Grassmannian Learning for Facial Expression Recognition from Video

by

Anirudh Yellamraju

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved November 2014 by the
Graduate Supervisory Committee:

Chaitali Chakrabarti, Co-Chair
Pavan Turaga, Co-Chair
Lina Karam

ARIZONA STATE UNIVERSITY

December 2014

ABSTRACT

In this thesis we consider the problem of facial expression recognition (FER) from video sequences. Our method is based on subspace representations and Grassmann manifold based learning. We use Local Binary Pattern (LBP) at the frame level for representing the facial features. Next we develop a model to represent the video sequence in a lower dimensional expression subspace and also as a linear dynamical system using Autoregressive Moving Average (ARMA) model. As these subspaces lie on Grassmann space, we use Grassmann manifold based learning techniques such as kernel Fisher Discriminant Analysis with Grassmann kernels for classification. We consider six expressions namely, Angry (AN), Disgust (Di), Fear (Fe), Happy (Ha), Sadness (Sa) and Surprise (Su) for classification. We perform experiments on extended Cohn-Kanade (CK+) facial expression database to evaluate the expression recognition performance. Our method demonstrates good expression recognition performance outperforming other state of the art FER algorithms. We achieve an average recognition accuracy of 97.41% using a method based on expression subspace, kernel-FDA and Support Vector Machines (SVM) classifier. By using a simpler classifier, 1-Nearest Neighbor (1-NN) along with kernel-FDA, we achieve a recognition accuracy of 97.09%. We find that to process a group of 19 frames in a video sequence, LBP feature extraction requires majority of computation time (97 %) which is about 1.662 seconds on the Intel Core i3, dual core platform. However when only 3 frames (onset, middle and peak) of a video sequence are used, the computational complexity is reduced by about 83.75 % to 260 milliseconds at the expense of drop in the recognition accuracy to 92.88 %.

DEDICATION

To my Parents, Smt. Neppalli Katyayani Devi Yellamraju and Shri Y.V.Subbarao

ACKNOWLEDGMENTS

This thesis would not have been possible without the continuous support and guidance of my advisors Dr. Chaitali Chakrabarti and Dr. Pavan Turaga. I thank them for their help and patience especially during the initial phases of my research study. I would also like to thank my family and friends for their support throughout my research work.

TABLE OF CONTENTS

Page

CHAPTER

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Human expressions are very important cues for understanding various forms of non-verbal communication. They convey the information about a person's emotional state and so play a vital role in human computer interaction (HCI) based applications. Human expressions are primarily characterized through:

- Emotion recognition based on speech
- Expression recognition using visual information from facial images.

Facial expression recognition has received a lot of attention in recent times. In this work we focus on extracting spatio-temporal information from facial images in order to recognize facial expression from video sequences.

1.1 Related work in facial expression recognition

For facial expression recognition (FER) there are two key steps: 1) image registration, and 2) feature extraction. Image registration is a preprocessing step that includes face detection, image alignment and normalization. In the feature extraction step, either only appearance based features are extracted (static techniques) or temporal information (dynamic technique) in addition to static features is extracted. Most of the static techniques extract the shape features such as landmark points or combination of various facial action units (AU). The static algorithms use appearance based features that represent the textural information in facial images using Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG) feature descriptors etc.

Although these methods perform very well in FER [10], they do not consider the temporal behavior of expressions. Facial expressions are very dynamic in nature and evolve with time. So it is more appropriate to use the temporal information for real time scenarios as in video based FER. In this work we focus on developing a framework for FER that extracts spatio-temporal features from video sequences.

One of the earliest frameworks for video based expression recognition was proposed by Cohen et al. [7] using Hidden Markov Models (HMM) and Support Vector Machines (SVM) classifier. Cohn et al. [25] used active appearance model (AAM) based similarity normalized shape (SPTS) and canonical appearance (CAPP) features with SVM as the classifier. A popular algorithm with good recognition performance was proposed by Zhao et al. [11] using volume local binary patterns (VLBP) and LBP from three orthogonal planes (LBP-TOP/3D LBP, a simplified version of VLBP) to model the dynamic textures found in video sequences. Jain et al. [27] proposed a method for temporal modelling of shapes using latent-dynamic conditional random fields (LDCRF).

Liu et al. proposed a method for video based human emotion recognition using partial least squares (PLS) regression on Grassmannian manifold [32]. Similar to our approach, all the frames in a video sequence are represented as a linear subspace. Shan et al. recently proposed a framework for dynamic FER using expressionlet features [33]. Each video sequence is modeled as a spatio-temporal manifold (STM) and each STM is statistically modelled on a universal manifold model (UMM).

Although the algorithms presented so far achieve a good recognition performance, there are still some challenges in the current approaches that we try to address in this work. Within class variation for some expressions is very high and at the same time inter class

variations is minimal. This leads to poor discrimination between visually similar looking expressions such as fear, sad and angry. The nonlinear structure of the features representing the video sequence and subsequently, the underlying geometry of the feature space is not used for choosing the classifier with appropriate metrics. To overcome these problems we propose to use subspace representations and Grassmann manifold based learning techniques.

1.2 Motivation for using subspace and Grassmann manifold based algorithms

Subspace structures are used in many applications of computer vision and video processing especially for face recognition. In one of the very early instances it was proved empirically that different variations in facial images can be modelled by a low dimensional subspace under certain physical constraints [18, 20]. Another very popular use of subspace representation is in the use of Eigen faces for face recognition [19, 29].

The main reason for the popularity of subspace based modelling of data especially in face recognition is that it provides a way to encapsulate the physical variations (such as different viewpoints of the subject or images with varying illumination) to a good degree of approximation in a single representation. Another advantage of using subspace based representation of a set of images characterized by a common property is that, it is computationally very efficient to just store a low dimensional subspace for an entire set of images or a video sequence and capturing the within class variations in a single subspace while removing the redundancy in the data.

Although subspace based representations have been in use for a while now, only recently has there been a surge in understanding and using the underlying geometry of the subspace structures. If we use the traditional Euclidean based distance or similarity metrics

for classification, it would not be effective as the space of subspaces has non-trivial geometrical properties. So it is very important to characterize the geometrical property of these subspaces while devising recognition algorithms. Several works have shown that linear subspaces can be formally defined as points on Grassmann manifolds and subsequently developed frameworks for subspace based learning. Hamm et al. [1] presented a robust methodology for subspace based learning on Grassmann manifolds and proposed Grassmann kernels which can be used in conjunction with discriminant analysis techniques such as kernel FDA or linear classifiers such as k-Nearest Neighbors (k-NN) and SVM. Various subspace distance and similarity metrics have been defined on manifolds using the concept of principal angles and canonical correlation analysis.

1.3 Thesis contribution

In this thesis we develop a framework for facial expression recognition from video sequences. Our method is based on subspace based representations for data involving facial expression video sequences using local binary patterns and performing the classification using Grassmann manifold based learning techniques. We consider six expressions namely, Angry (AN), Disgust (Di), Fear (Fe), Happy (Ha), Sadness (Sa) and Surprise (Su) for classification. We compare the performance of our approach with several other algorithms and show that overall our method outperforms these current state of the art methods for expression recognition from video sequences. Our specific contributions are as follows:

1) We present a framework for facial expression recognition from video sequences using Grassmann manifold based subspace learning techniques. We develop a

model to represent the video sequence in a lower dimensional expression subspace and also as a linear dynamical system using ARMA models.

2) We successfully show good expression recognition performance from video sequences using Grassmann kernel based classifiers such as kernel-SVM, kernel Fisher Discriminant Analysis (KFDA) and achieve an accuracy of 97.41 % on the extended Cohn-Kanade (CK+) database [25].
3) We show that our method has good recognition accuracy (92.88%) even when only 3 frames are used. Such an approach helps in significantly reducing the computation complexity; the runtime reduces by 83.75% on an Intel Core i3, dual core platform.

1.4 Thesis organization

The thesis is presented in five chapters and organized as follows:

In chapter 2, we discuss feature extraction using local binary patterns (LBP) from facial images, present the examples of subspace structures found in video sequences of time varying patterns and geometrical interpretation of these subspaces as points on Grassmann manifolds. We also describe kernel functions on Grassmann manifolds, also known as Grassmann kernels and how these kernel functions can be used in kernel Fisher discriminant analysis of data points on Grassmann manifolds (Grassmann discriminant analysis, GDA) and demonstrate their use in linear classifiers such as k-NN, SVM.

In chapter 3, we present the proposed framework for facial expression recognition from video sequences using subspace based representations and Grassmann manifold based learning techniques.

In chapter 4, the results obtained in various experiments using various Grassmann manifold based learning algorithms such as subspace modelling, ARMA linear dynamical system models, Grassmann discriminant analysis are presented.

In chapter 5, we conclude our work and present possible future directions of this research.

CHAPTER 2

BACKGROUND

In this chapter we present a brief overview of the subspace based representations and Grassmann manifold based learning techniques. The theory and mathematical concepts discussed in this chapter have been referred from well-known papers [1, 2, 4, 7, 10, 12, 15 and 31]. In section 2.1 we provide a background of local binary pattern (LBP) and discuss its applications. In sections 2.2, 2.3, we present an overview of subspace/ARMA models, methodology for computing expression subspace and estimating the parameters of ARMA model. In section 2.4 we introduce Grassmann manifolds and subsequently in section 2.5 we discuss various distance metrics that are used for similarity measurement on Grassmann manifolds. Sections 2.6 and 2.7 present an overview of kernel functions and discriminant analysis techniques. In sections 2.8 and 2.9 we discuss Grassmann kernels and Grassmann discriminant analysis (GDA).

2.1 Local Binary Patterns as facial feature descriptors

Local Binary Pattern (LBP) is a very popular technique for extracting texture and shape features from an image that has been primarily used for texture analysis [13]. It has also gained popularity in extracting facial features due its capability to encapsulate the texture information and hence has been used successfully for facial recognition tasks [12]. The major advantage of LBP is that it is invariant to illumination changes of the image and thus negates the effect of illumination variation. Also, LBP is computationally very simple, probably one of the simplest feature extraction methods in computer vision.

The original LBP operator was proposed by Ojala et al. in 1996. Since then different extensions to LBP have been proposed with better texture characterization. We use uniform pattern based LBP operator as our main facial feature extractor and refer the readers to the works presented in [11, 12] for more information on other variants of LBP.

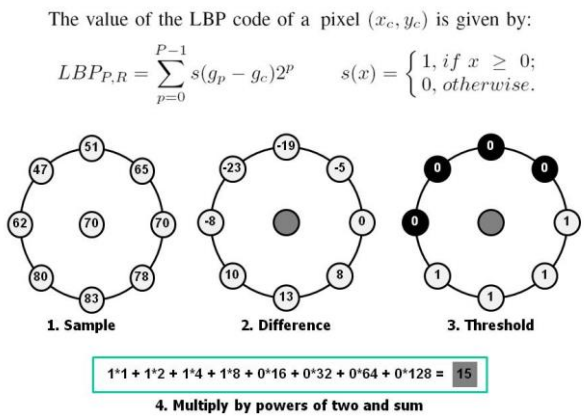


Figure 2.1: Illustration showing LBP computation. Picture is taken from scholarpedia introductory article on local binary patterns [8].

The uniform pattern based LBP operator basically labels each pixel of the image using a circular neighborhood (like a circular image kernel) as summarized in Figure 2.1. This circular neighborhood or kernel is parameterized by a) P, the number of sampling points on the circle and b) R, the radius of the circle. Generally for P sampling points, each pixel can take one of 2^P possible LBP codes. Ojala et al. observed that for facial images, 90% of the LBP codes are uniform patterns i.e. there are at most 2 transitions in a LBP pattern. For uniform pattern LBP with P=8, the total possible number of LBP codes is 59 (58 for uniform and 1 code for the rest of the non-uniform patterns). Uniform pattern LBP operator is denoted by $LBP_{P,R}^{u2}$.

Once LBP codes of an image are computed, the 59 bin histogram (for $LBP_{P,R}^{u2}$) forms the final feature vector which characterizes the facial image. However it has been observed in various psycho-visual experiments that to extract features characterizing facial expressions like various action units (action units are fundamental action individual muscles or group of muscles), it is more useful to divide the image into smaller blocks and compute LBP histogram of each block and concatenate all the histograms in a particular order to create the global feature vector. The main advantage of dividing the face into smaller blocks is that the global feature vector represents the local texture and the global shape of the face image. The process is summarized in Figure 2.2.

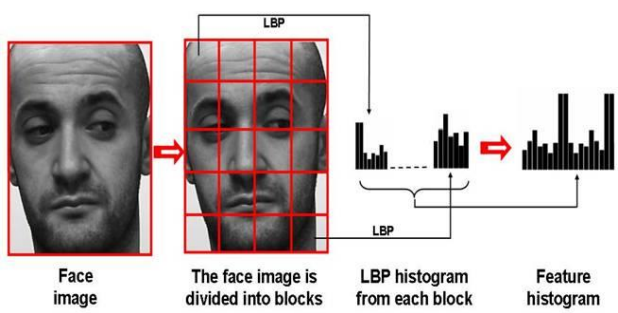


Figure 2.2: Illustration showing LBP histogram generation. Picture is taken from scholarpedia introductory article on local binary patterns [8].

2.2 Subspace structure in images and video sequences

In the field of facial recognition, historically, the images were modelled in high dimensional vector space (also called image space) where an image with n pixels was considered as a data point in $\mathbb{R}^n$, a very high dimensional vector space. For example in case of hand written digit recognition (consider MNIST database, where each image is of size $N \times N$ pixels), the raw image pixels are arranged in row scan order to form the feature

vector in $\mathbb{R}^{N^2}$ vector space. Using a similar approach in facial recognition poses a challenge since variations in lighting, expression, viewpoint result in lower recognition accuracy. Also it has been proved that for objects that exhibit approximately Lambertian reflectance properties such as faces, the set of all reflectance functions obtained under a wide variety of lighting conditions can be approximated with a low-dimensional linear subspace.

In the field of facial recognition, the data points lie on a very high dimensional space and hence pose a lot of challenges in organizing and modelling the data. Techniques that are robust against variations in pose, illumination and expression are based on dimensionality reduction of these data points. Note that the dimensionality of these images can vary from 2 to 9 depending upon the application.

Use of subspace representations of facial data improves the accuracy because, subspaces minimize the within class variations and maximize the inter class variation. It also results in significant reduction in complexity for performing learning and classification tasks. In section 2.5 we describe how the geometry of the subspace structures is used to choose appropriate distance metrics for classification.

In various approaches that involve learning of subspace structures, Hamm et al. [1] highlight a common problem. Feature extraction is done in non-Euclidean space using Euclidean distance metrics for similarity measurement. Hamm et al. [1] propose an alternative framework (which they refer to as Grassmann discriminant analysis), where the feature extraction and classification is performed in Grassmann manifolds. The advantage of such an approach is that appropriate distance metrics can be used for measuring the similarity between data points. The relationship between these low dimensional subspaces

and the underlying geometry of the data with Grassmann manifolds is very well studied [31, 3, and 4]. So Grassmann manifold based learning algorithms can be efficiently used on data containing linear subspace structures, especially to tackle variations in illumination, expression, facial features alignment and pose.

Next we describe different subspace based modelling techniques for facial expression recognition from video sequences.

2.3 Computing subspaces

We present two techniques for modelling the data in low dimensional subspaces. 1) Computing the expression subspaces for each video sequence of a subject exhibiting a particular emotion under a constant pose, 2) Modelling the spatio-temporal dynamics of facial expression video sequences using linear dynamical system models such as Auto-Regressive Moving Average (ARMA).

2.3.1 Computing expression subspaces from video sequences

Consider any database which contains video sequences of different emotions expressed by various subjects. As a pre-processing step, each image is first converted into grayscale format and then is cropped and aligned using facial feature points such as eyebrows, mouth or landmark points so that only the facial region is used for further analysis. For every frame in the sequence, a LBP histogram (computed from user specified LBP parameters) is generated and used as a feature descriptor encapsulating the properties of facial texture as explained in earlier sections. Let us assume that this feature vector is of length D and that there are N number of frames in a particular sequence. The number of frames can vary

from subject to subject, but there needs to be at least 2 frames (neutral and peak frame) per sequence.

The ensemble of all image data is organized into a data matrix X of size $D \times N$, where each column vector is the LBP feature vector of the corresponding frame. Using Singular Value Decomposition (SVD), the $m-$ dimensional orthonormal basis vectors (obtained from m largest singular values) of X matrix are computed. The corresponding orthonormal basis vector matrix U is of dimension $D \times m$ which represents the $m-$ dimensional subspace of the video sequence.

2.3.2 Estimating the parameters of ARMA model representation of a video sequence

Time varying patterns or textures such as video sequences of human expressions can be modelled as a linear dynamical system (LDS). LDS models are very useful to embed both the spatial and temporal information from the data and are hence used for a variety of tasks such as changing actions/gestures etc. There are several LDS based models such as Auto-Regressive (AR) model, Auto-Regressive Moving Average (ARMA) model etc. We consider ARMA model for our work. For more details about ARMA model readers can refer to [3].

The generalized continuous time domain ARMA model can be represented mathematically as:

$$Y(t+1) = A_t X(t) + V(t) \tag{2.1}$$

$$Y(t) = C_t X(t) + W(t) \tag{2.2}$$

where t is the time instant, $V(t)$ and $W(t)$ are noise components modelled as zero mean white Gaussian noise. As we are not synthesizing the data, we ignore the noise components in our model. The time instant can be used as an indexing parameter for the frames in the

sequence starting from the neutral frame to the peak frame. $Y(t)$ is the $D \times 1$ observation vector, $X(t)$ is the hidden state vector, A_t is the transition matrix, C_t is measurement matrix. The closed form procedure for building an ARMA model can be described as below:

For a given sequence the feature vectors extracted from each frame $f(1), f(2), \ldots \ , f(t)$ can be organized as a data matrix Z with size D x N where N is number of frames, D is the length of the feature vector and t is the time instant. First we compute the closest rank m approximation to Z using SVD, where m is the number of dimensions in the subspace. Let $[U, S, V]$ be the SVD of the data matrix Z. Then the model parameters A, C in closed form for each sequence is given by:

$C = U, A = SVD_1V'(VD_2V')^{-1}S^{-1}$, where $D_1 = [0\ 0\ ; I_{N-1}\ 0]\ and\ D_2 = [I_{N-1}\ 0; 0\ 0]$, and I_N is an identity matrix of size $N \times N$.

The observability matrix is given by:

$$O = [C; CA; CA^2; \ldots; CA^{m-1}] \tag{2.3}$$

2.4 Stiefel and Grassmann manifolds

In this section we briefly introduce the Stiefel and Grassmann manifolds followed by the required tools for enabling recognition algorithms.

2.4.1 Stiefel manifold

Let Y be a $D \times k$ matrix whose elements are real numbers and Y is an orthonormal matrix i.e. $Y'Y = I_k$. The Stiefel manifold is defined as follows:

Stiefel manifold $S(k, D)$ is the set of $k-$frames in $\mathbb{R}^D$, where a $k-$frame is a set of k orthonormal vectors in $\mathbb{R}^D$. Each element of $S(k, D)$ provides an orthonormal basis for a $k-$dimensional subspace of $\mathbb{R}^D$.

2.4.2 Grassmann manifold

Let I be a feature vector extracted from an image (which belongs to a set of k images) with a resolution of D, i.e. $I \in \mathbb{R}^D$ where I is represented as a column vector of size D. A Grassmann manifold denoted by $G_{k,D}$ or $G(k, D)$ is defined as the set of all $k-$ dimensional linear subspaces of $\mathbb{R}^D$. A linear subspace formed from a set (k) of images in $\mathbb{R}^D$ represented by orthonormal column matrix Y of dimension D by k can be identified as a point on Grassmann manifold. Two elements $Y_1, Y_2 \in G_{k,D}$ are equivalent iff $span(Y_1) = span(Y_2)$.

Since we encode the sets of images characterized by a common property as points on the Grassmann manifold, in the next section we will explore the distance metrics on Grassmann manifolds which can be used from similarity measurement of different subspaces.

2.5 Subspace distances metrics for similarity measurement on Grassmann manifolds

Consider a Grassmann manifold $G(k, D)$ representing $k-$dimensional linear subspace of $\mathbb{R}^D$ obtained by SVD of data matrix X of size D by m (m distinct points or set of images organized as a data matrix X where each column is a data vector). As mentioned earlier, by representing the set of images as linear subspaces, a group of images or a video sequence

represented as points on Grassmann manifolds and hence distance measures specific to Grassmann manifold can be used.

It is very popular to use principal angles to measure the similarity (closeness) or variation between the subspaces. This method of analysis is also known as canonical correlation analysis. Using principal angles, various distance measures have been defined. Before discussing the subspace distance metrics, let us understand the mathematical representation of points on Grassmann manifold.

Consider a set of images characterized by a common property such as a video sequence with m image frames, where the subject's expression changes from neutral to peak emotion. Let the size of feature vector representing each frame be D and hence each image can be seen as a data point in $\mathbb{R}^D$. The collection of feature vectors corresponding to m images is organized as a $D \times m$ matrix X. The $m-$ dimensional linear subspace spanned by these m images is obtained by computing the k -dimensional orthonormal basis vectors (using Eigen value decomposition or SVD of X). Let Y be the matrix that contains these orthonormal vectors as its columns.

Let Y_i and Y_j be two such orthonormal matrices of size $D \times m$ representing 2 video sequences or 2 sets of images. The principal angles between these two subspaces span(Y_i) and span(Y_j) can be computed from the SVD of the product of the two matrices i.e. the covariance matrix $Y_i'Y_j$. From the SVD of $Y_i'Y_j$ principal angles $\theta = [\theta_1, \theta_2, \dots, \theta_m]$ can be computed as:

$$Y_i'Y_j = USV',\ where\ U = [u_i \dots u_m], V = [v_i \dots v_m], S = diag(\cos\theta_i \ \dots \cos\theta_m), \quad (2.1)$$

$$0 \le \theta_1 \le \cdots \le \theta_m \le \frac{\pi}{2},$$

$$1 \geq \cos\theta_i \geq \cdots \geq \cos\theta_m \geq 0$$

As it can be observed the 1st principal angle θ_1 is the smallest and the corresponding cosine i.e. the 1st canonical correlation is the largest. The Riemannian distance between the two subspaces $span(Y_i)\ and\ span(Y_j)$, i.e. geodesic or arc length distance between two points on the Grassmann manifold is given by

$$d(Y_i, Y_j) = \sum_{\forall\, k} \theta_k^2 = \|\theta\|_2 \tag{2.4}$$

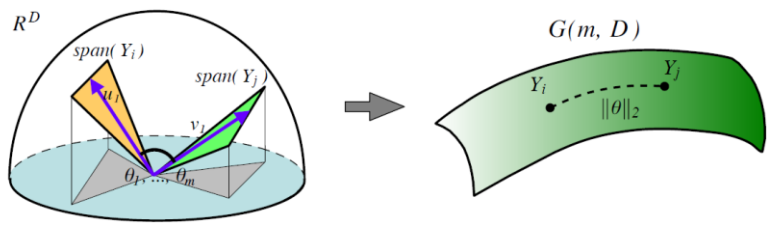


Figure 2.3: Subspaces spanned by Y_i, Y_j in input space $\mathbb{R}^D$ can be visualized as points on Grassmann manifold $G(m, D)$. The picture is taken from [1].

For further details about various other valid distance metrics and the relevant proofs, readers can refer to [1]. Different types of distances on Grassmann manifold are summarized in Table 2.1. The projection kernel which satisfies the properties of Grassmann kernel and discussed in later sections is derived using projection distance metric.

Distance metric	Expression in terms of principal angles
Projection	$d_{proj} = \left(\sum_{i=1}^{m} (\sin^2 \theta_i)\right)^{\frac{1}{2}}$
Binet-Cauchy	$d_{BC} = \left(1 - \prod_{i=1}^{m} \cos^2 \theta_i\right)^{\frac{1}{2}}$
Procrustes 1	$d_{P1} = 2\left(\sum_{i=1}^{m} (\sin^2(\theta_i/2))\right)^{\frac{1}{2}}$
Procrustes 2	$d_{P2} = 2\sin(\frac{\theta_m}{2})$
Max correlation	$\sqrt{2}\sin\theta_1$
Min correlation	$\sin(\theta_m)$

Table 2.1: Subspace distance metrics in terms of principal angles

For Grassmann manifolds or for any geometrical structure in general, using only distance based metrics for similarity measurement between the data points in that data space limits the amount of statistical analysis that can be performed with the data. An alternative approach can be adopted using positive definite kernel functions on the manifold. Using the kernel functions we can transform the existing nonlinear space of the data to higher dimension linear spaces such as Hilbert space. More about kernel functions, kernels on manifolds which are called as Grassmann kernels and the associated subspace based learning techniques are explored and discussed in the next few sections.

2.6 Kernel functions

Kernel functions are generally used whenever the original space on which the data points lie is complex or if the decision boundary for separation of the data points belonging to different classes is nonlinear. As an example, if the original data points lie on a low dimensional vector space but is complex in structure, we can do a nonlinear transformation of the data into a higher dimensional vector space. Such a transformation make it easy for the learning algorithms to classify or automatically assign the data points to a particular cluster in the higher dimensional *feature space*.

For instance, in Figure 2.4 in the original data space $[x_1, x_2]$, the decision boundary is nonlinear and hence it is not possible to classify the data using a linear classifier. However if we use the feature map ϕ, we can transform the data into a 3 dimensional feature space where the decision boundary is now a 2 dimensional hyperplane and the classification in this transformed space is a much easier task.

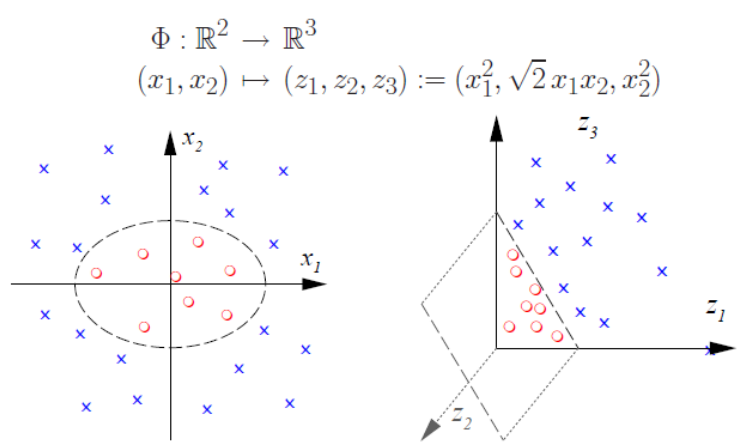


Figure 2.4: An example to illustrate feature maps. Image captured from [15]

For most of the linear classifiers such as SVM, Linear Discriminant Analysis (LDA), k-nearest neighbors (k-NN) etc., the important aspect is to find the decision boundary either in input space or some higher dimensional feature space. However it is computationally very inefficient to transform each point in the input space to the corresponding data point in a higher dimensional feature space. Since a hyperplane can be defined using inner products of the data points in the new feature space we can just compute the inner products in the new feature space. These inner products in the feature space can be mathematically expressed as a function of the data points in the original input space and is called a kernel function. A kernel function can also be thought as a nonlinear similarity measure of data in the original space that corresponds to a linear similarity measure in feature space. Every kernel function needs to satisfy certain properties that are briefly discussed in the subsequent sections.

2.6.1 Symmetric positive definite kernel property: A function $k : X \times X \rightarrow \mathbb{R}$ is a positive definite kernel iff it is symmetric, i.e. $k(x_i, x_j) = k(x_j, x_i)$ for any 2 data points $x_i, x_j \in X$ and positive definite, i.e.

$$\sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j k(x_i, x_j) \geq 0 \tag{2.5}$$

For any $n > 0$, any n objects $x_i, \dots, x_n \in X$, and any real numbers $c_i, \dots, c_n \in \mathbb{R}$.

2.6.2 Mercer's theorem:

Mercer's theorem is a fundamental theorem which every kernel function has to satisfy.

A kernel function $K(x, y)$ is a symmetric function that can be expressed as inner product in the feature space i.e.

$$K(x,y) = \langle\, \phi(x), \phi(y)\, \rangle \tag{2.6}$$

for some ϕ if $K(x,y)$ is positive semi definite, i.e.

$$\int K(x,y)g(x)g(y)dxdy \geq 0 \quad \forall g \tag{2.7}$$

Or, equivalently the kernel matrix,

$\begin{bmatrix} K(x_1,x_1) & K(x_1,x_2) & \cdots \\ K(x_2,x_1) & \ddots & \\ \vdots & & \end{bmatrix}$, is positive semi-definite for any collection $\{x_1, x_2, \dots x_n\}$.

Mercer theorem is a simple extension of the above discussed kernel properties to a compact subspace $\mathcal{X}$ in $\mathbb{R}^D$. A kernel function can be expressed in terms of Eigen values λ_i and Eigen functions ψ_i as:

$$k(x,y) = \sum_{i=1}^{\infty} \lambda_i \psi_i(x)\psi_i(y) \tag{2.8}$$

The above condition is true if the kernel function $k: \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive definite symmetric and continuous kernel function of the integral operator $T_k : \mathcal{L}_2(\mathcal{X}) \to \mathcal{L}_2(\mathcal{X})$, $(T_k f) = \int_{\mathcal{X}} k(x,y)f(y)dy$, and it satisfies the following property:

$$\int_{\mathcal{X}^2} k(x,y)f(x)f(y)dxdy \geq 0 \quad \forall f \in \mathcal{L}(\mathcal{X}) \tag{2.9}$$

As a corollary, if a kernel function satisfies Mercer's theorem, then there is always a feature map $\phi: \mathcal{X} \to \mathcal{H}$, such that k becomes an inner product in the feature space. Here $\mathcal{X}$ is the input space and $\mathcal{H}$ is the feature space (Hilbert space).

2.7 Kernel discriminant analysis

Linear discriminant analysis (LDA) is a technique for computing a low dimensional subspace of the input space while preserving the discriminant features of multi-class data.

It can be also visualized as a projection where the class separation is maximized and in-class variation minimized. LDA and different extensions of LDA like non parametric discriminant analysis (NDA) find a subspace that maximizes the ratio of between class scatter S_B and within class scatter S_W after the data is projected onto the subspace.

Consider a data set of vectors $\{x_1, \dots, x_N\}$ each with a dimension of D, with a corresponding labels vector $\{y_1, \dots, y_N\}$.The labels take values ranging from 1 to C corresponding to each class. The assumption is that each class denoted by c has N_c number of samples.

$\mu_c = \frac{1}{N_c} \sum_{\{i|y_i = c\}} x_i$, is then mean of the class c.

$\mu = \frac{1}{N} \sum_i x_i$, is the global mean of all the data vectors.

With such a distribution of data vectors and labels, the between-class and within class scatter matrices of linear discriminant analysis (or Fisher discriminant analysis, FDA) can be mathematically expressed as follows:

$$S_B = \frac{1}{N} \sum_{c=1}^{C} N_c(\mu_c - \mu)(\mu_c - \mu)' \tag{2.10}$$

$$S_W = \frac{1}{N} \sum_{c=1}^{C} \sum_{\{i|y_i = c\}} (x_i - \mu_c)(x_i - \mu_c)' \tag{2.11}$$

Generally the objective function for multi-class data is given by multi-class Rayleigh quotient

$$J(W) = tr[(W'S_W W)^{-1}(W'S_B W)] \tag{2.12}$$

W is called as the projection matrix of size D × d, d is the subspace dimension to which we are projecting the data. The optimal W can be found by Eigen value decomposition

of $S_W^{-1}S_B$. The maximum number dimension that we can project to is limited to C-1 as the rank of the $S_W^{-1}S_B$ matrix is C -1. So, effectively we have achieved the dimensionality reduction by projecting the data onto the subspace spanned by column vectors of W.

2.7.1 Kernel Fisher Discriminant Analysis

Kernels discussed earlier can be used with LDA to perform classification on nonlinear data. A nonlinear extension to LDA is called as Kernel Fisher Discriminant Analysis or also known as Nonlinear Discriminant Analysis.

$\phi : \mathcal{F} \rightarrow \mathcal{H}$ is a feature map from the input space to a higher dimensional Hilbert space. Let K be the kernel matrix representing this matrix of size $N \times N$ i.e. there are N data points in the input space. The projection matrix for the new feature space can be written as a function of K i.e. $W = K\alpha$ and the transformed objective function described earlier (i.e. Rayleigh quotient) is given by:

$$J(W) = J(K\alpha) = J(\alpha) = \frac{\alpha' K' S_B K\alpha}{\alpha' K' S_w K\alpha} \tag{2.13}$$

$$= \frac{\alpha' K\left(V - \frac{1}{N}1_N 1_N'\right)K\alpha}{\alpha'(K(I_N - V)K + \sigma^2 I_N)\alpha}$$

In the above equation, 1_N is a uniform unit vector of length N, V is a block diagonal matrix whose Cth element is the uniform matrix $\frac{1}{N_C}1_{N_C}1_{N_C}'$ .Similar to the procedure of computing the optimal W for LDA from the eigen value decomposition of $S_W^{-1}S_B$, the optimal value of α is computed from the eigen vectors of $K_W^{-1}K_B$,where K_W and K_B are given by:

$$K_B = K\left(V - \frac{1}{N}1_N 1_N'\right)K \tag{2.14}$$

$$K_W = (K(I_N - V)K + \sigma^2 I_N) \tag{2.15}$$

2.8 Grassmann Kernels

We discussed about the subspace distances in Grassmann manifold for similarity measurement in earlier sections. But it can be very useful to use kernel functions for similarity measurement as we can effectively use linear classifiers. Even for nonlinear structures such as Grassmann manifold, valid kernel functions have been defined, using which we can transform the nonlinear manifold structure to linear Hilbert space.

Projection and Binet-Cauchy distances discussed in the earlier sections satisfy the condition of positive definite kernels. It has been shown that these subspace based distance metrics discussed earlier can be extended to define positive definite kernel functions on the manifold and subsequently transform the manifold structure to Hilbert space by using the RKHS (Reproducing kernel Hilbert space) theory [15]. Also it has been successfully shown that Binet-Cauchy kernel and projection kernel can be used as a similarity measure for various applications such as facial recognition [1].

A Grassmann kernel has to satisfy the following properties:

Let $k : \mathbb{R}^{D \times m} \times \mathbb{R}^{D \times m} \rightarrow \mathbb{R}$ be a real valued symmetric function $k(Y_1, Y_2)$ i.e. $k(Y_1, Y_2) = k(Y_2, Y_1)$. The function k is a Grassmann kernel if

1) k is positive definite i.e.

   If $\sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j k(x_i, x_j) \geq 0, \quad \forall\, x_i | x_j \in R^{D \times m} \;\&\; \forall\, a_i |\, a_j \in \mathbb{R}$

2) k is invariant to different representations such as:

   $k(Y_1, Y_2) = k(Y_1 R_1, Y_2 R_2), \qquad \forall\, R_1 R_2 \in O(m)$

Projection and Binet-Cauchy kernels have been proved to satisfy the above mentioned fundamental kernel properties.

2.8.1 Projection Kernel

The Projection kernel is defined as the Frobenius norm

$$k_{proj}(Y_1, Y_2) = \|Y_1' Y_2\|_F^2 \tag{2.16}$$

2.8.2 Binet-Cauchy Kernel

The Binet-Cauchy kernel is defined as:

$$k_{BC}(Y_1, Y_2) = (\det Y_1' Y_2)^2 = (\det(Y_1' Y_2 Y_2' Y_1)) \tag{2.17}$$

2.9 Grassmann Discriminant Analysis (Projection kernel + kernel FDA)

The Grassmann discriminant analysis (GDA) was proposed by Hamm et.al [1]. They have shown that the traditional kernel based techniques can be extended to Grassmann manifolds by using positive definite Grassmann kernels such as Projection kernel, Binet-Cauchy kernel. Grassmann Discriminant analysis is basically Kernel Fisher Discriminant Analysis using one of the Grassmann kernels as the kernel function.

GDA in conjunction with linear classifiers such as k-NN, SVM for subspace based data models have been used successfully in various applications such as illumination/pose invariant face recognition, activity and gesture recognition etc. [1, 2, and 3]. In this thesis we use GDA technique for learning the expression subspaces and ARMA linear dynamical system models from video sequences. Out approach achieves very good expression recognition accuracy as will be shown in chapter 4.

CHAPTER 3

PROPOSED FRAMEWORK

In this chapter we present the proposed framework for facial expression recognition using subspace representations and Grassmann manifold based learning algorithms for classification. The six classes of expressions are Angry (AN), Disgust (Di), Fear (Fe), Happy (Ha), Sadness (Sa) and Surprise (Su). We describe our framework from LBP feature extraction of each frame to subspace/ARMA modelling followed by GDA (along with k-NN / SVM classifier) based classification. We also specify the choice of parameters used for different algorithms along with the criterion for selecting them. The steps of this framework are summarized in Figure 3.1 and described in detail in section 3.1.

Proposed framework for facial expression recognition (FER) from video sequences

---

Input: M frames of a video sequence

Step 1: Image preprocessing, alignment and normalization.

Step 2: Frame level feature extraction (LBP) and subspace computation (SVD).

Step 3: Kernel trick (Grassmann kernel) and feature space transformation (LDA).

Step 4: Classification (1-NN, SVM, kernel-SVM).

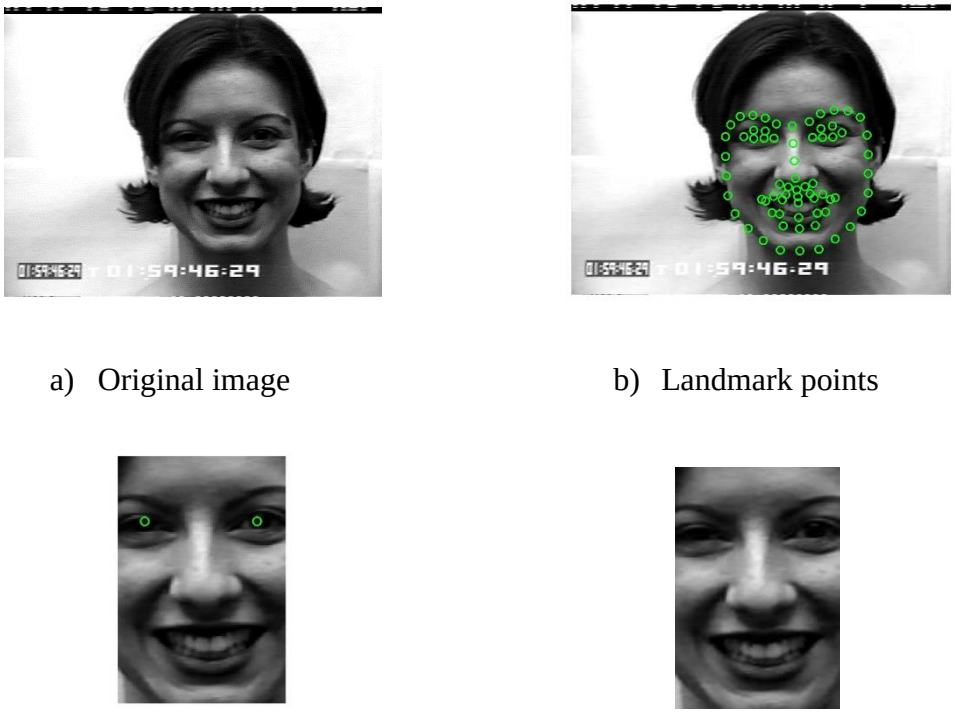
Output: The video sequence is classified into one of six expressions.

Figure 3.1: Summary of the proposed framework

3.1 Framework description

Input: Images of a video sequence where M is the number of images/frames in a sequence.

Step 1 - Image pre-processing and alignment: Since the images can be of any resolution, the images in the sequences have to be preprocessed before feature extraction. This can be done using several ways. We use the ground truth information about the images such as landmark points provided in the database. We use these landmark points as a bounding box and then normalize the image with respect to fixed eye distance and finally resize the normalized image to a fixed resolution of 150×102. The process has been summarized in Figure 3.2. Although we have used landmark points for image registration, we have also tested our framework on facial images which were detected and aligned automatically.



a) Original image

b) Landmark points

c) Cropped image

d) Normalized image wrt eye distance

Figure 3.2: Image preprocessing: image cropping, alignment and normalization

Step 2 - Feature extraction and subspace computation: We use local binary patterns (LBP) as facial feature descriptors. As discussed in Chapter 2 [10], each image is first divided into smaller blocks and LBP histogram is computed for each block. We performed a sweep analysis to arrive at a block size of 5×6 for best recognition performance. The concatenated histogram feature vectors of each frame in a video sequence are organized as the row vectors of global feature matrix X of size $D \times M$, where D is the size of uniform local binary pattern based histogram feature vector which is extracted for each frame of the video sequence and M is number of frames in the video sequence.

The expression subspace spanned by images of the video sequence is then computed by computing the k-dimensional orthonormal basis vectors by singular value decomposition (SVD) of the feature matrix Y. The dimension of this matrix is of size $D \times k$ and effectively this matrix projects the subspace spanned by this video sequence as a point on the Grassmann manifold.

We also model the video sequence as a linear-dynamical system such as an ARMA model [3] by following the procedure mentioned in Chapter 2. Let O be the observability matrix of size $kD \times k$ representing the ARMA model. Then a video sequence can be represented by either an expression subspace (matrix Y) or as an ARMA model (matrix O). The next few steps are same irrespective of these two approaches. For simplicity, we represent the video sequence by the orthonormal matrix Y which can be seen as a point on Grassmann manifold. Y is of size $D \times k$, where D is the feature vector length of each frame and k is subspace dimension.

Step 3 - Kernel trick and feature space transformation: The subspace or the column space spanned by the observability matrix (from ARMA model) lies on the Grassmann manifold which is a nonlinear space. To transform this feature space into a higher dimensional linear space we use kernel trick for optimal performance of the linear classifiers (SVM, k-NN). We use the Projection kernel (Grassmann kernel) discussed in Chapter 2 for this transformation. The kernel trick can be used with a variety of statistical techniques and classifiers. After using the kernel trick, a nonlinear classifier such as kernel-SVM can be directly used for classifying the data points.

Another approach is to use discriminant analysis techniques like kernel LDA followed by a linear classifier such as k-NN and SVM. Kernel LDA is a combination of kernel trick and LDA algorithm, where LDA is a supervised dimensionality reduction technique which projects the data onto a lower dimension Euclidean space such that within class variation is minimized and inter class variation is maximized. The kernel LDA algorithm when used with Grassmann kernels like projection kernel is called as Grassmann discriminant analysis (GDA). So from here on we refer the kernel LDA in our framework as GDA. We implemented our framework using both the approaches i.e. kernel SVM and GDA.

Step 4 - Classification: After the training and test sample feature vector matrices F_{train}, F_{test} are computed (described in Section 3.2), any linear classifier like k-nearest neighbor (k-NN) or Support vector machines (SVM) can be used. We use 1-NN classifier (Euclidean distance) and SVM kernel with parameters: polynomial kernel, degree-1, cost-0.035 and gamma-0.2. We also use pre-computed kernel with SVM i.e. projection kernels computed on training and test samples i.e. K_{train}, K_{test}.

3.2 Training and testing

First the kernel matrix needs to be computed from the training samples. Assume training set contains N_{train} samples, then the training kernel matrix K_{train} is computed using projection kernel between each pair of matrices $\{Y_i\}$ in the training set i.e.

$[K_{train}]_{i,j} = k_{proj}(Y_i, Y_j) = \|Y_i' Y_j\|_F \quad \forall\, Y_i, Y_j \; in\; the\; training\; set.$

From equation 2.11, the optimal value of α is computed using $N_c - 1$ largest eigen values and corresponding eigen vectors of $K_w^{-1} K_B$, where N_c is number of classes. The local optima α can be represented as a matrix i.e. $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_{C-1}\}$, where each column of matrix α is a eigen vector of $K_w^{-1} K_B$. So size of matrix α is $N_{train} \times (N_c - 1)$.

For Grassmann discriminant analysis, α and training kernel matrix K_{train} are used such that each training sample (Y_i matrix, which is a point on Grassmann manifold) can be projected on a $N_c - 1$ dimensional subspace spanned by α. So effectively each sample is now represented as a vector of length $N_c - 1$, all the training samples can now be represented by a new matrix F_{train} of size $N_{train} \times N_c - 1$ using the relation $F_{train} = K_{train} * \alpha$.

For testing, we use the same set of optimal eigen vectors α obtained from training set. Assume test set contains N_{test} samples. Each test sample is projected on a $N_c - 1$ dimensional vector space. All the test samples are represented by the matrix F_{test} of size $N_{test} \times N_c - 1$ using the relation $F_{test} = K_{test} * \alpha$.

The testing set kernel matrix K_{test} is computed (using projection kernel) between each matrix $\{Y_i\}$ in the test set with each matrix $\{Y_j\}$ in training set i.e.

$[K_{test}]_{i,j} = k_{proj}(Y_i, Y_j) = \|Y_i' Y_j\|_F \quad \forall\, Y_i, Y_j \; in\; the\; test\; set.$

The parameters that we use in various stages of the framework are summarized in Table 3.1.

Stage	Algorithm	Parameter
Image pre-processing		Crop the images using the landmark points. Normalize all images wrt eye distance to a resolution of 150×102
Facial features	LBP	Block size = 5×6, 59 bin uniform patterns, radius R = 2, sampling points P = 8
Subspace	SVD	Subspace dimension = 2
Discriminant	GDA	Kernel = Projection
Classifiers	k-NN SVM	1-nearest neighbor with Euclidean distance metric SVM → polynomial kernel, with g = 0.2, c = 0.035, d = 1

Table 3.1: Parameters used in different stages of the framework

3.3 Experimental evaluation

We consider six methods based on the choice of classifier and the steps in feature transformation stage depend upon the feature space in which the data is modelled.

1) Three methods using expression subspace for temporal modelling and using different classifiers such as kernel LDA + 1-NN, kernel SVM, kernel LDA + SVM.
2) Three methods using ARMA model for temporal modelling and using classifiers kernel LDA + 1NN, kernel SVM, kernel LDA + SVM.

We evaluate our framework primarily in terms of classification performance demonstrated using metrics like recognition accuracy and confusion matrix. The evaluation results are included in the next chapter.

CHAPTER 4

EXPERIMENTAL RESULTS

In this chapter, we describe the experimental framework followed by the performance results. We also present the complexity results.

4.1 Dataset

We have performed the experiments on Cohn-Kanade (CK+) facial expression database [25] which is quite popular for facial expression recognition. The dataset includes 593 sequences from 123 subjects with varying number of frames (6 to 60) per sequence. The video sequence contains images from neutral (first frame) to peak expression of the subject (last frame). All the images are of frontal pose. Each image is of resolution 640x490 (8 bit grayscale) or 640x480 (24 bit RGB) pixels. Out of 593 sequences only 309 sequences are validated by FACS coders, so we used only 309 sequences for training and testing of our algorithm.

The 309 sequences are from 106 subjects with 6 classes of expression i.e. Angry (AN), Disgust (Di), Fear (Fe), Happy (Ha), Sadness (Sa) and Surprise (Su). We randomly partition the dataset into 10 equal training (roughly 279 sequences) and testing sets (roughly 30 sequences) sampled uniformly across all the classes. We use 10 fold cross validation protocol, where we divide the entire dataset (i.e. set of all Y_i matrices) into 10 sets uniformly across all the classes. For each iteration of testing the algorithm, the training is performed using 9 sets and validation is done on 1 set, the process is repeated 10 times. The recognition accuracy is then measured as the average of accuracies in each iteration. Before evaluating the recognition accuracy for every run, the random number generator in MATLAB is reset, so that the sequences that are used for testing are randomized. For every

trial, the same set of sequences are chosen for testing and training so that the results reported for various algorithms are consistent. In the rest of the chapter we report the recognition accuracy as a ratio of total number of sequences classified correctly with respect to total number of sequences available. The distribution of various classes is presented in Table 4.1.

Expression class	Number of sequences
Angry (AN)	45
Disgust (Di)	59
Fear (Fe)	25
Happy (Ha)	69
Sadness (Sa)	28
Surprise (Su)	83
Total	309

Table 4.1: Distribution of samples in CK+ database [25]

4.2 Performance Results

4.2.1 Results on original video sequences

The following algorithms were implemented and tested for facial expression recognition (FER) in terms of recognition accuracy on CK+ database [25]:

Method 1.1: Subspace modelling from entire video sequence + Grassmann discriminant analysis (GDA with Projection kernel) + 1-NN.

Method 1.2: Subspace modelling from entire video sequence + Grassmann discriminant analysis (GDA with Projection kernel) + SVM.

Method 1.3: Subspace modelling from entire video sequence + Kernel-SVM.

Method 2.1: ARMA model from entire video sequence + Grassmann discriminant analysis (GDA with Projection kernel) + 1-NN.

Method 2.2: ARMA model from entire video sequence + Grassmann discriminant analysis (GDA with Projection kernel) + SVM.

Method 2.3: ARMA model from entire video sequence + Kernel-SVM.

The confusion matrices of all the methods are presented in Tables 4.2-4.7.

	Angry	Disgust	Fear	Happy	Sad	Surprise	Class Accuracy
Angry	45	0	0	0	0	0	100 %
Disgust	0	58	0	0	1	0	98.31 %
Fear	0	0	21	1	2	1	84 %
Happy	0	0	0	69	0	0	100 %
Sad	2	0	0	0	25	1	89.29 %
Surprise	0	0	0	1	0	82	98.80 %
Total Accuracy	97.09 %						

Table 4.2: Confusion matrix for FER using Method 1.1

	Angry	Disgust	Fear	Happy	Sad	Surprise	Class Accuracy
Angry	45	0	0	0	0	0	100 %
Disgust	0	58	0	0	1	0	98.31 %
Fear	0	0	21	1	2	1	84 %
Happy	0	0	0	69	0	0	100 %
Sad	2	0	0	0	26	0	92.86 %
Surprise	0	0	0	1	0	82	98.80 %
Total Accuracy	97.41 %						

Table 4.3: Confusion matrix for FER using Method 1.2

	Angry	Disgust	Fear	Happy	Sad	Surprise	Class Accuracy
Angry	45	0	0	0	0	0	100 %
Disgust	0	58	0	0	1	0	98.31 %
Fear	0	0	17	4	2	2	68 %
Happy	0	0	0	69	0	0	100 %
Sad	4	0	1	0	21	2	75 %
Surprise	0	0	0	1	0	82	98.80 %
Total Accuracy	94.50 %						

Table 4.4: Confusion matrix for FER using Method 1.3

	Angry	Disgust	Fear	Happy	Sad	Surprise	Class Accuracy
Angry	45	0	0	0	0	0	100 %
Disgust	0	58	0	0	1	0	98.31 %
Fear	0	0	21	1	2	1	84 %
Happy	0	0	0	69	0	0	100 %
Sad	2	0	0	0	25	1	89.29 %
Surprise	0	0	0	1	0	82	98.80 %
Total Accuracy	97.09 %						

Table 4.5: Confusion matrix for FER using Method 2.1

	Angry	Disgust	Fear	Happy	Sad	Surprise	Class Accuracy
Angry	45	0	0	0	0	0	100 %
Disgust	0	58	0	0	1	0	98.31 %
Fear	0	0	21	1	2	1	84 %
Happy	0	0	0	69	0	0	100 %
Sad	2	0	0	0	25	1	89.29 %
Surprise	0	0	0	1	0	82	98.80 %
Total Accuracy	97.09 %						

Table 4.6: Confusion matrix for FER using Method 2.2

	Angry	Disgust	Fear	Happy	Sad	Surprise	Class Accuracy
Angry	45	0	0	0	0	0	100 %
Disgust	0	58	0	0	1	0	98.31 %
Fear	0	0	18	4	2	1	72 %
Happy	0	0	0	69	0	0	100 %
Sad	4	0	1	0	21	2	75 %
Surprise	0	0	0	1	0	82	98.80 %
Total Accuracy	94.82 %						

Table 4.7: Confusion matrix for FER using Method 2.3

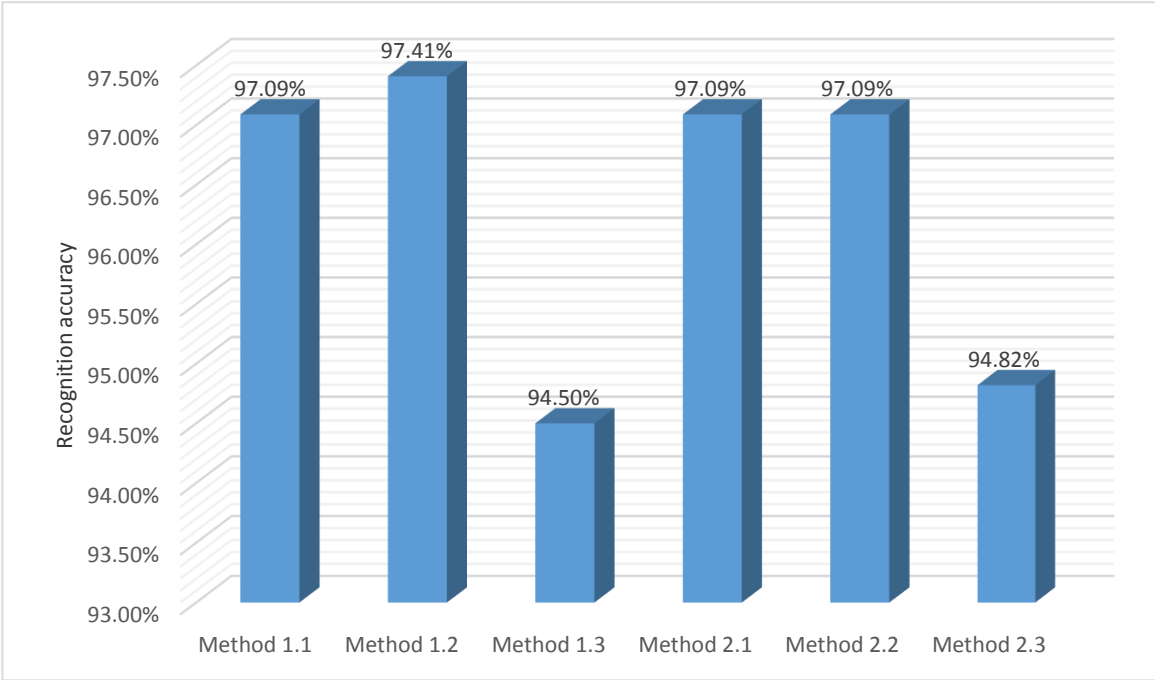


Figure 4.1: Recognition performance of different algorithms for original set.

The comparison of different methods in terms of recognition accuracy is presented in Figure 4.1. Methods 1.1, 1.1, 2.1 and 2.2 give the best results in terms of recognition performance with an accuracy of 97.09 %, 97.41%, 97.09 % and 97.09 %, respectively. A maximum recognition accuracy of 97.41% was achieved when Method 1.2 (SVM classifier) was used. However even with a simple 1-NN classifier, Method 1.1 achieves a recognition accuracy of 97.09 %. We also implemented the static technique for FER using LBP patterns of peak frame in each video sequence [10]. The static technique achieves an accuracy of 57.6% with a 1-NN classifier and increases to 90.29 % when SVM classifier is used. However in our approach there is not much difference in the recognition performance between the nearest neighbor classifier and SVM, which clearly shows that the features obtained by subspace/ARMA model of the video sequence are very good.

Another trend that is observed across all the approaches is that the recognition accuracy is 100 % for Angry and Happy emotions and is the least for Fear and Sad emotions. This matches the trend in most of the current state of the art approaches. A comparison of all the methods in terms of overall FER accuracy and individual class recognition performance is presented in Table 4.8.

	Method 1.1	Method 1.2	Method 1.3	Method 2.1	Method 2.2	Method 2.3
Angry	100 %	100 %	100 %	100 %	100 %	100 %
Disgust	98.31 %	98.31 %	98.31 %	98.31 %	98.31 %	98.31 %
Fear	84 %	84 %	68 %	84 %	84 %	72 %
Happy	100 %	100 %	100 %	100 %	100 %	100 %
Sad	89.29 %	92.86 %	75 %	89.29 %	89.29 %	75 %
Surprise	98.80 %	98.80 %	98.80 %	98.80 %	98.80 %	98.80 %
Avg. Accuracy	97.09 %	97.46 %	94.50 %	97.09 %	97.09 %	94.82 %

Table 4.8: Overall FER performance comparison of various methods on original set along with individual class accuracies.

4.2.2 Results on different video sequence organizations

In a video sequence, there is not much change in the facial expression between adjacent frames and so fewer frames could be used to reduce the computational complexity. We show that even with fewer frames, the recognition accuracy is decent.

To test the recognition performance for different organizations, we have restructured the video sequences of CK+ database into 4 categories. These are 1) Original set, which contains the images in the same order as provided in the CK+ database 2) Extended set, in which the original sequence is extended such that, the new sequence contains all the images from neutral to peak emotion followed by peak to neutral emotions, 3) Apex set, in which the first frame (neutral) and all the frames after the middle frame are

used and 4) 3-frame set, in which only 3 frames (neutral, middle and peak) are used as shown in Figure 4.3. We observe that, when the lesser number of frames are processed in a video sequence, there is a dip in the recognition performance by about 3-6%. However there is also reduction in complexity as fewer frames are processed with reasonable performance. Overall the performance of our algorithm is good for these different organizations of the video sequences. The recognition performance of Method 1.1 for various dataset organizations is presented in Figure 4.2. Recognition accuracy is highest for original set with 97.09 % followed by 96.76% for the extended set and 94.17 % for the apex set. The performance for 3-frame set drops, but still the accuracy is about 92.88 %, which is decent.

It can also be noted that as the subspace model inherently tracks the emotion of the subject in a video sequence, the direction of the sequence has no impact over the recognition performance i.e. if the sequence begins with peak emotion and changes to neutral emotion or vice-versa there is no change in the recognition performance.

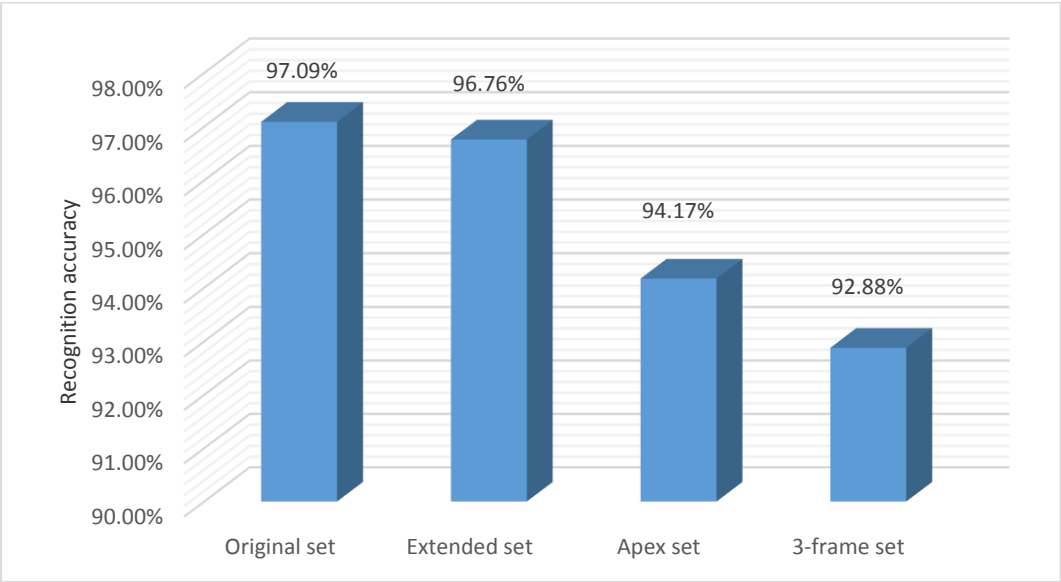


Figure 4.2: Recognition performance of method 1.1 for different dataset organizations.

1) Original set: All the images in a video sequence are as is in the CK+ database. Images start from neutral to peak and contains all the images between peak frames. Number of frames, N = 13.

2) Extended set: All the images in a video sequence are arranged such that a video sequence which originally ended in the peak frame (peak emotion) is now extended so that sequence contains all the frames from neutral to onset to peak and again back to neutral frame in reverse order. N = 25

3) Apex set: The video sequence is reorganized such that it starts with the neutral frame followed by only apex frames, i.e. middle frame to peak frame. N = 8

4) 3-frame set: 3 frames are used for the entire video sequence, i.e. the first, middle and neutral frame. N=3

Figure 4.3: Different organizations of video sequences used for testing the performance

4.3 Performance with respect to other state-of-the-art approaches

Different algorithms have been proposed for facial expression recognition using video sequences. One of the earliest and better performing algorithms was proposed by Zhao et al. [11] using VLBP (an extension to LBP in temporal domain) and LBP-TOP feature extractors. Although the performance is very good for this approach, it requires accurate alignment of images in a video sequence. Also the complexity of this algorithm is very high.

Jain et al. proposed a method for temporal modelling of shapes using latent-dynamic conditional random fields (LDCRF) [27]. They use uniform LBP operator for feature extraction for each frame. So this approach is similar to our approach in frame level feature extraction but the temporal modelling is based on LDCRF.

Liu et al. proposed a method for video based human emotion recognition using partial least squares (PLS) regression on Grassmann manifold [32]. The frame level features are extracted using action unit aware deep networks (AUDN). Similar to our approach, all the frames in a video sequence are represented as a linear subspace and Grassmann kernels are used for discriminant analysis. Partial least squares (PLS) is used for the classification of the expressions. They achieve a recognition accuracy of 32.07 % on Emotion Recognition in the Wild Challenge (EmotiW 2013) dataset [34]. As they evaluate the recognition performance on a different dataset, we do not compare the performance of our approach with their method.

Shan et al. recently proposed a framework for dynamic FER using expressionlet features [33]. Each video sequence is modeled as a spatio-temporal manifold (STM) and

each STM is statistically modelled on a universal manifold model (UMM). Discriminant analysis followed by SVM is used for classification.

Most of the facial expression recognition algorithms extract either appearance based features (LBP, HAAR, SIFT, HOG, Gabor filters) or shape based features such as landmark points, active appearance models, FACS based action units etc. The recognition accuracy of these methods is presented in Table 4.9. Note that our approach performs better than all these algorithms.

Group	Algorithm	Number of sequences	Dynamic	Protocol	Recognition accuracy (%)
[11]	LBP-TOP+VLBP	374	Yes	10-fold	96.26
[27]	LDCRF, PCA +SVM	309	Yes	4-fold	95.79
[28]	PHOG	309	Yes	10-fold	95.30
[33]	STM-ExpLet	309	Yes	10-fold	94.19
Ours	LBP+subspace model + SVM	309	Yes	10-fold	97.46
Ours	LBP + ARMA model + 1-NN	309	Yes	10-fold	97.09

Table 4.9: Recognition performance comparison with other algorithms

4.4 Complexity analysis

We perform experiments to study the execution times of different stages in our framework to process a group of 18 frames. All the measurements were done on Intel Core i3 (dual core, 2.13 GHz) processor using MATLAB profiler. Method 1.2 is chosen to demonstrate the complexity. It is observed that LBP feature extraction takes majority of the computation time (97 %), the subspace and kernel computations require about 3 % of the total computation time.

The complexity of our methods is presented in Table 4.10. It can be observed that feature extraction followed by subspace/ARMA model and feature space transformation (kernel computations and/or LDA) majorly contribute to the execution time. When compared to subspace modelling, ARMA model computations require 15 % more time.

The complexity of the best competing method for FER from video sequences using LBP-TOP features [11] in comparison with our methods is presented in Table 4.11.

	Method 1.1	Method 1.2	Method 1.3	Method 2.1	Method 2.2	Method 2.3
LBP Feature extraction (entire sequence of about 18 frames)	1.662 s	1.662 s	1.662 s	1.662 s	1.662 s	1.662 s
Subspace/ARMA model computation	32.1 ms	32.1 ms	32.1 ms	48.2 ms	48.2 ms	48.2 ms
Kernel computation	13.3 ms	13.3 ms	13.3 ms	19.17 ms	19.17 ms	19.17 ms
Classification	3.3 ms	1 ms	0.5 ms	5.15 ms	0.1 ms	0.4 ms
Total execution time	1.710 s	1.708 s	1.707 s	1.734 s	1.729 s	1.729 s
Recognition accuracy	97.09 %	97.41 %	94.5 %	97.09 %	97.09 %	94.82 %

Table 4.10: Execution times of different methods

LBP-TOP algorithm [11] encapsulates the temporal information in the feature extraction stage itself. Since it processes a volume of blocks at a time and then repeats across all the blocks of a frame, the complexity is very high.

The method proposed by Jain et al. [27] is similar to our approach in frame level feature extraction (LBP). However the temporal modelling is performed using latent-dynamic conditional random fields (LDCRF). As we could not get an optimized implementation of LDCRF, we did not measure the complexity of this approach. However

we think that the complexity of their approach would be similar to ours as LBP features are extracted for each frame.

	Method 1.1	Method 1.2	Competing method [11]
LBP Feature extraction	1.662 s	1.662 s	38.32 s
Subspace/ARMA computation	32.1 ms	32.1 ms	
Kernel computation	13.3 ms	13.3 ms	-
Classification	3.3 ms	1 ms	8 ms
Total execution time	1.710 s	1.708 s	38.328 s

Table 4.11: Complexity comparison with the best competing method

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this thesis we present a framework for facial expression recognition (FER) from video sequences that achieves high recognition accuracy.

First, we develop a model to represent a video sequence of facial expressions as a lower dimensional expression subspace and also as a linear dynamical system using ARMA model. We consider six expressions namely, Angry (AN), Disgust (Di), Fear (Fe), Happy (Ha), Sadness (Sa) and Surprise (Su) for classification. We use Grassmann kernels in kernel based classifiers such as kernel-SVM, kernel-FDA. Our method achieves an average recognition accuracy of 97.41% when expression subspace and kernel-FDA with SVM classifier is used.

One of the advantages of the proposed framework is that the order of the sequence is not important as the expression subspace efficiently captures temporal information. Also by using this framework the within class variance is minimized and inter class variation is maximized due to which similarly looking expressions are classified with greater accuracy. We find that frame level LBP feature extraction and kernel computations require majority of the computation time (99%). On average this takes about 1.662 seconds for extracting these features from a group of 19 frames, on a dual core Intel Core i3 machine.We also show good recognition performance when lesser number of frames (atleast 3) per sequence are used for extracting facial features. Such a method significantly reduces the computational complexity by about 83.75 % (260 milliseconds). Overall using this

framework we show good recognition performance outperforming the state of the art FER algorithms.

Some of the potential future directions of this research are as follows:

1) Long term expression modelling of video sequences contatining a combination of expressions: Such time varying actions can be modeled as a collection of time invariant linear dynamical systems. The sequence can be divided into small temporal neighborhoods (10-15 frames), and in each neighborhood, the sequence can be modeled by time invariant dynamical systems. So a long sequence can now be seen as a sequence of subspaces, which can be represented as trajectory on a Grassmann manifolds. Trajectories on Grassmann manifold can be compared using techniques like dynamic time warping, switching linear dynamical systems etc.
2) Optimizing the implementation of LBP feature extraction, Riemanninan and kernel computations: From our complexity analysis, we see that feature extraction at frame level using LBP and temporal modelling using subspace/ARMA techniques require 99 % of the total computation time. So techniques to optimize these computations or map them onto custom hardware implementations or through multi-core implementations will help in reducing the computation time and make such a system be used in real-time scenarios.

REFERENCES

[1] Hamm, Jihun, and Daniel D. Lee. "Grassmann discriminant analysis: a unifying view on subspace-based learning." In *Proceedings of the 25th international conference on Machine learning*, pp. 376-383. ACM, 2008.

[2] Turaga, Pavan, Ashok Veeraraghavan, and Rama Chellappa. "Statistical analysis on Stiefel and Grassmann manifolds with applications in computer vision." In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, pp. 1-8.*

[3] Turaga, Pavan, et al. "Statistical computations on grassmann and stiefel manifolds for image and video-based recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence (2011), pp.* 2273-2286.

[4] Jen-Mei Chang. 2008. *Classification on the Grassmannians: Theory and Applications*. Ph.D. Dissertation. Colorado State University, Fort Collins, CO, USA.

[5] Chang, Jen-Mei, et al. "Illumination Face Spaces Are Idiosyncratic." International Conference on Image Processing, Computer Vision, and Pattern Recognition, *IPCV* 2006, pp. 390-396.

[6] Liu, X., Srivastava, A., & Gallivan, K. (2003, June). Optimal linear representations of images for object recognition. *IEEE Conference on Computer Vision and Pattern Recognition, 2003,* Vol. 1, pp. I-229.

[7] Cohen, I., Sebe, N., Garg, A., Lew, M. S., & Huang, T. S. (2002). Facial expression recognition from video sequences. *IEEE International Conference on Multimedia and Expo, ICME'02,* Vol. 2, pp. 121-124.

[8] Matti Pietikäinen (2010) Local Binary Patterns. Scholarpedia, 5(3):9775.

[9] Shan, Caifeng, Shaogang Gong, and Peter W. McOwan. "Robust facial expression recognition using local binary patterns." In *IEEE International Conference on Image Processing, ICIP 2005*, vol. 2, pp. II-370.

[10] Shan, Caifeng, Shaogang Gong, and Peter W. McOwan. "Facial expression recognition based on local binary patterns: A comprehensive study." *Image and Vision Computing* 27, no. 6, 2009, pp. 803-816.

[11] Zhao, Guoying, and Matti Pietikainen. "Dynamic texture recognition using local binary patterns with an application to facial expressions." *IEEE Transactions on Pattern Analysis and Machine Intelligence (*29.6), 2007, pp. 915-928.

[12] Ahonen, Timo, Abdenour Hadid, and Matti Pietikainen. "Face description with local binary patterns: Application to face recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence,* (28.12), 2006, pp. 2037-2041.

[13] Ojala, Timo, Matti Pietikainen, and Topi Maenpaa. "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns." *IEEE Transactions on Pattern Analysis and Machine Intelligence,* (24.7), 2002, pp. 971-987.

[14] Vert, Jean-Philippe, Koji Tsuda, and Bernhard Schölkopf. "A primer on kernel methods." *Kernel Methods in Computational Biology* (2004), pp. 35-70.

[15] Schölkopf, Bernhard. "Introduction to Kernel Methods." *Analysis of Patterns Workshop, Erice, Italy*. 2005.

[16] Basri, Ronen, and David W. Jacobs. "Lambertian reflectance and linear subspaces." *IEEE Transactions on Pattern Analysis and Machine Intelligence, (*25.2), 2003, pp. 218-233.

[17] Lee, K. C., Ho, J., & Kriegman, D. (2001). Nine points of light: Acquiring subspaces for face recognition under variable lighting. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2001,* Vol. 1, pp. I-519.

[18] Hallinan, P. W. (1994, June). A low-dimensional representation of human faces for arbitrary lighting conditions. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 1994,* pp. 995-999.

[19] Epstein, R., Hallinan, P. W., & Yuille, A. L. (1995, June). 5/spl plusmn/2 eigenimages suffice: an empirical investigation of low-dimensional lighting models. IEEE *Proceedings of the Workshop on Physics-Based Modeling in Computer Vision, 1995,* p. 108.

[20] Sirovich, Lawrence, and Michael Kirby. "Low-dimensional procedure for the characterization of human faces." *JOSA A* 4.3, 1987, pp. 519-524.

[21] Belhumeur, Peter N., and David J. Kriegman. "What is the set of images of an object under all possible illumination conditions?" *International Journal of Computer Vision*, 28.3, 1998, pp. 245-260.

[22] Doretto, Gianfranco, et al. "Dynamic textures." *International Journal of Computer Vision,* 51.2, 2003, pp. 91-109.

[23] Tian, Ying-li. "Evaluation of face resolution for expression analysis." In IEEE *Conference on Computer Vision and Pattern Recognition Workshop, CVPRW'04*, pp. 82-82, 2004.

[24] Lin, D., Yan, S., & Tang, X. “Pursuing informative projection on grassmann manifold”. *IEEE Conference on Computer Vision and Pattern Recognition, 2006,* Vol. 2, pp. 1727-1734.

[25] Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., & Matthews, I. “The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression”. *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2010,* pp. 94-101.

[26] Anirudh, Rushil. *Low complexity differential geometric computations with applications to activity analysis*. Diss. Arizona State University, 2012.

[27] Jain, Suyog, Changbo Hu, and Jake K. Aggarwal. "Facial expression recognition with temporal modeling of shapes." *IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 2011*, pp. 1642-1649.

[28] Khan, Rizwan Ahmed, et al. "Human vision inspired framework for facial expressions recognition." *9th IEEE International Conference on Image Processing (ICIP), 2012,* pp. 2593-2596

[29] Turk, Matthew, and Alex Pentland. "Eigenfaces for recognition." *Journal of cognitive neuroscience* 3.1, 1991, pp. 71-86.

[30] Warner, Frank W. *Foundations of differentiable manifolds and Lie groups*. Vol. 94. Springer, 1971.

[31] Srivastava, Anuj, and Eric Klassen. "Bayesian and geometric subspace tracking." *Advances in Applied Probability*, 2004, pp. 43-56.

[32] Liu, M., Wang, R., Huang, Z., Shan, S., & Chen, X. (2013, December). Partial least squares regression on grassmannian manifold for emotion recognition. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pp. 525-530.

[33] Liu, M., Shan, S., Wang, R., & Chen, X. (2014, June). Learning expressionlets on spatio-temporal manifold for dynamic facial expression recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014,* pp. 1749-1756.

[34] Dhall, A., Goecke, R., Joshi, J., Wagner, M., & Gedeon, T. (2013, December). Emotion recognition in the wild challenge 2013. In *Proceedings of the 15th ACM on International conference on multimodal interaction* pp. 509-516.