structures such as corners, edges and blobs. A feature descriptor can be built by computing the histogram $H(k) = \sum_{x,y} S[I_{LBP}(x,y) = k], k = 0, \ldots, 255$ of the LBP labels. An extended, spatially enhanced descriptor that captures large scale spatial relationships consists of several histograms, each computed on different regions of the image.

In a later publication, Ojala et al. extended the basic operator to be able to capture larger structures [OPM02]. Instead of only considering a 3×3 neighborhood, P equally spaced pixels on a circle with radius R are thresholded against the center pixel. Because this formulation allows for non-integer pixel coordinates, bi-linear interpolation is used to compute the brightness values. The basic LBP operator is an instance of the general operator with $P = 8$ and $R = 1$.

A second expansion introduced *uniform patterns*. A binary pattern is called uniform, if it and all its circular shifts contain at most two transitions from 0 to 1 and vice versa. For example, the patterns 11111111 (no transition), 00011110 (two transitions) and 11100000 (two transitions, considering the circular shifts) are uniform, whereas 11010011 (four transitions) and 01011101 (six transitions, again considering the circular shifts) are not.

Ojala et al. noticed that in their experiments uniform patterns accounted for nearly 90% of all occurrences [OPM02]. Ahonen et al. confirmed these findings with experiments on the FERET database [AHP06, PWHR98]. This suggests to reduce the histogram descriptor by assigning the same label to all non-uniform patterns and thus collect all those patterns into the same bin. The resulting uniform histogram consists of only 59 bins, but contains nearly the same amount of information as the 256-bin histogram.

The feature descriptor used in this system is built by first transforming the input image with the basic LBP operator (i.e. $(P, R) = (8, 1)$) and then computing the histograms of uniformly labeled LBP images $I^u_{LBP}(x, y)$ in regions $\mathcal{R}_i$ defined by equation (3.3):

$$H^u_i(k) = \sum_{(x,y)\in\mathcal{R}_i} S\left[I^u_{LBP}(x, y) = k\right], \quad k = 0, \ldots, 58. \tag{3.8}$$

The histograms are normalized so that $\sum_k |H^u_i(k)| = 1$ and concatenated to form a spatially enhanced feature descriptor. Figure 3.4 illustrates this process.

3.1.4 Gabor Feature

In 1985 Daugman was able to show that cells of the human visual cortex could be modelled using Gabor filters [Dau85]. Despite their relatively high computational complexity, they have since been applied numerous times in computer vision showing remarkably good results.

The impulse response of a Gabor filter is the product of an uni-directional complex harmonic function and a Gaussian:

$$g(x, y; f, \sigma_x, \sigma_y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right) + (2\pi f x)\, i\right]. \tag{3.9}$$

The parameter f defines the frequency of the complex wave, whereas σ_x and σ_y control the ellipticity of the Gaussian. Figure 3.5 shows the plot of the real and imaginary part of the function.

A family of filters $g_{mn}(x, y)$ can be derived from equation (3.9) by systematically applying different filter-scales and rotation angles. With $m = 0, 1, \ldots, M-1$ denoting the scale and $n = 0, 1, \ldots, N-1$ denoting the orientation, the impulse response is computed as

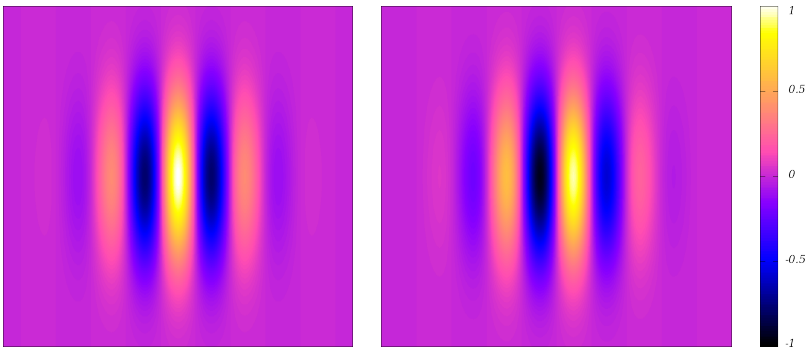
$$g_{mn}(x, y; f, \sigma_x, \sigma_y) = a^{-2m} g(\tilde{x}, \tilde{y}; f, \sigma_x, \sigma_y), \tag{3.10}$$

where

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} = a^{-m} \begin{pmatrix} \cos\frac{n\pi}{N} & -\sin\frac{n\pi}{N} \\ \sin\frac{n\pi}{N} & \cos\frac{n\pi}{N} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{3.11}$$

are the rotated and scaled pixel coordinates.

The Gabor transformation on an image $I(x, y)$ is the convolution of $I(x, y)$ with all filters of a given family. The resulting $M \cdot N$ complex output images denoted by $\mathcal{G}_{mn}(x, y)$ are often reduced to real-valued magnitude images $\|\mathcal{G}_{mn}(x, y)\|$, since the phase $\arg(\mathcal{G}_{mn}(x, y))$ only carries little additional information.



(a) Real part. (b) Imaginary part.

Figure 3.5: Plot of the Gabor function $g(x, y; f, \sigma_x, \sigma_y)$.

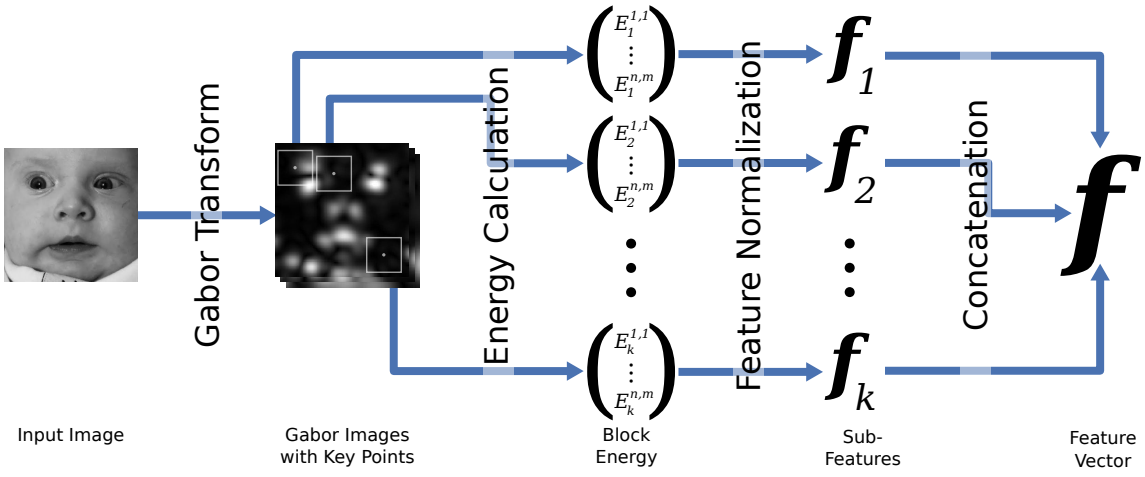


Figure 3.6: Extraction pipeline of the Gabor filter feature descriptor.

The Gabor feature descriptor is built by concatenating several sub-features $\mathbf{f}_i$. Each $\mathbf{f}_i$ corresponds to a region $\mathcal{R}_i$ as defined in eq. (3.3) and is the collection of the energy content in that region given the Gabor images of N scales and M orientations:

$$\mathbf{f}_i = [E^i_{0,0}, E^i_{0,1}, \ldots, E^i_{M-1,N-1}]^T, \tag{3.12}$$

where

$$E^i_{m,n} = \sum_{(x,y)\in\mathcal{R}_i} \|\mathcal{G}_{mn}(x, y)\|. \tag{3.13}$$

The sub-features are normalized so that $\|\mathbf{f}_i\|_{L2} = 1$ and concatenated to build the final descriptor $\mathbf{f}$. The feature extraction process is depicted in Figure 3.6.

The Gabor feature descriptor used by Bartlett [BLFM03] and that is often found in literature considers only the filter responses of some pixels of the input image. This descriptor can be derived from the energy-content formulation by considering regions of only one pixel, i.e. $\mathcal{R}_i = \{(x_i, y_i)\}$.

3.2 AdaBoost

Boosting is a simple yet remarkably powerful machine learning technique that combines several classifiers to a committee, whose joint performance can be significantly better than the individual classifiers. The most popular boosting algorithm is *AdaBoost* (for "Adaptive Boosting") which was developed by Freund and Schapire for solving classification problems [FS95, FS96]. Since then it has been subject to numerous extensions, such as the application to regression [Fri00].

The key idea of AdaBoost is to iteratively select classifiers from a set of *weak classifiers* that barely perform better than chance, so that each new classifier reduces the classification error with respect to a distribution. After each selection round, the distribution is updated to give weight towards samples that have been mis-classified by the selected classifier. Classifiers selected in the following rounds will be chosen to correctly classify these samples and thus reduce the overall classification error.

Algorithm 1 shows the implementation of AdaBoost for a two label classification problem. Given a training set $\mathcal{T} = \{(x, y)|x \in \mathcal{X}, y \in \{-1, 1\}\}$ of labeled features and a set of weak classifiers $\mathcal{H} = \{h : x \to h(x) = \pm 1\}$, the algorithm computes weights ω_t for the most discriminating classifiers $h_t \in \mathcal{H}$ in T iterations. Once finished, the class of a feature $x \in \mathcal{X}$ is determined using the sign of

$$H(x) = \sum_{t=1}^{T} \omega_t h_t(x). \tag{3.14}$$

Algorithm 1 The AdaBoost algorithm.

Input: Number of iterations T

Input: Training set $\{(x_i, y_i)|i = 1, \ldots, m,\ x_i \in \mathcal{X},\ y_i \in \{-1, +1\}\}$

Input: Weak classifiers $h_i \in \mathcal{H}, h_i : x \in \mathcal{X} \to h(x) = \pm 1$

Output: Committee $(h_t, \omega_t) \in \mathcal{H} \times \mathbb{R}, t = 1, \ldots, T$

1: $W_1(i) \leftarrow \frac{1}{m}, \quad i = 1, \ldots, m$

2: **for** $t = 1$ to T **do**

3: $\quad h_t = \arg\min_{h_t \in \mathcal{H}} \sum_{i=1}^{m} W_t(i)[h_t(x_i) \neq y_i]$

4: $\quad \omega_t = \Psi(\underbrace{\sum_{i=1}^{m} W_t(i)[h_t(x_i) \neq y_i]}_{classification\ error})$

5: $\quad W'_{t+1}(i) = W_t(i) \exp(-\omega_t y_i h_t(x_i)), \quad i = 1, \ldots, m$

6: $\quad W_{t+1}(i) = W'_t(i)/Z_t, \quad Z_t = \sum_{i=1}^{m} W'_t(i)$

7: **end for**

The weight update $\Psi(x)$ in line 4 of the algorithm can be chosen freely, provided that

$$\omega_t y_i h_t(x_i) = \begin{cases} > 0, & if\ h_t(x_i) = y_i \\ < 0, & if\ h_t(x_i) \neq y_i. \end{cases} \tag{3.15}$$

A commonly used function that rewards classifiers with low mis-classification rate is

$$\Psi(x) = \frac{1}{2} \ln \frac{1-x}{x}. \tag{3.16}$$

If the weak classifiers do not change their classification result while boosting, the algorithm can be optimized by computing an error-matrix $(e_{ik})_{n \times m}$ that encodes classification results for each classifier $h_1, \ldots, h_n$ and training sample $(x_1, y_1), \ldots, (x_m, y_m)$,

$$e_{ik} = \begin{cases} 1, & \text{if } h_i(x_k) \neq y_k \\ 0, & \text{if } h_i(x_k) = y_k. \end{cases} \tag{3.17}$$

Finding the classifier h_t and weight ω_t in line 3 and 4 can be achieved by finding the smallest element of the error-vector

$$\mathbf{err}_t = \begin{pmatrix} e_{11} & e_{12} & \ldots & e_{1m} \\ e_{21} & e_{22} & \ldots & e_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ e_{n1} & e_{n2} & \ldots & e_{nm} \end{pmatrix} \begin{pmatrix} W_t(1) \\ W_t(2) \\ \vdots \\ W_t(m) \end{pmatrix}. \tag{3.18}$$

3.3 Support Vector Machines

Support vector machines are a class of supervised, discriminative machine learning methods. Given a set of labeled features $\mathcal{T} = \{(\mathbf{x}_i, y_i) | \mathbf{x}_i \in \mathcal{X}, y_i \in \{-1, 1\}\}$, a SVM finds a hyperplane

$$\mathbf{h} = \mathbf{w}^T \phi(\mathbf{x}) + b \tag{3.19}$$

that maximizes the margin $2\|\mathbf{w}\|$ between the $\phi(\mathbf{x}_i)$ with $y_i = -1$ and $y_i = 1$ respectively, i.e. the hyperplane that separates both classes and has the largest distance to the features subject to a mapping ϕ.

3.3.1 Linear Classifiers

Assuming that $\phi(\mathbf{x}) = \mathbf{x}$ and that the classes in $\mathcal{X}$ are linearly separable, $\mathbf{h}$ can be determined by solving the optimization problem

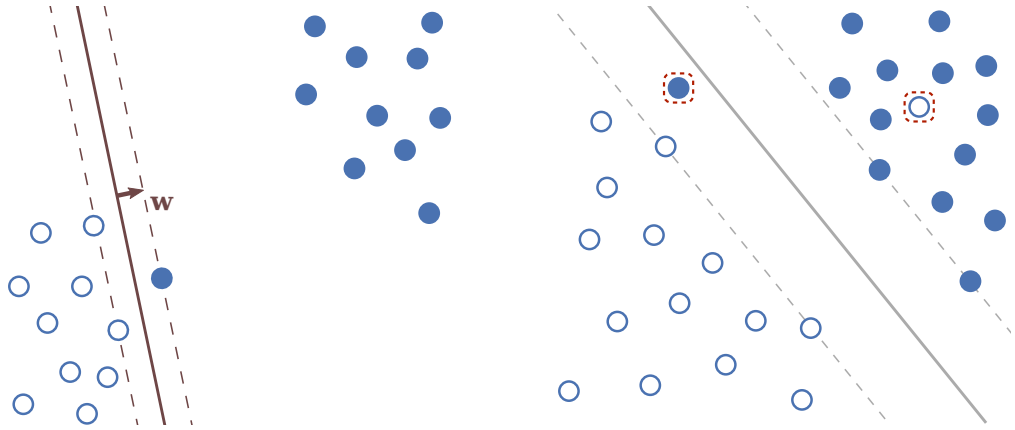
$$\arg\min_{\mathbf{w},b} \frac{1}{2} \|\mathbf{w}\|^2 \tag{3.20}$$

subject to the constraints

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1. \tag{3.21}$$

Since $\mathcal{X}$ is linearly separable, this quadratic programming problem has exactly one global solution, producing a *hard margin* classifier.

Figure 3.7 illustrates that such a solution may not always be desirable. Noisy data can result in too narrow margins and thus poor generalization performance. Overlapping, non linearly separable classes even prevent a solution. To deal with this problem, Cortes



(a) Poor generalization. (b) Linearly inseparable data.

Figure 3.7: Problems of hard margin classifiers with noisy data.

and Vapnik relaxed the conditions in equation (3.21) by introducing a slack parameter ξ_i [CV95]:

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i. \tag{3.22}$$

ξ_i can be interpreted as measuring the classification error of sample x_i. $\xi_i = 0$ corresponds to the original constraints and means that sample was classified correctly and lies outside the margin. If $0 < \xi_i < 1$ the sample was classified correctly, but lies inside the margin, and $\xi_i > 1$ implies that x_i was classified incorrectly. The optimisation problem is extended to penalize classification errors:

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_i. \tag{3.23}$$

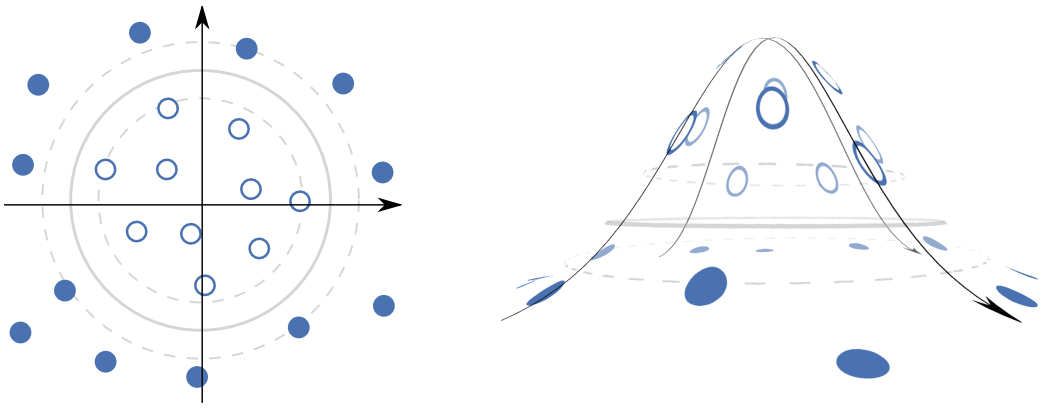
The parameter $C > 0$ controls the trade-off between minimizing the classification error and obtaining a large margin.

3.3.2 Nonlinear Classifiers

Some data may be linearly inseparable not due to noisy labels, but as intrinsic property of the dataset, as shown in Figure 3.8. Boser et al. suggest to apply the kernel trick to create a nonlinear classifier that successfully separates the data [BGV92]. The features $\mathbf{x_i}$ are transformed to a different feature space $\mathcal{H}$ using some mapping $\phi : \mathcal{X} \mapsto \mathcal{H}$. Using Lagrange multipliers, the SVM algorithm can be reformulated so that the mappings only appear in scalar products $\langle\phi((\mathbf{x}_i), \phi(\mathbf{x}_k)\rangle$, which can be computed implicitly using a kernel function

$$k(\mathbf{x}_i, \mathbf{x}_k) = \langle\phi(\mathbf{x}_i), \phi(\mathbf{x}_k)\rangle. \tag{3.24}$$

The feature space $\mathcal{H}$ becomes an implicit property of the kernel, which makes it possible to use very high or even infinite dimensional feature spaces and thus arbitrary decision boundaries with only little additional computation cost due to the dual representation. Some common kernel functions are listed in Table 3.1. For a detailed discussion of support vector machines and the kernel trick see [Bis06b].



(a) Linearly inseparable data.

(b) Linearly separable when mapped to a higher dimension.

Figure 3.8: Illustration of the kernel trick in context of SVM classifiers.

Description	**Kernel Function**
Linear	$k(\mathbf{x}_i, \mathbf{x}_k) = \mathbf{x}_i^T \mathbf{x}_k$
Polynomial	$k(\mathbf{x}_i, \mathbf{x}_k) = (\gamma \mathbf{x}_i^T \mathbf{x}_k + c)^d$
Radial Basis Function	$k(\mathbf{x}_i, \mathbf{x}_k) = \exp\{-\gamma(\mathbf{x}_i^2 + \mathbf{x}_k^2 - 2\mathbf{x}_i^T \mathbf{x}_k)\}$
Sigmoid	$k(\mathbf{x}_i, \mathbf{x}_k) = \tanh(\gamma \mathbf{x}_i^T \mathbf{x}_k + c)$

Table 3.1: Popular kernel functions used with support vector machines.

3.4 Key Point Selection

As mentioned in Section 3.1, a good feature extraction method should select the most discriminative information to build a descriptor. In case of the descriptors used in this thesis, maximising information content is equivalent to choosing key points that add discrimination criteria to the descriptor.

This process can be formulated as following: First every pixel of the input image is considered to be a key point. The resulting feature descriptor

$$\mathcal{F} = (\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{W \times H})^T \tag{3.25}$$

consists of sub-features $\mathbf{f}_i$, where $\mathbf{f}_1$ corresponds to the key point $(0, 0)$, $\mathbf{f}_1$ to $(1, 0)$, and so on. In total, there are $W \times H$ sub-features corresponding to the $W \times H$ pixels of the input image. Because this allows for regions to cover pixels outside of the image area, the preprocessing step in Section 3.1.1 has to be modified to include a border of size r.

The problem of selecting the most discriminative $\mathbf{f}_i$ can be solved by associating classifiers h_i to each $\mathbf{f}_i$. Using a base classifier $h(\mathbf{x}) = \pm 1$, this mapping can be defined by forwarding the corresponding sub-feature to h, i.e.

$$h_i(\mathcal{F}) = h(\mathbf{f}_i). \tag{3.26}$$

Using the h_i as weak classifiers, AdaBoost can be used to select a committee $h_t, t = 1, \ldots, n$ of the n most discriminative classifiers, which can be mapped back to n most discriminative

sub-features and thus to relevant key point locations. Note that because AdaBoost requires the weak classifiers to perform better than chance, this scheme can only work if the base classifier h meets this requirement. Other than that, the implementation of h may be chosen freely, although the computational complexity of the method suggests to choose classifiers that are able to provide fast decisions, e.g. probabilistic models or linear SVMs.

4. Evaluation

The previous chapter introduced several feature extraction methods and machine learning algorithms suitable for facial expression analysis. In this chapter these methods are evaluated using the image database described in Chapter 2.

4.1 Cross-Validation

Cross-validation is a technique to thoroughly evaluate a classification system using a limited amount of data in several rounds. In each round, the ground truth is partitioned into two complementary sets. The first set is used to train the classifier while the remaining data is used for evaluation. In order to increase variation, different partitions are chosen each round.

In *k-fold cross-validation*, the initial set is divided into k partitions of equal size. In the i^{th} of k rounds, partition i is used as evaluation data, while the remaining $k-1$ are used for training. In case that the amount of ground truth is highly limited, it can be useful to split the data into n partitions, where n is the number of samples, and proceeding as with k-fold cross-validation. This is known as *leave-one-out cross-validation*.

4.2 Metrics

Several quantitative measurements are available to evaluate classification systems in an objective and systematic way.

4.2.1 Two Class Problem

Given a two class problem, most metrics can be derived from the confusion matrix shown in Table 4.1.

		actual class A	actual class B
prediction	A	TP (true positive)	FP (false positive)
prediction	B	FN (false negative)	TN (true negative)

Table 4.1: Confusion matrix for two class classification.

The fraction of samples that have been classified correctly is known as *accuracy*,

$$\text{accuracy} = \frac{TP + TN}{TP + FP + FN + TN}. \tag{4.1}$$

High accuracy indicates that the underlying classification model is a good approximation of reality. However, high accuracy does only indicate high quality of all predictions.

Precision measures the positive prediction quality of a classifier as the fraction of accepted relevant samples,

$$\text{precision} = \frac{TP}{TP + FP}. \tag{4.2}$$

High precision indicates a low probability of false positive predictions. Statements about the classification completeness cannot be derived. The measure of completeness, i.e. the fraction of positive samples that have been accepted by the classifier, is called *true positive rate* or *recall*,

$$\text{recall} = \frac{TP}{TP + FN}. \tag{4.3}$$

Since 100% recall can be achieved by simply accepting every sample, the true positive rate is not a sound quality measure on its own. However, in combination precision and recall are powerful means to evaluate classification systems.

The harmonic mean of precision and recall, i.e.

$$F_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{4.4}$$

is known as F_1 *score*. It can be interpreted as an accuracy measure that focuses on true positive while neglecting true negative predictions.

Closely related to the true positive rate is the *false positive rate*, which is the fraction of incorrectly accepted negative samples,

$$\text{fpr} = \frac{FP}{FP + TN}. \tag{4.5}$$

Plotting the true positive rate against the false positive rate while varying a discrimination threshold yields the *receiver operating characteristic* (ROC) curve. The ROC curve is a

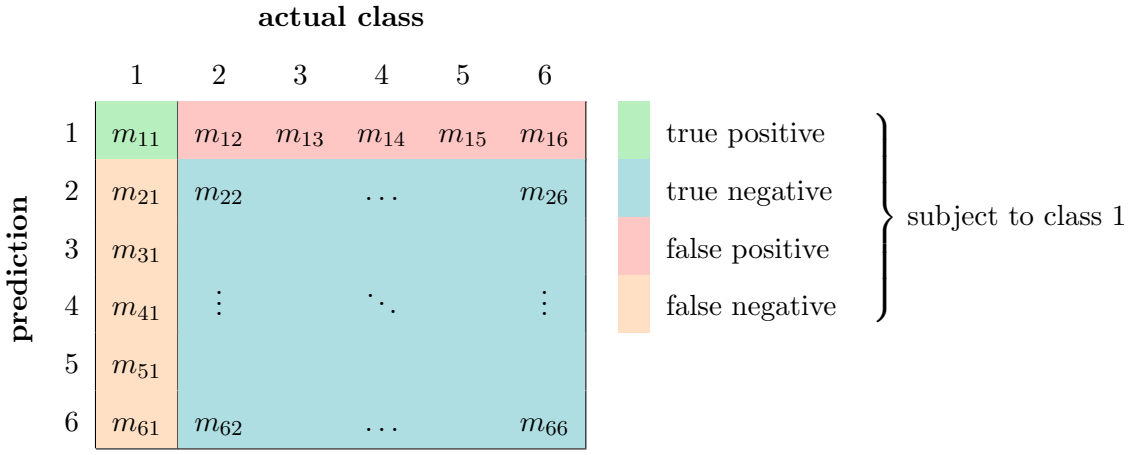


Table 4.2: A 6-class confusion matrix.

valuable tool to intuitively assess the quality of a classifier: A curve converging to the diagonal $y = x$ (the *no-discrimination-line*) attests a classification performance close to random guessing. A good classifier on the other hand will show a curve that leans towards the perfect classification point $(0, 1)$, where there are no faulty predictions. A curve leaning towards the point $(1, 0)$, i.e. a curve lying under the no-discrimination-line means that the classifier performs worse than random. Such a classifier can be converted into a good classifier by inverting the classification criterion.

4.2.2 Multi-Class Problem

The presented metrics only consider the two-class problem, but expression recognition is a multi-class problem. Re-examining Table 4.1 and the definitions of precision and recall suggest a simple extension. The confusion matrix M is built by considering the classifications results $H(\mathbf{x}) = h$ of a sample $\mathbf{x}$ given its class $C(\mathbf{x}) = c$. Each cell m_{hc} of the confusion matrix M contains the count of samples of class c being classified as h. The diagonal of M corresponds to true positive classifications, while cells in the same row as m_{cc} mark false positives and respectively cells in the same column as m_{cc} mark false negatives given class c.

For example, in the confusion matrix in Table 4.2, m_{11} denotes the true positive classifications of class 1, while $\sum_{k=2}^{6} m_{1k}$ is the number of false positives and $\sum_{i=2}^{6} m_{i1}$ is the number of false negatives given class 1.

More formally, the definition of true positive, false positive, false negative and true negative given class c can be calculated according to

$$TP(c) = m_{cc}, \tag{4.6}$$

$$FN(c) = \sum_{i \neq c} m_{ic}, \tag{4.7}$$

$$FP(c) = \sum_{k \neq c} m_{ck} \text{ and} \tag{4.8}$$

$$TN(c) = \sum_{i \neq c, k \neq c} m_{ik}. \tag{4.9}$$

Using those definitions, the dual-class metrics can be extended to their class-dependent multi-class equivalents:

$$\text{accuracy}(c) = \frac{TP(c) + TN(c)}{TP(c) + FP(c) + TN(c) + FN(c)}, \tag{4.10}$$

$$\text{precision}(c) = \frac{TP(c)}{TP(c) + FP(c)}, \tag{4.11}$$

$$\text{recall}(c) = \frac{TP(c)}{TP(c) + FN(c)}, \text{ and} \tag{4.12}$$

$$F_1(c) = \frac{2 \cdot TP(c)}{2 \cdot TP(C) + FP(c) + FN(c)}. \tag{4.13}$$

4.3 Evaluation System

Figure 4.1 shows the high level structure of the system used to evaluate the different classification parameters. The ground truth is balanced so that there is an equal amount of samples for each expression and then split into three non overlapping sets to use in cross-validation. Using the web image database from Chapter 2, this results in 288 samples per expression and 96 samples per set. In each of the six evaluation rounds, one of the sets is used for key point selection, while the remaining two are used to train and evaluate the classifiers. In case that a regular grid is used to place the key points, the selection phase can be omitted and the corresponding set can be ignored.

In preparation of feature extraction, the facial images are registered to have a size of $(96 + 2r) \times (96 + 2r)\,px$ (see Sec. 3.4) and so that the eyes have a distance of $44\,px$ to each other and $(22 + r)\,px$ to the top. No further image deformation is conducted.

The key point selection uses linear soft-margin SVMs with $C = 1$ as base classifier h. Half of the key point selection set is used to train the classifiers while the other half is used for boosting. In *per-expression* key point selection, positive samples are those of the target expression, while negative samples are randomly selected from all other classes. The sets are balanced, so that there is an equal amount of training data for both cases. The intuition is that this method will select the salient features of each individual expression. In contrast, in *expressive-vs-neutral* selection, negative samples are provided by the neutral expression, while positive training data is selected from all other expressions. In both cases the region radius varies between $r = 4$, $r = 6$, $r = 8$ and $r = 12$, which corresponds to region-sizes of $8 \times 8px$ up to $24 \times 24px$. 5, 20, 32, 64, 96 or 144 points were selected as described in section 3.4. In case that no key point selection is conducted, i.e. when the regions are placed on a regular grid, the number of points depends on the region radius and is either 16 ($r = 12$), 36, 64 or 144 ($r = 4$).

The remaining feature extraction parameters are fixed to reduce the amount of validation rounds. The DCT descriptor parameters are chosen to drop the first $k_1 = 1$ and collect the next $k_2 = 10$ coefficients. The feature dimension ranges from $|\mathbf{f}| = 10 \cdot 5 = 50$ to $|\mathbf{f}| = 10 \cdot 144 = 1440$ depending on the number of key points. The LBP descriptor is built using the basic LBP operator, i.e. $P = 8$ and $R = 1$. The descriptor size ranges from $|\mathbf{f}| = 59 \cdot 5 = 295$ to $|\mathbf{f}| = 59 \cdot 144 = 8496$. The Gabor filter bank consists of filters in $M = 5$ scales and $N = 8$ orientations, resulting in feature descriptors of size $|\mathbf{f}| = 40 \cdot 5 = 200$ to

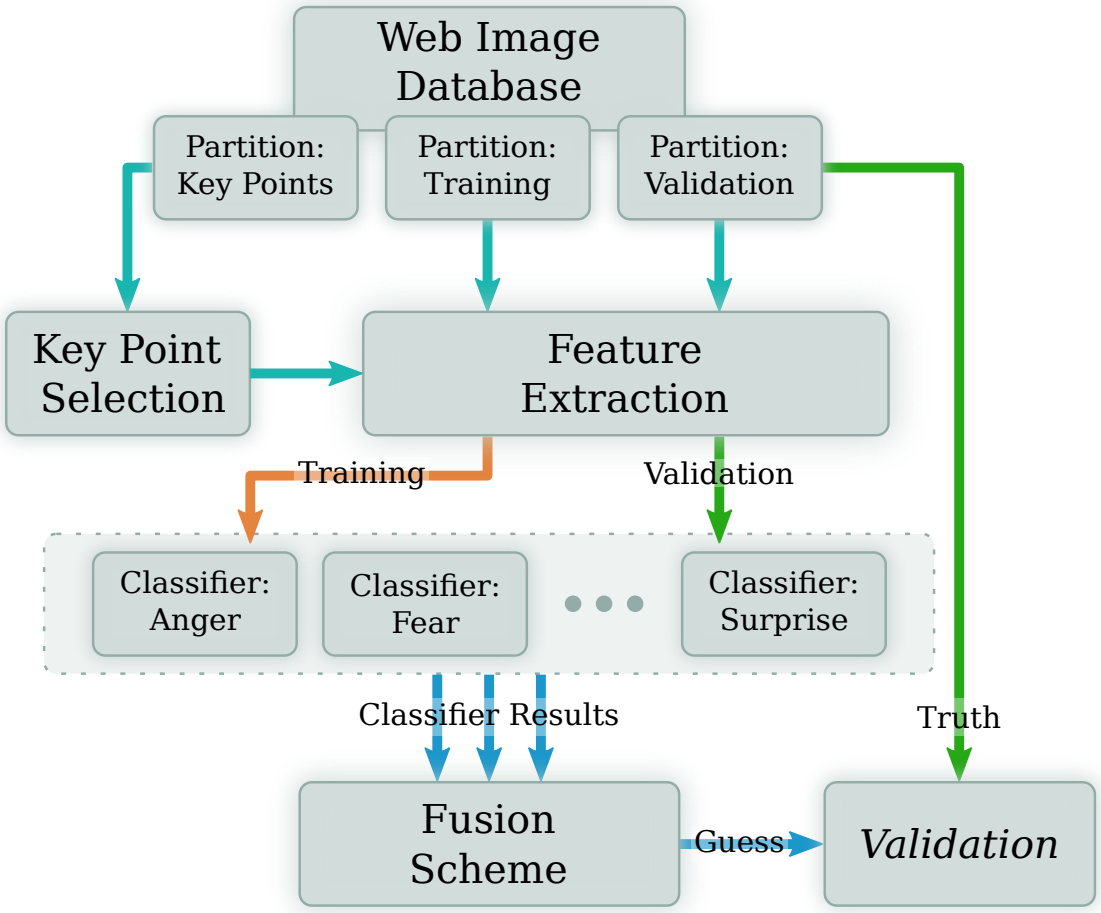


Figure 4.1: High level structure of the evaluation system.

$|\mathbf{f}| = 40 \cdot 144 = 5760$. It is possible that the performance can be improved by choosing different parameters.

The classification process is divided into several one-vs-all classification tasks performed by either AdaBoost or SVM classifiers. The AdaBoost classifiers can be adopted from the key point selection phase and require no further training. The SVM classifiers use third degree polynomial kernels $k(\mathbf{x}_i, \mathbf{x}_k) = (\gamma \mathbf{x}_i^T \mathbf{x}_k)^3$, where the parameters $C = 2^s$ and $\gamma = 2^t$ are estimated in a grid search by choosing $s = -6, \ldots, 0$ and $t = -14, \ldots, -5$ to yield highest accuracy in a 5-fold cross-validation. Each class and feature extraction method may produce a different set of parameters.

The class of a given sample $\mathbf{x}$ is determined by the maximum binary classifier output. The output of an AdaBoost classifier is given by equation (3.14), the output of a SVM classifier is the signed distance to the hyper-plane in equation (3.19). An alternative would be to follow Bartlett and transform the outputs into a probability distribution using a softmax competition:

$$p(\mathbf{x} = i) = \frac{\exp(H_i(\mathbf{x}))}{\sum_{j=1}^{n} \exp(H_j(\mathbf{x}))}. \tag{4.14}$$

Classifier	Key-Points	Mean Accuracy $r=4$	$r=6$	$r=8$	$r=12$
AdaBoost	5 (per-class)	0.734	**0.735**	0.732	0.732
AdaBoost	20 (per-class)	0.731	0.729	0.730	0.728
AdaBoost	32 (per-class)	0.729	0.728	0.729	0.729
AdaBoost	64 (per-class)	0.730	0.729	0.728	0.728
AdaBoost	96 (per-class)	0.729	0.728	0.728	0.730
AdaBoost	144 (per-class)	0.730	0.730	0.728	0.729
SVM	5 (per-class)	0.803	0.809	0.803	0.810
SVM	20 (per-class)	0.834	0.835	0.835	0.838
SVM	32 (per-class)	0.834	0.841	0.840	0.834
SVM	64 (per-class)	0.839	0.842	0.841	0.837
SVM	96 (per-class)	0.839	**0.844**	0.841	0.837
SVM	144 (per-class)	0.838	0.843	0.838	0.835
SVM	5 (expressive)	0.803	0.791	0.804	0.793
SVM	20 (expressive)	0.831	0.830	0.832	0.826
SVM	32 (expressive)	0.831	0.831	0.833	0.828
SVM	64 (expressive)	0.834	0.835	0.833	0.830
SVM	96 (expressive)	0.837	0.836	0.834	0.833
SVM	144 (expressive)	**0.841**	0.839	0.834	0.834
SVM	16 (grid)	-	-	-	0.828
SVM	36 (grid)	-	-	0.835	-
SVM	64 (grid)	-	**0.843**	-	-
SVM	144 (grid)	**0.843**	-	-	-

Table 4.3: Mean accuracy for all DCT feature configurations.

However, choosing the class of $\mathbf{x}$ according to $p(\mathbf{x})$ produces the same result as simply selecting the class according to the highest classifier output.

4.4 Results

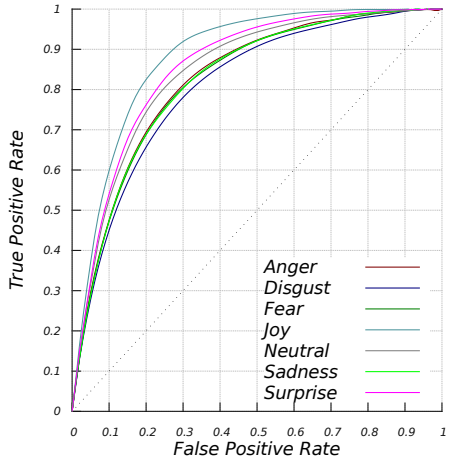
In the following the performance of the three different feature extraction methods is analyzed using the metrics described in section 4.2.

4.4.1 DCT Descriptor Performance

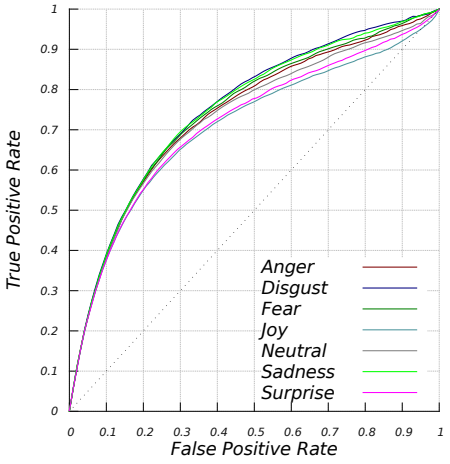
Figure 4.2 shows the ROC curves for different configurations using the DCT feature. Figures 4.2(a), (b) and (c) hint the superiority of SVM classification compared to AdaBoost.

	Anger	Disgust	Fear	Joy	Neutral	Sadness	Surprise	*Mean*
Precision	0.383	0.342	0.429	**0.616**	0.498	0.392	0.541	0.457
Recall	0.411	0.392	0.370	**0.634**	0.509	0.380	0.484	0.454
F_1 score	0.397	0.365	0.397	**0.625**	0.503	0.386	0.511	0.456
Accuracy	0.821	0.806	0.840	**0.891**	0.857	0.827	0.868	0.844

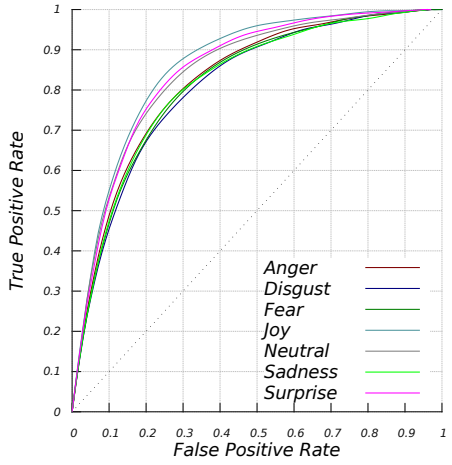
Table 4.4: Highest mean accuracy DCT classifier's metrics by expression.



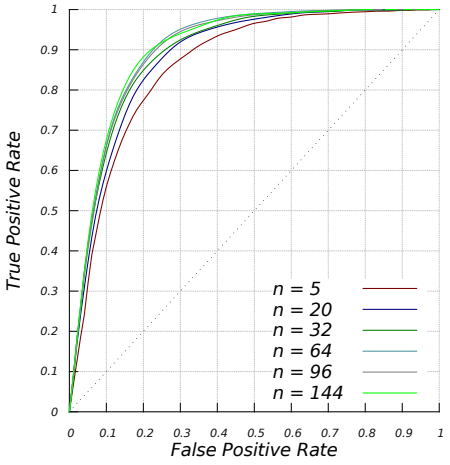
(a) SVM, $r = 6$, 20 key points (per-class)



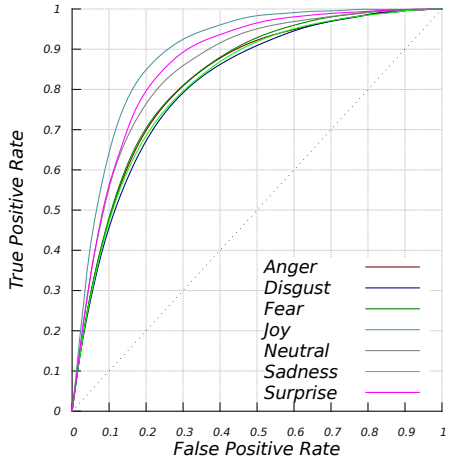
(b) AdaBoost, $r = 6$, 20 key points (per-class)



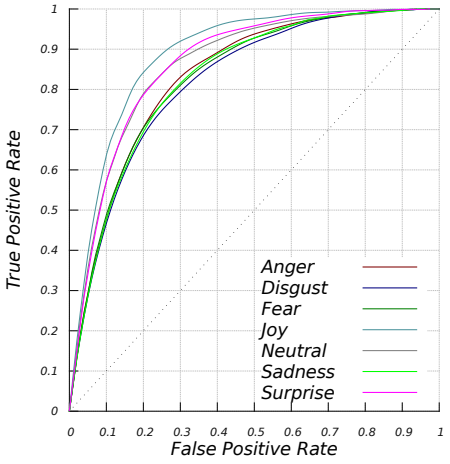
(c) SVM, $r = 6$, 20 key points (expressive)



(d) "Joy", SVM classifier, $r = 6$



(e) SVM, $r = 6$, 32 key points (per-class)



(f) SVM, $r = 6$, 64 key points (grid)

Figure 4.2: DCT feature ROC curves for different system configurations.

	Anger	Disgust	Fear	Joy	Neutral	Sadness	Surprise
Anger	**237**	97	59	45	55	78	48
Disgust	104	**226**	75	60	55	97	43
Fear	38	42	**213**	18	41	50	95
Joy	40	40	37	**365**	30	52	29
Neutral	63	55	43	32	**293**	57	45
Sadness	63	87	54	39	60	**219**	37
Surprise	31	29	95	17	42	23	**279**

Table 4.5: Confusion matrix of the highest mean accuracy DCT feature classifier.

As shown in Fig. 4.2(d), the usefulness of added key points decreases with the amount of regions taken into account. While there is a large gap between the curves for $n = 5$ and $n = 20$ key points, the curves for $n = 64$, $n = 96$ and $n = 144$ lie very close together. Comparison of Figures 4.2(e) and (f) confirms this finding.

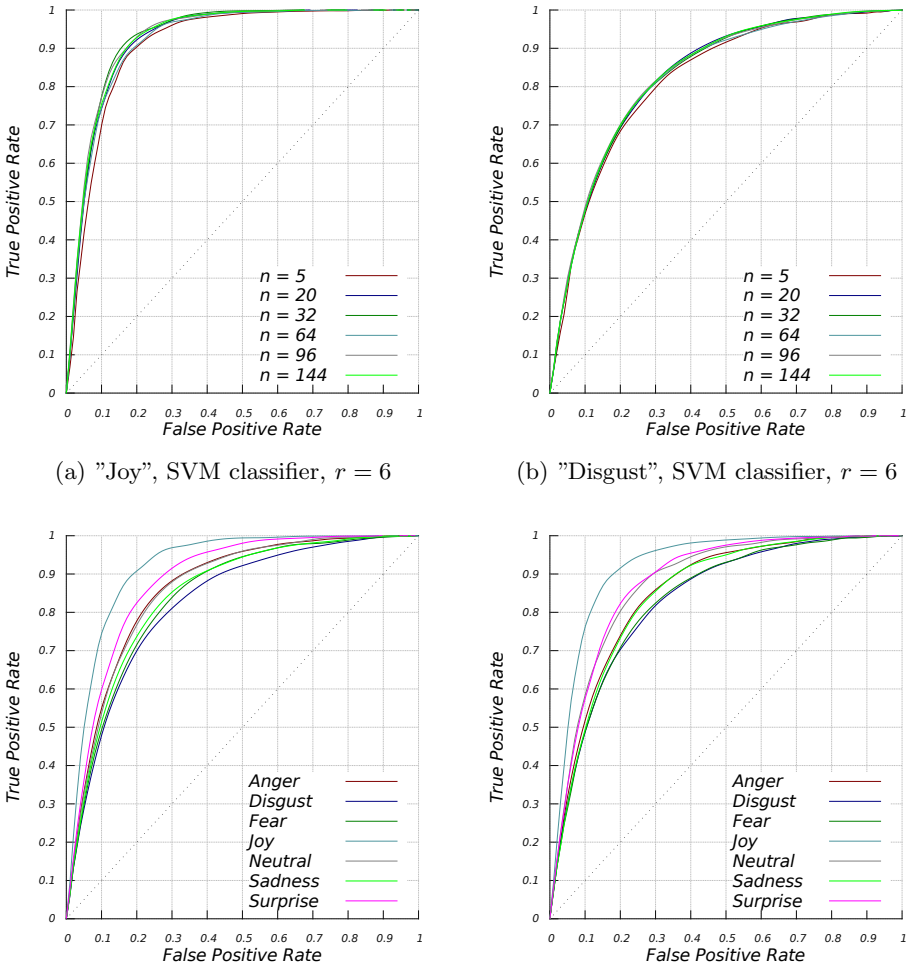
Table 4.3 lists the mean accuracy for all configurations using the DCT feature. The table confirms what was suggested by the ROC curves: AdaBoost performs about 10% worse than the SVM classifier. Surprisingly, AdaBoost performs best with only 5 key points regardless of the region size. As anticipated, per-class key point selection performs better than expressive-vs-neutral selection. The best result – 84.4% mean accuracy – is achieved using 96 key points and a region size of $12 \times 12px$. A higher number of key points shows lower performance suggesting that too many features can hurt classification performance. Interestingly, the grid-approach outperforms express-vs-neutral selection and even per-class selection given a region size of $8 \times 8px$.

A detailed performance description of the configuration yielding highest mean accuracy can be found in Table 4.4. "Joy" is the expression yielding highest precision, recall and accuracy. "Anger", "Disgust" and "Sadness" are difficult classification subjects and perform the worst. This finding is consistent with the ROC curves in Figure 4.2(a) as well as previous results by Bartlett and Shan [BLF+05, SGM09]. The confusion matrix 4.5 highlights the difficulties in discriminating "Anger" from "Disgust", "Disgust" and "Sadness" as well as "Fear" and "Surprise" – expressions that share important features like raised lips or wide-opened eyes.

4.4.2 LBP Descriptor Performance

Figure 4.3(a) and (b) show the performance impact of the number of selected key points on recognition systems using the LBP feature. As with the DCT feature, the cost-benefit ratio of additional regions worsens with increasing number of key points. Unlike with the DCT descriptor, even a very small number of key points and thus a relatively low-dimensional feature descriptor shows a performance close to high dimensional features. As demonstrated in Figures 4.3(c) and (d), per-class selection produces only slightly better results than a grid approach given the same number of key points and region size.

Table 4.6 provides a comparison of the mean accuracy given different configurations. The general results of the DCT feature descriptor analysis are repeated: AdaBoost performs



(a) "Joy", SVM classifier, $r = 6$

(b) "Disgust", SVM classifier, $r = 6$

(c) SVM, $r = 6$, 64 key points (per-class)

(d) SVM, $r = 6$, 64 key points (grid)

Figure 4.3: Impact of key point selection on LBP feature performance.

much worse than SVM classification, while the best result is achieved with 96 per-class selected key points and a $12 \times 12px$ region. Selecting more key points decreases performance. The best performing AdaBoost configurations again use only 5 key points. Per-class selection shows superior performance to expressive-vs-neutral selection as well as a grid-based approach. Unlike with DCT descriptors, per-class selection is always superior to the grid-approach regardless of the region size. This is consistent with Shan's findings [SGM09].

Table 4.7 lists precision, recall and accuracy by class for the best performing feature descriptor. Again "Joy" is with 92.5% accuracy and 71.5% recall the by far most recognizable expression, while "Disgust" is the most difficult expression with only 83% accuracy and 36.1% recall, followed by "Fear" and "Sadness'. The confusion matrix in Table 4.8 reasserts this result by showing high confusion of "Anger" and "Disgust", "Disgust" and "Sadness", as well as "Fear" and "Surprise", but very high confidence in "Joy".

Classifier	Key-Points	Mean Accuracy $r = 4$	$r = 6$	$r = 8$	$r = 12$
AdaBoost	5 (per-class)	0.724	0.723	0.725	**0.728**
AdaBoost	20 (per-class)	0.723	0.721	0.721	0.725
AdaBoost	32 (per-class)	0.721	0.722	0.723	0.724
AdaBoost	64 (per-class)	0.723	0.722	0.724	0.727
AdaBoost	96 (per-class)	0.722	0.722	0.723	0.725
AdaBoost	144 (per-class)	0.721	0.720	0.720	0.723
SVM	5 (per-class)	0.832	0.831	0.831	0.825
SVM	20 (per-class)	0.852	0.854	0.850	0.841
SVM	32 (per-class)	0.857	0.855	0.852	0.841
SVM	64 (per-class)	0.857	0.857	0.854	0.838
SVM	96 (per-class)	0.857	**0.859**	0.850	0.839
SVM	144 (per-class)	0.855	0.852	0.852	0.838
SVM	5 (expressive)	0.822	0.828	0.824	0.825
SVM	20 (expressive)	0.839	0.840	0.842	0.839
SVM	32 (expressive)	0.844	0.849	0.843	0.836
SVM	64 (expressive)	0.854	0.854	0.850	0.840
SVM	96 (expressive)	0.847	0.852	0.848	0.842
SVM	144 (expressive)	0.853	**0.855**	0.852	0.844
SVM	16 (grid)	-	-	-	0.837
SVM	36 (grid)	-	-	0.848	-
SVM	64 (grid)	-	0.854	-	-
SVM	144 (grid)	**0.857**	-	-	-

Table 4.6: Mean accuracy for all LBP feature configurations.

	Anger	Disgust	Fear	Joy	Neutral	Sadness	Surprise	*Mean*
Precision	0.460	0.396	0.417	**0.746**	0.515	0.423	0.605	0.509
Recall	0.503	0.361	0.399	**0.715**	0.568	0.451	0.549	0.507
F_1 score	0.480	0.378	0.408	**0.730**	0.540	0.437	0.576	0.508
Accuracy	0.844	0.830	0.834	**0.925**	0.862	0.834	0.884	0.859

Table 4.7: Highest mean accuracy LBP classifier's metrics by expression.

	Anger	Disgust	Fear	Joy	Neutral	Sadness	Surprise
Anger	**290**	125	41	34	52	65	24
Disgust	84	**208**	56	33	46	70	28
Fear	42	38	**230**	28	47	63	104
Joy	23	21	29	**412**	15	37	15
Neutral	60	59	51	21	**327**	65	52
Sadness	61	98	71	34	54	**260**	37
Surprise	16	27	98	14	35	16	**316**

Table 4.8: Confusion matrix of the highest mean accuracy LBP feature classifier.

4.4.3 Gabor Descriptor Performance

Figures 4.4(a)-(d) show ROC curves of the best performing classifier configurations using the Gabor feature. While the SVM classifiers show comparable results, AdaBoost based classification shows a significantly lower performance. For "Joy" the curve even crosses the no-discrimination-line when reaching 90% true and false positive rate. The relative unimportance of a high number of key points is shown again in Figures 4.4(e) and (f), but in contrast to the LBP descriptor, the performance difference between $n = 5$ and $n = 20$ key points is significant.

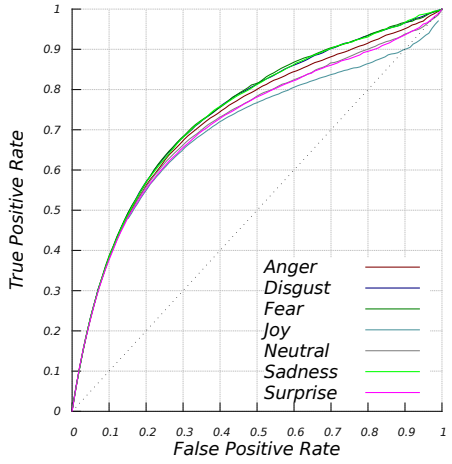
Table 4.9 repeats the findings with DCT and LBP descriptors. Per-class key point selection outperforms expressive-vs-neutral selection and a high number of regions results in a performance drop. Unlike with the other features, the turning point is reached sooner. Also unlike with the DCT and LBP descriptors, highest mean accuracy is achieved by grid-based feature extraction with a region size of $8 \times 8px$. This indicates that considering all regions may be superior to computing energy content in large regions. The result is also contradicting to Bartlett's research [BLF$^+$05], where AdaSVMs outperformed SVM classification on the global descriptor. However, this may be attributed to the differences in the feature description and feature selection methods. The next best result can be achieved with 32 per-class selected key points and shows an only 0.2% lower mean accuracy, but has a significantly smaller feature descriptor.

To investigate this trend, additional experiments with region sizes of $r = 1$ and and $r = 0$ (consideration of only the key point pixel) were conducted. Per-class key point selection can be slightly improved with smaller region size. Still, the best result of 86% mean accuracy is achieved by selecting the 32 most meaningful regions. Expressive-vs-neutral selection too benefits from small region sizes, achieving 85.9% mean accuracy with 144 key-points. It is likely that the performance of grid-based feature extraction would also increase with smaller region radius. However, due to the very high feature dimensions of $|\mathbf{f}| = 92160$ ($r = 1$) and $|\mathbf{f}| = 368640$ ($r = 0$) a comparison with key point selection ($|\mathbf{f}| = 5760$ with 144 regions) is – if at all – only partially possible.

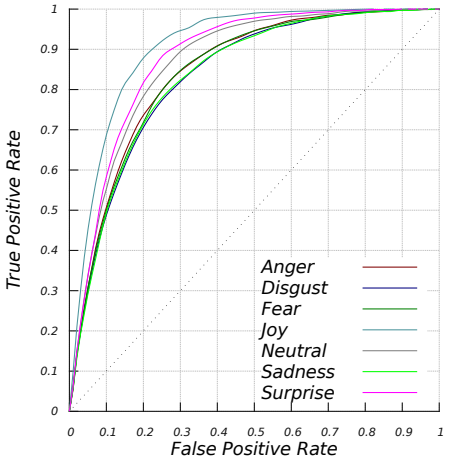
Detailed performance measurements of the grid-based configuration are shown in table 4.10. With 93% accuracy and 72.4% recall, "Joy" is again the most recognizable expression. Unlike with DCT and LBP features, the Gabor descriptor performs worst in recognizing "Sadness" and not "Disgust". The confusion matrix in Table 4.11 shows once more high confusion of "Anger" and "Disgust", "Disgust" and "Sadness" and "Fear" and "Surprise" while "Joy" shows the least amount of mis-classifications.

4.4.4 Key Point Selection

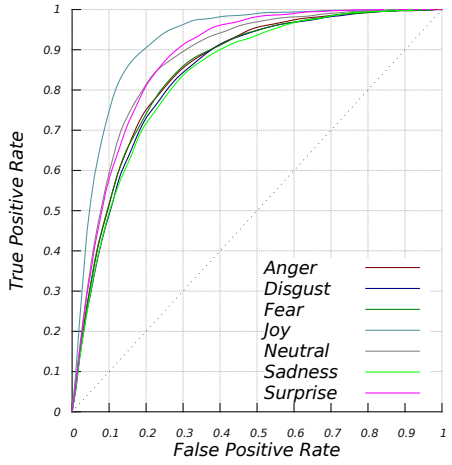
Figure 4.5 shows the locations of 96 key points extracted using the different feature descriptors given the same region size. The regions of the DCT descriptor are spread over the whole image area, while some faint clusters are forming around eye and mouth regions. In contrast, the LBP descriptor tends to concentrate the key points around certain interest points, e.g. around the inner brows and upper lip for "Anger" and around the mouth and eyes given a surprised expression. The Gabor descriptor exhibits inconsistent behavior, scattering the selected regions for "Anger" and "Disgust", but concentrating around eyes and mouth for "Joy" and "Surprise".



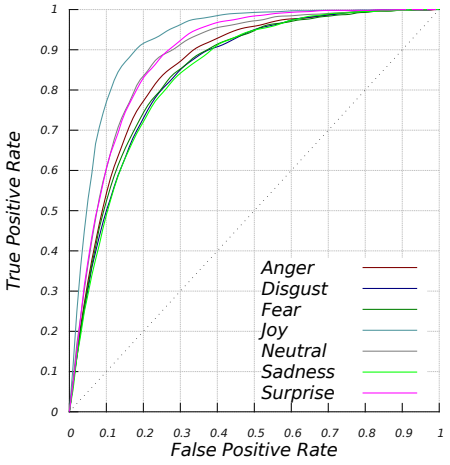
(a) AdaBoost, $r = 8$, 32 key points (per-class)



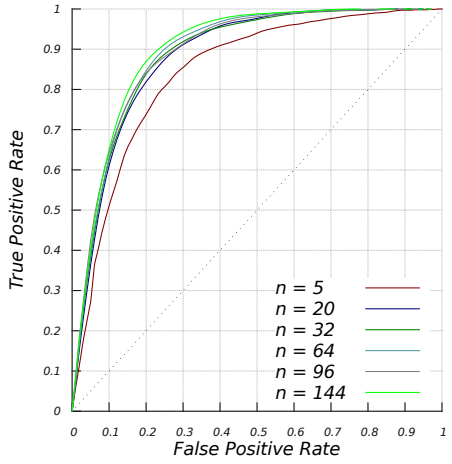
(b) SVM, $r = 4$, 32 key points (expressive)



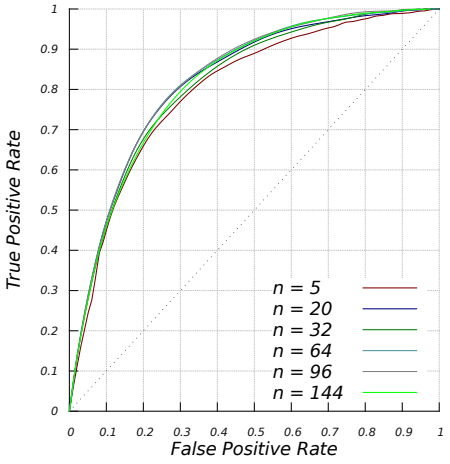
(c) SVM, $r = 4$, 96 key points (per-class)



(d) SVM, $r = 4$, 144 key points (grid)



(e) "Joy", SVM classifier, $r = 4$



(f) "Sadness", SVM classifier, $r = 6$

Figure 4.4: ROC curves for several system configurations using the Gabor feature.

Classifier	Key-Points	Mean Accuracy					
		$r = 0$	$r = 1$	$r = 4$	$r = 6$	$r = 8$	$r = 12$
AdaBoost	5 (per-class)	-	-	0.726	0.725	0.725	-
AdaBoost	20 (per-class)	-	-	0.725	0.726	0.724	-
AdaBoost	32 (per-class)	-	-	0.726	0.726	**0.727**	-
AdaBoost	64 (per-class)	-	-	0.724	0.724	0.723	-
AdaBoost	96 (per-class)	-	-	0.723	0.726	0.726	-
AdaBoost	144 (per-class)	-	-	0.726	0.725	**0.727**	-
SVM	5 (per-class)	0.839	0.836	0.836	0.835	0.832	0.829
SVM	20 (per-class)	0.850	0.852	0.850	0.850	0.847	0.836
SVM	32 (per-class)	**0.860**	0.857	**0.860**	0.855	0.853	0.843
SVM	64 (per-class)	0.858	0.857	0.857	0.856	0.852	0.842
SVM	96 (per-class)	0.859	0.859	0.858	0.855	0.849	0.842
SVM	144 (per-class)	0.857	**0.860**	0.856	0.854	0.854	0.842
SVM	5 (expressive)	0.819	0.824	0.824	0.826	0.819	0.816
SVM	20 (expressive)	0.848	0.850	0.846	0.846	0.842	0.833
SVM	32 (expressive)	0.852	0.851	0.849	0.848	0.843	0.835
SVM	64 (expressive)	0.851	0.852	0.852	0.848	0.845	0.836
SVM	96 (expressive)	0.853	0.853	0.853	0.849	0.844	0.839
SVM	144 (expressive)	**0.859**	0.856	0.852	0.851	0.847	0.840
SVM	16 (grid)	-	-	-	-	-	0.843
SVM	36 (grid)	-	-	-	-	0.854	-
SVM	64 (grid)	-	-	-	0.860	-	-
SVM	144 (grid)	-	-	**0.862**	-	-	-

Table 4.9: Mean accuracy for all Gabor feature configurations.

	Anger	Disgust	Fear	Joy	Neutral	Sadness	Surprise	*Mean*
Precision	0.495	0.408	0.451	**0.771**	0.562	0.399	0.553	0.520
Recall	0.441	0.411	0.425	**0.724**	0.641	0.422	0.564	0.518
F_1 **score**	0.466	0.410	0.438	**0.747**	0.599	0.410	0.559	0.519
Accuracy	0.856	0.831	0.844	**0.930**	0.877	0.827	0.873	0.862

Table 4.10: Highest mean accuracy Gabor classifier's metrics by expression.

	Anger	Disgust	Fear	Joy	Neutral	Sadness	Surprise
Anger	**254**	91	39	19	30	47	33
Disgust	112	**237**	45	33	32	95	27
Fear	29	45	**245**	18	39	56	111
Joy	14	37	18	**417**	16	25	14
Neutral	69	41	36	28	**369**	76	38
Sadness	65	104	73	45	51	**243**	28
Surprise	33	21	120	16	39	34	**325**

Table 4.11: Confusion matrix of the highest mean accuracy Gabor feature classifier.

(a) DCT descriptor, $r = 6$, 96 key points.

(b) LBP descriptor, $r = 6$, 96 key points.

(c) Gabor descriptor, $r = 6$, 96 key points.

Figure 4.5: Key point locations using per-expression selection. Expressions from left to right: Anger, Disgust, Fear, Joy, Neutral, Sadness, Surprise.

The key point locations of expressive-vs-neutral selection are shown in Figure 4.6. Using the DCT descriptor, key points are concentrated mostly around the mouth and eye regions, regardless of the region size. The LBP feature scatters key points around the mouth and eye regions for a radius of $r = 4$ and concentrates the regions around the inner brows and the mouth for $r = 6$. With higher radius, key point locations diverge towards the image border leaving the face only sparsely covered. This indicates multiple selection of the same sub-features, meaning that the base classifier used for feature selection may not perform better than random for large LBP feature regions. This result becomes even more apparent using the Gabor feature. It is noticeable that most of the key points are located in the lower half of the facial image. Given $r = 8$, a small number of regions concentrates around the mouth, while only 7 regions account for the eyes, cheek and inner brows. Once

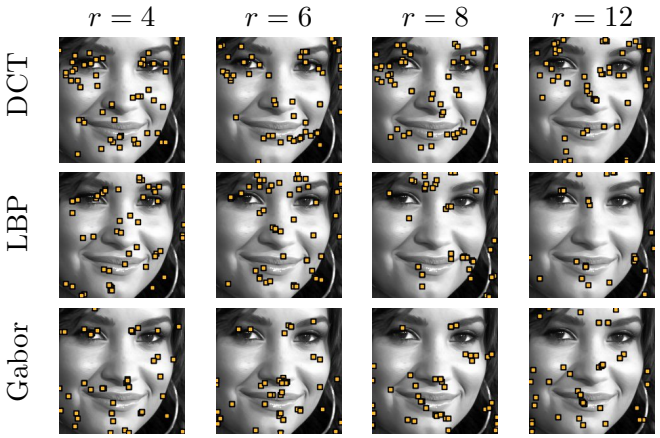


Figure 4.6: Key point locations using expressive-vs-neutral selection.

more, this may be caused by poor performance of the key point selection base classifier given the Gabor feature.

These results differ from Shan's experiments, where the selected regions distributed mostly around the mouth and eye area [SGM09]. The difference may be explained by the use of a broader selection process as well as the use of the Cohn-Kanade dataset, which does not include as much variation in pose and facial structure as the web image database used in this study.

5. Conclusion

The purpose of this study was twofold. First, focus was placed on building a database suitable for creating and evaluating expression recognition systems. The database was compiled from web-images and features 4761 labeled faces of male and female subjects of different ethnicities and age groups ranging from infants to elderly people. Face bounding boxes, eye positions and displayed facial expression were manually labeled. While the expression's intensities are not encoded, the additional meta-information provided by Google image search could be utilized in multi-modal classification systems.

Due to variations of expression intensity, head pose and lighting conditions as well as partial face occlusions and overlaying watermarks make the dataset an interesting challenge for classification systems with focus on web- or magazine images.

In the second part a modular system for facial expression recognition was developed. Though the goal was to classify Ekman's basic expressions, the modular approach is open to any number of classes, including AU detection.

Feature descriptors based on the discrete cosine transform, local binary patterns and Gabor filters were formulated in terms of pixels around key points. This approach can be seen as generalization of the local appearance based approaches often found in literature that divide the image into non-overlapping blocks of the same size. Three different methods to select significant key point locations were explored: Placing the regions on a regular grid – corresponding to the subdivision approach – as well as boosting for meaningful key points in per-class and expressive-vs-neutral contexts.

It has been shown that the per-class approach is generally superior to expressive-vs-neutral selection, which was expected. Consistent with Shan's results [SGM09], LBP features performed best with key point selection and SVM classifiers. This result also applies to DCT based features. In both cases, 96 key points with a region size of $12 \times 12px$ had shown best results. In contrast, Gabor features excelled with non-overlapping regions of $8 \times 8px$. This is a surprising result, because it is inconsistent with Bartlett's findings that *AdaSVMs* show superior performance to non-boosted SVM classification [BLF$^+$05]. However, the feature descriptors and feature selection used by Bartlett is different than the methods used in this thesis.

Feature	Regions	Size	|Feature|	Precision	Recall	Accuracy
DCT	96 (per-class)	$r = 6$	960	0.457	0.454	0.844
LBP	96 (per-class)	$r = 6$	5664	0.509	0.507	0.859
Gabor	144 (grid)	$r = 4$	5760	0.520	0.518	0.862

Table 5.1: Comparison of best performing configurations by feature descriptor.

As summarized in Table 5.1, Gabor features performed best, while DCT features fall behind with 1.8% less mean accuracy. However, the DCT descriptor has a much lower dimensionality and is faster to compute than both LBP and Gabor features and thus more suitable for real time applications. The LBP feature stands on middle ground regarding computational complexity as well as recognition performance. Depending on the requirements, DCT features should be used when fast reaction time is more important than accurate results and Gabor based descriptors should be chosen if the priorities are inversed. The LBP descriptor offers reasonable performance with little computational complexity and can be considered a good choice for interactive applications with focus on reasonable recognition performance.

When designing a classification system based on key point selection, the impact of the number of key points on the recognition performance has to be taken into account. As shown in tables 4.3, 4.6 and 4.9, too many key points may drop performance at a certain point. Careful analysis is needed to find optimal selection parameters. Neither Bartlett nor Shan account for this phenomenon in their experiments: Bartlett's system continues to select features until a termination criterion is reached, while Shan et al. always extract 50 meaningful regions [BLF+05, SGM09]. It remains to be seen if their systems could be improved by choosing an appropriate amount of boosted features.

5.1 Research Prospects

The proposed system offers several starting points for future enhancements, mostly concerning the feature selection process.

The LBP descriptor might be improved by using a pyramid of several operators with different radii to capture larger scale structures that elude the basic LBP operator. Similar to the Gabor feature, relevant sub-features would be selected using a boosting method. However, such an approach would lessen the speed advantage of local binary patterns over Gabor filters.

Following Shan, the key point selection could be extended by allowing rectangular (non-square) regions of variable size. Other feature extraction parameters like the number of DCT coefficients to drop and take could be varied as well.

To further improve the quality of selected regions, extended boosting algorithms like mutual information (MI) boosting [SBBW05] or FloatBoost [LZ04] could be employed. MI boosting introduces the notion of mutual information as a measure of independence of two random variables. Instead of choosing the weak classifier h_t that minimizes classification error under the distribution W_t, MI boosting accepts h_t only if it contributes with a certain amount of additional information. FloatBoost on the other hand applies a back-tracking

mechanism in each iteration of the algorithm to remove non-effective weak classifiers from the committee. In both cases the resulting strong classifier consists of fewer, more significant weak classifiers. Applied to feature selection, the resulting descriptor would be both smaller and more expressive. However, both algorithms have significantly higher computational complexity than AdaBoost.

Another enhancement could select smaller sub-features that do not correspond to facial regions anymore. In case of LBP features such an approach could be interpreted as only considering meaningful textons in a given region.

As mentioned above, the system could also be modified to recognize action units. Similar to CERT, Ekman's basic expressions could then be inferred by combining the output of the several AU detectors. In the same vein, recognition could be performed on video sequences rather than on static images. If so, the contribution of temporal relationships to the recognition performance should be explored.

Lastly, the system output could be fused to describe a point in *activation-evaluation space* (see [CDCT+01]), which has been shown to be a surprisingly powerful device to model emotion. Such an output would especially be useful in human computer interaction, e.g. for social robots and virtual driving assistants.

Bibliography

[AHP04] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face Recognition with Local Binary Patterns," in *Computer Vision - ECCV 2004*, ser. Lecture Notes in Computer Science, T. Pajdla and J. Matas, Eds. Berlin, Heidelberg: Springer-Verlag, 2004, vol. 3021, pp. 469–481.

[AHP06] ——, "Face Description with Local Binary Patterns: Application to Face Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[BGV92] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," in *Proc. Fifth Annual ACM Workshop on Computational Learning Theory*. ACM Press, 1992, pp. 144–152.

[BHES99] M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski, "Measuring facial expressions by computer image analysis," *Psychophysiology*, vol. 36, no. 2, pp. 253–263, 1999.

[Bis06a] C. M. Bishop, *Pattern Recognition and Machine Learning*, B. S. M. Jordan, J. Kleinberg, Ed. Springer, 2006.

[Bis06b] ——, "Sparse Kernel Machines," in *Pattern Recognition and Machine Learning*, B. S. M. Jordan, J. Kleinberg, Ed. Springer, 2006, ch. 7, pp. 326–331.

[BLF+05] M. S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan, "Recognizing facial expression: machine learning and application to spontaneous behavior," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2005*, vol. 2, 2005, pp. 568–573.

[BLFF96] T. Berners-Lee, R. Fielding, and H. Frystyk, "Hypertext Transfer Protocol – HTTP/1.0," RFC 1945 (Informational), Internet Engineering Task Force, May 1996.

[BLFM03] M. S. Bartlett, G. Littlewort, I. Fasel, and J. R. Movellan, "Real Time Face Detection and Facial Expression Recognition: Development and Applications to Human Computer Interaction." in *Proc. Conference on Computer Vision and Pattern Recognition Workshop CVPRW '03*, vol. 5, 2003.

[BS10] T. Bänziger and K. R. Scherer, "Introducing the Geneva Multimodal Emotion Portrayal (GEMEP) Corpus," *Blueprint for affective computing A sourcebook*, pp. 271–294, 2010.

[CDCT+01] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, and J. G. Taylor, "Emotion recognition in human-computer interaction," *Signal Processing Magazine, IEEE*, vol. 18, no. 1, pp. 32–80, January 2001.

[Cro06] D. Crockford, "The application/json Media Type for JavaScript Object Notation (JSON)," RFC 4627 (Informational), Internet Engineering Task Force, July 2006.

[CV95] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[Dar09] C. Darwin, *The Expression of the Emotions in Man and Animals*, anniversary ed., P. Ekman, Ed. Harper Perennial, 1872/2009.

[Dau85] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *Journal of the Optical Society of America A: Optics, Image Science, and Vision*, vol. 2, no. 7, pp. 1160–1169, July 1985.

[EF78] P. Ekman and W. Friesen, *Facial Action Coding System: A Technique for the Measurement of Facial Movement.* Palo Alto: Consulting Psychologists Press, 1978.

[Eke09] H. K. Ekenel, "A Robust Face Recognition Algorithm for Real-World Applications," Ph.D. dissertation, Universität Karlsruhe (TH), Fakultät für Informatik, February 2009.

[Ekm99] P. Ekman, "Basic emotions," in *Handbook of cognition and emotion.* John Wiley & Sons, 1999, vol. 98, no. 1992, ch. 3, pp. 45–60.

[FE04] B. Fröba and A. Ernst, "Face Detection with the Modified Census Transform," in *FGR' 04 Proc. Sixth IEEE International Conference on Automatic Face and Gesture Recognition.* Washington, DC, USA: IEEE Computer Society, 2004, pp. 91–96.

[Fel98] C. Fellbaum, *WordNet: An Electronical Lexical Database.* Cambridge, MA: The MIT Press, 1998.

[FL03] B. Fasel and J. Luettin, "Automatic facial expression analysis: a survey," *Pattern Recognition*, vol. 36, no. 1, pp. 259–275, 2003.

[Fri00] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," *Annals of Statistics*, vol. 29, pp. 1189–1232, 2000.

[FS95] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting," in *EuroCOLT '95 Proc. Second European Conference on Computational Learning Theory.* London, UK: Springer-Verlag, 1995, pp. 23–37.

[FS96] ——, "Experiments with a New Boosting Algorithm," in *Proc. International Conference on Machine Learning*, 1996, pp. 148–156.

[KCT00] T. Kanade, J. Cohn, and Y. Tian, "Comprehensive Database for Facial Expression Analysis," in *Proc. Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG'00)*, 2000, pp. 46–53.

[LBL07] G. C. Littlewort, M. S. Bartlett, and K. Lee, "Faces of pain: Automated measurement of spontaneous facial expressions of genuine and posed pain," in *ICMI '07: Proc. Ninth International Conference on Multimodal interfaces*. New York, NY, USA: ACM, 2007, pp. 15–21.

[LBSR11] G. C. Littlewort, M. S. Bartlett, L. P. Salamanca, and J. Reilly, "Automated measurement of children's facial expressions during problem solving tasks," in *Proc. IEEE International Conference on Automatic Face & Gesture Recognition and Workshops (FG 2011)*, 2011, pp. 30–35.

[LWW+11] G. Littlewort, J. Whitehill, T. Wu, I. Fasel, M. Frank, J. Movellan, and M. Bartlett, "The computer expression recognition toolbox (CERT)," in *Proc. IEEE International Conference on Automatic Face & Gesture Recognition and Workshops (FG 2011)*, 2011, pp. 298–305.

[LZ04] S. Z. Li and Z. Zhang, "FloatBoost Learning and Statistical Face Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1112–23, 2004.

[OPH96] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51 – 59, 1996.

[OPM02] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 971–987, 2002.

[Pic11] R. W. Picard, "Measuring affect in the wild," in *ACII'11 Proc. Fourth International Conference on Affective Computing and Intelligent Interaction - Volume Part I.* Berlin, Heidelberg: Springer-Verlag, 2011, pp. 3–3.

[PR00] M. Pantic and L. Rothkrantz, "Automatic Analysis of Facial Expressions: The State of the Art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1424–1445, December 2000.

[PWHR98] P. Phillips, H. Wechsler, J. Huang, and P. Rauss, "The FERET Database and Evaluation Procedure for Face-Recognition Algorithms," *Image Vision Comput.*, vol. 16, no. 5, pp. 295–306, April 1998.

[SBBW05] L. Shen, L. Bai, D. Bardsley, and Y. Wang, "Gabor feature selection for face recognition using improved adaboost learning," in *Proceedings of International Workshop on Biometric Recognition System, in conjunction with ICCV'05*, 2005.

[SGM09] C. Shan, S. Gong, and P. W. McOwan, "Facial expression recognition based on Local Binary Patterns: A comprehensive study," *Image and Vision Computing*, vol. 27, no. 6, pp. 803–816, 2009.