The Thesis Committee for Shaohua Wan
Certifies that this is the approved version of the following thesis:

A Scalable Metric Learning Based Voting Method for Expression Recognition

APPROVED BY
SUPERVISING COMMITTEE:

Supervisor: ____________________________________
J. K. Aggarwal

____
Kristen Grauman

A Scalable Metric Learning Based Voting Method for Expression Recognition

by

Shaohua Wan, B.E.

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

May 2013

Dedication

Dedicated to my family

Acknowledgements

This work has been made possible because of the following special people who have always been a source of inspiration for me to do research and pursue my dreams.

I offer my greatest regards to Professor J. K. Aggarwal for providing me the opportunity to work with him. It is he who introduced me into the world of computer vision. Without his continuous support and guidance, I would not have completed my research work at UT.

I also deeply thank Professor Kristen Grauman. Her academic competence has always been a source of inspiration and a model to emulate.

I would like to thank Dr. Changbo Hu, Birgi Tamersoy, Suyog Jain, and Lu Xia who made my stay here a memorable one. I will always cherish the numerous philosophical discussions on evolution and existence which were a part of most lunches we had together.

Finally, I would like to thank my friends who enriched my everyday life in Austin. I own my greatest debt to my parents for raising me up and educating me. I attribute all my success to them.

Abstract

A Scalable Metric Learning Based Voting Method for Expression Recognition

Shaohua Wan, M.S.E.

The University of Texas at Austin, 2013

Supervisor: J. K. Aggarwal

In this research work, we propose a facial expression classification method using metric learning-based k-nearest neighbor voting. To achieve accurate classification of a facial expression from frontal face images, we first learn a distance metric structure from training data that characterizes the feature space pattern, then use this metric to retrieve the nearest neighbors from the training dataset, and finally output the classification decision accordingly. An expression is represented as a fusion of face shape and texture. This representation is based on registering a face image with a landmarking shape model and extracting Gabor features from local patches around landmarks. This type of representation achieves robustness and effectiveness by using an ensemble of local patch feature detectors at a global shape level. A naive implementation of the metric learning-based k-nearest neighbor would incur a time complexity proportional to the size of the training dataset, which precludes this method being used with enormous datasets. To scale to potential larger databases, a similar approach to that in [24] is used to achieve an approximate yet efficient ML-based kNN voting based on Locality Sensitive Hashing

(LSH). A query example is directly hashed to the bucket of a pre-computed hash table where candidate nearest neighbors can be found, and there is no need to search the entire database for nearest neighbors. Experimental results on the Cohn-Kanade database and the Moving Faces and People database show that both ML-based kNN voting and its LSH approximation outperform the state-of-the-art, demonstrating the superiority and scalability of our method.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Motivation

Human Computer Interaction (HCI) is a popular research area that lies at the interaction between computer science, behavioral sciences, design and several other fields of study. Two channels have been distinguished in the study of HCI [29]: one transmits explicit messages, which may be about anything or nothing; the other transmits implicit messages about the speakers themselves. Understanding the emotional state of a human is one of the key tasks associated with the second, implicit channel.

Unlike other types of non-verbal communication, the human face is truly expressive and facial expressions are closely tied to emotional state. Ekman has recognized some basic categories of emotions (i.e., neutral, fear, anger, sadness, surprise, disgust and happiness) [30]. Every one of these emotions corresponds to a unique facial expression. Knowing the emotional state of the user makes machines communicate and interact with humans in a natural way: intelligent entertaining systems for kids, interactive computers, intelligent sensors, social robots, to name just a few.

The automated recognition of facial expressions is a necessary first step for machines to understand human emotions. The promising future of emotion-aware machine intelligence has propelled researchers to build computer systems to understand and use this natural form of human communication [3].

In this thesis, a scalable metric learning-based voting method for expression recognition will be presented. In the remaining of this chapter, a brief discussion of the techniques used in the facial expression recognition system are presented.

1.2 EXPRESSION REPRESENTATION

An effective representation of facial expressions is a vital component of any successful facial expression recognition system. Various models and methods have been proposed to attack this problem. Seen from a geometric perspective, model-based approaches, as in [8, 9, 10, 11], iteratively register the face with a deformable shape model and capture the holistic geometric variation aspects of an expression. Appearance-based approaches consider the varying pattern of pixel intensities as the distinguishing traits of an expression and design feature detectors for local skin patches, examples of which include SIFT [12], Local Binary Pattern [13], Local Directional Pattern [14], etc. To draw on the descriptive power of both shape and texture, [27, 28] use a combination of shape and texture models.

In our method, a hybrid representation of facial expressions is also used by fusing a shape model of landmark points and an underlying appearance model of local patches of face images. Chapter 3 will discuss the details of our choice of representation of facial expressions.

1.3 EXPRESSION CLASSIFICATION

A good expression recognition methodology should consider classification as well as representation issues [1]. Donohue et al. [4] used the back-propagation algorithm to train a neural network, and a recognition rate of 85% based on 20 test cases was reported. Kotsia et al. [5] used Support Vector Machine to classify geometric deformation features. In [22], Condition Random Fields are used to model the temporal variations of face shapes and make classifications accordingly. Previous methods have demonstrated satisfactory categorization performance on exaggerated expressions. As categorization samples the semantic space more densely and naturally induced expressions are involved, expression classification becomes quite difficult and previous methods are subject to severe accuracy degradation.

We consider facial expression classification in the framework of measuring similarities mainly for the reason that it is conceptually simple and does not require direct access to the features of the samples -- it only requires the similarity function to be defined for any pair of samples. While nearest neighbor is a natural choice in this setting, its classification resolution is limited since there exists much overlapping between subtle expressions. Thus, a metric structure that adapts to the feature space embeddings is preferred over the default Euclidean metric as a measure of similarity. Inspired by the success of Metric Learning (ML) in learning domain specific distance metrics [24] [53-57], we propose an expression classification method based on ML. In particular, a generalized Mahalanobis distance matrix is learned that satisfies pairwise similarity/dissimilarity constraints on distance between expression feature vectors. Afterwards, a kNN classifier equipped with this distance metric is used to assign the majority class label to the query expression.

As far as we know, this is the first time ML has been used for expression recognition. Our ML-based kNN classifier can be used either as an expression detector, where a single category is discriminatively trained against all other categories, or as a multi-class classifier, where the metric structure for multiple expressions are simultaneously learned. The latter approach has the advantage of sharing useful metric structures across different expression categories.

kNN classifier tends to Bayesian optimal as the dataset size tends to infinity [6]. However, the time cost of searching all examples for nearest neighbors would become the bottleneck as more instances are added to the database. To scale up to potential larger database, a variant of the ML-based kNN voting method [24] based on Locality Sensitive

Hashing (LSH) [7] is used to speed up the kNN search process. This variant achieves efficiency by narrowing the search for candidate nearest neighbors down to only items sharing the same hash key in a pre-computed hash table.

1.4 CONTRIBUTIONS

The main contributions of this research work are summarized below:

1. Metric learning is employed to train a domain-specific distance metric that is able to capture the inherent embeddings of feature space and yields more accurate and robust expression classification results;
2. An approximation scheme based on LSH is used to speed up the nearest neighbor search process of ML-based kNN, making our method a real-time one and applicable to a large database;
3. We also perform comparative experimental studies of various expression classification algorithms on two databases. Our ML-based kNN voting method compares favorably in terms of recognition rate, especially when it comes to subtle expressions.

This thesis is structured in the following way. Chapter 2 will summarize the past research that is related to expression recognition. Chapter 3 will introduce the feature representation of facial expression. We then formulate expression classification as a metric learning problem in Chapter 4. In Chapter 5, how ML-based kNN can be sped up via the use of LSH is described. In Chapter 6, we present various experimental results and discussions followed by conclusions.

Chapter 2: Related Work

2.1 ANATOMY OF AN EXPRESSION RECOGNITION SYSTEM

Automated facial expression recognition has been an area of interest for several decades in the community of computer vision. It basically involves recognizing basic human expressions from 2-D or 3-D images. Six basic expressions defined by Ekman [2] are shown in Figure 2.1. A comprehensive survey of techniques to this purpose that have been developed can be found in [33].

The problem of facial expression recognition from images presents various challenges. A robust expression recognition system should be able to handle intra-class variations since people can have different skull sizes and expressions of the same type can vary in degree. Occlusion and pose variations could cause much difficulty in localizing landmarks. Lighting condition is another important factor that affects the performance of an expression recognition system. Also, much information is lost when working with 2-D face images instead of 3-D face images, which may potentially harm the recognition process. An automated expression recognition system which is able to combat all these obstacles is yet to be developed; hence facial expression recognition remains an open research problem today.

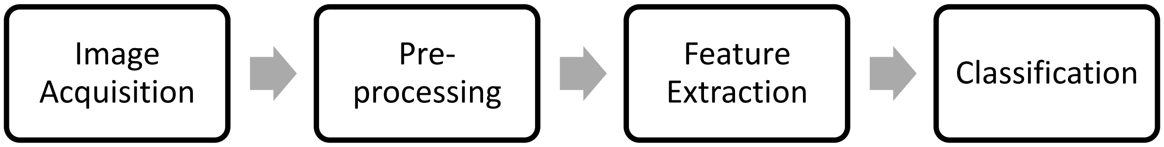
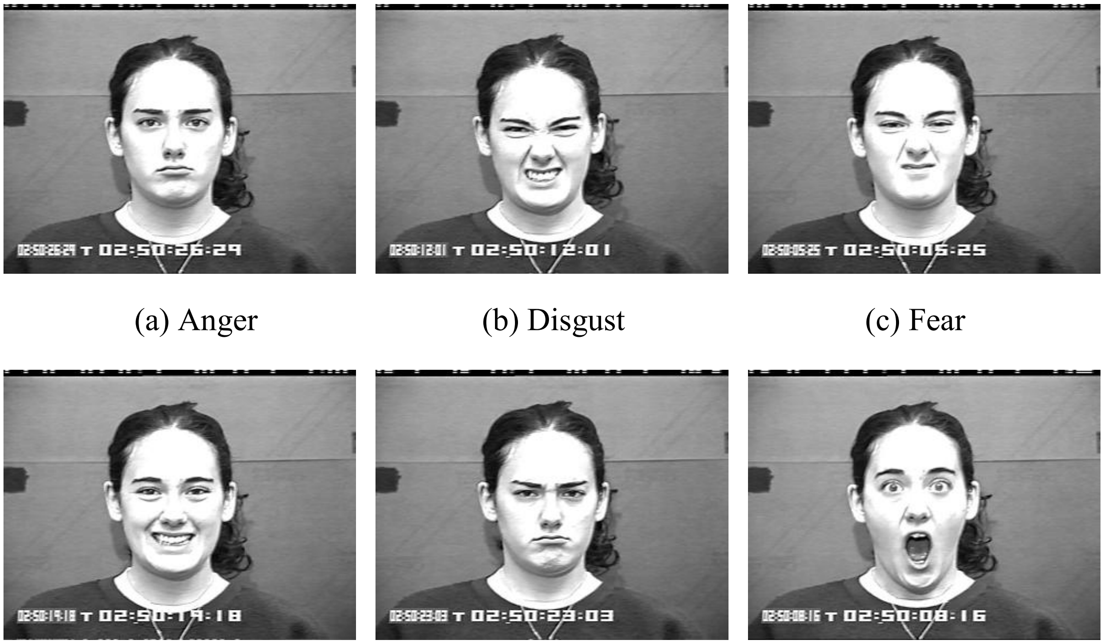


Figure 2.1: The building blocks of a typical facial expression recognition system.

An automatic facial expression recognition system usually takes the form of a sequential configuration of processing blocks, which adheres to a classical pattern recognition model (see Figure 2.1) [30]. The main blocks are: image acquisition, pre-processing, feature extraction, and classification. In the following, the research work related to expression recognition will be presented in the order of these four building blocks.



(a) Anger (b) Disgust (c) Fear

(d) Happiness (e) Sadness (f) Surprise

Figure 2.2: Examples of six basic facial expressions.

2.2 IMAGE ACQUISITION

With respect to the spatial, chromatic, and temporal dimensionality of input images, 2-D monochrome (gray-scale) facial image sequences are the most popular type of images used for automatic expression recognition [33-37]. In addition to that, there has been work that feeds other forms of input to the expression recognition system.

Y. Yoshitomi [51] proposes a method that recognizes expressions from facial thermal images. The method is based on 2-dimensional detection of temperature distribution of the face, using infrared rays. The front-view face in the input image is normalized in terms of the size and the location, followed by measurement of the local temperature difference between the averaged neutral and the unknown expressive faces.

In [52], the depth image of a 3D facial point cloud is captured and combined with Zernike moments to tackle the problem of facial expression recognition, and proves to be robust to affine transformations of the data, such as translation, rotation and scaling.

In the following, we will mainly focus on discussing facial expression recognition using 2-D frontal face images.

2.3 PRE-PROCESSING

Pre-processing is an important step before the extraction of features, and it is often done for the purpose of registration and normalization. A number of algorithms have been proposed to register a human face in an image.

The Active Appearance Model [31] is a popular framework for face registration. It is a statistical model of face shape and appearance and finds the optimal match between the model instance and the input image by iteratively solving for incremental additive updates to the parameters. The Constrained Local Model [32] is another popular class of face registration method. Instead of registering a whole face at once, it makes independent predictions regarding locations of the model's landmarks, which are combined by enforcing a prior over their joint motion.

In order to remove variations of face shape in position, orientation, and size, Procrustes analysis is used to geometrically normalize the registered face. This normalization is usually based on iterative alignment to the mean shape of all training examples [60].

Robust localization of face or its parts is difficult to attain in many real-world settings. Tracking is often implemented as localization of the face or its parts within an image sequence, whereby previously determined landmarks are typically used for estimating landmarks in subsequent image frames [58, 59].

2.4 FEATURE EXTRACTION.

Feature extraction builds a high-level representation of expression from pixel information. Multiple types of feature descriptors can be extracted, including shape, texture, and motion.

In [33], an automatic system is developed for analyzing subtle changes in facial expressions based on permanent facial features (brows, eyes, mouth) and transient facial features (deepening of facial furrows) in a nearly frontal image sequence. Continuous and discrete representation of magnitude and direction for motion of face parts (lips, eyes, brows, cheeks), as well as state-based (present/absent) representation of transient features (furrows and wrinkles) are extracted as characterizing features.

Feature extraction in [34] is based on the Facial Action Coding System (FACS), which separates expressions into upper and lower face actions. Three types of facial expression information are extracted using the following approaches: (1) facial feature point tracking; (2) dense flow tracking with principal component analysis (PCA); and (3) high gradient component detection (i.e. furrow detection).

The more successful approach has been to localize facial landmarks in order to obtain precise information about the face. These landmark points can then be used to derive geometric features [35]. Shape information encodes the geometric features of the face by using the localized landmark points or by computing various distances and angles between them [36] [37]. The appearance features can be computed over the entire face or face patches around the landmark points. The appearance features usually represent the texture on the face. Several appearance features like Gabor Filters [38] [39] and Local Binary Pattern features [40] [41] have been used to recognize facial expressions successfully.

2.5 EXPRESSION CLASSIFICATION

There are basically two approaches to the problem of facial expression recognition, a static one and a dynamic one. In a static approach, an expression label is assigned according to the features extracted from a single face image. In contrast, a dynamic approach tries to model the temporal dynamics of facial expressions by looking at the entire sequence, and an expression label is assigned based on adjacent images in an image sequence.

In [42], the expression recognition system uses a rank-weighted k-nearest neighbor classifier. First, the k nearest neighbors to the pattern being classified are found by calculating the inter-pattern distance. Then the rank order of these neighbors is reversed. Finally, the score for each known class of expression is calculated as the sum of the reverse rank of each nearest neighbor belonging to the class.

In [45], Bartlett et al presents some preliminary results on a task of facial action detection in spontaneous facial expressions. Spontaneous facial expressions differ from posed expressions in terms of which muscles are moved, and in the dynamics of the movement, and are usually harder to recognize. In their work, a fully automatic system for recognition of facial actions from the Facial Action Coding System (FACS) [46] is used to detect facial muscle movements and can work in real time. Two machine learning methods, support vector machines and AdaBoost, are applied to texture-based image representations to predict action unit intensity. Such a real time system enables frame-by-frame intensity measurements and help a great deal in investigation of the spontaneous expression dynamics.

Irene Kotsia, et al [5] uses a dynamic approach that calculates the difference of landmark positions in adjacent frames in an image sequence. In particular, one has to manually place some of Candide grid nodes to landmarks at the first frame of the image sequence under examination. The grid is tracked across consecutive video frames over time using a deformable model based tracking system, as the facial expression evolves, until it reaches the frame that corresponds to the greatest facial expression intensity. The geometrical displacement of certain selected Candide nodes, defined as the difference of the node coordinates between the first and the greatest facial expression intensity frame, is used as an input to a multi-class Support Vector Machine (SVM) system of classifiers, which are used to recognize either the six basic facial expressions or a set of chosen Facial Action Units (FAUs).

T. Otsuka and J. Ohya [44] use Hidden Markov Model to model the temporal sequence of a feature vector obtained from image processing. Image processing is performed in two steps. First, a velocity vector is estimated from every two successive frames in an image sequence using an optical flow algorithm. Then, a two dimensional Fourier transform is applied to the velocity vector field at the regions around the eyes and mouth. The coefficients for the lower frequencies are selected to form a feature vector.

Although there is much work on facial expression recognition, little attention has been paid to the problem of expression recognition under occlusion. For example, long hair can hide eyebrows or eyes from image capturing systems, or only one half of the face is visible when seeing a person from the side. In these cases, a facial expression recognition system should be robust to partial occlusion. Bourel et al. [47] presents a data fusion approach to facial expression recognition in the presence of occlusion. Kanade-

Lucas tracker [48] is used to detect one of 12 facial points that can be lost due to variation in lighting conditions, translation, motion, or head orientation. Then, multiple data and knowledge are integrated together by means of data fusion to form a collective representation of facial expression. In particular, a reference point based heuristic approach is performed to the visual properties of the face to get the lost facial points. A final classification result is produced by summing up the weighted cumulative scores output by each local classifiers.

[61] proposes a view-invariant expression recognition method based on analytic shape manifolds. They use the equivalence class of shapes in a proper shape-space to remove the need for a pre-processing step to align the data to a common coordinate frame. It is showed that the affine shape-space for the facial landmark configurations has Grassmannian properties and therefore nonrigid facial deformations due to various expressions can be represented as points on the Grassmann manifold. The advantage of modeling the facial expressions on this manifold is that the variability being computed is from shape changes only and not the coordinate frame, thus achieving invariance to view point changes.

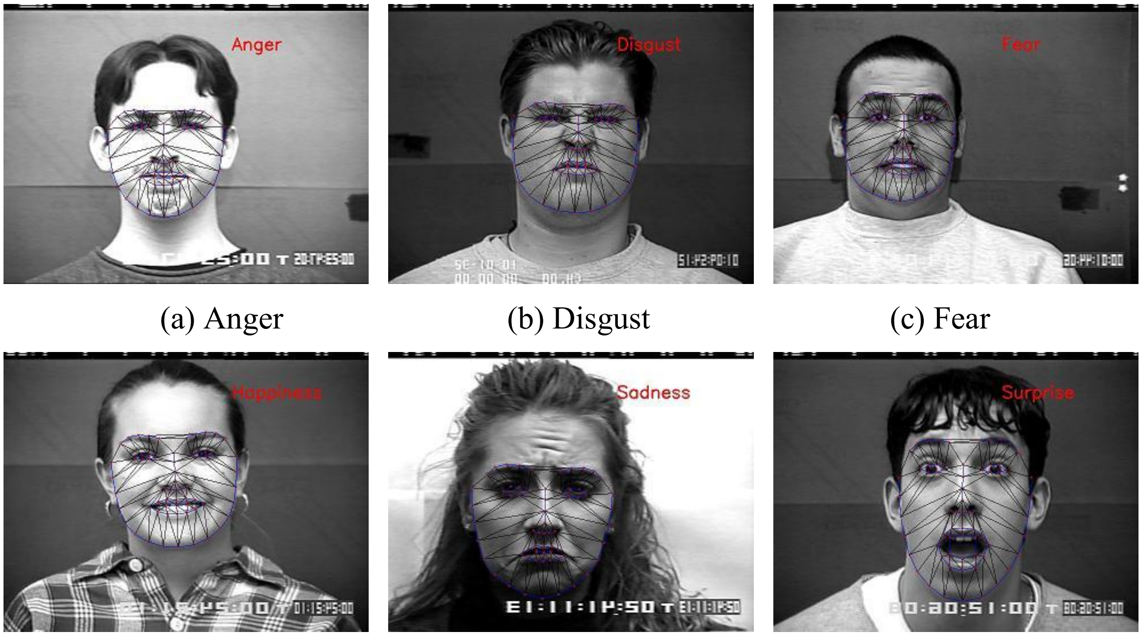
K-Nearest Neighbors (K-NN) is a widely used example-based classification algorithm [62]. One of the advantages of K-NN is that it is well suited for multi-modal classes as its classification decision is based on a small neighborhood of similar objects. As a result, even if the target class is multimodal (i.e., consists of objects whose independent variables have different characteristics for different subsets), it can still lead to good classification accuracy. [63] classifies facial expressions using k-Nearest Neighbor classifier. However, the testing dataset used in the experiment is quite small and its classification accuracy remains to be coroborated. In this work, the proposed facial expression classification method is evaluated on two large-scale datasets with the aim of achieving satisfactory classification accuracy. Moreover, this work differs from [63] in that Metric Learning is employed to reduce the confusion between overlapping classes, thus potentially giving more accurate classification result.

Chapter 3: Expression Representation

This chapter outlines the method for representing facial expression in our work. We use a fusion of face shape and texture as the representation of facial expression. This hybrid representation is able to incorporate local pixel intensity variation pattern while still adhering to shape constraint at a global level, proving to be robust and effective. Necessary preprocessing steps prior to constructing such a representation are also described.

3.1 Face Shape

In our work, a face shape is represented by a set of 68 points known as landmarks. These landmarks are distributed along the contour of the eyebrows, eyes, nose, mouth, and chin. Landmarks corresponding to six basic expressions are show in Figure 3.1.



(a) Anger (b) Disgust (c) Fear

(d) Happiness (e) Sadness (f) Surprise

Figure 3.1: 68 landmarks that are used as characteristic points of the face shape of an expression. They are selected to lie on the contour of the eyebrows, eyes, nose, inner mouth lip, outer mouth lip, and chin.

As one can see, the face shapes denoted by these landmarks vary substantially in reference point, orientations and scale. Before comparing these shapes to each other, they must be first optimally superimposed, i.e. they must be optimally translated, uniformly scaled, and rotated to obtain a similar placement and size, by minimizing a measure of shape difference.

In this work, Generalized Procrutes Analysis (GPA) is employed to align these face shapes to an optimally determined mean shape. The algorithmic steps are described as follows:

Generalized Procrutes Analysis

1. Set the reference shape as the mean shape of the current set of shapes;

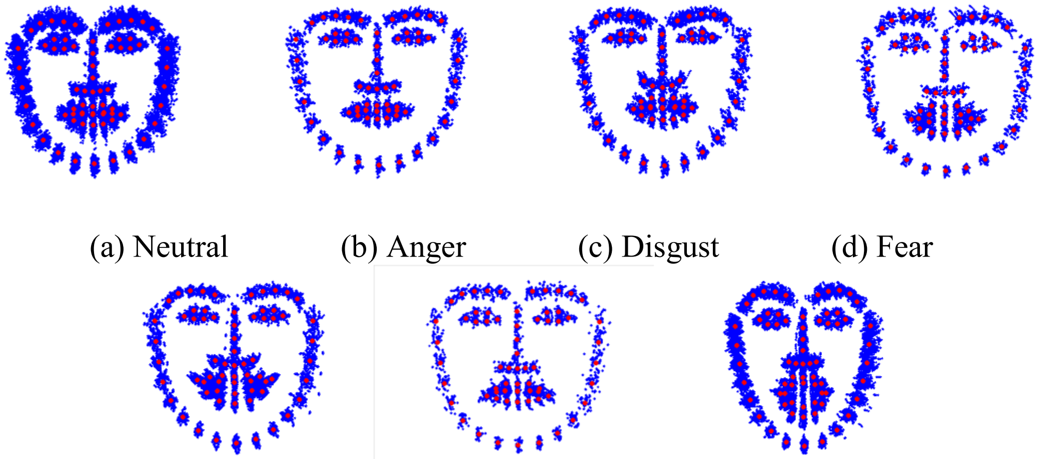
2. Update the current set of shapes by superimposing them to the reference shape; Compute the mean shape of the current set;

3. If the Procrustes distance between the reference shape and the mean shape is above a threshold, set the reference shape to the mean shape and continue to step 2.

Example Procrustes superimposition results from the CK+ dataset [20] are shown in Figure 3.2. The face shape landmarks are obtained using the automatic face image registrator from [9]. In total, shapes of seven basic expressions (including neutral) defined in [2] are given in seven subfigures [15]. Each subfigure is a superimposition of face shapes of all examples of a specific expression from the CK+ dataset [20]. Red landmarks denote the mean shape. In the following sections, we denote face shape feature vector s as

$$s = [s_x^1, s_y^1, s_x^2, s_y^2, \dots, s_x^n, s_y^n]$$

where (s_x^i, s_y^i) denotes the x and y coordinates of the i-th landmark point.



(a) Neutral (b) Anger (c) Disgust (d) Fear

(e) Happiness (f) Sadness (g) Surprise

Figure 3.2: Face shapes of seven basic expressions (including neutral) from the Cohn-Kanade dataset after being aligned using Generalized Procrustes Analysis. Red denotes the mean shape of that particular expression.

3.2 Texture Feature

The Gabor filter is a good model of simple cell receptive fields in cat striate cortex [49, 50]. A number of research works on expression recognition using Gabor features have reported improved recognition rate [16, 17, 18]. In our work, Gabor features are used as the texture feature extractor due to its optimal localization properties in both spatial and frequency domain. In this section, the procedures for extracting Gabor features are described.

3.2.1 Face Image Normalization

Face appearance can vary greatly among instances of subjects due to skull sizes, lighting conditions, image noise, and intrinsic sources of variability. To minimize geometric and luminance variances, two normalization techniques are applied to raw images before the actual extraction of Gabor features.

First, the face image is shifted, scaled and rotated so that the face shape in this image is aligned to the mean shape using affine transformation. An example affine transformation result of a face image is given in Figure 3.3.

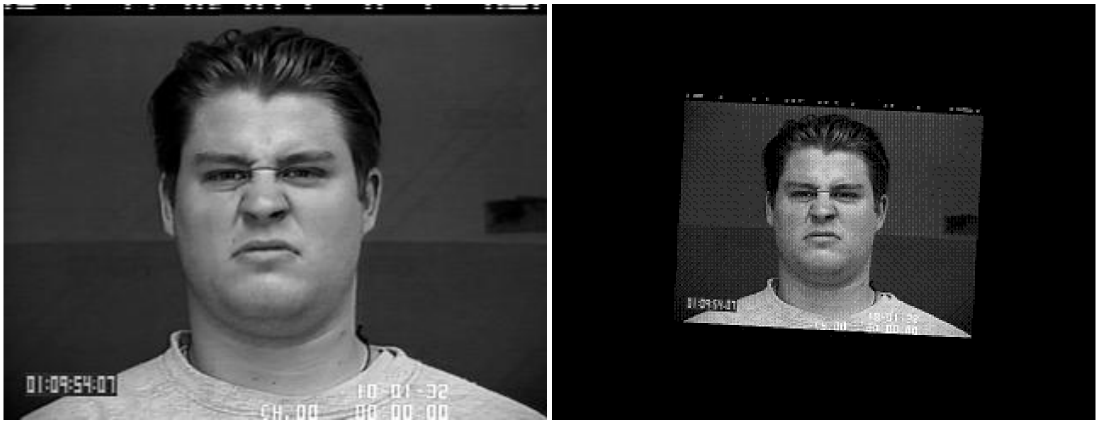


Figure 3.3: Example affine transformation result of a face image. The original image is scaled, shifted, and rotated so that the new face shape aligns to the optimally determined mean shape.

Then from this affine-transformed image we calculate the self-quotient image to attenuate variation of illumination. This is accomplished by first convolving the transformed image with a Gaussian smooth filter and then dividing it by its smoothed version. Figure 3.4 illustrates how self-quotient image is obtained.

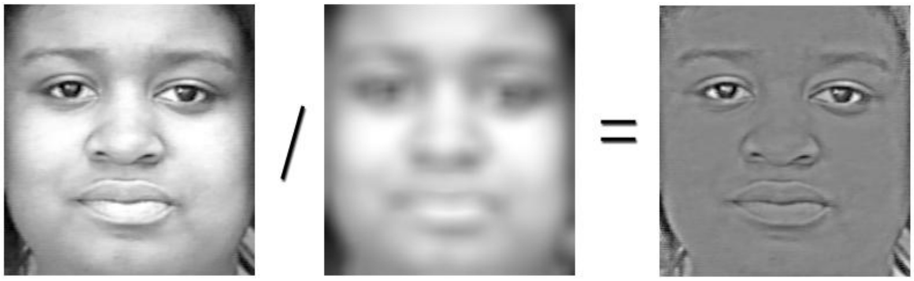


Figure 3.4: Computation of self-quotient image.

3.2.2 Gabor Feature Extraction

A family of Gabor filter can be expressed as a Gaussian modulated sinusoid in the spatial domain. Mathematically, a Gabor filter is defined as

$$g(\boldsymbol{z})_{u,v} = \frac{\left\|k_{u,v}\right\|^2}{\sigma^2} exp\left\{-\frac{\left\|k_{u,v}\right\|^2 \|\boldsymbol{z}\|^2}{2\sigma^2}\right\}\{exp\left(i\boldsymbol{z}k_{u,v}\right) - exp\,(-\frac{\sigma^2}{2})\}$$

where $k_{u,v} = \frac{k_v cos\phi_u}{k_v sin\phi_u}$, $k_v = 2^{-\frac{v+2}{2}}\pi$ and $\phi_u = \frac{\pi u}{k}$ are separately modulating frequency and modulating orientation, $\boldsymbol{z} = (x, y)$ are the pixel coordinates in the spatial domain. u is the orientation of a Gabor filter and v is the scale of a Gabor filter. The wavelength is decided by v. Furthermore, the second term of the Gabor filter $exp\,(-\frac{\sigma^2}{2})$ compensates for the direct current component value, because the cosine component has a nonzero mean while the sine component has a zero mean.

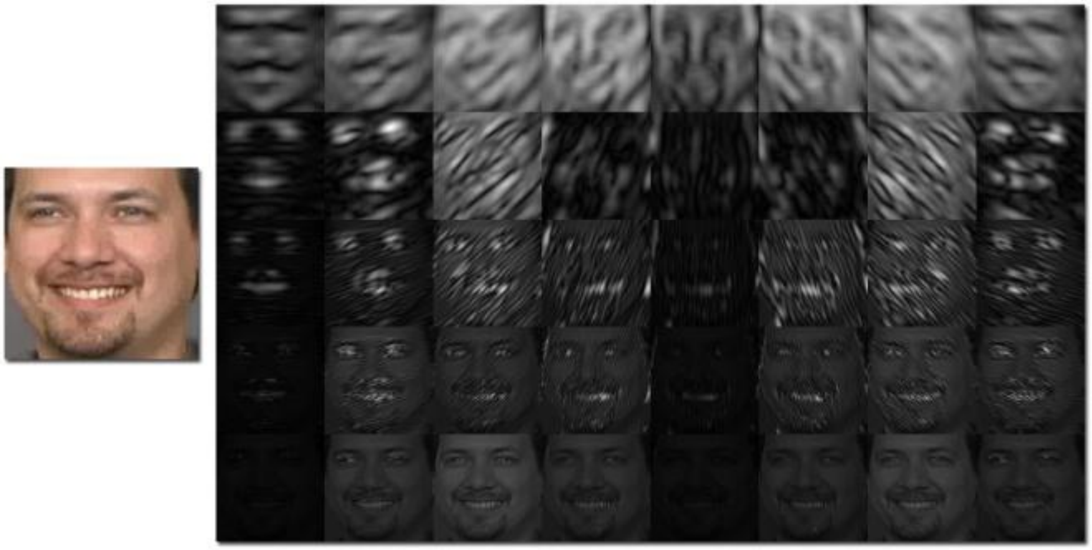


Figure 3.5: Original face image (left) and the set of output images of Gabor filter bank (right). The face image is from the Moving Faces and People dataset. It can be observed, from the figure, how the changes in orientation and wave-factor in the Gabor filter affect the response of the image.

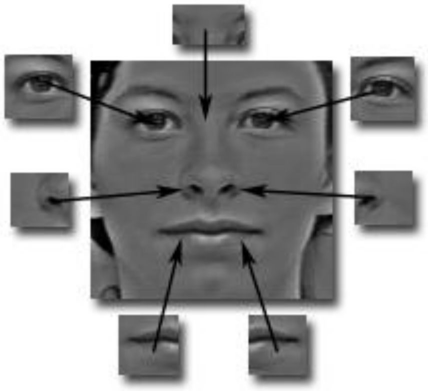


Figure 3.6: Face image patches where Gabor features are extracted.

The response image output by a Gabor filter will have peak value at the orientation and frequency that matches with the Gabor filter. By using a Gabor filter bank of several scales and orientations, we can obtain a fairly high resolution in both the spatial and spectral domain. Our Gabor filter bank $G = g(\boldsymbol{z})_{m,n}$ consists of filters at 5 scales $(m = 1,2,\dots,5)$ and 8 orientations$(n = 1,2,\dots,8)$. Then, there are 5×8 Gabor wavelet kernel filters. An example of Gabor filtered images $I(x,y)_{(m,n)}$ is shown in Figure 3.5.

Since it has been shown that the mouth contributes the most to a particular expression, followed by the canthus and eyebrows, we crop a total number of 7 patches from the self-quotient image to serve as expression identification regions, as shown in Figure 3.6. Gabor filter bank is then applied to these 7 patches respectively, resulting in a Gabor feature vector of dimension 560. To remove redundancy, Principal Component Analysis (PCA) is employed to reduce data dimension to 80 while retaining 98% of energy.

Denoting the face shape vector and the Gabor feature vector as s and g respectively, a particular expression could be represented as a concatenation s of g:

$$\mathrm{x} = [\mathrm{s}^{\mathrm{T}}, \lambda \cdot \mathrm{g}^{\mathrm{T}}]^{\mathrm{T}}$$

where λ is a weighting factor balancing the relative importance of shape and texture. To further reduce data dimension, PCA is performed on x to derive the final representation of facial expression. Without causing confusion, we will still use x to represent facial expression in the later parts of this thesis.

To select a proper λ such that s and g are commensurate, we estimate the effect of varying s on g using a similar method in [8]. To do this, we displace s from its ground truth position and the RMS change in g per unit RMS change in s is recorded. The weighting factor λ is set as the inverse of the average value of RMS change of all training examples.

Chapter 4: Distance Metric Learning

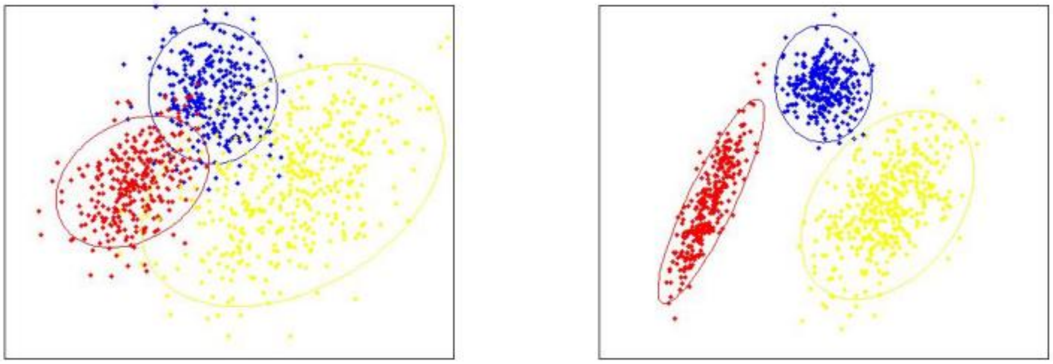
Many algorithms in pattern recognition rely on some distance metric for measuring the similarity between two objects. A good metric should supply high similarity for objects of the same category, and a low one for those of different categories. L_p norm is a frequently used metric due to its simplicity. Kernel methods can be seen as an attempt to transform default Euclidean geometry with a non-linear kernel operation to a high dimensional feature space and has achieved wide applicability in the pattern recognition community. Other methods like Linear Discriminant Analysis seek to project data to a subspace that maximizes inter-class variance while keeping intra-class variance as small as possible.

For a kNN classifier, the class label is determined by the consensus of k nearest neighbors. Traditionally, in the absence of prior knowledge on the statistical regularities in data, the Euclidean distance is used to measure the dissimilarity between instances. However, as shown by some researchers [25, 26], kNN performance can be significantly improved by exploiting the inherent data embeddings and learning a distance metric accordingly.

Distance metric learning is an emerging method that allows for more flexible transformation of feature space so that in the derived feature space, similar examples are closer to each other while dissimilar examples are separated by a large margin.

Distance metric learning can be supervised or semi-supervised, depending on the side-information that is available. In supervised metric learning, each input carries a label indicating the class it belongs to. Inputs that belong to the same class are considered similar; otherwise they are considered dissimilar. In semi-supervised metric learning, both labeled and unlabeled data is available. And for unlabeled data, the similarity/dissimilarity relationship is directly given as a priori. In the following, we will mainly focus on discussing supervised metric learning.

In supervised metric learning, each training example is annotated with its class label. Pairwise constraints on the data are inferred from the label information. There are two types of pairwise constraints: (1) equivalence constraints, which require that the given pair is semantically-similar because they have the same label and should be close together in the new feature space derived by the learned metric; and (2) inequivalence constraints, which state that the given points are semantically-dissimilar because they have different class labels and should be far away from each other in the deirved metric space.



(a) Before metric learning (b) After metric learning

Figure 4.1: Schematic illustration of data distribution before and after metric learning. Class label is denoted by the color. The distance metric is optimized so that similarly labeled data is tightly clustered and differently labeled data is separated by a large margin.

Let $C = \{(x_1, c_1), (x_2, c_2), \dots, (x_n, c_n)\}$ be the training data set of n data points, where each data point $x_i \in \mathbb{R}^m (i = 1,2, \dots, n)$ is a m-dimensional feature vector, and c_i is the corresponding class label. Let the set of similar examples be denoted by

$$S = \{(x_i, x_j) |\ c_i = c_j\}$$

and the set of dissimilar examples denoted by

$$D = \{(x_i, x_j) |\ c_i \neq c_j\}$$

The learning dynamics of distance metric learning are illustrated in Figure 4.1. Particularly, we seek a matrix $A \in \mathbb{R}^{m \times m}$ that parameterizes the (squared) generalized Mahalanobis distance between two examples x_i and x_j:

$$d_A\ (x_i, x_j)\ =\ \left(x_i - x_j\right)^T A(x_i - x_j)$$

Metric learning can be realized in various ways. In Large Margin Nearest Neighbor Metric Learning (LMNN) [53], metric learning is realized. The goal is to learn a metric matrix $A \in \mathbb{R}^{m \times m}$ such that, in the derived feature space, each input x_i and its k nearest neighbors share the same label while inputs labeled differently are separated from x_i by a large margin. In particular, this procedure requires k target neighbors to be identified. Usually the k nearest neighbors within the same class under Euclidean geometry are selected to be the k target neighbors and should become the k nearest neighbors in the derived feature space.

As compared to LMNN, Information Theoretic Metric Learning (ITML) [19] applies regularization to the distance metric during the learning process, making it less prone to overfitting. Thus, we choose ITML as the metric learning algorithm in this work.

4.1 INFORMATION THEORETIC METRIC LEARNING

In Information Theoretic Metric Learning, the objective is to learn a distance matrix $A \in \mathbb{R}^{m \times m}$ regularized by A_0 that satisfies pairwise constraints imposed by label information:

$$\min_{A} LogDet(A, A_0)$$

$$s.t.\, d_A(x_i, x_j) \leq l_{i,j} \ \ if\ (i,j) \in \ S$$

$$d_A(x_i, x_j) \geq \ u_{i,j} \ \ if\ (i,j) \in \ D$$

$$A \geq 0$$

where A_0 is a known matrix and serves to regularize A so that overfitting is alleviated, $LogDet(A, A_0)$ is the LogDet divergence between A and A_0 , $A \geq 0$ requires A to be semi-positive definite, and $l_{i,j}$ and $u_{i,j}$ are the lower and upper bound for similar and dissimilar pairs respectively.

4.2 Expression Classification Using ITML

In our work, A_0 is chosen to be the inverse of the covariance matrix of the training data. For examples from two different categories, $u_{i,j}$ is set as the 80^{th} percentiles of the sample histogram of distances between all dissimilar pairs of these two categories. For examples from the same category, $l_{i,j}$ is set as the 20^{th} percentile of the sample histogram of distances between similar pairs within category. In total, we use 7 lower bounds and 21 upper bounds for a 7-class expression classification problem. To classify an unseen example, k nearest neighbors are first retrieved based on the learned metric, and weighting is further applied to the votes of these k nearest neighbors to determine the final wining expression category.

The algorithmic steps used to perform the ML-based kNN voting are described as a two-stage process. In the very first stage, the bootstrapping stage, the model is constructed and the distance metric is learned. After the training is finished, our voting method proceeds to stage 2, where the classification decision is made.

In implementation, k is chosen to be 100 and w_i is chosen to be geometric progression with a common ratio of 0.9. This makes intuitive sense since the vote of a nearest neighbor with lower ranking should be weighted progressively less. (In our experiment, k and the common ratio is selected by linear grid search at regular intervals. We find that the common ratio affects the classification accuracy the most whereas k only has a slight impact on the classification accuracy.)

Stage 1 Metric Learning

Input: The training set of face images $I = \{I_i\}$ with different expressions, the set of face shapes $S = \{s_i\}$ and the set of expression labels $C = \{c_i\}$, the weighting factor λ to balance the relative importance of face shape and texture.

Output: Distance metric A.

1: Compute the optimally determined mean shape $\overline{m}$ from S using Generalized Procrustes Analysis;

2: For each $s_i \in S$, align s_i to $\overline{m}$ using affine transform $aftr_i$, then transform $I_i \in I$ with $aftr_i$. Still denote the resulting face image set and face shape set as I and S respectively for convenience;

3: For each $I_i \in I$, extract Gabor features g_i;

4: Concatenate s_i and g_i to derive the final representation of expression $\mathrm{x} = [\mathrm{s}^\mathrm{T}, \lambda \cdot \mathrm{g}^\mathrm{T}]^\mathrm{T}$;

5: According to expression label information C, form the set of similar pairs S and the set of dissimilar pairs D and calculate pairwise constraints $u_{i,j}$ and $l_{i,j}$;

6: Optimize the distance metric A with the goal of satisfying pairwise constraints imposed by $u_{i,j}$ and $l_{i,j}$ using the ITML algorithm.

Stage 2 Classification

Input: Distance metric A, query facial expression x, the number k of nearest neighbors to extract, weighting sequence $\{w_i(i = 1, 2, \dots, k)\}$ controlling the decay of k votes of the nearest neighbors.

Output: Winning expression label c.

1: For a new expression x, calculate its distance to all examples in the dataset using A;

2: Retrieve the k nearest neighbors, rank them according to their similarity to x. Denote the corresponding expression label sequence as $\{c_i'(i = 1, 2, \dots, k)\}$;

3: Denote the score for each expression label as $score_j$ and initialize each score to 0.0. Apply $\{w_i(i = 1, 2, \dots, k)\}$ to the votes of $\{c_i'(i = 1, 2, \dots, k)\}$ to derive the final score for each expression using the following routine:

```
for $c_i'$ in $\{c_i'(i = 1, 2, \dots, k)\}$
    $score_{c_i'} = score_{c_i'} + w_i$
```

4: Set the winning label c as the one that has the maximum score.

Chapter 5: Scaled ML-based KNN via LSH

A k Nearest Neighbor classifier would require a linear scan of all examples in the database in order to find the most similar examples and produce a classification decision, thus being computationally expensive. While our ML-based k-NN classifier working at the scale of 10,000 facial images can reach a processing speed of 17fps on a Dell desktop with 3.6GHz Intel Core i7 CPU and 4G memory, this method would virtually become computationally infeasible as a REAL-TIME algorithm as new data-rich collections with more facial images and expression categories are continuously being introduced. To gear towards future large-scale data set, Locality Sensitive Hashing (LSH) [64] is adopted to increase the computation speed at the expense of acceptable loss in classification accuracy.

Definition A locality sensitive hashing scheme is a distribution on a family F of hash functions operating on a collection of objects, such that for two objects x_1, x_2,

$$Pr_{\hbar\in F}\,[\hbar(x_1) = \hbar(x_2)] \ = sim(x_1, x_2)$$

where $sim(x_1, x_2)$ is some similarity function defined on the collection of objects. Note that [7] gives a slightly different definition of LSH than that in [64], although in the same spirit. In [7], a family F is said to be (r_1, r_2, p_1, p_2)-sensitive for a similarity measure $sim(x_1, x_2)$ if $Pr_{h\in F}\,[h(x_1) = h(x_2)] \geq p_1$ when $sim(x_1, x_2) \geq r_1$ and $Pr_{h\in F}\,[h(x_1) = h(x_2)] \leq p_2$ when $sim(x_1, x_2) \leq r_2$. In this work, we adhere to the definition in [64]. Given a locality sensitive hash function family F that corresponds to similarity function $sim(x_1, x_2)$, it is showed that using such a hashing scheme could construct efficient data structures for retrieving approximate nearest-neighbor on the collection of objects.

The basic idea of LSH is to compute a hash key for each example x in the database using a family of hashing functions $h(x) \in F$ so that similar examples will have a higher probability of collision in the hash table. The hash function $h(x)$ should satisfy the locality sensitive hashing property.

Commonly used similarity function and corresponding hash function family are inner product similarity and random hyperplane projection defined as follows [64]:

$$\mathrm{sim}(\mathrm{x_i}, \mathrm{x_j}) \;=\; 1 - \frac{1}{\pi}\cos^{-1}\left(\frac{\mathrm{x_i^T\,x_j}}{\|\mathrm{x_i}\|\|\mathrm{x_j}\|}\right)$$

$$\mathrm{h_r(x)} = \begin{cases} 1 & \text{if } \mathrm{r^T x} > 0 \\ 0, & \text{otherwise} \end{cases}$$

where r is a random vector drawn from multivariate normal distribution with zero mean and identity variance $N(0, I)$. To estimate the similarity between two examples, the hash keys are formed by concatenating the output of l hash functions drawn from F, and the hamming distance between these two l-bit keys are calculated.

To account for the effect of the learned metric from the previous chapter, we take the similar approach as proposed in [24] and adapt the similarity function and hash function to have the following form:

$$\mathrm{sim}(\mathrm{x_i}, \mathrm{x_j}) \;=\; 1 - \frac{1}{\pi}\cos^{-1}\left(\frac{\mathrm{x_i^T\,Ax_j}}{\|\mathrm{Lx_i}\|\|\mathrm{Lx_j}\|}\right)$$

$$\mathrm{h_r(x)} = \begin{cases} 1 & \text{if } \mathrm{r^T Lx} > 0 \\ 0, & \text{otherwise} \end{cases}$$

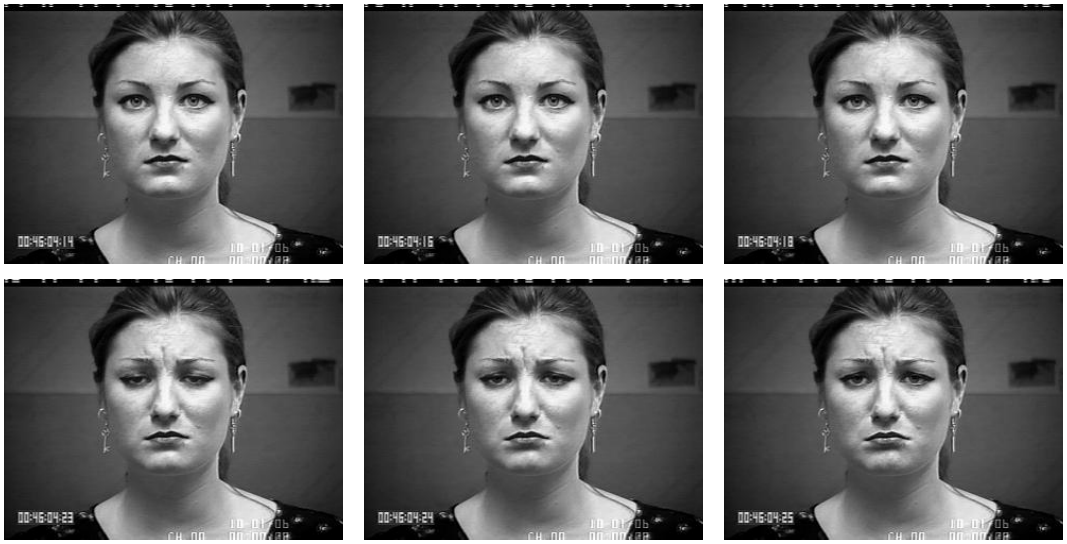
where L is the Cholesky decomposition of A satisfying $A = LL^T$.

At query time, the vector representing an expression is hashed directly to a specific position in the hash table and all examples that are in the same place as the query vector are returned as similar candidates. The k nearest neighbors are then selected via a linear search through similar candidates. The computational cost is incurred the most when taking the sequence of examples that collided and sorting them by their similarity to the query. Since the range of search for nearest neighbors is significantly reduced, LSH makes possible the faster search of high-dimensional feature space.

Chapter 6: Experiments and Results

Our primary goal is to verify that the proposed metric learning method can indeed learn a metric that adapts to the feature embeddings of facial expressions, and improve the “hit rate” when retrieving nearest neighbors. To this end, we perform a comparative study of several widely used methods, including standard kNN [63], SVM [5], and LDCRF [15] (LDCRF is reported to give the highest recognition rate for a 7-class expression recognition problem). Experiments show that our method outperforms the state-of-the-art. In particular, we empirically derive the average recognition rate of 5 classification methods on 2 different datasets and contrast the confusion matrix obtained from LDCRF [15] and our ML-based kNN. An interesting plot of the first 3 principal components of facial expressions before and after metric learning further demonstrates the discriminative power of our method on subtle expressions. We use LIBSVM [23] for experimenting on SVM, and implementation of LDCRF is based on [15].

6.1 Overview of the Dataset



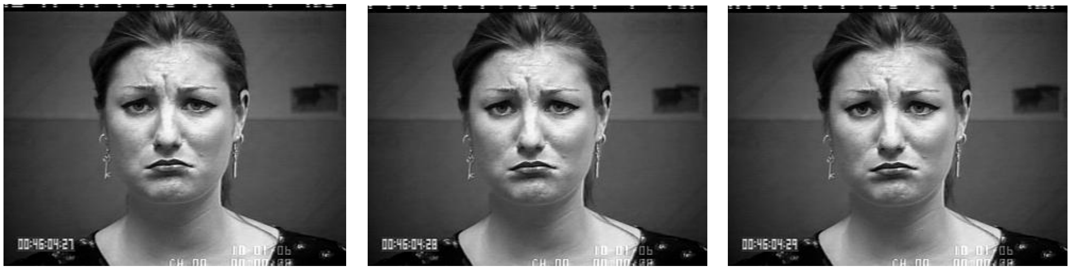


Figure 6.1: Image sequences from CK+ database showing the formation of sadness from onset to peak.

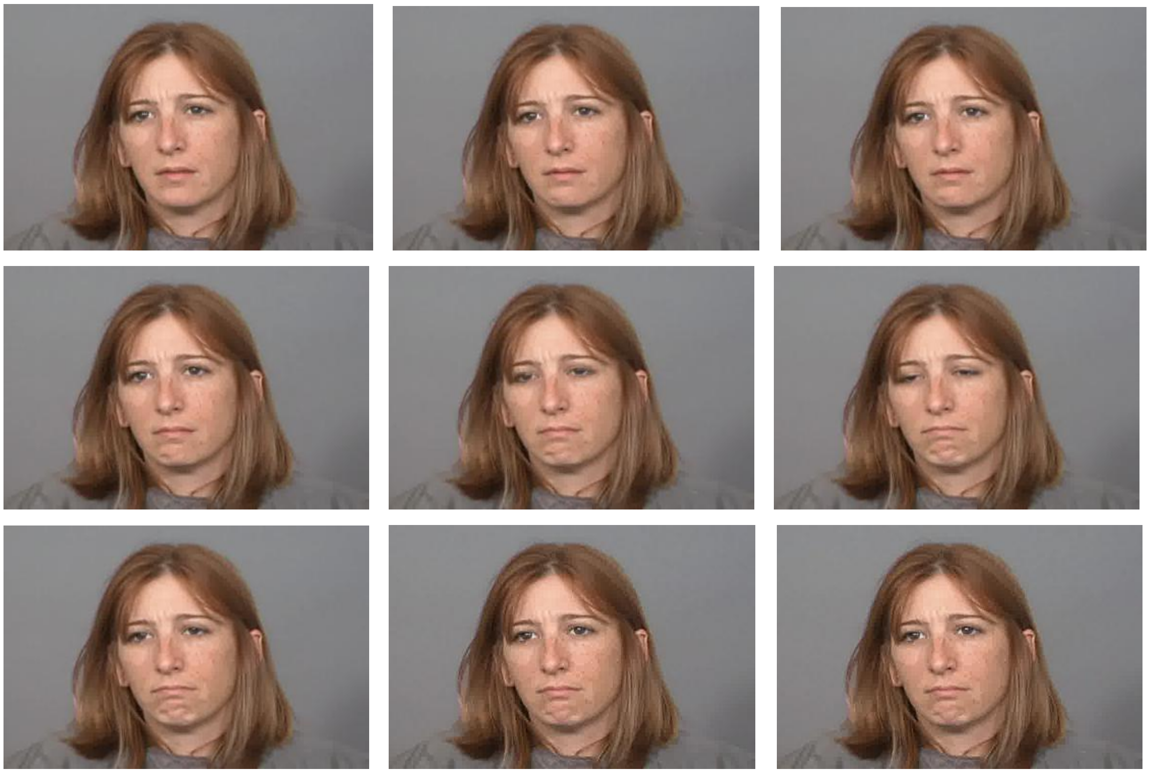


Figure 6.2: Image sequences from MFP database showing the formation of sadness from onset to peak.

We evaluate our algorithm on two datasets, the Extended Cohn-Kanade (CK+) dataset [20] and the Moving Faces and People (MFP) dataset [21]. The first one, CK+, is one of the most widely used test-bed for face analysis algorithms and consists of AU-coded and expression-labeled face images of individuals, taken under relatively controlled viewpoints and illumination conditions. Current state-of-the-art expression recognition systems have saturated in performance on this dataset, and we include evaluation on it for the purpose of comparison of our method against other systems.

We also evaluate our algorithm on a more challenging dataset, the MFP dataset. It contains a variety of still images and videos of individuals in natural context. Human expressions exhibited in MFP are all naturally induced by scenes from movies and television programs rather than posed ones, thus being subtle and more difficult to recognize. Figure 6.1 and Figure 6.2 show two image sequences demonstrating the formation of sadness from onset to peak, with the first from CK+ and the second from MFP. To the best of our knowledge, no previous experimental results of expression recognition on MFP are available, most probably due to the fact that it is a dataset of spontaneous, hard-to-classify facial expressions.

One difference between CK+ and MFP motivates us to take different approaches when evaluating the performance of our method: CK+ carries ground-truth landmarks with itself but MFP does not. Hence, we implement the following three approaches of training and testing: 1) Training and testing on CK+ with ground truth facial landmarks and expression labels provided by the dataset itself; 2) Training and testing on MFP with ground truth facial landmarks and expression labels obtained by manual annotation; and 3) Training and testing on MFP with ground truth facial landmarks and expression labels obtained from automatic annotation based on [9]. (It should be noted that a real facial expression recognition system need to automatically detect landmarks in test images.) To maximize the amount of training and testing data, a five-fold cross-validation configuration is used.

6.2 Experimental Results

6.2.1 CK+

CK+ contains 593 sequences from 123 subjects. Out of the 593 sequences, 309 were labeled as one of the six basic expressions. Since all the sequences start from the neutral pose to the peak formation of the expression, to train a discriminative model, we split each continuous sequence into two halves: the first half is labeled as neutral, and the second half is labeled as expressive. As a result, we set up a 7-class classification experiment for CK+.

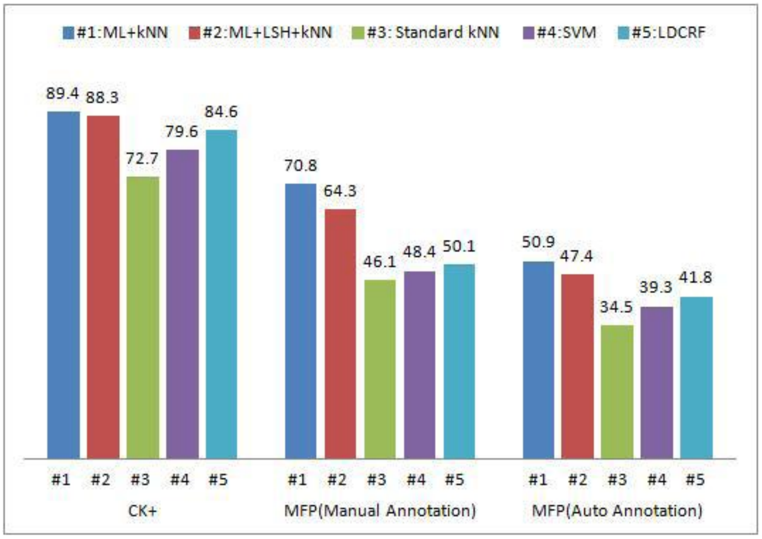


Figure 6.3: Average recognition rate of various methods for expression classification. See text for detailed discussion.

The average recognition rate of our method as well as the state-of-the-art methods is given in Figure 6.3. The confusion matrices of our method and LDCRF are given in Table 1 and Table 2 respectively (LDCRF [15] reports the best performance among the state of the art, so other methods' confusion matrices are not given here).

	Neutral	Anger	Disgust	Fear	Happiness	Sadness	Surprise
Neutral	**96.7**	0.7	0.0	0.3	1.0	1.3	0.0
Anger	11.1	**84.9**	0.0	0.0	0.0	0.0	0.0
Disgust	15.2	0.0	**83.1**	0.0	1.7	0.0	0.0
Fear	20.0	0.0	0.0	**80.0**	0.0	0.0	0.0
Happiness	4.3	0.0	0.0	0.0	**95.7**	0.0	0.0
Sadness	7.2	0.0	0.0	0.0	0.0	**92.8**	0.0
Surprise	5.6	0.0	0.0	0.0	0.0	0.0	**94.4**

Table 6.1: Confusion Matrix for 7-Class Expression Classification Using ML-based KNN on CK+.

	Neutral	Anger	Disgust	Fear	Happiness	Sadness	Surprise
Neutral	**73.5**	6.0	1.6	1.9	2.6	9.2	5.2
Anger	20.6	**76.6**	1.1	0.0	1.6	0.0	0.0
Disgust	2.7	6.2	**81.5**	0.0	9.6	0.0	0.0
Fear	0.0	0.0	0.0	**94.4**	0.0	4.2	1.4
Happiness	0.5	1.0	0.0	0.0	**98.6**	0.0	0.0
Sadness	21.5	0.0	0.0	1.3	0.0	**77.2**	0.0
Surprise	0.9	0.0	0.0	0.0	0.0	0.0	**99.1**

Table 6.2: Confusion Matrix for 7-Class Expression Classification Using LDCRF on CK+.

From the comparison between Table 6.1 and Table 6.2, it is clear that ML-based kNN achieves less confusion between subtle expressions such as neutrality, anger, fear, etc, which is exactly the reason why our method outperforms others with a recognition rate of 89.4%. Of course, LDCRF is a probabilistic method of modeling dynamically varying patterns whereas our method tries to uncover the interrelationships between different examples in a static manner. The optimal way for the classification task at hand is to incorporate metric learning into the LDCRF model in a unified framework; this is subject for future research.

6.2.2 MFP

MFP is a database of static images and video clips of human faces and people. Of more interest to our investigation are the Dynamic Facial Expressions video clips that show spontaneous expressions of subjects watching a 10 minute video clip. There are several drawbacks to working directly with these video clips: a) Expressions in each video vary in length. Some occur over a few frames, others may last many seconds; b) These expressions are not verified and subjects may respond to stimulus with a non-intended expression (e.g. aiming at inducing fear but actually getting disgust; c) Some clips contain more than one expression (e.g. a fear expression may be accompanied by a surprise); and d) None of these clips come with landmarks nor expression labels.

Expression	Anger	Disgust	Fear	Happiness	Sadness	Surprise
Number of Examples	N/A	96	15	134	30	99

Table 6.3: Statistics of the MFP Dataset after Validation.

	Neutral	Disgust	Fear	Happiness	Sadness	Surprise
Neutral	**94.6**	0.0	0.0	2.1	0.0	0.0
Disgust	40.6	**56.2**	0.0	3.2	0.0	0.0
Fear	80	0.0	**20.0**	0.0	0.0	0.0
Happiness	3.5	0.0	0.0	**96.5**	0.0	0.0
Sadness	0.0	0.0	0.0	0.0	**100.0**	0.0
Surprise	15.2	0.0	0.0	0.0	0.0	**84.8**

Table 6.4: Confusion Matrix for 6-Class Expression Classification Using ML-based KNN Trained on the Manually Annotated MFP.

	Neutral	Disgust	Fear	Happiness	Sadness	Surprise
Neutral	**68.6**	1.3	13.4	3.1	4.1	9.5
Disgust	8.4	**81.9**	1.2	0	7.7	0.8
Fear	56.0	0.0	**27.5**	10.1	6.4	0.0
Happiness	0.9	1.0	0.0	**98.1**	0.0	0.0
Sadness	25.7	17.3	9.7	6.8	**40.5**	0.0
Surprise	15.2	27.7	12.9	4.8	6.8	**32.6**

Table 6.5: Confusion Matrix for 6-Class Expression Classification Using LDCRF Trained on the Manually Annotated MFP.

To suit our needs, the following steps are taken to validate the dataset: 1) Manually examine each clip and discard those containing non-intended expressions; 2) For each valid clip, cut it short so that it contains exactly the formation of an expression from onset to peak; and 3) Manually annotate frames in each cut-short clip. Table 6.3

gives detailed statistics about the validated dataset. Note that anger is excluded from our experiment, and we only perform a 6-class classification on the MFP dataset since we found no valid examples of anger at all.

With a validated MFP dataset, we first test the 5 expression recognition methods with the manually annotated landmarks, i.e. training and testing using approach 2. Furthermore, we also test the 5 expression recognition methods in a real-time setting using the auto face image annotator from [9], i.e. training and testing using approach 3. The average recognition rates of these 5 methods are given in Figure 6.3. Confusion matrices for ML-based kNN and LDCRF obtained using approach 2 are given in Table 6.4 and Table 6.5 respectively (Confusion matrices obtained using approach 3 are not given here).

Since MFP is a database of spontaneous expressions rather than acted ones, one should not be surprised that recognition rates on MFP decline significantly compared to those on CK+. However, ML-based kNN still gives much better results than other methods when tested on MFP. Through careful examination of Table 6.4 and Table 6.5, it is again evidenced that our method is conducive to high classification accuracy by removing indiscrimination between dissimilar examples while reinforcing alikeness between similar examples.

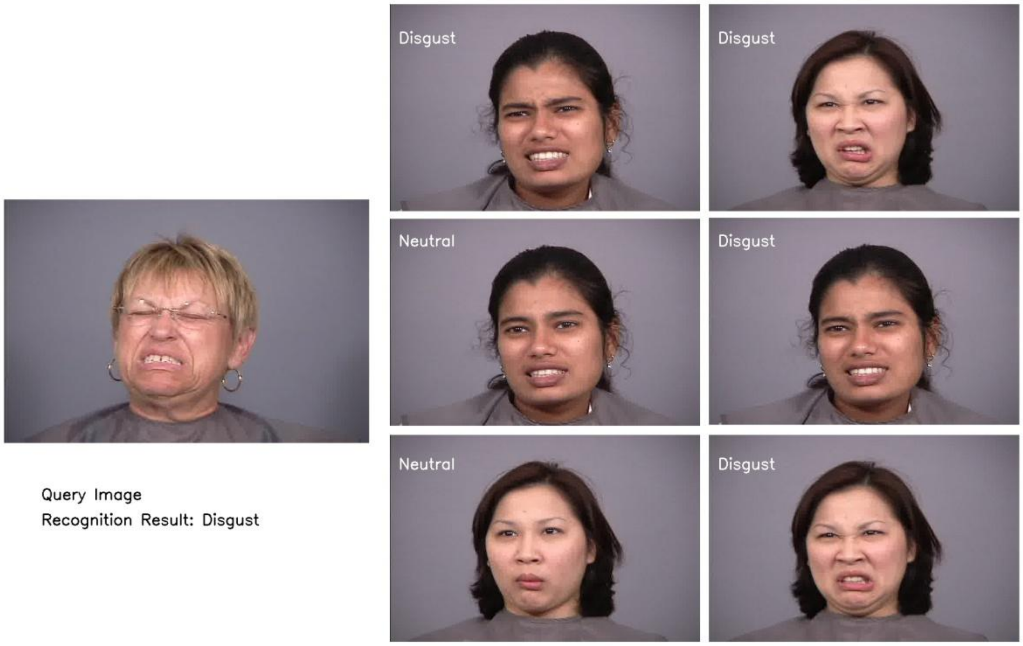
We could also see in Figure 6.3 that approach 3 yields the lowest recognition accuracy among the three training/testing approaches. This is in part due to the subtlety of expression and in part due to the inferior CLM-based auto face registration. For this very reason, we call for improvement on current face registration and alignment techniques.

What is noteworthy is that the recognition rate of LSH approximation of ML-based kNN voting is consistently better than all but ML-based kNN. In our experiment, the bit length of the hash key is selected to be 100, for which a frame rate of around 25 fps could be attained. In fact, the length of hash key controls the tradeoff between accuracy and speed. We could potentially improve the accuracy of LSH approximation by using a longer hash key, at the expense of higher computational complexity and lower frame rate. It should be noted that too long we do not observe any significant performance improvement beyond bit length of 130, where the program runs at a frame rate of 20 fps. The superior performance of the LSH approximation scheme demonstrates the scalability of our ML-based kNN voting method and qualifies itself as applicable to real world problems.

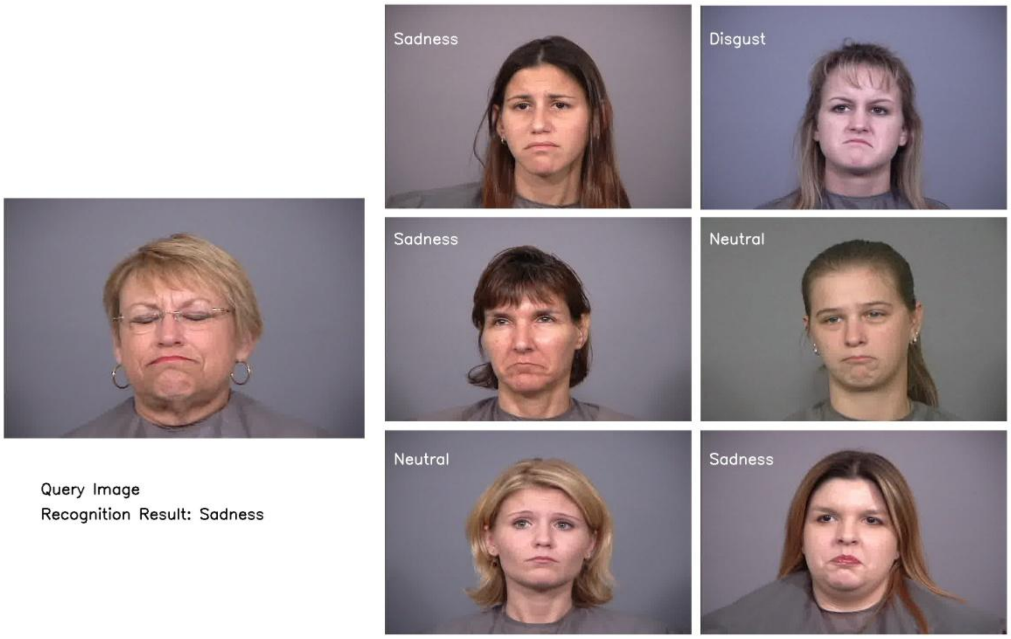
6.3 VISUALIZATION OF FEATURE EMBEDDING TRANSFORMATION

To exemplify the retrieval performance of ML-based kNN, we show in Figure 6.4 and Figure 6.5 the nearest neighbors retrieved by ML-based kNN and standard kNN respectively given the same query images. Disgust, sadness and surprise are used as the query expressions. As can be seen, ML-based kNN gives quite satisfactory results whereas disgust, sadness and surprise are confused with happiness and neutrality by the standard kNN.

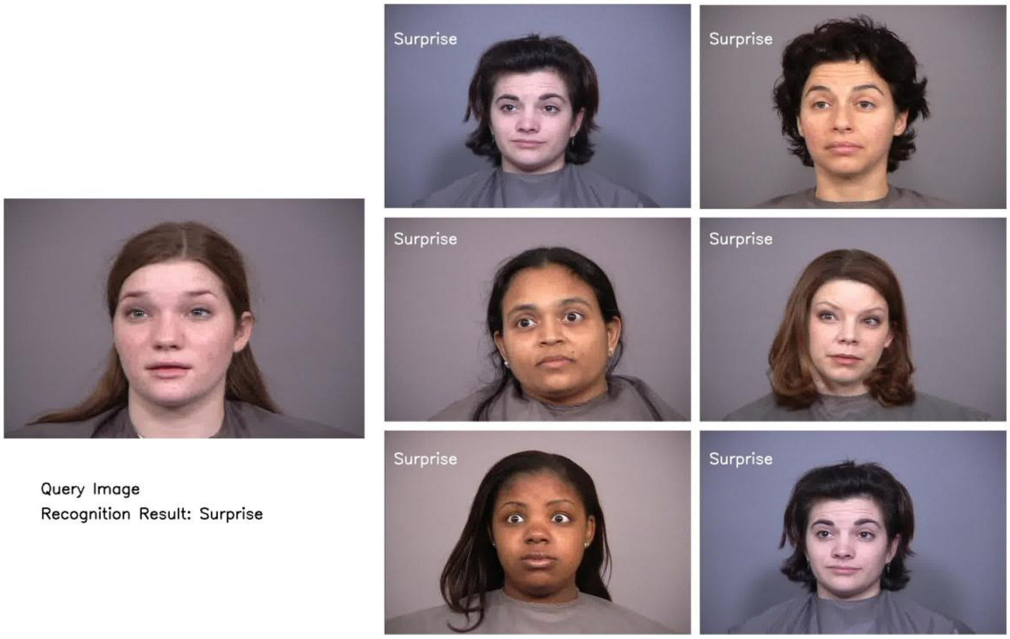
To visualize the transformation effect of metric learning on feature space, we further show a comparative plot of the first 3 principal components of facial expressions in CK+ before and after applying ML-based transformation in Figure 6.6 and Figure 6.7 respectively. Much overlap could be observed in Figure 6.6 between dissimilar expressions. The overlap is even exacerbated for subtle expressions such as anger, disgust, fear, and sadness. In contrast, Figure 6.7, which gives a PCA plot of expressions after ML-based transformation, shows a much better separated point distribution for differently labeled expressions, consequently facilitating higher recognition accuracy.



(a) Query image: disgust

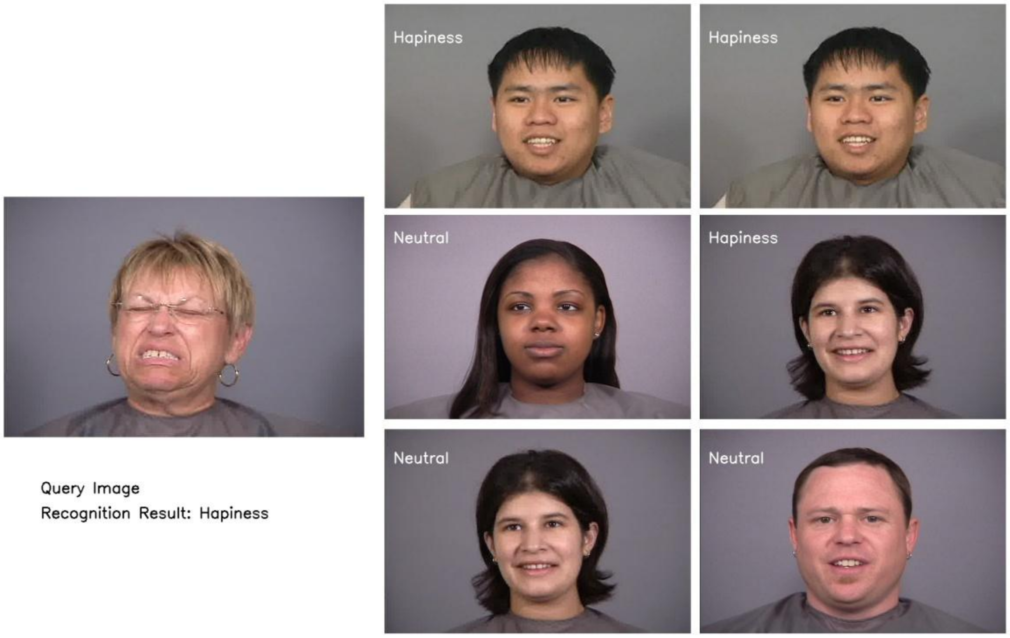


(b) Query image: sadness

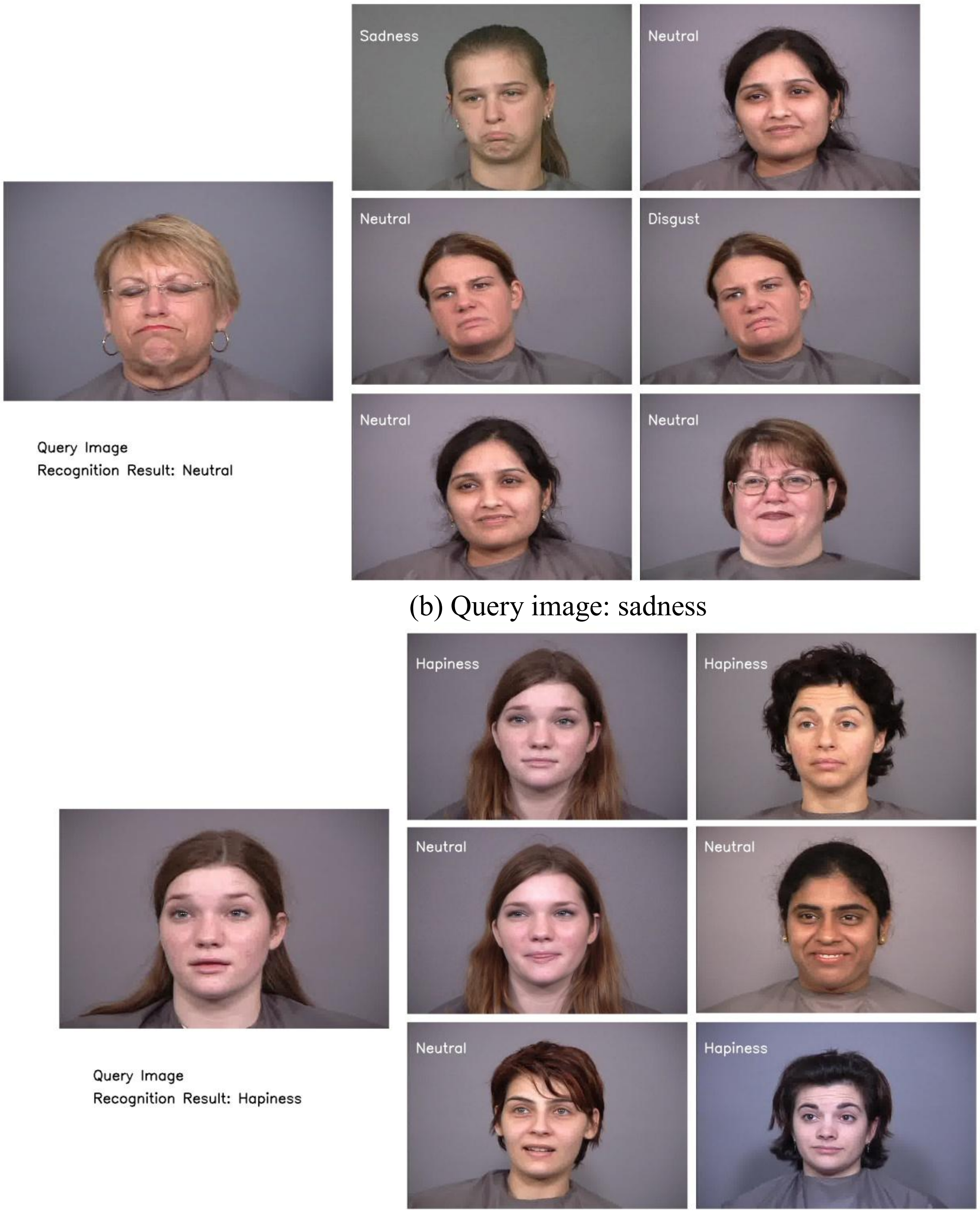


(c) Query image: surprise

Figure 6.4: Ranked list of nearest neighbors obtained using ML-based kNN along with the groundtruth labels. The ranked list is sorted from left to right, from top to bottom.



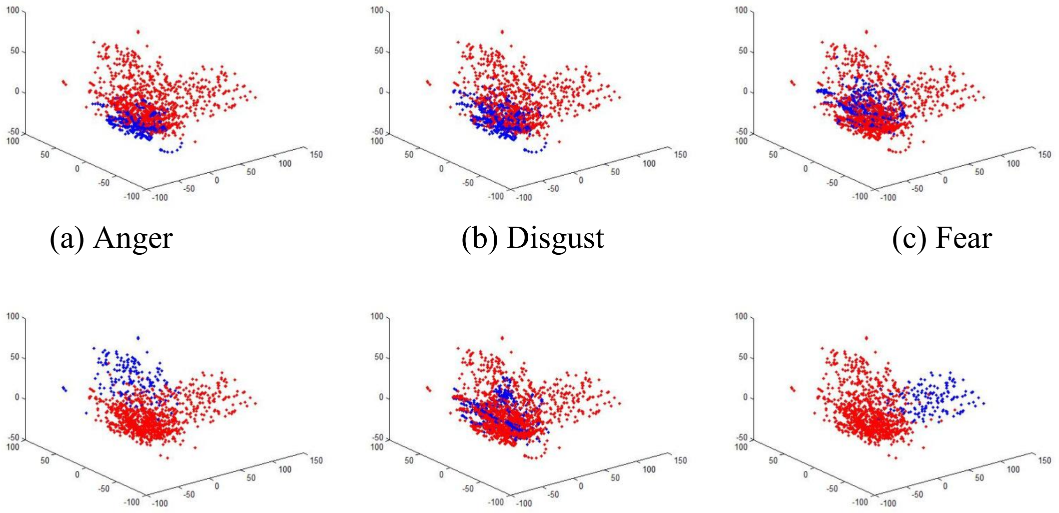
(a) Query image: disgust



(b) Query image: sadness

(c) Query image: surprise

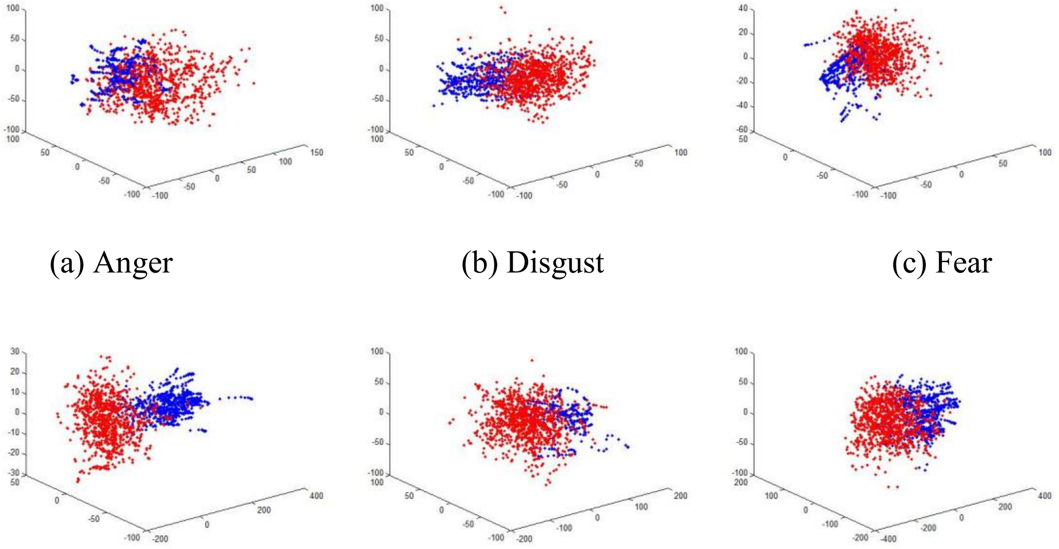
Figure 6.5: Ranked list of nearest neighbors obtained using standard kNN along with the groundtruth labels. . The ranked list is sorted from left to right, from top to bottom.



(a) Anger (b) Disgust (c) Fear

(d) Happiness (e) Sadness (f) Surprise

Figure 6.6: Plot of the first 3 principal components of different expression vectors BEFORE applying ML-based transformation. Blue denotes the expression of interest. Red denotes all expressions of non-interest.



(a) Anger (b) Disgust (c) Fear

(d) Happiness (e) Sadness (f) Surprise

Figure 6.7: Plot of the first 3 principal components of different expression vectors AFTER applying ML-based transformation. Blue denotes the expression of interest. Red denotes all expressions of non-interest.

Chapter 7: Conclusions

We present a new expression classification method using Metric Learning-based k-Nearest Neighbor voting. The metric is optimized with the goal that all similarly labeled inputs have small pairwise distances, while all differently labeled inputs have large pairwise distances. This method alleviates confusion between subtle expressions such as neutral, angry, fear, etc., thus outperforming the state-of-the-art methods. To speed up our method, an approximate yet efficient variant scheme of ML-based kNN voting taken from [24] is used. LSH allows fast indexing of similar examples with the help of a precomputed hash table and significantly accelerates the nearest neighbor matching process.

Experiments show that ML-based kNN demonstrates better classification especially when it comes to subtle expressions. Also, our LSH approximation scheme gives superior classification performance than the state-of-the-art, and more importantly, works at a faster speed, demonstrating the scalability and capability of our method.

Bibliography

[1] Chengjun Liu and Harry Wechsler. Gabor Feature Based Classification Using the Enhanced Fisher Linear Discriminant Model for Face Recognition. IEEE Trans. Image Processing, vol. 11, no. 4, pp. 467-476, 2002.

[2] Ekman P., Friesen W.V. Unmasking the Face: A guide to recognizing emotions from facial clues. Consulting Psychologists Press 1975.

[3] Dornaika, Fadi and Bogdan Raducanu. "Facial Expression Recognition for HCI Applications." Encyclopedia of Artificial Intelligence. IGI Global, 2009. 625-631. Web. 6 Jul. 2012. doi:10.4018/978-1-59904-849-9.ch095.

[4] B.A. Donohue, J.D. Bronzino, J.H. DiLiberti, D.P. Olson, L.R. Schweitzer, P. Walsh, Application of a neural network in recognizing facial expression, in: IEEE Proceedings of the Seventh Annual Northeast Bioengineering Conference, Hartford, CT, USA, 1991, pp. 206-207.

[5] Irene Kotsia, Ioannis Pitas. Facial Expression Recognition in Image Sequences Using Geometric Deformation Features and Support Vector Machines. IEEE Transactions on Image Processing 16(1): 172-187 (2007).

[6] T. M. Cover. Estimation by the Nearest Neighbor Rule. IEEE Trans. on Information Theory, 14(1):50-55, 1968.

[7] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In Proceedings of the thirtieth annual ACM symposium on Theory of computing, STOC'98, pages 604-613, New York, NY, USA, 1998. ACM.

[8] T.F. Cootes, G.J Edwards, and C,J. Taylor. Active Appearance Models, Proc. European Conference on Computer Vision 1998.

[9] Jason M. Saragih, Simon Lucey, and Jeffrey Cohn, Face Alignment through Subspace Constrained Mean-Shifts, International Conference of Computer Vision (ICCV), September, 2009.

[10] D. Cristinacce and T. F. Cootes. Feature Detection and Tracking with Constrained Local Models. In EMCV, pages 929-938, 2004.

[11] L. Gu and T. Kanade. A Generative Shape Regularization Model for Robust Face Alignment. In ECCV'08, 2008.

[12] Stefano Berretti, Alberto Del Bimbo, Pietro Pala, Boulbaba Ben Amor, Mohamed Daoudi: A Set of Selected SIFT Features for 3D Facial Expression Recognition. ICPR 2010: 4125-4128.

[13] Yuxiao Hu; Zhihong Zeng; Lijun Yin; Xiaozhou Wei; Xi Zhou; Huang, T.S. Multi-view facial expression recognition. FG 2008. Page(s): 1-6.

[14] Taskeed Jabid, Md. Hasanul Kabir, and Oksam Chae. Robust Facial Expression Recognition Based on Local Directional Pattern. ETRI Journal, vol.32, no.5, Oct. 2010, pp.784-794.

[15] Suyog Jain, Changbo Hu, J. K. Aggarwal. Facial Expression Recognition with Temporal Modeling of Shapes. 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), page(s): 1642-1649.

[16] S.M. Lajevardi, M. Lech. Averaged Gabor Filter Features for Facial Expression Recognition. Digital Image Computing: Techniques and Applications, 2008 (DICTA '08). 71-76.

[17] W. Fellenz, J. Taylor, N. Tsapatsoulis, S. Kollias, Comparing template-based, feature-based and supervised classification of facialexpressions from static images, Proceedings of Circuits, Systems, Communications and Computers (CSCC'99), Nugata, Japan, 1999, pp. 5331-5336.

[18] Shishir Bashyal, Ganesh K. Venayagamoorthy: Recognition of facial expressions using Gabor wavelets and learning vector quantization. Eng. Appl. of AI 21(7): 1056-1064 (2008).

[19] Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, Inderjit S. Dhillon: Information-theoretic metric learning. ICML 2007: 209-216.

[20] Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., \& Matthews, I. (2010). The Extended Cohn-Kande Dataset (CK+): A complete facial expression dataset for action unit and emotion-specified expression. Third IEEE Workshop on CVPR for Human Communicative Behavior Analysis (CVPR4HB 2010).

[21] Alice J. O'Toole, Joshua Harms, Sarah L. Snow, Dawn R. Hurst, Matthew R. Pappas, Janet H. Ayyad, Herve´ Abdi. A Video Database of Moving Faces and People. IEEE Transactions on Pattern Analysis And Machine Intelligence, Vol. 27, No. 5, May 2005.

[22] A. Kanaujia and D. N. Metaxas. Recognizing facial expressions by tracking feature shapes. In International Conference on Pattern Recognition, 33-38, 2006.

[23] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[24] Prateek Jain, Brian Kulis, Kristen Grauman. Fast Image Search for Learned Metrics. CVPR 2008.

[25] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In L. K. Saul, Y. Weiss, and L. Bottou, editors, Advances in Neural Information Processing Systems 17, 513-520, Cambridge, MA, 2005. MIT Press.

[26] C. Domeniconi, D. Gunopulos, and J. Peng. Large margin nearest neighbor classifiers. IEEE Transactions on Neural Networks, 16(4):899-909, 2005.

[27] Xudong Xie, Kin-Man Lam. Facial expression recognition based on shape and texture. Pattern Recognition. Volume 42, Issue 5, May 2009, Pages 1003-1011.

[28] I. Kotsia and I. Pitas. Facial expression recognition using shape and texture information. IFIP International Federation for Information Processing, 2006, Volume 217, Artificial Intelligence in Theory and Practice, Pages 365-374.

[29] R. Cowie and E. Douglas Cowie. Speakers and hearers are people: Reflections on speech deterioration as a consequence of acquired deafness, in Profound Deafness and Speech Communication, K-E. Spens and G. Plant, Eds. London, UK: Whurr, 1995, pp. 510-527.

[30] Chibelushi, C.C., Bourel, F., Facial Expression Recognition: A Brief Tutorial Overview. Available: "http://www.dai.ed.ac.uk/cgi-bin/rbf/CVONLINE/entries.pl?TAG878". In CVonline: On-Line Compendium of Computer Vision. R. Fisher (ed). Available: "http://www.dai.ed.ac.uk/CVonline/" . [9 January 2003].

[31] Iain Matthews, Simon Baker: Active Appearance Models Revisited. International Journal of Computer Vision 60(2): 135-164 (2004).

[32] Jason M. Saragih, Simon Lucey, Jeffrey F. Cohn: Face alignment through subspace constrained mean-shifts. ICCV 2009: 1034-1041.

[33] Y-L. Tian, T. Kanade, J.F. Cohn, Recognizing Action Units for Facial Expression Analysis, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 23, No. 2, pp. 97-115, 2001.

[34] J.J. Lien, T. Kanade, J.F. Cohn, C-C. Li, Automated Facial Expression Recognition Based on FACS Action Units, Proc. Third IEEE Int. Conf. Automatic Face and Gesture Recognition, pp. 390-395, 1998.

[35] Suyog Jain, Facial Expression Recognition with Temporal Modeling of Shapes. Master Thesis. The University of Texas at Austin.

[36] Atul Kanaujia and Dimitris N. Metaxas. Recognizing facial expressions by tracking feature shapes. In International Conference on Pattern Recognition, pages 33-38, 2006.

[37] Nicu Sebe, Michael S. Lew, Ira Cohen, Yafei Sun, Theo Gevers, and Thomas S. Huang. Authentic facial expression analysis. In IEEE International Conference on Automatic Face and Gesture Recognition, pages 517-522, 2004.

[38] Marian Stewart Bartlett, Gwen Littlewort, Claudia Lainscsek, Ian Fasel, and Javier Movellan. Machine learning methods for fully automatic recognition of facial expressions and facial actions. In Proc. IEEE Intl. Conf. Systems, Man and Cybernetics, pages 592-597, 2004.

[39] Gwen Littlewort, Jacob Whitehill, Tingfan Wu, Ian R. Fasel, Mark G. Frank, Javier R. Movellan, and Marian Stewart Bartlett. The computer expression recognition toolbox (cert). In IEEE International Conference on Automatic Face and Gesture Recognition, pages 298-1305, 2011.

[40] Caifeng Shan, Shaogang Gong, and Peter W. McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. Image Vision Comput., 27:803-816, May 2009.

[41] Guoying Zhao and Matti Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29:915-928, 2007.

[42] F. Bourel, C.C. Chibelushi, A.A. Low, Robust Facial Expression Recognition Using a State-Based Model of Spatially-Localised Facial Dynamics, Proc. Fifth IEEE Int. Conf. Automatic Face and Gesture Recognition, pp. 106-111, 2002

[43] Y. Zhu, Liyanage C. De Silva, and Chi Chung Ko. Using moment invariants and hmm in facial expression recognition. Pattern Recognition Letters, 23(1-3):83{91, 2002.

[44] T. Otsuka and J. Ohya. Recognizing multiple persons facial expressions using HMM based on automatic extraction of significant frames from image sequences. In Proc. Int. Conf. on Image Processing (ICIP-97), pages 546–549, Santa Barbara, CA, USA, Oct. 26-29, 1997.

[45] Marian Stewart Bartlett, Gwen C. Littlewort, Mark G. Frank, Claudia Lainscsek, Ian R. Fasel, and Javier R. Movellan. Automatic recognition of facial actions in spontaneous expressions, 2006.

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

[47] F. Bourel, C. Chibelushi, and A. Low. Recognition of facial expressions in the presence of occlusion. In Proceedings of the Twelfth British Machine Vision Conference, volume 1, pages 213{222. Citeseer, 2001.

[48] B. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. In Proceedings of the 7th international joint conference on Artificial intelligence, 1981.

[49] J. G. Daugman, Complete Discrete 2-D Gabor Transforms by Neural Networks for Image Analysis and Compression, IEEE Trans. Acoustic, speech and signal processing, Vol. 36 1988, pp.1169-1179.

[50] P. Ekman and W. V. Friesen. Facial Action Coding System: Investigator’s Guide. Palo Alto, CA: Consulting Psychologists Press, 1978.

[51] Yoshitomi, Y., Miyawaki, N., Tomita, S. & Kimura, S. (1997). Facial expression recognition using thermal image processing and neural network. , 380—385.

[52] Nicholas Vretos, Nikos Nikolaidis, Ioannis Pitas: 3D facial expression recognition using Zernike moments on depth images. ICIP 2011: 773-776.

[53] Kilian Q. Weinberger, John Blitzer, Lawrence K. Saul: Distance Metric Learning for Large Margin Nearest Neighbor Classification. NIPS 2005.

[54] E. Xing, A. Ng, M. Jordan, and S. Russell. Distance Metric Learning, with Application to Clustering with Side-Information. In NIPS, 2002.

[55] A. Globerson and S. Roweis. Metric Learning by Collapsing Classes. In NIPS, 2005.

[56] A. Frome, Y. Singer, and J. Malik. Image retrieval and classification using local distance functions. In B. Scholkopf, J. Platt, and T. Hofmann, editors, Advances in Neural Information Processing Systems 19, Cambridge, MA, 2007. MIT Press.

[57] J. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon. Information-Theoretic Metric Learning. In ICML, 2007.

[58] Jason M. Saragih, Simon Lucey, and Jeffrey Cohn International Conference of Computer Vision (ICCV), September, 2009.

[59] Rogerio Schmidt Feris, Jim Gemmell, Kentaro Toyama, and Volker Krüger. Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on. Page(s): 118 - 123. 20-21 May 2002.

[60] I.L.Dryden, K.V. Mardia, Statistical Shape Analysis, Wiley, Chichester, (1998).

[61] Sima Taheri, Pavan K. Turaga, Rama Chellappa: Towards view-invariant expression analysis using analytic shape manifolds. FG 2011: 306-313.

[62] Dasarathy, B.V. (eds.): Nearest Neighbor: Pattern Classification Techniques (Nn Norms: Nn Pattern Classification Techniques). IEEE Computer Society Press (1991).

[63] Abu Sayeed Md. Sohail, Prabir Bhattacharya. Classification of Facial Expressions Using K-Nearest Neighbor Classifier. Computer Vision/Computer Graphics Collaboration Techniques Lecture Notes in Computer Science Volume 4418, 2007, pp 555-566.

[64] Charikar, Moses 2002. Similarity Estimation Techniques from Rounding Algorithms In Proceedings of the 34th Annual ACM Symposium on Theory of Computing.

Vita

Shaohua Wan received the Bachelor of Science in Communication Engineering from Beijing University of Posts and Telecommunications, School of Information and Communication Engineering, in 2011. During his undergraduate studies, he was awarded with the National Scholarship in two consecutive years, the most prestigious scholarship in China awarded to the top 1% college students nation-wide. He was nominated as the Excellent Graduate in Beijing Area in 2011. He entered the University of Texas at Austin in Fall 2011, in the Department of Electrical and Computer Engineering.

Email: shaohuawan@utexas.edu

This thesis was typed by Shaohua Wan.