SOME STUDIES ON THE FACIAL EXPRESSION RECOGNITON USING GRAPH SIGNAL PROCESSING

Ph.D. Thesis

HEMANT KUMAR MEENA

ID No. 2013REC9543

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING
MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY, JAIPUR

October 2018

Some studies on the facial expression recognition using graph signal processing

By

Hemant Kumar Meena

(2013REC9543)

Under the Guidance of

Prof. Kamalesh Kumar Sharma

Dept. of ECE, MNIT Jaipur

and

Prof. Shiv Dutt Joshi

Dept. of EE, IIT Delhi

Submitted

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

to the

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING
MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY, JAIPUR
October-2018

Dedicated to My Parents and Wife

Declaration

I, **Hemant Kumar Meena**, declare that this thesis titled **"Some studies on the facial expression recognition using graph signal processing"** and the work presented in it is my own. The work has been carried out under the supervision of **Prof. Kamalesh Kumar Sharma** and **Prof. Shiv Dutt Joshi**. I confirm that:

1. This work was done wholly or mainly while in candidature for a Ph.D. degree at MNIT Jaipur.

2. Where any part of this thesis has previously been submitted for a degree or any other qualification at MNIT Jaipur or any other institution, this has been clearly stated.

3. Where I have consulted the published work of others, this is clearly attributed.

4. Where I have quoted from the works of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.

5. I have acknowledged all main sources of help.

6. Where the thesis is based on work done by myself, jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Hemant Kumar Meena

Department of Electronics and Communication Engineering
MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY JAIPUR

CERTIFICATE

This is to certify that the thesis entitled "**Some studies on the facial expression recognition using graph signal processing"** submitted by **Hemant Kumar Meena** (ID. No.2013REC9543) to the Department of Electronics and Communication Engineering, Malaviya National Institute of Technology, Jaipur, for the award of the degree of **Doctor of Philosophy,** is a bonafide research work carried out by him under my supervision and guidance. The results obtained in this thesis have not been submitted to any other university or institute for the award of any other degree.

Prof. Kamalesh Kumar Sharma

Department of Electronics and Communication Engineering
Malaviya National Institute of Technology Jaipur
Jaipur 302017, Rajasthan (India)

Prof. Shiv Dutt Joshi

Department of Electrical Engineering,
Indian Institute of Technology Delhi,
New Delhi-110016 (India)

Acknowledgements

First of all, I would like to thank GOD, the Almighty, for his showers of blessing throughout my research work. Then I would like to express my deep and sincere gratitude to my supervisors, **Prof. Kamalesh Kumar Sharma** and **Prof. Shiv Dutt Joshi**, for accepting to be my advisors and guides. They trusted me and gave me the freedom to choose my own research topics. They have always been the pillars of support. Their advice-always encouraging, their ideas-always useful and the environment they provided-exceptionally friendly and professional.

I express my gratitude to DREC members, Dr. Mohammed Salim, Dr. R C Soni, and Dr. Satyasai Jagannath Nanda for their constructive feedback in critically examining and reviewing my work. Further, I would like to acknowledge Dr. K R Ramakrishnan, Indian Institute of Science, Banglaore for inspiring me to explore the GSP concepts. I also extend my deep sense of gratitude to Prof. Udaykumar R Yaragatti, Director, MNIT, Jaipur for strengthening the research environment of the institute by providing all necessary facilities to the research scholar.

Further, I would also like to thank all colleagues and Ph.D. scholars from Electrical Engineering Department for their friendship and support time to time.

My thesis would not have been possible without extensive support from my wife *Nisha*. I am very much thankful for her love, understanding and support during my research work. I am grateful to my parents who continues to have full faith and love irrespective of the outcomes in my life. I am also thankful to my sister *Manvi (Naanu)* for her patience and blessings during this time.

Last, I would like to thank my gratitude to my friends and many familiar persons whose cooperation made the Ph.D. a remarkable journey for me.

Hemant Kumar Meena

Abstract

Facial expression recognition (FER) has always popular as the fundamental basis of the social interaction. In the recent years, FER is getting special attention because of its interdisciplinary nature from the behavioral science, neurology and artificial intelligence. Its application ranges from human computer interaction to the design of social intelligent systems.

In FER, the main challenge is to effectively encode the facial expression in the form of the feature vector so that the accurate classification of the expression can be performed with minimum computational complexity. We have addressed this challenge by the use of the emerging approach named graph signal processing (GSP). GSP has established itself as a powerful tool for a wide variety of applications particularly dealing with the analysis of the multidimensional and multivariate signals.

In the thesis, we have represented the facial expression regions in the form of the graph structure to encode their relationship with the help of the weighted edges. The key idea behind this representation is that the GSP based approach helps in exploiting the higher level dependence in those facial expression regions. Thus, this additional information of the intrinsic connectivity has been used to enrich the analysis of the multi dimensional signals and multivariate signals lying in the facial expression regions.

In the first part of the thesis, we focus upon the existing schemes of FER using GSP to design composite schemes. We start by using existing feature vectors lying on the nodes of a graph. The weighted edges between these nodes is formulated to capture their relationship. Then, the graph signal is represented as a sum of harmonic modes by using the eigendecomposition of the Laplacian (a second derivative operator). In the form of these newly represented signals, not only

the dimension of the feature vector is reduced but also the accuracy of FER is improved.

In the second part of the thesis, we design the independent schemes using GSP. Rather than using the derived information of facial expression regions, we directly work upon these regions assuming that the information of facial expression lies in the distribution of the pixel intensities. We construct a graph to represent the facial regions as a graph signal by defining the weight between the parts of the regions using their intensities. Then, to extract the information from that graph signal, we have designed algorithms using the concepts of spectral graph wavelet transform and graph Fourier transform. In order to reduce the dimension of the resultant graph feature vectors, the spectral analysis in the graph frequency domain has been carried out to select the main frequency components, as performed in principal component analysis. Finally, we evaluate the different graph structures to find the optimum graph structure for the better FER.

Key words: Facial expression recognition, feature vector, graph, graph signal processing, spectral graph wavelet transform, graph Fourier transform, graph structure.

Contents

List of Figures

List of Tables

Abbreviations and Symbols

The symbols used in the text have been defined at appropriate places, however for easy reference, the list of principle symbols is given below.

Abbreviation/Symbol	**Description**
CT	Curvelet Transform
DWT	Discrete Wavelet Transform
FRFT	Fractional Fourier Transform
FER	Facial Expression Recognition
GFT	Graph Fourier Transform
GSP	Graph Signal Processing
HOG	Histogram of Oriented Gradients
kNN	k- Nearest Neighbors
SGWT	Spectral Graph Wavelet Transform
$\mathbb{R}$	Set of real numbers
$\mathbb{R}^+$	Positive set of real numbers
I	Image
G	An arbitrary graph
V	Set of vertices in a graph
E	Set of vertices in a graph
$e_{\mathrm{i,j}}$	Edge connecting vertex i and vertex j
$\mathbf{W}$	Weight matrix of a graph
W_{ij}	Weight of the edge between node i and node j
N	Total number of nodes in a graph
$\mathbf{D}$	Degree matrix
$\mathbf{x}$	Graph signal vector

Continued on next page

Continued from previous page

Abbreviation/Symbol	**Description**
$\mathbf{X}$	Graph signal matrix
$\mathbf{L}$	Graph Laplacian matrix
$\mathbf{U}$	Graph Fourier matrix
λ_{l}	$l-th$ eigenvalue of Laplacian matrix
u_{l}	$l-th$ eigenvector of Laplacian matrix
$\hat{\mathbf{x}}$	GFT vector of $\mathbf{x}$
$\hat{\mathbf{X}}$	GFT matrix of $\mathbf{X}$
$\|\mathbf{x}\|$	L^2 norm of $\mathbf{x}$
$\|\mathbf{x}\|$	L^1 norm of $\mathbf{x}$
$\mathbf{x}_{\mathrm{i}}$	Graph signal vector of $i-th$ node(face)
$\mathbf{x}_{\mathrm{i}}{}^{\mathrm{M}}$	Graph signal vector of the mouth region of $i-th$ face
$\mathbf{x}_{\mathrm{i}}{}^{\mathrm{E}}$	Graph signal vector of the eyes region of $i-th$ face
$\mathbf{y}_{\mathrm{i}}$	Reduced graph signal vector of $i-th$ node (face)
δ_{i}	Kronecker delta localized at node i
$\psi_{\mathrm{t,n}}$	SGWT of graph signal at scale t at node n

Chapter 1

Introduction

1.1 Introduction

In our communication, the non-verbal cues provide more important information than the verbal statements. Among the non verbal cues, the facial expression is one of the most significant components. Seeing the utility of the facial expressions recognition (FER), it has been interpreted not only in informal social networks (Facebook, Whatsapp) but also in advanced applications of patient monitoring, surveillance, neuroscience [1] , lie detection etc. The approach to the FER may be, in general, sub-divided into the following steps: the face registration, representation, dimensionality reduction and recognition [2].

In the literature, various approaches have been proposed based on the feature extraction of the facial expression using Gabor filtering [3], Principal Component Analysis(PCA) [4], Independent Component Analysis(ICA) [5], Linear Discriminant Analysis(LDA) [6], LBP(Local Binary patterns) [7] and Discrete Wavelet Transform (DWT) [8]. Moreover, the dimension reduction is carried out by the PCA by taking the eigenvectors in the order of decreasing variance whereas the LDA maximizes the separation between the feature vectors in the subspace. Apart from the dimension reduction, some approaches focus on finding the distinct patterns like the LBP used for capturing the textures of the facial regions [9], HOG (Histogram of the Orientation Gradient) [10] for the computation of the gradient at cellular level where all the gradient of the cells are concatenated to form the feature vector etc. In the last step,the classifier using methods such as Sup-

port Vector Machines (SVM) [11], k-Nearest Neighbors (kNN) and Deep Learning such as Convolutional Neural Networks [12], [13] categorizes the expression into different labels.

Recently, the GSP has been increasingly used by the researchers for the analysis of many multi-dimensional signals like brain signals [14], [15], fMRI [16] etc. The GSP, explained in [17], deals with the signals represented on the structure of a graph. A graph is built incorporating the neighborhood information of the high-dimensional signals. The graph structure thus encodes the inherent relationship of the signal lying between its components located on the vertices of the graph. The main step for the GSP method are to define a weight between the vertices and generalize the classical signal processing techniques such as filtering, multiscale transforms by using a different notion of the graph frequency. Such processing of the signals represented on the graphs proves to be efficiently extracting the information from the high dimensional signals. Similar to the GSP, the concept of Locality Preserving Projection (LPP) was introduced in [18] and it was used for face recognition [19].

1.1.1 Motivation for the present work

Beauty with GSP based approach is the flexibility in interpreting the underlying signal as a graph. There are various ways in which a signal can be represented as a graph. And, this thesis has interpreted signal as a graph in a variety of ways. In any context employing GSP, the scheme would decompose the problem into two stages. In first stage, the local properties (features) of the signal would be extracted such as correlation, edges, curvature, spectral features, intensity etc. and each local region could correspond to one node of the graph. In stage two, GSP would exploit the higher level dependence between the local features, which is modeled as the adjacency matrix. It is pertinent to note that this holistic GSP approach can be applied in any context. In this thesis, our context has been the FER.

Facial expression can be understood as a visual pattern recognition problem in which three-dimensional object is identified on the basis of its two-dimensional image. Using the two dimensional image, the features are extracted and then clas-

sified by the machine learning algorithms. From the viewpoint of signal processing, the main challenge lies in the selection of a feature vector with optimum dimension and maximum accuracy. In this regard, the objectives of the proposed work are as follows:

i) Existing schemes of the FER including DWT, HOG, Curvelet transform (CT) and Fractional Fourier transform (FRFT) are quite effective in correctly classifying the expressions but their feature dimensions become significantly higher which results into their restricted use. There is clearly a need to develop algorithms for reducing the feature dimension without any compromise on the recognition rate.

ii) As the facial image is a high dimensional signal, the challenge is to extract information for the effective classification. The interrelationship between the components of facial image has not been satisfactorily utilised for the better FER. Although the focus of the FER schemes remain upto finding the difference in the different facial parts (say, the eyes region, the lips region etc.) individually at the local level but, at the higher (global) level, the changes in one part of the facial region (lips region) are also related to the changes in the shape of the another facial region (eyes region) during the facial expressions. Hence, the graph signal representation of the facial image seems to be the natural choice to take into account the unexplored relationship between the different facial regions.

A typical feature based approach for FER only makes use of local features. The interdependence that may exist among the local regions is generally not exploited. A typical GSP based approach is the suitable choice to capture that interrelationships. The local features would form nodes of a graph. The weights of the edges between these nodes provide the estimate of the bonding. Such weighted graphs present an extremely flexible model for approximating the data domains of a large class of problems.

Similarity relationships between "point cloud of vectors" may be represented by a graph. Graphs associate data instance with a collection of the feature vectors that hopefully encapsulate sufficient information about the data points e.g. for the object recognition, SIFT features [20] are calculated after extracting key points. In view of the above observations, we choose to investigate the existing schemes of the FER and how their feature vectors can be represented in the form of the

graph signal structure. After their representation, we designed the approaches based upon GSP concepts to extract out meaningful information from these graph signals. Our contributions in this regard are explained in the next section.

1.1.2 Contributions

The contribution of this thesis is the improvement of the FER by considering different facial feature vectors constructed using tools from the GSP framework. To be more specific, the contributions are (also refer Fig. (1.1)):

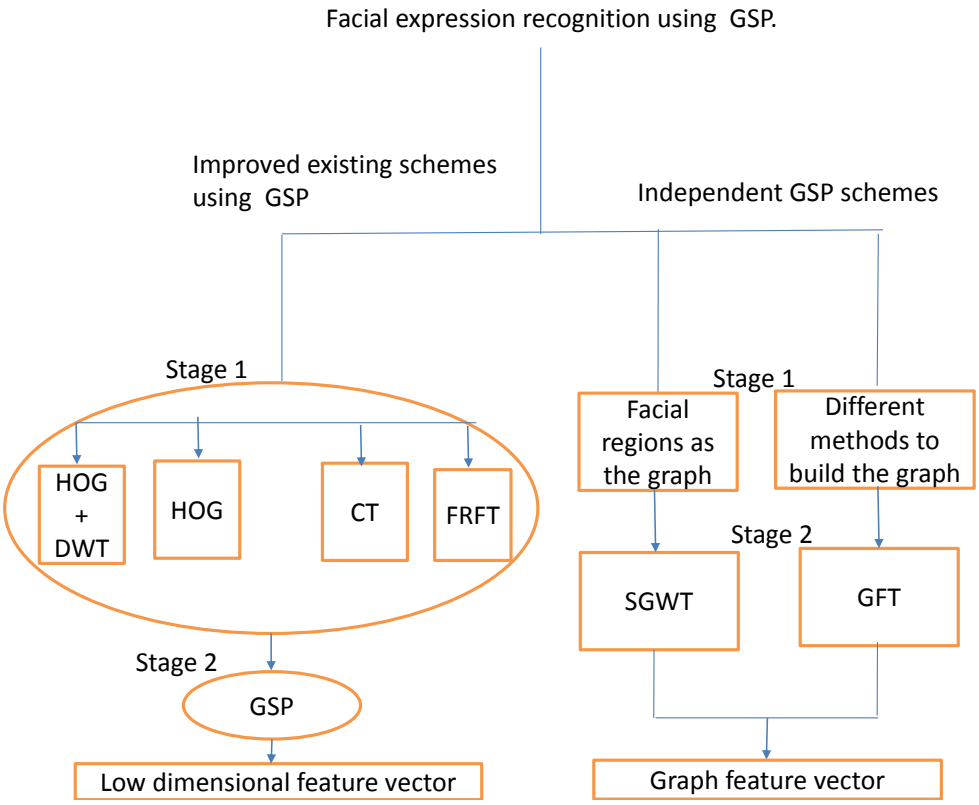

Figure 1.1: Summary of our contribution in the FER using GSP

1) Improved existing FER schemes using GSP: GSP has been used to investigate whether the existing schemes of FER may be improved. The improved schemes are as follows:

a) GSP based approach for FER using DWT and HOG features has been proposed by extracting out the relationship among the existing feature vectors. Each feature vector is represented on the vertex of a graph and the interrelationship between these feature vectors has been represented by the weights. Further, it has

been utilized to reduce the dimension of the feature vector and also resulting in a better recognition rate. In order to reduce the computational complexity, the GSP based approach for FER using HOG has been evaluated.

b) CT is preferred over DWT because of the improved directionality and better representation but it also leads to computationally expensive due to their large feature dimension. GSP approach has been used with the existing feature vectors of the CT to find the lower dimension feature vector and improve the accuracy of classifying expressions.

c) Since FRFT is a generalized family of transforms, where the conventional Fourier Transform (FT) is a special case, it is clearly expected to provide better compared to FT. GSP based approach is proposed to find out the increase in expression recognition rate and reduce the dimension of the feature vector.

ii) Independent schemes: These schemes are based on GSP only without employing any of the existing feature selection for FER methods.

a) **SGWT for FER**: A graph structure is formulated independently on the facial image itself. Then the concept of the SGWT has been used to propose a new GSP based scheme. This uses the facial image itself to find out the effective feature vector for FER.

b) **GFT for FER**: For the independent GSP schemes for FER, dimension reduction technique using GFT has been proposed. Then the key frequency components are selected in such a way to optimize the accuracy with the lower dimension feature vector.

c) **Optimum graph structure for FER**: Different type of methods to build the graph have been evaluated to find the optimum graph structure for the better FER.

1.1.3 Thesis outline

The second chapter focuses on studying the basics of the FER and provides a brief introduction to GSP. The reviewed topics include the notion of graph frequency and the definitions of GFT, SGWT.

Chapter 3 presents a new GSP approach for enhanced FER using DWT and HOG. Then, the proposed methodology of the GSP approach is described. The

facial images from Cohn - Kanade (CK+) and JAFFE databases are used to evaluate the performance of the proposed method. The performance results are compared with the existing wavelet based techniques. Further, in order to decrease the computational complexity, the former approach has been investigated without using DWT i.e. by using the GSP approach with HOG only. The advantages and the limitations of the proposed GSP based techniques are discussed.

Chapter 4 considers the problem of large dimensional feature vector in the existing schemes of CT and FRFT. To reduce the dimension of the feature vector, the proposed GSP based approaches using CT and FRFT are presented. First, the GSP based CT approach is compared with the existing CT methods on JAFFE dataset. Later, the GSP based FRFT approach is compared with the existing FRFT methods.

In the next two chapters, we propose approaches for FER exclusively based on GSP concepts only i.e. without employing any of the existing feature selection schemes. **Chapter 5** is devoted to leveraging of the spectral graph wavelet transform (SGWT) for the automatic FER. In this chapter, the algorithms using the graph signal from the facial image is presented. the experimental results are obtained on CK+ and JAFFE databases. The effect of the different filter-banks with the different weights to their channel is observed on the performance of the algorithm. Next, the experimental results are compared with the existing state-of-the-art in the literature.

Chapter 6 deals with the dimension reduction of the feature vectors using the concept of Graph Fourier Transform (GFT). Then, the proposed GFT method is described and its performance is observed on CK+ and JAFFE databases by selecting the few frequencies among all the frequencies. Furthermore, the different graph structures are evaluated to find out the best results of GSP approach for the FER. The experimental performance of the optimum graph structure is compared (using on CK+ and JAFFE databases) with the existing methods based on PCA.

The thesis concludes in **Chapter 7** where a summary of our work is presented. Finally, the future scope of our work regarding the application of the GSP in the field of FER is also mentioned.

Chapter 2

Literature Review

Facial expression recognition (FER) is required for Human-Computer interaction as the facial expression provides important basis to understand the non-verbal cues. Apart from the technology facilitation, FER has the significant applications e.g. in getting the signal from the specially privileged person with autism or with speech disorder, building the socially aware systems [21], improving the e-learning experiences by detecting the frustration of the students [22], better experience of gaming and in monitoring drowsiness of a driver [23] etc.

FER is usually confused with human emotion recognition. While in FER the facial motion and facial feature deformation are classified into abstract classes that are purely based on visual information, human emotion results from different sources and that state is revealed through different channels such as emotional voice, postures, gaze direction and facial expression. In contrast to FER, emotion recognition requires the understanding of a given situation, together with the availability of full contextual information. Overall, emotion recognition is far more complex to detect than the FER. Moreover, from the originating view-point, facial expressions are affected by many sources, as shown in Fig. 2.1.

2.1 Basics of FER

For the input as a single facial image for spatial representations, the fundamental stages include the facial registration, representation, dimensionality reduction and recognition. Face registration may be categorised into the whole face, part and

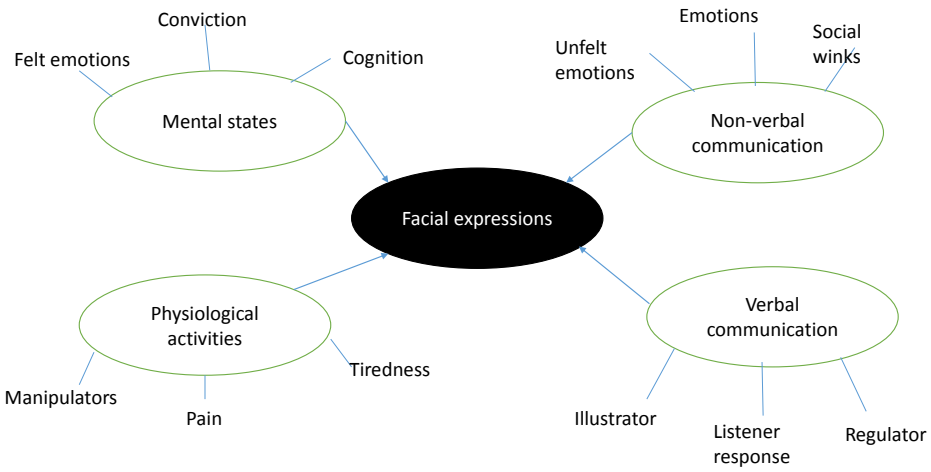

Figure 2.1: Sources of Facial expressions [24].

point registration. The whole face used for the posed images include techniques such as Active Appearance models [25], Robust FFT [26] and Lucas-Kanade [27] approaches. In parts registration, the face is processed in term of the eyes and mouth. Useful for shape representation, the point registration involves the localization of fiducial points. Representations encode the information: lower level information by Gabor representation, LBP, HOG and bag-of-words (BOW), and higher level information by non-negative matrix factorisation (NMF) or sparse coding.

Feature extraction can be done by predesigned and learned features. Predesigned features are individually framed while learned features require automatically learning from the trained data e.g. in deep learning. Predesigned features may be categorized into appearance and geometrical. In the appearance features, the intensity information is used while geometrical features use distances, deformations, curvatures etc. The advantage of the appearance features is the detection of micro-expressions (through finding characteristics like wrinkles, furrows or skin texture), where the geometrical features fail to detect.

Learned features, trained by a joint feature learning and classification, can't be classified as local or global e.g. in the case of CNNs, either higher level features comprising the whole face, or a pool of local features, may be obtained. There-

after, the dimensionality reduction stage is to reduce the dimension of the encoded feature vector. Some of the approaches are such as discrete cosine transformation (DCT), principal component analysis (PCA) [4] and linear discriminant analysis (LDA).

Finally, the recognition of the expression from the reduced feature vector is performed using the static approaches and the dynamic approaches. Static approaches work on each frame independently such as the support vector machine (SVM), k-nearest neighbor (kNN), Bayesian network classifiers (BNC), neural networks etc. Dynamic models process features extracted independently from each frame to model the evolution of the expression over time. It includes hidden markov models (HMMs) and variable-state latent condition random fields (VSL-CRF).

2.2 Feature Extraction

As the work done in thesis deals mainly with the feature extraction of the facial images, a brief discussion of the feature extraction follows.

Initially a geometric face model of 30 facial characteristic points was proposed in [28]. Here, an affine transformation was used to normalize an input image to fix the distance between irises. Thereafter, these distances were used to empirically determine the length of the vertical lines. Similarly, a point based model using the frontal and the side view of the facial image was given in [29]. In the frontal view face model, a set of facial points were used to determine the specific shapes of the mouth and the chin. The side-view of face image was used to find the curvature of the profile contour function from the facial points. Multiple feature detectors were applied for each prominent facial feature (eyebrows, eyes, mouth, nose and profile) to localize the contours of the prominent facial features and extract the model features in an input dual–view. Due to the accurate requirement of extracting the reference points, the geometric approach does not perform well for low quality and complex background images. In addition, extraction of geometrical features ignores the information of the skin texture changes, which limits it to distinguish the expression of the subtle changes.

Active Appearance Model (AAM) is another widely used facial feature extrac-

tion method [30, 31]. The facial images that were manually labeled with more than hundred points to represent the facial features in [25]. In another AAM based method, 70 images were selected as the AAM training set with the marking of 57 feature points in each image [32]. In [33], an input testing face image was changed into its corresponding neutral expression image. As a testing facial image is queried, it is represented by a feature vector using the active appearance model (AAM). Using a point distribution model (PDM), 90 facial feature points were localized to show basic six emotions [34].

2.2.1 Feature Extraction with Gabor

For the past one decade or so, Gabor wavelet-based methods preferred for the FER feature extraction since they can detect multi-scale, multi-direction texture changes and can handle the effects of illumination. This model is motivated by the biological model of simple cell receptive fields present in the cortex of cat. A Gabor filter can be formulated as follows [35]:

$$\psi_{\mathrm{u,v}} = \frac{\|k_{\mathrm{u,v}}\|^2}{\sigma^2} exp(-\frac{\|k_{\mathrm{u,v}}\|^2\|z\|^2}{2\sigma^2})[exp(izk_{\mathrm{u,v}}) - exp(-\frac{\sigma^2}{2})] \tag{2.1}$$

where u, v are the direction and scale of Gabor kernel respectively; z=(x,y) represents the spatial position of image pixel; σ is related to the width parameter of the Gaussian kernel; $k_{\mathrm{u,v}} = k_{\mathrm{v}} exp(i\phi_{\mathrm{u}})$ indicates the responses of the filter in different directions and scales, where $k_{\mathrm{v}} = \frac{k_{\mathrm{max}}}{f^v}$ and $\phi_{\mathrm{u}} = \frac{\pi u}{8}$, k_{max} is the maximum frequency and f is the spacing factor between kernels in the frequency domain. Gabor filters have good resolution both in frequency and space. Furthermore, they have directional selectivity. In 2006, Yu and Bhanu in [36] used the Gabor wavelet representation for primitive features and linear/nonlinear operators to synthesize new features . In [37], Gabor transform was combined with the hierarchical histogram for the facial features extraction. Here, the hierarchical representation of the change of texture in local regions was used to extract the intrinsic facial features. In addition, as the Gabor transform is relatively less sensitive to the change of lighting conditions, it can tolerate certain rotation and deformation of images. In this way, the 2-D Gabor transform is better than traditional representation

scheme using 1-D Gabor coefficients.

2.2.2 Feature Extraction with HOG

The Histogram of Orientation Gradients (HOG), proposed by Dalal and Triggs [10], is well suited to robustly extract features for visual object recognition. In HOG descriptor computation, the gradient of the image is find out and a predefined number of orientation intervals are used to quantize the phase. Then, by the division of the image into small regions 'cells', the histogram is constructed from the quantized orientation of each pixel. The magnitude of the gradient for the pixels act as the weight for the orientation. Thereafter cells are combined to form blocks which are the normalization units of that scheme. The purpose of normalization is to reduce the variation effect of the gradient in local areas due to illumination and object/background contrast. Finally the descriptor is created by the concatenation of the block-normalized histograms of all the cells.

2.2.3 Feature Extraction with Wavelet Transform

An approach of multi-resolution signal decomposition was presented in [38]. That has been found to analyze the information content of images. This decomposition defines an orthogonal multi-resolution representation called Discrete wavelet transform (DWT). For images, the wavelet representation differentiates several spatial orientations. This representation is used for data compression in image coding, texture discrimination and fractal analysis, to name a few.

At the first level of filtering each block image, low-low (LL), low-high (LH), high-low (HL), and high-high (HH) bands are obtained. Then, at the next level, LL band is further decomposed into four smaller images at 2-level. That 2-level 2-dimensional DWT was used to extract information from an input image. Finally, the result of the segmented 128×128 pixels region is an arrangement like that of

the Haar transform.

$$L_{\mathrm{n}}(i,j) = [L_{\mathrm{x}} * [L_{\mathrm{y}} * L_{\mathrm{n\text{-}1}}]_{\downarrow 2,1}]_{\downarrow 1,2}(i,j)$$
$$D_{\mathrm{n1}}(i,j) = [L_{\mathrm{x}} * [H_{\mathrm{y}} * L_{\mathrm{n\text{-}1}}]_{\downarrow 2,1}]_{\downarrow 1,2}(i,j)$$
$$D_{\mathrm{n2}}(i,j) = [H_{\mathrm{x}} * [L_{\mathrm{y}} * L_{\mathrm{n\text{-}1}}]_{\downarrow 2,1}]_{\downarrow 1,2}(i,j)$$
$$D_{\mathrm{n3}}(i,j) = [H_{\mathrm{x}} * [H_{\mathrm{y}} * L_{\mathrm{n\text{-}1}}]_{\downarrow 2,1}]_{\downarrow 1,2}(i,j)$$

where * denotes the convolution,$\downarrow$2,1 and $\downarrow$1,2 denotes the subsampling along the columns and rows,respectively. L_0 represents the original image. L and H indicates the low pass and the high pass filter, respectively. We obtain the $\mathrm{L_n}$ by low pass filtering and is considered as the lower resolution image at level n. To obtain the detail images $\mathrm{D_{ni}}$, the band pass filters are used in a specific direction. These images have the directional detail information at level n. The original image can be represented by a set of subimages at several levels $\{\mathrm{L_d},\mathrm{D_{n,i}}\}$ i=1,2,3 and n=1,2,...d,which is a multilevel representation of the depth d of image I[39].

It is to be noted that the 'LL' band features associated with low-frequency components are used mainly for the face recognition whereas the high frequency components including 'LH', 'HL' and 'HH' are used to recognize the face expressions. Selecting the number of levels (stages) of the wavelet decomposition further leads to the better distinguishing power if the image is not too small. Finally, the optimum number of levels of decomposition is decided based upon the experiments conducted.

2.2.4 Feature Extraction with CT

CT was proposed by Candes and Donoho in 1999 to analyze images [40]. The transform has improved the directional capability, ability to represent edges and other singularities along curves as compared to the traditional wavelet transform. After its introduction in 1999, Curvelet construction has been under revision to make it simpler to understand and use. The second generation CT introduced [41] is simpler, faster and less redundant compared to its first generation version [42]. The CT is multiscale and multidirectional. Curvelet exhibits a highly anisotropic

shape obeying parabolic scaling relationship. For implementing the CT, the first step is to compute the 2D FFT of the image. Then the 2D Fourier frequency plane is divided into parabolic wedges. Finally, an inverse FFT of each wedge is taken to find the curvelet coefficients at each scale j and angle ℓ. The division of the wedges of the Fourier frequency plane is shown in Fig.2.2. The wedges result from partitioning the Fourier plane in radial (concentric circles) and angular divisions. Concentric circles are responsible for the decomposition of the image in multiple scales (used to bandpass the image). Angular divisions correspond to different angles or orientation. Hence, the angle and scale are required to define a particular wedge. In the spatial domain, each wedge corresponds to a particular curvelet at the given scale and angle. Curvelets in spatial Cartesian grid related with a given scale and angle are shown in Fig. 2.2. The second generation Fast Digital Curvelet

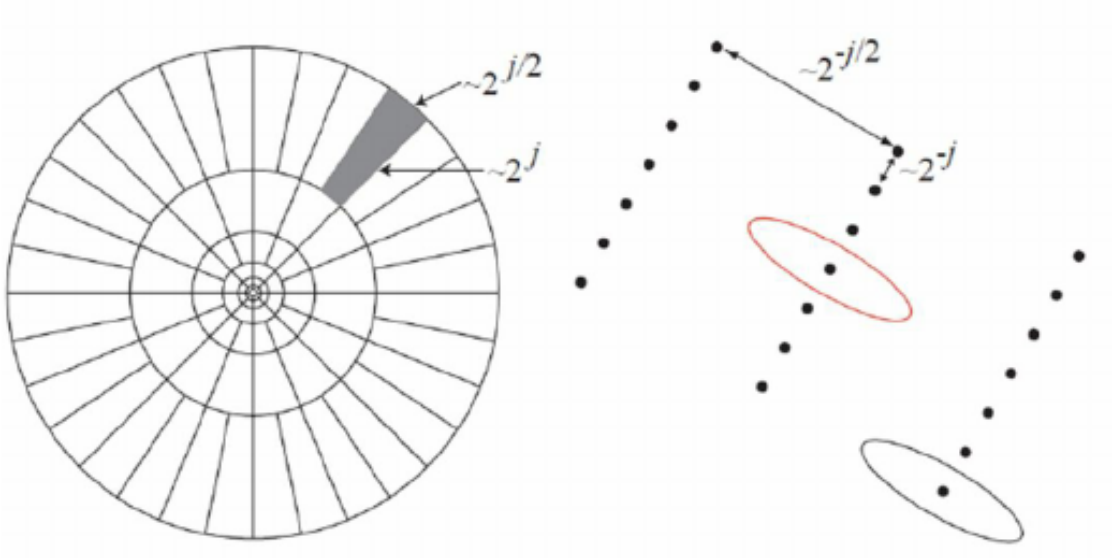

Figure 2.2: Curvelet tiling in the frequency domain and the spatial domain [41].

Transform (FDCT) is implemented in two different ways, which are as follows [41]: Curvelet via USFFT (Unequally Spaced Fast Fourier Transform) and Curvelets via Wrapping. These transforms are linear and take the input of Cartesian array $f[t_1,t_2],0 \leq t_1,t_2 <$ n. They use different spatial grids to translate into the curvelets. Both the FDCTs have a computational cost of $\mathcal{O}(n^2 \log n)$ for an (n×n) image. In case of wrapping, a rectangular grid is assumed. Due to its fast implementation, FDCT via wrapping is used [41].

The implementation of FDCT via Wrapping is shown as below[41]:

1) The 2D FFT is applied to find Fourier samples $\hat{f}[\mathrm{n}_1,\mathrm{n}_2]$,where $\frac{-n}{2} \leq \mathrm{n}_1,\mathrm{n}_2 < \frac{n}{2}$

2), The obtained Fourier samples are multiplied with the $\tilde{U}_{\mathrm{j},\ell}$ (which represents the discrete localizing window) to find the product $\tilde{U}_{\mathrm{j},\ell}\hat{\mathrm{f}}[\mathrm{n}_1,\mathrm{n}_2]$ for each scale j and angle ℓ.

3) The above computed product is wrapped around the origin to find

$$\hat{f}_{\mathrm{j},\ell}[n_1, n_2] = W(\tilde{\mathcal{U}}_{\mathrm{j},\ell}\hat{f})[n_1, n_2] \tag{2.2}$$

where the range n_1 and n_2 is now $0 \leq n_1 < L_{1,\mathrm{j}}$ and $0 \leq n_2 < L_{2,\mathrm{j}}$

4) The inverse 2D FFT of each $\hat{f}_{\mathrm{j},\ell}$ is computed to obtain the discrete coefficients.

Initially, the Fourier frequency plane of the image is converted into radial and angular wedges. A wedge represents the curvelet coefficients at particular scale and angle. Then, the data is wrapped around the origin. Finally, the inverse FFT is applied to find the discrete curvelet coefficients in the spatial domain.

2.2.5 Feature Extraction with Two-dimensional FRFT

The two-dimensional FRFT (2D-FRFT) of a signal $f(s,t)$ is defined as follows[43]:

$$F_{a_1,a_2}(u,v) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(s,t)K_{a_1,a_2}(s,t,u,v)\,ds\,dt \tag{2.3}$$

where the $K_{a_1,a_2}(s,t,u,v)$ is the 2D-FRFT kernel defined in [43] as follows:

$$K_{a_1,a_2}(s,t,u,v) = \frac{\sqrt{(1-j\cot\alpha)}\sqrt{(1-j\cot\beta)}}{2\pi} exp(\frac{j(s^2+u^2}{2\tan\alpha} - \frac{jsu}{\sin\alpha})exp(\frac{j(t^2+v^2}{2\tan\beta} - \frac{jtv}{\sin\beta}) \tag{2.4}$$

Here, $\alpha = \frac{a_1\pi}{2}$ and $\beta = \frac{a_2\pi}{2}$ denote the rotation angle in the FRFT domain and a_1,a_2 are the transform orders along row and column respectively. The separable kernel in the 2D-FRFT is represented as follows:

$$K_{a_1,a_2}(s,t,u,v) = K_{a_1}(s,u) \cdot K_{a_2}(t,v) \tag{2.5}$$

As a result, 2D-FRFT is computed by one- dimensional FRFTs, first along each column and then along each row of the result, while dealing with images.

From the earlier discussion, it is clearly evident that the typical feature size is very large leading to high computational complexity. Hence, there is need to reduce the dimensionality of the feature vectors. For that purpose generally the following methods are used:

a) PCA: PCA is performed on the images by considering each image as high dimensional observation with the gray level of each pixel as the measure.The principal component axes are the eigenvectors with the pixelwise covariance matrix of the dataset.These component axes are template images that resemble ghost like faces (eigenfaces). A low dimensional representation of the face images with minimum reconstruction error is obtained by projecting the images onto the first few principal component axes,corresponding to axes with highest eigenvalues.

b) ICA: In ICA, the orthogonal bases are not necessary to represent the data. In [44], the ICA was used to extract the facial features and it performed better than PCA.

c) LBP: The LBP operator was proposed in [45] which encodes the micro-level information of edges, spots and other local features in an image. It computes the edge response values in different directions and use these to encode the image texture. After computing all the local directional pattern (LDP) code for each pixel, the input image is represented by an LDP histogram which represents a descriptor of that image.

2.2.6 Requirement of the GSP

As it is clear from the discussion so far, the FER problem can be considered at two stages. At the first stage, the local features are extracted and at the second stage, the interrelationships between the local features are extracted. From our experiences, GSP based approach reduces the dimensionality of the feature set. Before further proceeding, the basics of the GSP and the related concepts have been explained with its mathematical representation.

2.3 Graph Signal Processing (GSP)

Graph signal processing (GSP) involves the modelling, representation and processing of the signals defined on the graphs. The geometric structure of data domain is represented by the graphs. A graph signal or function may be defined as a vector $\mathbf{x} \in \mathbb{R}^{\mathrm{N}}$ lying on the vertices of the graph, where the i^{th} component of the vector $\mathbf{x}$ represents the value at the i^{th} vertex [17].

A graph tuple G=(V,E,$\mathbf{W}$) can be defined as a set of vertices V={ v_1, v_2,.... v_{N}} and a set of edges$E$=$\{e_{\mathrm{ij}}$:v_{i},$v_{\mathrm{j}}\in$ V$\}$, along with a weighted adjacency matrix. e_{ij} represents the edge between vertex i and vertex j. The size of the graph, $N = |V|$, is given by the number of vertices and the weight W_{ij} is the weight of the edge between node i and node j. The weight W_{ij} is equal to zero if there is no edge. Further, it is assumed that there are no self loops in the graphs ($W_{\mathrm{ii}} = 0$). Along with the degree matrix (defined as $\mathbf{D}$ =diag$\{d_1$,d_2,...$d_{\mathrm{N}}\}$, where each d_{i} is the sum of the weights of all edges connected to v_{i}), the graph Laplacian matrix $\mathbf{L}$ can be obtained, as given in the next section.

2.3.1 The Graph Laplacian matrix

The concept of graph Laplacian has been used in clustering, link prediction, community detection etc. In GSP, the eigenvalue and the eigenvectors of graph Laplacian are used for the frequency analysis of graph signals. The graph Laplacian is given as [17]:

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \tag{2.6}$$

This graph Laplacian, defined in 2.6, is also called as the non-normalized graph Laplacian. As the non-normalized graph Laplacian matrix is symmetric and positive semi-definite, its eigenvalues are real, and the eigenvectors are orthonormal. Because the eigenvalues and the eigenvectors of graph Laplacian can be regarded as frequency components, one can talk about Laplacian spectrum which in turn becomes the key component of GSP. In addition, the use of the eigenvalues as the graph frequency and the eigenvectors as the graph Fourier basis make GSP analogous to the classical signal processing.

A notion of frequency is provided by the eigenvalues and the eigenvectors of

the graph Laplacian [17]. Zero eigenvalue corresponds to zero frequency and the associated eigenvector $\mathbf{u}_0$ is constant. The eigenvectors corresponding to low frequencies λ_l vary slowly across the graph i.e. the values of the eigenvectors at the vertices connected by the large weight are likely to be similar. While the eigenvectors corresponding to larger eigenvalues vary more rapidly on the vertices connected by a large weight. In this manner, the smaller eigenvalues correspond to low frequencies and the large eigenvalues correspond to high frequencies [17].

Apart from the non-normalized graph Laplacian, another popular option is to use the normalized weight $\frac{W_\mathrm{ij}}{\sqrt{d_\mathrm{i} d_\mathrm{j}}}$ where d_i and d_j are the i^{th} and the j^{th} diagonal elements of the degree matrix $\mathbf{D}$. It leads to the normalized graph Laplacian, which is given as [17]:

$$\tilde{\mathbf{L}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} \tag{2.7}$$

As similar to the non-normalized graph Laplacian, the spectrum of the normalized graph Laplacian $\tilde{\mathbf{L}}$ represents the graph frequency. However, its zero eigenvector $\tilde{\mathbf{u}}_0$ associated with the zero eigenvalue is not constant vector. Its advantage is that its spectrum is always contained in the interval [0,2]. Regarding the choice to use either of the normalized or the non-normalized Laplacian for representing the graph frequencies, there is not a clear answer.

2.3.2 Graph Fourier Transform (GFT)

The eigenvalues of an N node graph are denoted as $\lambda_0 \leqslant ..\lambda_{\mathrm{N\text{-}1}}$. Then $\mathbf{U} = [\mathbf{u}_0...\mathbf{u}_{\mathrm{N\text{-}1}}]$ can be defined as the matrix by stacking the corresponding eigenvectors as the columns. On finding the orthogonal set of eigenvectors and their normalization,the graph signal $\mathbf{x} \in \mathbb{R}^\mathrm{N}$, can be represented in the form of orthonormal basis. In order to project $\mathbf{x}$, the graph Fourier transform $\hat{\mathbf{x}}$ is evaluated as follows [17]:

$$\hat{x}(\lambda_\mathrm{l}) = \langle \mathbf{x}, \mathbf{u}_\mathrm{l} \rangle = \sum_{n=0}^{N-1} x(n)(u_\mathrm{l})^*(n) \tag{2.8}$$

Because the graph Fourier matrix $\mathbf{U}$ is non-sparse in general, it requires $\mathcal{O}(N^2)$ arithmetic operations and does not allow to rapidly apply GFT. In [46], a method based on multi-layer sparse factorizations was provided to obtain approximate GFT which can be applied rapidly and stored efficiently. Here, the graph Fourier

matrix $\mathbf{U}$ is approximated as a product of sparse and orthogonal matrices as follows [46]:

$$\hat{\mathbf{U}} = \mathbf{S}_1\mathbf{S}_2.....\mathbf{S}_\mathrm{J} \tag{2.9}$$

where the matrices $\mathbf{S}_1.....\mathbf{S}_\mathrm{J}$ are sparse and orthogonal. In this fast GFT computation, a modified version of the famous Jacobi eigenvalues algorithm is used.

In [47], different novel concept is given to address the problem of sketching graph signals. Instead of using the full signal to compute a few frequency coefficients and then implement a given classifier, the idea is to select a few samples of the signal and compute jointly the frequency transformation and the classification. Here, a sketch of the linear model is drawn resulting in the lower dimensional transform. Assuming that the graph signal $\mathbf{x}$ may be represented with 'k' samples only (where k << n). Here, the 'p' samples are drawn (where k $\leq$ p) by the sampling matrix $\mathbf{C}$ of dimension (p $\times$ n). Further, for the processing of the lower dimensional sampled input, a sketch $\mathbf{H}_\mathrm{s} \in \mathbb{R}^{\mathrm{m} \times \mathrm{p}}$ of the linear transform ($\mathbf{H} = \mathbf{U}$) is designed to ensure $\hat{\mathbf{y}} = \mathbf{H}_\mathrm{s}\mathbf{C}\mathbf{x}$ as an accurate estimate of the intended transformation $\mathbf{y} = \mathbf{H}\mathbf{x}$ [47]. As the cost of forming $\hat{\mathbf{y}}$ is of order $\mathcal{O}(mp)$, the computation cost is reduced by a factor of $\frac{p}{n}$.

GFT provides global information about a graph signal which covers the entire graph. However, to find the local details of a graph signal at different scales, graph wavelet transform (GWT) is used.

2.3.3 Graph Wavelet Transform (GWT)

GWT localizes the graph signal contents in both the vertex and the spectral domain. However, as given in [17], the operation of translation and scaling is not straight forward as in classical signal processing. In [48], the graph wavelet transform was designed in vertex domain while in spectral domain, GWT was explained as the diffusion wavelets [49] and the spectral graph wavelets [50]. In diffusion wavelets, diffusion is used as a scaling tool for multiscale analysis. Then, the wavelet is constructed by the compressed representations of powers of the diffusion operator. However, the spectral graph wavelets transform (SGWT) is simpler

than diffusion wavelets. In addition, SGWT provides better control over the selection of wavelet scales and gives highly redundant transform. Moreover, the fast SGWT algorithm with the reduction in computational complexity can be done using Chebyshev polynomials.

Spectral Graph Wavelet Transform (SGWT)

SGWT is based on the spectral decomposition of the graph Laplacian $\mathbf{L}$ [50], where the scaling functions are introduced to extract the features at local level. To find out each graph spectral wavelet, a kernel function g: $\mathbb{R}^+ \rightarrow \mathbb{R}^+$ is used with scaling its domain by a scalar t. Then the result is localized by its convolution with an impulse $\delta_{\mathrm{n}} \in \mathbb{R}^+$, where

$$\delta_{\mathrm{n}} = \begin{cases} 1, \text{for vertex } n \\ 0, \text{ elsewhere} \end{cases} \tag{2.10}$$

In vector form, a spectral graph wavelet $\psi_{\mathrm{t,n}} \in \mathbb{R}^{\mathrm{N}}$ at scale t localized around vertex n can be shown as [50]:

$$\psi_{\mathrm{t,n}}(m) = \sum_{l=0}^{N-1} g(t\lambda_{\mathrm{l}}) u_{\mathrm{l}}(n) u_{\mathrm{l}}(m) \tag{2.11}$$

Then, the SGWT coefficients of a given graph signal $\mathbf{f}$ is produced by inner product $<\psi_{\mathrm{t,n}},\mathbf{x}>$. The kernel g is selected as the following band-pass filter [50]:

$$g(r) = \begin{cases} {r_1}^{-\alpha} r^{\alpha}, \text{for } \mathrm{r}<\mathrm{r}_1 \\ s(r), \text{for } \mathrm{r}_1 \leq \mathrm{r} \leq \mathrm{r}_2 \\ {r_2}^{\beta} r^{-\beta}, \text{for } \mathrm{r}>\mathrm{r}_2 \end{cases} \tag{2.12}$$

where α=β=1, r_1=1 and r_2=2 and s(r) is a unique cubic spline that follows the curvature of g. Now, the high frequency information is localized around a vertex by the coefficients for smaller scale (small t) while low frequency information is captured by larger scales. In order to stably represent the low frequency content in the graph, a scaling kernel h: $\mathbb{R}^+ \rightarrow \mathbb{R}$, is included for creating a scalar function ϕ_{n}. γ is a parameter used to equalize the value of $h(0)$ to the maxima of g and

the design parameter where $\lambda_{\max}$ is an upper bound of the maximum eigenvalue of the graph Laplacian. The scaling vector ϕ_{n} is defined as follows:

$$\phi_{\mathrm{n}}(m) = \sum_{l=0}^{N-1} h(\lambda_{\mathrm{l}}) u_{\mathrm{l}}(n) u_{\mathrm{l}}(m) \tag{2.13}$$

Let J denote an integer such that the set of wavelet scales is $\{\mathrm{t_j}\}_{\mathrm{j=1,....J}}$. Then, the SGWT is transformed with $J+1$ scales; J wavelets and one scaling function. Finally, the transform coefficients is mapped as $(J+1)N$ dimensional vector $\mathbf{c} = T^{\mathrm{T}}\mathbf{f}$ by collecting the wavelet and scaling function vectors in a transformation matrix $\mathbf{T} = [\Psi_{t_1}.....\Psi_{t_J}, \Phi]$.

Since the SGWT is an overcomplete transform as there are more wavelet coefficients than vertices in the graph. For the signal which is represented using only a few wavelet scales, the SGWT seems quite similar to sparse coding[51], where each wavelet acts as an atom in a sparse dictionary[52]. However, the SGWT can be computed more efficiently computed than sparse coding transformations due to its fixed mathematical structure. Even, if graph structure is embedded in the learning dictionary, an efficient implementation is not guaranteed. While, SGWT has also the advantage in efficient implementation by performing the operations directly in the vertex domain without the need of diagonalizing the Laplacian.The details of that implementation is followed in the next section.

Fast Approximate SGWT

As the direct computation of the SGWT has the complexity of $O(N^3)$ with $O(N^2)$ memory, it is feasible only for the graphs with few thousand nodes [50]. As a solution, another method based on truncated Chebyshev polynomials is introduced in [50] which has the computational complexity of $O(M|E|+NJ)$ where $|E|$ is the number of non-zero edges in the graph and M is the degree of the polynomial. The kernels g and h using low dimensional Chebyshev polynomials is approximated as [50]:

$$g(t_{\mathrm{j}}\lambda) \approx \frac{1}{2} c_{\mathrm{j,0}} + \sum_{k=1}^{M_{\mathrm{j}}} c_{\mathrm{j,k}} \overline{T_{\mathrm{k}}}(\lambda) \tag{2.14}$$

where $\mathrm{M_j}$ is the degree of the approximation, typically $\mathrm{M_j} = 50$. The expansion $\overline{T_\mathrm{k}}(\lambda)$ is equivalent to $T_\mathrm{k}(\lambda - 1)$, the shifted Chebyshev polynomial of order k which satisfies the following relation: $T_\mathrm{k}(\lambda) = 2\lambda T_{\mathrm{k\text{-}1}}(\lambda) - T_{\mathrm{k\text{-}2}}(\lambda)$. Moreover, $c_{\mathrm{j,k}}$ are the Chebyshev coeffcients to be estimated by a spectrum upperbound $\lambda_{\max}$[53].

The approximated transforms are given as:

$$\Psi_{\mathrm{t_j}}{}^{\mathrm{T}}\mathbf{x} \approx \frac{1}{2}c_{\mathrm{j},0}\mathbf{x} + \sum_{k=1}^{M_\mathrm{j}} c_{\mathrm{j,k}}\overline{T_\mathrm{k}}(L)\mathbf{x} \tag{2.15}$$

$$\Phi_{\mathrm{t_j}}{}^{\mathrm{T}}\mathbf{x} \approx \frac{1}{2}c_{0,0}\mathbf{x} + \sum_{k=1}^{M_\mathrm{j}} c_{\mathrm{j,k}}\overline{T_\mathrm{k}}(L)\mathbf{x} \tag{2.16}$$

with $\overline{T_0}(L) = I$ and $\overline{T_1}(L) = L - I$. Using the matrix-vector multiplication, the approximation yields $\mathbf{L}$ and thus, becomes fast for sparse graph.

2.3.4 Application of GSP in the image processing

Recent advances of GSP has given rise to the intensive studies of signals that live naturally on irregular data kernels described by graphs e.g. social networks, wireless sensor networks. Though a digital image contains pixels that reside on a regularly sampled 2-D grid, an appropriate undelrlying graph can be designed connecting pixels with weights that reflect the image structure. In this manner, the image can be interpreted as a signal on a graph and GSP tools for processing and analysis of the signal in graph spectral domain can be applied. The new insights have been achieved in a number of image processing areas like image compression, image restoration, filtering and segmentation [54].

For image compression, GFT and its variants have been applied in coding of piecewise smooth and natural images. As the different graph structures can be used to define GFT, it is required to consider both the cost of describing the graph as well as the cst of coding GFT coefficients to represent the signal.

In image restoration including denoising and deblurring, designing appropriate signal is a major challenge. Wiener filter has been applied for graph signals to minimize the denoising problem [55]. In another image denoising approach [56], an observed signal has been projected to a low dimensional Krylov subspace of the

graph Laplacian via a conjugate gradient method, resulting in a fast image filtering operation that provides an alternative to Chebyshev polynomial approximation for the same order.

For image filtering, extraction of smooth components of the image i.e. low graph frequency components is a significant issue. In [57], heat kernel in the spectral domain has been proposed to find the graph spectral filter.By the graph spectral representation of bilateral filter, the bilateral filter has been implemented by a combination of graph Fourier basis and a graph low pass filter [58].

Image segmentation refers to the partition of an image into different regions where each region has its own meaning or characteristic in the image. In [59], the use of combinatorial graph cut algorithms has been done to solve variational image segmentation problems.

2.4 Summary

In this chapter, we presented the review of the basic stages of FER and the techniques used in these stages. Here, we have discussed the brief relevance of the mathematical preliminaries associated with these techniques. The techniques include Gabor filter, HOG, DWT, CT and FRFT. We observe that the interrelationship among their components is not utilized. We have also provided a brief review of the basics of GSP and its related concepts which have been used in our work of FER. We propose to use the GSP in the following chapter in order to capture the interrelationship to improve the feature extraction in term of accuracy as well as the dimensionality reduction.

Chapter 3

GSP based approach for FER using HOG and DWT features

In the previous chapter, the basics of the DWT and HOG have been discussed. In this chapter, GSP based approach for FER using DWT and HOG features is proposed and it is shown that how the application of GSP reduces the dimension of the existing feature vector and improves the accuracy of the FER. But the drawback of this algorithm is the increase in computational complexity due to the combined use of DWT and the HOG. In order to reduce the computational complexity, we evaluate this approach without DWT. For compensating the role of DWT in the accuracy, the parameter of the HOG (cell of size) has been adjusted to optimize the accuracy. Although using the small values for the cell of size increases the accuracy of the FER, the feature dimension also increases significantly. However, due to the large feature length of the facial expression, there is a challenge to decrease the size of the feature vector. Here, we investigate how the use of GSP tools can help to handle this challenge. This proposed method is also demonstrated on CK+ and JAFFE database and the experimental performance is compared with the proposed method in Chapter 2 to find out its advantages and limitations.

3.1 Introduction

DWT has been found to be an effective method for the face detection as well as the facial expression recognition. It has been used along with PCA for the feature

extraction and the dimensionality reduction in [8], [60] and [61]. In [60], each image is decomposed by DWT and then, linear discriminant analysis (LDA) approach is used to extract features from the decomposed low frequency and high-frequency components. In [61], the method of wavelet decomposition has been used with neural network to extract the feature of facial expression.

To extract the information of the facial expression lying in the form of different edges, HOG has been used for facial expression recognition [62], [63]. The application of HOG descriptor has been shown with adjusting different parameters to consider this technique as one of the most suitable for characterizing the facial expression peculiarities in [62]. In [63], the real-value-coded genetic algorithm (GA) was used for the selection of HOG descriptor parameters.

Seeing the utility for FER, we have constructed the feature vector to take into account both the DWT and HOG features. By such combination of DWT and HOG, we will able to collectively take the advantage of them. However, this leads to the increased dimensionality of the feature vector. As we referred in Chapter 2 about the GSP utility for the analysis of multi-dimensional signals, a graph may be build incorporating the neighborhood information of that signal. Similar to the GSP, the concept of Locality Preserving Projection (LPP) is discussed in [18] and it was used for face recognition in [64]. The main step for the GSP method is to define the weight between the nodes (using the feature vector computed from DWT-HOG) and compute the final lower dimensional feature vector by projection of the initial feature vector on the generalized eigenvector basis [18].

3.2 Feature Extraction Review

3.2.1 Feature extraction with DWT and HOG

As referred in Chapter-2, the DWT features can be extracted of the facial image at different scales with the components of 'LL', 'LH', 'HL' and 'HH'. It is to be noted that the 'LL'band features due to low frequency are used mainly for the face recognition whereas the high frequency components including 'LH','HL' and 'HH' are used to recognize the face expressions. Selecting the level of wavelet decomposition further leads to the better decomposition in case the image is not too

small,as shown in Fig. 3.1 and Fig. 3.2 for JAFFE database image. However,for the small image,the higher level decomposition weakens the distinguished power of recognition.

Similarly, the HOG features can be computed for the facial image divided into the cells by drawing the histogram for the gradients of these cells, as referred in Chapter-2.

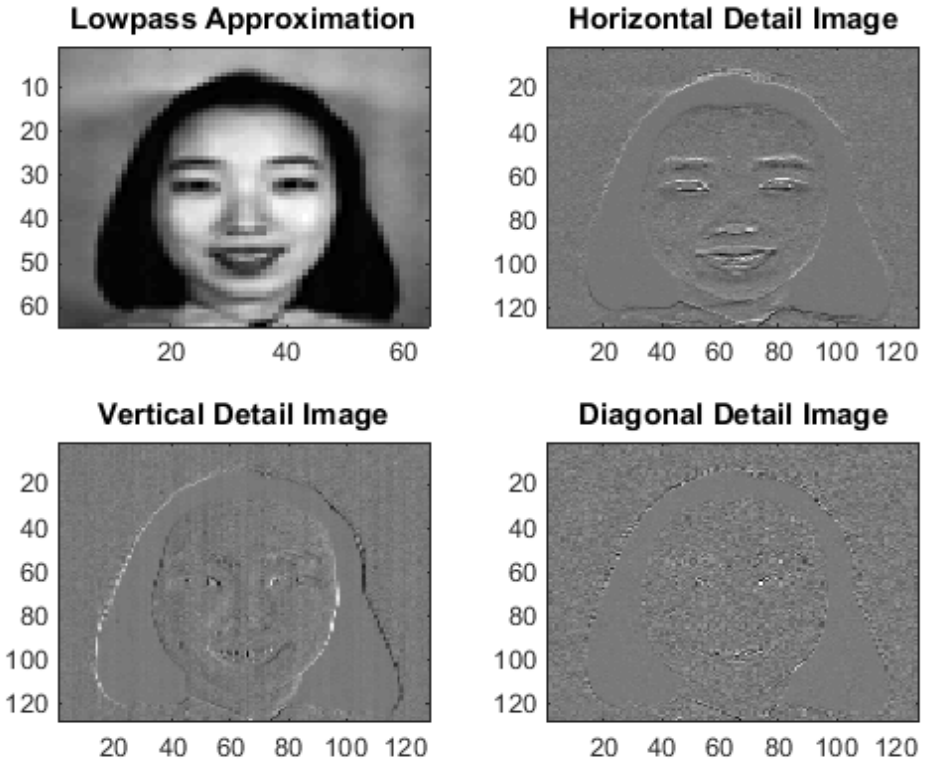

Figure 3.1: 1st level wavelet decomposition of JAFFE image

3.2.2 Feature Extraction with GSP

In order to reduce the dimension of the above computed feature vector $\mathbf{x}_\mathrm{i}$ using the DWT and the HOG, the GSP is applied. The GSP requires the signal to be represented as the graph, as shown in Fig. 3.3 .

For building a graph, let each facial image be considered as the vertex (denoted as red in Fig. 3.3) and $\mathbf{X}$ be the graph signal with N nodes on the vertices V={ v_1, v_2,.... v_N} connected by the edges E=$\{e_\mathrm{ij}$:v_i,$v_\mathrm{j}\in V\}$. Here, the computed feature vector $\mathbf{x}_\mathrm{i}$ is taken as the i-th component of the graph signal $\mathbf{X}$ lying on vertex

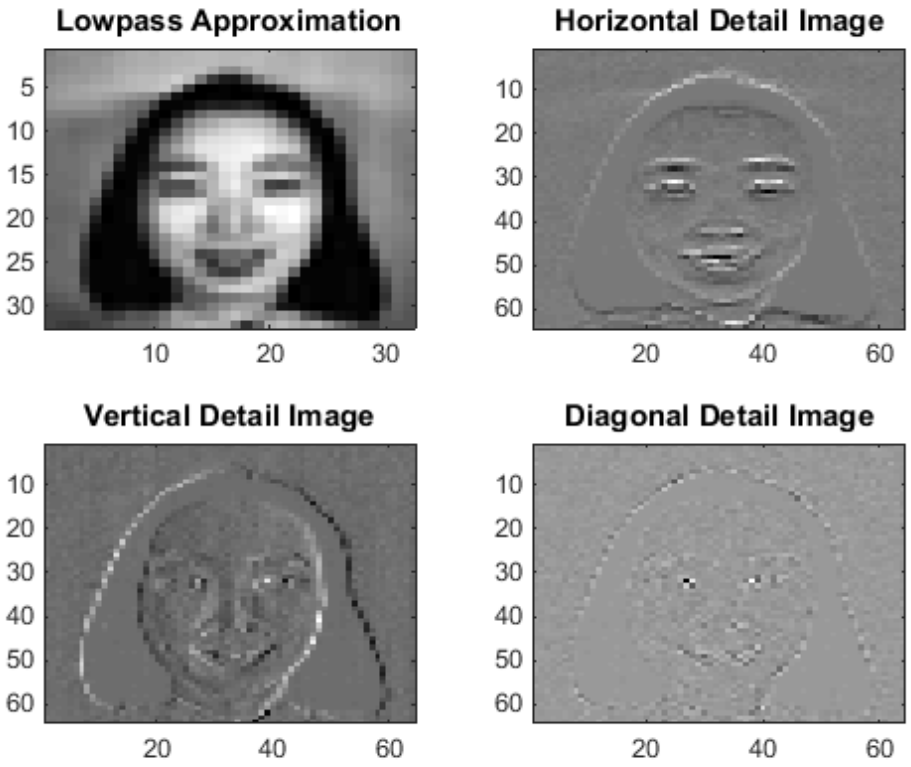

Figure 3.2: 2nd level wavelet decomposition of JAFFE image

v_{i} , shown as vertical in Fig. 3.3. The edges of the graph are undirected here. The level of connectivity for an edge joining vertices v_{i} and v_{j} is specified by the associated weight W_{ij} and it is defined as [18]

$$W_{\mathrm{ij}} = \begin{cases} exp(-\frac{\|\mathbf{x}_{\mathrm{i}}-\mathbf{x}_{\mathrm{j}}\|^2}{t}), \text{ if } \|\mathbf{x}_{\mathrm{i}}-\mathbf{x}_{\mathrm{j}}\|^2 < \varepsilon \\ 0, \text{ elsewhere} \end{cases} \tag{3.1}$$

where the symbol $\|\mathbf{x}\|$ stands for L^2 norm of a vector.

A graph is compactly represented by its adjacency matrix W where each entry is given by W_{ij}. Along with the degree matrix (defined as D =diag$\{d_1,d_2,...d_{\mathrm{N}}\}$, where each d_{i} is the sum of the weights of all edges connected to v_{i}), the matrix $\mathbf{L}$ gives the graph Laplacian: $\boldsymbol{L}=\boldsymbol{D}-\boldsymbol{W}$.

Then, the eigenvectors and eigenvalues for the generalized eigenvector problem

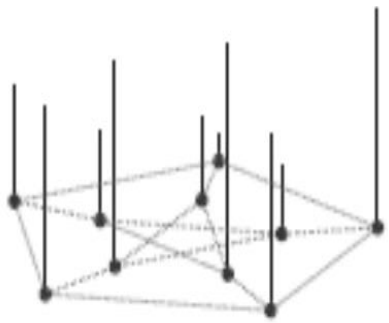

Figure 3.3: A random positive graph signal on the vertices of the Petersen graph [17]

are computed as follows [18]:

$$XLX^{\intercal}\mathbf{a} = \lambda XDX^{\intercal}\mathbf{a} \tag{3.2}$$

Let the column vectors $\mathbf{a}_0$,...,$\mathbf{a}_{l\text{-}1}$ be the solution of (3.2) corresponding to the eigenvalues, λ_0<..<$\lambda_{l\text{-}1}$. The reduced vector $\mathbf{y}_i$ is given as follows:

$$\mathbf{y}_i = A^{\intercal}\mathbf{x}_i, A = (\mathbf{a}_0, ..., \mathbf{a}_{l\text{-}1}) \tag{3.3}$$

where $\mathbf{y}_i$ is an l-dimensional vector ($l << n$), and A is an $n \times l$ matrix. Finally, the kNN classifier is trained and tested to classify each facial expression using the feature vector $\mathbf{y}_i$.

3.2.3 kNN classifier

kNN is a non-parametric lazy learning algorithm i.e. it does not make any assumptions on the underlying data distribution. The term 'lazy' implies that it does not use the training data points to do any generalization. Thus, it keeps all the training data.The cost is high in terms of both time and memory. Yet it is preferred due to its simplicity in implementation.

Suppose there are 'C' known pattern classes and the number of samples in each

class are n_{i} (i=1,2,.....C). For a given new sample 'Y', the nearest neighbor can be found out as follows:

$$d_{\mathrm{i}}(Y) = \parallel Y - Y_{\mathrm{ir}} \parallel^2 \tag{3.4}$$

where Y_{ir} is the training sample of class i as well as the nearest neighbor to the sample Y. d_{i}(Y) is the distance based upon the similarity (generally, the Euclidean distance is selected) in (3.4). 'Y' is considered as belonging to the class 'q' in case the distance between 'Y' and the class 'q' is minimal, as shown in (3.5).

$$d_{\mathrm{q}}(Y) = \min_{i} d_{\mathrm{i}}(Y) \tag{3.5}$$

3.3 Proposed GSP approach with the combination of DWT and HOG

Since most of the facial expressions are distinguishable on the basis of edge information lying in the region of the mouth and the eye pair together, the edge information is extracted from these regions. This requires cropping of the the mouth and the eye pair in the face images and we have used the Viola-Jones algorithm. In order to extract the edge information from the regions $\mathrm{I_m}$ (for mouth) and $\mathrm{I_e}$ (for eyes), HOG descriptors [10] ($\mathrm{H_m}$ and $\mathrm{H_e}$) are computed for the corner points of the mouth and the eye pair respectively. For reducing the computation cost and the length of the descriptors, the corner points have been selected in place of the whole mouth and eye pair using the FAST(Features from accelerated segment test) algorithm [65]. Simultaneously, the 2D-DWT Haar wavelets are used to decompose the frequency components of the regions ($\mathrm{I_m}$ and $\mathrm{I_e}$) into different bands- '$\mathrm{LL_m}$', '$\mathrm{LH_m}$', '$\mathrm{HL_m}$', '$\mathrm{HH_m}$' and '$\mathrm{LL_e}$', '$\mathrm{LH_e}$', '$\mathrm{HL_e}$' and '$\mathrm{HH_e}$' . Since the details of the contour, edge and texture features of the images are characterized by the high frequency components [60], zero weight is assigned to the low frequency components. Moreover, due to the elliptical shape of the mouth (around lips region) as well as the eyes, the horizontal component (HL) cover the more edges than the vertical and the diagonal component. Thus, the components are weighted with $w_{\mathrm{LL}} = 0, w_{\mathrm{LH}} > w_{\mathrm{HL}} > w_{\mathrm{HH}}$ to form the feature vector $\mathrm{D_m}$ and $\mathrm{D_e}$.

Next, the features extracted from the mouth and the eye pair using the DWT and the HOG are to be combined together for every face. In order to reduce the dimension of the combined feature vector, only the mean of the H_m and H_e is used and added to the every element of the feature vector D_m and D_e respectively. Then, the resultant DH_m and DH_e are concatenated into the single feature vector $\mathbf{x}_i$ of the dimension n for the i-th face. Thereafter, as per our holistic GSP approach of two stages, stage two is followed after computing the feature vector $\mathbf{x}_i$ by using the weight based interrelationship between these feature vectors. Now, from the weights defined in (3.1), the graph is constructed with the obtained feature vector $\mathbf{x_i}$ as the vertex and the Laplacian $\mathbf{L}$ matrix is calculated. Thereafter, the reduced dimension feature vector $\mathbf{y_i}$ is computed by solving the generalized eigenvector problem, as per (3.2) and (3.3). Finally, the kNN classifier is trained and tested to classify each facial expression using the feature vector $\mathbf{y}_i$. The proposed GSP based DWT-HOG algorithm for FER is illustrated in Fig. 3.4.

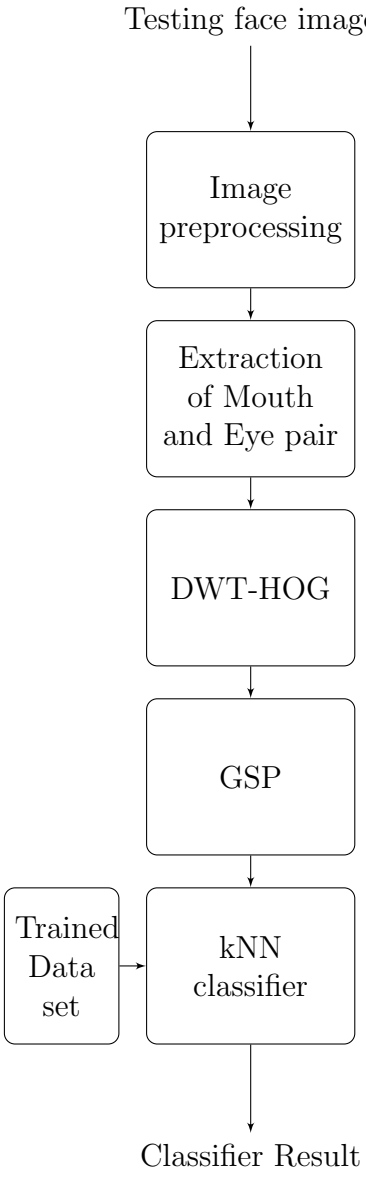

Figure 3.4: *The block diagram of GSP based DWT-HOG method*

3.4 Simulation results

Two standard facial databases have been used to study the performance of our method. The two databases are as follows: JAFFE database [66] and Extended Cohn-Kanade (CK+) database [67].

3.4.1 JAFFE database

In JAFFE database, there are 213 images of 10 subjects with 7 expressions viz. “anger”, “disgust”, “fear”, “happy”, “neutral”, “sad” and “surprise”. Every sub-

ject has 3-4 samples for each of the expression. For our experiments, training images are taken using leave one out approach i.e for each expression one expression from the samples is left for testing and the remaining are taken for the training set.

3.4.2 CK+ database

CK+ database includes around 593 image sequences (posed expressions) of around 123 subjects, and consists of same expressions like JAFFE except "contempt" in place of the "neutral". For each expression of a subject, multiple frames are arranged in the order of the increasing intensity of emotion. The last frames at the last position exhibiting peak intensity of emotion were chosen and overall, 248 frames have been selected. Here, total samples around 164 have been randomly taken for the training and the remaining samples were used for testing using a leave-one-out-approach.

The sample images of the JAFFE subjects and CK+ subjects are shown in Fig. 3.5 and Fig.3.6 respectively.

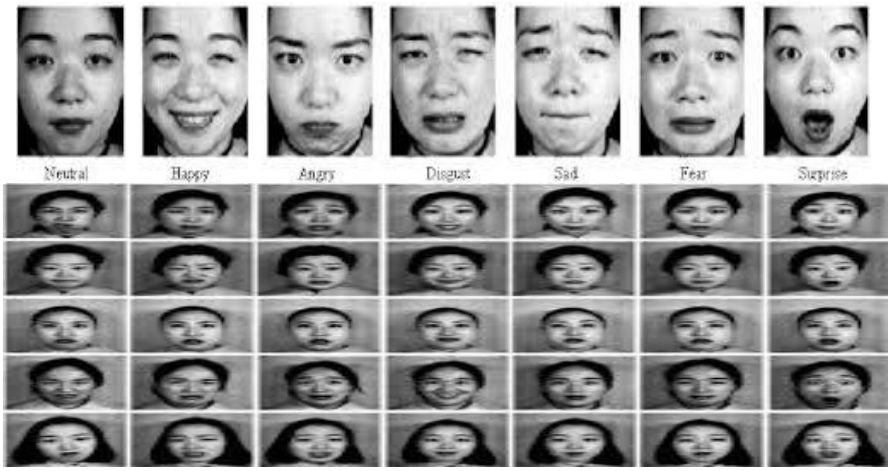

Figure 3.5: Sample images from JAFFE database

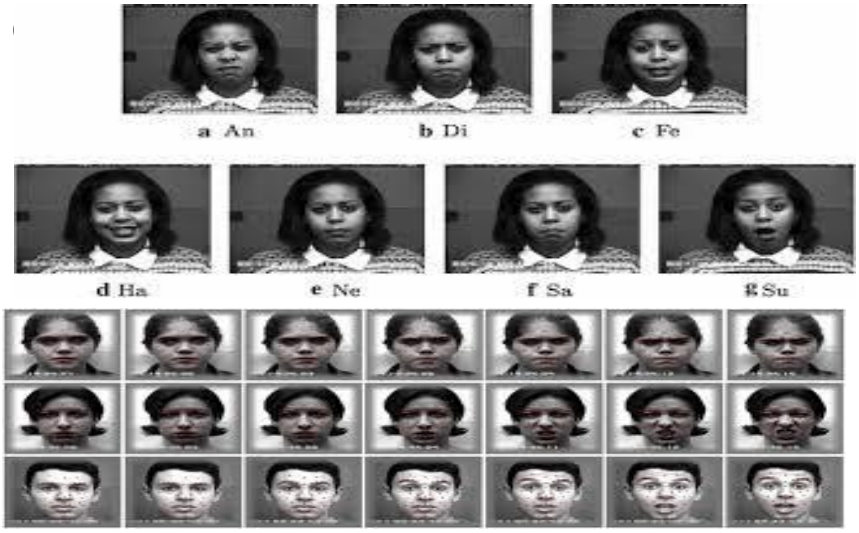

Figure 3.6: Sample images from CK+ database

In the preprocessing step, each image is normalized to the size of 256 × 256. Using different weight combinations for the different level of DWT, the simulations are performed to find the suitable weights, which are as follows:

$$w_{\mathrm{LL}} = 0, w_{\mathrm{LH}} = 3, w_{\mathrm{HL}} = 2, w_{\mathrm{HH}} = 1$$

The overall recognition rates using HOG and DWT features alone are shown in Table 3.1 and Table 3.2 for JAFFE and CK+ respectively. It is clear that the different levels of DWT affect the length of the feature vector and the overall recognition rate of the facial expression.

Table 3.1: The impact of different level of DWT on the length of the feature vector and the FER for JAFFE database

Method	DWT Level for Mouth	DWT Level for Eyes	Feature dimension	Recognition rate JAFFE (%)
DWT with HOG	2	2	456	48.14
DWT with HOG	3	3	126	66.16
DWT with HOG	3	4	84	74.04
DWT with HOG	4	3	76	66.66
DWT with HOG	4	4	34	62.96

Table 3.2: The impact of different level of DWT on the length of the feature vector and the FER for CK+ database

Method	DWT Level for Mouth	DWT Level for Eyes	Feature dimension	Recognition rate CK+ (%)
DWT with HOG	2	2	420	77.38
DWT with HOG	3	3	115	72.61
DWT with HOG	3	4	64	77.38
DWT with HOG	4	3	87	73.8
DWT with HOG	4	4	36	72.61

Table 3.3 shows the confusion matrix for the proposed method for JAFFE database. Moreover, the proposed method is compared with the initial approach of DWT-HOG in Table 3.4 for evaluating the effect of using GSP in terms of the recognition accuracy and the length of the feature vector. Finally, the proposed approach is compared with the wavelet based approaches, shown in Table 3.5, in terms of the accuracy. It is clear from the Table 3.5 that the use of GSP in

the proposed method improves the recognition accuracy with the decrease in the dimensionality of the feature vector.

Table 3.3: Confusion matrix of the GSP-HOG method for the JAFFE database

	Anger	Disgust	Fear	Happy	Sad	Surprise
Anger	90	0	0	0	10	0
Disgust	0	85.71	0	0	14.29	0
Fear	0	0	100	0	0	0
Happy	0	0	0	100	0	0
Sad	0	14.29	0	0	85.71	0
Surprise	0	0	0	0	9.1	90.9

Table 3.4: Comparison of the DWT-HOG with and without GSP

Method	JAFFE Feature dimension	CK+ Feature dimension	Accuracy JAFFE (%)	Accuracy CK+ (%)
DWT-HOG	84	64	74.04	77.38
DWT-HOG with GSP	14	35	92.1	82.14

Table 3.5: Comparison of JAFFE with the existing DWT methods

Method	Accuracy (%)
Wavelet packet decomposition[61]	83.7
DWT + PCA/LDA[60]	88.89
Proposed method using GSP	92.1

3.4.3 Discussion

The application of GSP has decreased the dimension of the feature vector significantly and increased the accuracy in comparison to the DWT-HOG scheme. However, we realize that the combined use of DWT and the HOG increases the computational complexity to obtain the feature vector. Then, to reduce the computational complexity, we evaluate the former approach without DWT. Without

using DWT, the resultant feature vector will not be effective as earlier. To increase the effectiveness of the feature vector, we observe that the parameter of the HOG, cell of size, can be adjusted to increase the distinctiveness among the feature vectors.

3.5 Experimental Performance for GSP-HOG method

We have taken the same databases of JAFFE and CK+ for analysing the effect of the simplified approach. In the experiment, the optimum size of the eye pair and the mouth is selected as 35× 120 and 40×60 to take into account the different patterns in the eyes and the mouth respectively. Next step is to decide the 'Cell Size' for the mouth and the eyes. As the size of the cell is small, fine details of the image are extracted while the capturing of the large scale spatial information requires the increase in the size of the cell. In addition, the dimensionality of the corresponding HOG feature vector $\mathbf{x_i}$ depends upon the cell size. For example, on taking the cell size of [4 4] for the mouth and [8 8] for the eyes, the dimension of the H_{m} and H_{e} becomes 1×1512 and 1×4536 respectively. Thus, the overall dimension of $\mathbf{x_i}$, which is formed by the concatenation of H_{m} and H_{e}, becomes 1×6048. Depending upon the different cell size, the length of the HOG based feature vector $\mathbf{x_i}$ is shown in Table 3.6.

On extracting the features from HOG algorithm, every feature vector $\mathbf{x}_{\mathrm{i}}$ is considered as the signal on the *i-th* vertex of a graph. For applying the GSP methodology, the weight is calculated as per 3.1, between the different vertices of the graph to find the weight matrix $\mathbf{W}$. The value of t is selected as 0.6. Thereafter, the Laplacian matrix L is computed to find out the eigenbasis A as shown in (3.2). Finally, the $\mathbf{x}_{\mathrm{i}}$ is projected on the A to get the reduced feature vector $\mathbf{y}_{\mathrm{i}}$. The kNN classifier is trained with that reduced vector. In order to analyze the effect of GSP on the length of the feature vector and the accuracy of the facial expression recognition, the parameter of HOG descriptor named 'Cell Size' has been varied, as shown below in Table 3.6.

Table 3.6: Effect of the 'Cell size' for CK+ dataset

Mouth-Cell size	Eyes- Cell size	HOG-Feature length	GSP-Feature length	HOG-Accuracy (in %)	GSP-Accuracy (in %)
[4 4]	[4 4]	11844	47	91.66	90.47
[8 8]	[4 4]	8172	50	85.71	88.09
[12 12]	[4 4]	7596	50	84.52	88.09
[4 4]	[8 8]	6048	47	90.47	92.85
[8 8]	[8 8]	2376	50	89.28	94.04
[12 12]	[8 8]	1800	50	80.95	90.47
[4 4]	[12 12]	4860	50	86.40	92.85
[8 8]	[12 12]	1188	50	96.42	**98.03**
[12 12]	[12 12]	612	50	92.85	95.23

Based on the different values of the 'cell size' in Table 3.6, Table 3.7 shows the confusion matrix for the proposed method with the best accuracy. Further, the proposed method has been compared with the State-of-the-art approaches in Table 3.8 in terms of the recognition rate of the six prototypic emotions.

Table 3.7: Confusion matrix for FER using GSP-HOG method for CK+ dataset (Average Recognition Rate=98.03%)

	Anger	Disgust	Fear	Happy	Sad	Surprise
Anger	100	0	0	10	0	0
Disgust	0	100	0	0	0	0
Fear	5.89	5.89	88.22	0	0	0
Happy	0	0	0	100	0	0
Sad	0	0	0	0	100	0
Surprise	0	0	0	0	0	100

Table 3.8: Performance comparison of our method vs different State-of-the-art approaches for CK+ 6 expressions

	[68]	[69]	[70]	[71]	[63]	[62]	Proposed Work
Anger	87.1	87.1	71.4	87.8	94.07	88.6	100
Disgust	90.2	91.6	95.3	93.3	98.31	89.0	100
Fear	92.0	91.0	81.1	94.3	82.67	100	88.22
Happy	98.1	96.9	95.4	94.2	100	100	100
Sad	91.5	84.6	88.0	96.4	100	100	100
Surprise	100	91.2	98.3	98.5	98.8	97.4	100
Average	93.1	90.4	88.3	94.1	95.64	95.8	98.03

Unlike CK+ database, all the seven expressions (including neutral) have been considered to provide the sufficient number of images for the training and the testing. Every subject has 3-4 images for each of the expression. The training images (N=147) are selected using a leave- one-out approach, i.e., for each expression one expression from the samples for every subject is left for testing and the remaining are taken for the training set. The value of t is selected as 1. On setting the different values of cell size, the best result for the accuracy has been obtained for the cell size of [12 12] for both the mouth and the eyes together. Here, the length of the HOG based feature vector has been reduced from 621 to 58. Table 3.9 shows the confusion matrix for the proposed method for the JAFFE dataset.

Table 3.9: Confusion matrix for FER using GSP-HOG for the JAFFE dataset (Average Recognition Rate=88.5%)

	Anger	Disgust	Fear	Happy	Neutral	Sad	Surprise
Anger	90	0	0	10	0	0	0
Disgust	0	80	0	0	0	20	0
Fear	0	0	100	0	0	0	0
Happy	0	0	0	80	20	0	0
Neutral	0	0	0	0	100	0	0
Sad	0	0	0	0	10	90	0
Surprise	0	0	0	0	20	0	80

Further, in Table 3.10, the results of the proposed method on the JAFFE database are compared with the present methods using HOG. In [7], the Local Binary Pattern (LBP) descriptors are applied to represent the facial expression and the optimal features are selected using Adaboost. In [68], 3D Gabor features for obtaining the salient distance features were used.

In HOG based approaches,the HOG is used with the Genetic Algorithm in [63] and the LBP in [72] respectively. In [73], HOG features of corner points are computed, followed by Discrete Wavelet Transform (DWT) and the GSP. While the performance of our method is reasonably in line with most of the HOG method but comparatively less than [74]. However, the proposed method is better than [74] with respect to the feature length, as shown in Table 3.11.

Table 3.10: Comparison between the recognition accuracy of the GSP-HOG method with state-of-the-art methods using JAFFE database

Method	Facial features	Recognition rate(in %)
[7]	LBP	89.1
[68]	Patch based Gabor	92.3
[63]	HOG+GA	87.82
[72]	PHOG	87.43
[73]	HOG-DWT+GSP	92.1
[74]	HOG	**94.3**
Proposed method	HOG+GSP	88.57

Table 3.11: Effect of GSP on feature dimension

Method	Training set size	Tesing set size	Feature length
HOG [74]	190	23	5616
Proposed method	147	66	58

3.5.1 Discussion of GSP-HOG method

It is to be emphasized that our proposed method with the application of novel concept GSP makes the existing techniques (HOG here) further effective. Because the focus of the HOG remain upto finding the difference in the different facial parts (say, the eyes region, the lips region etc.) individually at the local level but, at the higher (global) level, the changes in one part of the facial region (lips region) are also related to the changes in the shape of the another facial region (eyes region) during the facial expressions, we have taken into account the unexplored relationship between the different facial regions using the GSP to find the better performance. And, the improved performance is clearly observed in Tables 3.6 for CK+. In addition, in Table 3.8, the performance of the proposed method is found to be better in most of the facial expressions individually.

For JAFFE, in Table 3.10, it is found that the performance of HOG in [74] is better than other HOG based methods. The underlying factor behind this better performance may be attributed to the larger training set (of 190 > 147, proposed method)) and the smaller testing set (of 23 < 66, proposed method). This larger ratio of training and testing set is likely to increase the accuracy in [74]. In addition, due to the comparative difference in the feature length, the computation time becomes very less in the proposed method. Apart from [74], [73] is another HOG method which performs better than the proposed method. However, it is important to notice here that the DWT has also been used in [73], which improves the accuracy but with the increase in computational complexity. Hence, the proposed method provides the comparatively better performance on taking into account the accuracy as well as the computation complexity criteria with respect to [73].

The proposed method provides significantly improved accuracy with the reduced size for the CK+ database but the improvement is not equally significant for the JAFFE database. As referred in [73], GSP along with the DWT has provided good accuracy along with the reduced size feature vector for JAFFE dataset. Thus, the direct application of GSP based approach ensures the reduction in the size of HOG based feature vector but it is not necessary that it will give the high amount of accuracy simultaneously for every database. Depending upon the specific database, the different type of processing (as DWT here) becomes necessary for the improved accuracy.

From the viewpoint of using the HOG, it is observed that in case of the lower cell size ([4 4]) for the mouth and the eye pair, the size of the feature vector becomes higher (11844 as shown in Table 3.6). In such cases, the size of the feature vector is remarkably reduced using GSP later but the computation of the eigenvector, as mentioned in (3.2), requires large processing time. Hence, there exists a trade off between the order of reduction in the size of the feature vector and the initial computational cost of the eigenbasis in the proposed method.

3.6 Summary

We have presented a GSP based DWT-HOG approach for the FER. Then, to decrease the computational complexity, we did not use the DWT. We apply the GSP on the HOG feature vector. Comparing the results of GSP-HOG with GSP based DWT-HOG method, we find that the use of DWT increases the accuracy for JAFFE database but do not improve the results of CK+. This indicates that the one scheme does not fit all the databases of FER. On getting the improved results with DWT and HOG, we are interested in further exploring the application of GSP with other existing FER methods, which are better than DWT, in the coming chapter.

Chapter 4

GSP based approach for FER using CT and FRFT

In the previous chapter, the GSP based approach for FER using DWT and HOG features was proposed which provides a significant improvement in the accuracy of FER. We began to explore the further application of GSP in existing FER schemes of CT and FRFT. First, we show how the joint application of GSP with CT can enhance the correct recognition rate of FER and decrease the dimension of the feature vector. Next, the GSP based approach with FRFT feature vector is proposed and demonstrated on the CK+ database.

As the CT remain limited to mainly to the facial recognition, it is less frequently used in the FER due to the large dimension of its feature vector. In this chapter, the proposed GSP- CT approach has decreased the existing feature vector size substantially (around 4 % of it). Moreover, that new reduced feature vector has been more effective with the improved recognition rate of the FER. Similarly, the proposed GSP-FRFT feature vector has performed better in the recognition of FER than the existing FRFT feature vectors. In addition, it results in the reduced computational time complexity due to its smaller length than the existing feature vectors.

4.1 Introduction

Features in the facial images are multi dimensional. The wavelet transform which has been extensively used as a tool for mathematical analysis of the facial images has the disadvantage of a poor directionality. Compared to the wavelet transform, CT is found to perform better in capturing directional information and representing the edges. Since the different facial expressions are characterized by the variability in orientations of facial curves, CT becomes more suitable for the feature extraction in the FER [75].

Using wavelets successfully, new multianalysis tools like ridgelets [76], contourlets were developed. CT, because of improved directional elements and better ability to represent edges, begins to be used. However, its application is limited to image denoising [77], image compression [78] and high quality image restoration [79]. However, the problems of pattern recognition are still under explored. Curvelet based face recognition has been discussed in [80] and [81]. It is shown that curvelets can exceed over wavelets in performance. In [81] the bit quantized images are used to extract curvelet features at different resolutions. However,the main problem in the curvelet transform is that they are computationally expensive. For the face recognition, dimension reduction techniques along with curvelet transform are shown in [82]. CT combined with PCA for facial expression recognition is introduced in [83]. Because the information in facial expression contains the joint interrelationship of the edges, GSP preserves the correlation of the edges at local neighborhood level and reduces the dimension of the feature vector significantly. In this chapter, our objective is to introduce the GSP for reducing the dimension and increase the accuracy of the facial expression recognition based on the Curvelet Transform.

This chapter is organized as follows: Section 4.2 reviews the basic concept of the CT and SVM in brief. In section 4.3, the proposed hybrid method of GSP-CT is explained. Section 4.4 discusses the experimental performance and the results of the proposed work.

4.2 Review of the CT

As referred in Chapter-2, the feature vector of the facial image using CT is computed. Then the graph is constructed with a feature vector for each facial image as the $i-th$ component of the graph signal similarly, as discussed in Chapter 3.

4.2.1 Support Vector Machine(SVM)

SVMs are used for the classification of the expression classes. The idea behind SVM is to build a hyperplane to separate the high dimensional space. As the distance between the hyperplane and the training data of any class becomes largest, the best separation is achieved. The LIBSVM [84] has been used in our experiment.

4.3 Proposed method of GSP approach with CT

It is well known that the information of the facial expression lies in the form of edges. The detail and fine coefficient of the CT can effectively capture that information. Moreover, the specific facial expression differentiated by its joint relationship of the edges can be well encoded in the graph signal form. The structure of the graph signal represented as graph encode the interrelationship of the edges effectively. Further, GSP allows to reduce the size of the feature vector. Because of these favorable aspects, the combination of CT and GSP has been employed here.

As shown in Fig 4.1, the representation of edges has been compared between wavelet (left) and the curvelet. Because the wavelets is better suited for the line discontinuity (piecewise approximation) while the curvelets take into account the curvature, hence the curvelets may take a bigger window, requiring the less features than the wavelets. At each scale, the number of required curvelets of elongated needle shape are smaller than the number of wavelets of square shape[86]. Curvelets are found to be very effective for representing objects with 'curve-punctuated smoothness'. These objects display smoothness except for discontinuity along a general curve e.g. facial images with edges. An image of size, say 32×32, when decomposed using curvelet transform at scale 3 (coarse,detail and fine) and angle 8, produce three subbands. Since the facial expression information lies in the high

Figure 4.1: Compare of two dimensional edge representation between wavelet and curvelet transform [85]. In the left figure of the wavelet transform, the edge is shown to be covered by the square boxes (represented as wavelet) and in the right figure of the curvelet transform, same edge is covered by the elongated needle shape (represented as curvelet).

frequency of the CT, the detailed and fine coefficients are also required to form a feature vector. The size of detailed and fine coefficients approximately includes the 8 subband of 21×10 and 16 subband of 22×18. In order to compute the final vector from the combination of the different coefficients, the coefficients have been given weight in the final vector as per their respective proportion in the total energy.

It becomes computationally expensive to work with such large feature vectors. Moreover, the joint relationship of the edges in the facial expression leads to a coherent relationship among them. Henceforth, GSP has been used here to capture that relationship by defining the graph with suitable weights to strengthen that relationship as well as to reduce the size of the feature vector. The proposed algorithm is given as follows:

And, fig. 4.2 illustrates the block diagram of the proposed FER method

Algorithm 1: Algorithm for the proposed GSP approach with CT

1 Load the testing facial image and convert it into the grayscale image;
2 Normalize the size of image to 256×256 in the preprocessing;
3 Extract the mouth and the eye pair from the image using the Viola-Jones algorithm ;
4 Compute the CT features for the mouth and the eye pair at the particular scale and angle;
5 Construct the feature vector $\mathbf{x_i}$ by concatenating the features of the mouth and the eye pair;
6 Find out the interrelationship among the computed CT feature vectors using the weight in (3.1);
7 Construct the graph signal with the feature vector $\mathbf{x_i}$ as the vertex and the Laplacian $\mathbf{L}$ matrix is calculated;
8 Compute the reduced dimension feature vector $\mathbf{y_i}$ by solving the generalized eigenvector problem, as per (3.2) and (3.3);
9 Feed the reduced vector $\mathbf{y_i}$ to the trained SVM classifier;
10 Recognize the classified expression based on the SVM classifier;

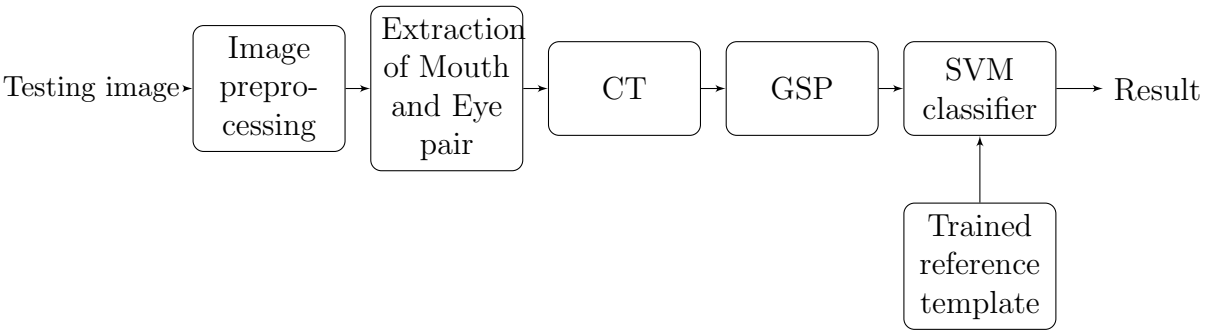

Figure 4.2: *The block diagram of the proposed GSP-CT method*

4.4 Experimental Performance

JAFFE database[66] is used to study the performance of our method in view of the available state-of-the-art for the assessment. There are 213 images of 10 subjects with 7 expressions viz. "anger", "disgust", "fear", "happy", "neutral", "sad" and "surprise". Every subject has 3-4 samples for each of the expression. For our experiments, training images are taken using leave one out approach i.e for each expression one expression from the samples is left for testing and the remaining are taken for the training set. The images are resized into the size of 256×256. The face detection is done through the Viola-Jones Face Detection algorithm[87]. The active face patches including mouth and eye pair are extracted from the faces.

Table 4.1: The effect of curvelet parameters on the length of the feature vector and the accuracy of FER

Scale	Angle	Length of the feature vector	Accuracy of FER(%)
2	8	1634	87.87
2	16	882	87.87
3	8	792	81.8
3	16	408	81.8
4	8	792	74.07
4	16	408	77.18

The extracted portions are resized to 32× 32 pixels. The curvelet transform for the mouth and eye pair is carried out separately. The resulted curvelet coefficients of mouth and eye pair are concatenated together to form the feature vector. The curvelet coefficients have been calculated for the different scales and angles. SVM classifier is used to carry out the recognition task. Table 4.1 demonstrates the effect of curvelet parameters including scale and angle on the length of the feature vector and the accuracy of the recognition.

For further reducing the size of the feature vector, the graph is created with the intial feature vector $\mathbf{x}_i$ as the vertex and the weight as defined in (3.1). The value of 't' is selected as 0.66 in (3.1). The eigenvectors are computed from the graph Laplacian matrix and the projection of the feature vector is evaluated to find the final feature vector $\mathbf{y}_i$ as per (3.3). Table 4.2 shows the comparison of the individual facial expression of the proposed method with [88] based on the LIBSVM.

Table 4.2: Comparison of the individual FER between CT and GSP-CT

Expression	Accuracy of CT method [88] (%)	Accuracy of GSP-CT method (%)
Anger	96.67	100
Disgust	89.66	100
Fear	78.13	88.89
Happy	83.87	100
Neutral	96.67	100
Sad	77.42	80
Surprise	86.67	80
Average	87.01	92.42

In Table 4.3, the proposed method has been compared to the other methods listed in [88] and [83] based on the JAFFE database and curvelet transform with respect to their feature dimension and the average expression recognition rate. It is clearly evident here that our proposed method is well matched to the other methods in view of the average recognition rate. Meanwhile, the feature dimension of our proposed method is least. However, since some of the references lack the index of the feature dimension directly, this comparison is carried out on the qualitative basis.

Table 4.3: Comparison of Curvelet with combined Curvelet GSP approach

Method	Feature dimension	Overall recognition rate (%)
Curvelet + SVM	792	81.8
Curvelet + SVM [88]	1850	87.01
Curvelet Experiment 1 [83]	Not available	88.32
Curvelet Experiment 2 [83]	Not available	94.74
Curvelet + PSO-SVM [88]	1850	94.94
Proposed method	48	92.5

4.4.1 Discussion of GSP-Curvelet method

CT is found to be better than wavelet in detecting edges and curves. It has been used in the face recognition, where coarse coefficents (of lower frequency) are used to form the feature vector. Moreover, in FER, the detail and the fine coefficients

(of high frequency) in addition to the coarse coefficients convey the information about the facial expression. Hence, the dimension of the feature vector for the FER increases significantly. That's why the use of CT is far limited in the facial expression recognition with respect to the facial recognition. Using the concept of GSP, the nearer neighboring points are bring closed together while the faraway points are distanced away. That rearrangement of the feature vectors based on the (3.1) preserve the neighborhood structure and sparsely represent the cluster of the feature vectors. Such sparse representation improves the classification of the facial expressions.

Seeing the utility of GSP in reducing the feature vector with improved accuracy of FER, we focus upon the FRFT, which is a generalized family of transforms. As the Fourier Transform (FT) is a special case of FRFT, it is clearly expected to provide better results compared to FT. We now apply the GSP on the FRFT based feature vector in the next section.

4.5 GSP approach with FRFT

FRFT is a general form of Fourier transform containing both time and frequency information [89]. In the Fourier transform, there is a rotation to the frequency axis while in the FRFT, a rotation of signal is performed on any angle. As the order 't' is changed from 0 to 1, all the features resulting from the different ratio of the time domain to the frequency domain are observed. [90] used the phase information of two dimensional FRFT as the feature vector for facial expression. [91] considered amplitude and complex information also for the feature extraction of smile recognition. They fused the information of FRFT in different orders but did not take into account the effective dimension reduction methods.

4.6 Proposed method of combining FRFT and GSP

Initially, each image is converted into its gray scale image. The face is localized from the images through the Viola-Jones Face Detection algorithm[87].Then, it is

normalized to the size of $M \times P$. As the mouth and the eye regions of the face carry the salient information of the facial expression, these regions are extracted using the Viola-Jones algorithm again. Further, these regions are resized to the $N_1 \times N_2$ and $N_3 \times N_4$ respectively. Moreover, it reduces the amount of input data for processing.

Then, the FRFT features of the mouth region are computed. Since the 2D-FRFT kernel depends upon the rotation angle 'α' and 'β' , the values of 'a_1' and 'a_2' are changed to cover the features at different angles of rotation. We have selected the equal values of $a_1 = a_2 = a$ to simplify the analysis and observe the impact of rotation angle directly on the feature vector performance. Due to the symmetry characteristics in the FRFT domain, it is sufficient to observe when the 'a' changes from 0 to 2.

As the FRFT of the image f(s,t) is $F_{a_1,a_2}(u,v)$, it can be further divided into amplitude and phase information as follows:

$$F_{a_1,a_2}(u,v) = |F_{a_1,a_2}(u,v)| \cdot P_{a_1,a_2}(u,v) \\ = M_{a_1,a_2}(u,v) \cdot P_{a_1,a_2}(u,v) \quad (4.1)$$

where $M_{a_1,a_2}(u,v)$ and $P_{a_1,a_2}(u,v)$ represent the amplitude and phase information of the mouth respectively. In [91], the amplitude based feature vector has the higher recognition rate than the phase and the complex based feature vectors. In addition, as the amplitude information depends upon the spatial distribution of the gray pixel values of the image, it is relatively smooth and steady to mark the variation in that feature vector. As a result, the feature vector for the $i-th$ image's mouth is:

$$\mathbf{x_i}^{\mathrm{M}} = [m_1, m_2, ..., m_{N_1 N_2}] \quad (4.2)$$

Similarly, the amplitude information based feature vector for eye region is computed as:

$$\mathbf{x_i}^{\mathrm{E}} = [e_1, e_2, ..., e_{N_3 N_4}] \quad (4.3)$$

Both the feature vectors of the mouth and the eyes are concatenated to form the

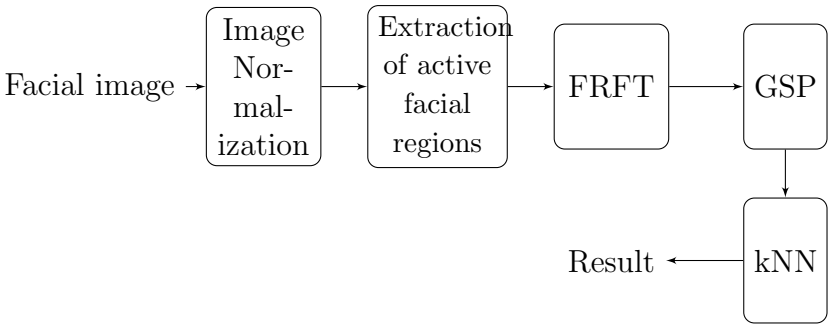

Figure 4.3: *Diagram of the proposed GSP-FRFT method*

resultant feature vector 'C_i' of the facial image as follows:

$$\mathbf{x}_i = [m_1, m_2, .., m_{N_1 N_2}, e_1, e_2, ...e_{N_3 N_4}] \tag{4.4}$$

The length of the feature vector '$\mathbf{x}_i$' thus becomes equal to $N_1N_2 + N_3N_4$. Then, based upon our holistic GSP approach of two stages, second stage is followed. Here, in order to reduce the dimension of the feature vector '$\mathbf{x}_i$', the GSP approach is used to find out the relationship among these FRFT computed feature vectors as per the weights defined in (3.1). Then, the graph signal is constructed with $\mathbf{X}$ as the graph signal with '$\mathbf{x}_i$' as the signal component lying on the $i-th$ vertex and the Laplacian matrix 'L' is calculated as: $\boldsymbol{L}=\boldsymbol{D}-\boldsymbol{W}$. Then, the eigenvalues (let 'l' be the total eigenvalues) and the corresponding eigenvectors $\mathbf{b}_0$,...,$\mathbf{b}_{l-1}$, are computed as follows:

$$\mathbf{X}L\mathbf{X}^\intercal\mathbf{b} = \lambda\mathbf{X}D\mathbf{X}^\intercal\mathbf{b} \tag{4.5}$$

The final reduced feature vector $\mathbf{y}_i$ of the $i-th$ facial image, is given below:

$$\mathbf{y}_i = B^\intercal\mathbf{x}_i, B = (\mathbf{b}_0, ..., \mathbf{b}_{l-1}) \tag{4.6}$$

The reduced feature vector $\mathbf{y}_i$ is passed to the trained kNN classifier and the classified expression is obtained from the classifier. The flowchart of the whole procedure is shown in fig 4.3.

Table 4.4: The overall comparison of FRFT and FRFT+ GSP

Transform Order 'a'	FRFT based accuracy of FER (%)	GSP based accuracy of FER (%)	GSP based feature dimension
0.1	94.04	94.04	68
0.3	89.28	91.66	68
0.5	94.04	95.23	69
0.7	92.85	94.04	69
0.9	92.85	96.63	67

4.7 Experimental Performance

The CK+ database[67] is used to study the performance of our method in view of the available state-of-the-art for the assessment. There are 593 images of 123 subjects with 7 expressions viz. "anger","disgust","fear","happy","contempt","sad" and "surprise". Out of the total image sequences, 274 sequences have been randomly selected. The last sequence with peak intensity from each expression was selected. For our experiments, training images (190) and testing images (84) are separated using leave-one-out-approach. Using $M = P = 256$, the images are normalized and the optimum size of the mouth and the eye regions for covering the salient parts of the expression are found to be 16×32 and 32×32 respectively. Thus , the length of the FRFT feature vector of each facial image becomes 1536. As the value of the transform order 'a' (on which the rotation angle depends) is changed by step of 0.2, the effect of the FRFT feature vector on the accuracy of the facial expression recognition rate is observed. The values above 0.9 have not been shown as the accuracy values are found to be repeated because of the symmetry around a=1. Thereafter, using the GSP approach, the performance of the GSP based feature vector is noted along with the new dimension of the feature vector, as shown in Table 4.4. The results shown in Table 4.4 show that not only the dimension of the FRFT feature vector has been reduced from 1536 to 67 with the application of GSP but also the accuracy of the FER has been improved for the every value of 'a'. The performance enhancement associated with the use of GSP tools is attributed to the preserving of local neighborhood information in the graph based representation. Hence, the resultant clustering provides the better

classification of the facial expressions. The confusion matrix using the proposed work on the CK+ database is shown in Table 4.5. Next, the proposed method is compared with the present state-of-the-art based on the FRFT in Table 4.6.

Table 4.5: Confusion matrix using FRFT-GSP on CK+ database (%)

	Anger	Disgust	Fear	Happy	Sad	Surprise
Anger	100	0	0	0	0	0
Disgust	0	100	0	0	0	0
Fear	5.88	0	94.12	0	0	0
Happy	0	0	0	100	0	0
Sad	0	0	0	0	100	0
Surprise	0	14.3	0	0	0	85.7

Table 4.6: The performance of different methods and proposed GSP-FRFT method

Method	FER accuracy (%)	Dimension of the feature vector
FRFT+ DM-CCA	89.6	N.A.
FRFT+FLDA	89.2	N.A.
FRFT	94.04	1536
[92]	92.5	90000
Proposed method	96.42	67

4.8 Summary

In this chapter we have presented a methodology of applying the GSP on the feature vectors of CT and the FRFT for improving the FER. Experimental performance on JAFFE and CK+ database demonstrated the effectiveness of the proposed methods of GSP-CT and GSP-FRFT respectively. On using the GSP, the neighborhood structure is preserved in the embedding which results in emphasizing the natural cluster leading to the better classification of the facial expressions. In addition, the feature vector length is also reduced substantially. However, in view of the computations requirement, we first construct the existing feature vector

and then put these feature vectors as a node lying on the vertex of the graph. As the dimension of the existing scheme's feature vectors increases, the computation to find the weight matrix of the graph structure also increases proportionately. Hence, rather than relying on the existing feature vectors, we work to find the GSP method without the use of any existing FER scheme in the coming chapter.

Chapter 5

FER using Spectral Graph Wavelet Transform

In the last two chapters, after finding out the composite schemes of GSP along with the existing FER methods of DWT, HOG, CT and FRFT, we observe that the GSP approach can be further beneficial to apply directly without using any large existing feature vector on considering the computational complexity. Moreover, we realize that the computational complexity can be decreased by minimizing the length of the graph signal component (preferably single value) lying on the vertex. Hence, we formulated a graph structure directly on the facial image itself where the pixel is considered as the vertex and the pixel intensity (single value) is used as the graph signal component on the vertex. Thereafter, we leverage the spectral graph wavelet transform (SGWT) [50] on the different type of formulated graph signals. A novel method of FER using SGWT is presented here to represent the expression patterns on the face by the graph signals. Then, the different filterbanks assigned with the different weights to their channels have been evaluated for the performance of the facial expression recognition.

In this chapter, the contribution of the proposed scheme lies in the novel application of the SGWT in the FER. Furhter, the proper filterbank and their related weights have been found out for the optimum FER from the given arrangement of the graph signals.

5.1 Introduction

The SGWT has not been satisfactorily applied in the facial expression recognition until now. We are interested in using SGWT for recognizing the facial expression by finding multi-scale information about the interactions between the interest points (lying in the active patches of the face). This multi scale representation allows us to capture the information about the active patches at different scales. Further, SGWT provides more flexibility than the classical wavelets due to the freedom of graph design.

The organization of this chapter is as follows: In section 5.2, the algorithm and the flowchart of the proposed SGWT method are given. Section 5.3 discusses the experimental performance using the proposed method. A summary of the main findings is provided in section 5.4.

5.2 GSP method using SGWT

As already explained, for the purpose of FER, most of the key information lies in the mouth and the eye regions. We refer these regions as the active patches. We require to extract the pattern of the edges present in these regions. Here, we propose the multi-scale decomposition of the graph signal in order to capture information about the edge pattern with respect to the graph structure.

For the construction of the graph of the active patches, the weights are required to be defined. Beginning with simple but the effective method, the pixel of the image region is considered as the vertex and the pixel intensity on the i^{th} vertex, x_{i}, represents the signal value at that vertex. Then, on taking the difference of the pixel intensity among the eight-point neighborhood, the weights between vertex i and vertex j, ${W^{\text{First}}}_{\text{ij}}$ and ${W^{\text{Second}}}_{\text{ij}}$, are defined as follows [50]:

$$ {W^{\text{First}}}_{\text{ij}} = \begin{cases} |x_{\text{i}} - x_{\text{j}}|, j \in N_{\text{i}} \\ 0, \text{ elsewhere} \end{cases} \tag{5.1} $$

$$W^{\text{Second}}{}_{\text{ij}} = \begin{cases} 1000 \times (|x_{\text{i}} - x_{\text{j}}|)^2, j \in N_{\text{i}} \\ 0, \text{ elsewhere} \end{cases} \tag{5.2}$$

where N_i is the 8-point neighborhood of vertex i. The rationale behind selecting the above defined weights is that the difference in the pixel intensities capture the distinctive edge patterns of the different expressions. In the $W^{\text{Second}}{}_{\text{ij}}$, the differences have been squared and positively scaled (by thousand times here) to make the weight further sensitive to the difference of the pixel intensity. As discussed in Sec. 2.3, the graph Laplacian is obtained as follows:

$$\mathbf{L}x_{\text{i}} = \sum_{v_{\text{j}}:e_{\text{ij}} \in E} W_{\text{ij}}[x_{\text{i}} - x_{\text{j}}] \tag{5.3}$$

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \tag{5.4}$$

Let the mouth region be of the size $A \times B$. Then, the total number of vertex N in the graph representation of the mouth region is equal to AB. Then, the eigenvalues of the graph Laplacian $\mathbf{L}$ and the corresponding eigenvectors are represented as $\lambda = \lambda^{\text{M}}{}_0 .... \lambda^{\text{M}}{}_{\text{AB-1}}$ and $\mathbf{U} = \mathbf{u}^{\text{M}}{}_0 .... \mathbf{u}^{\text{M}}{}_{\text{AB-1}}$ respectively. The eigenvalues signify the graph frequency and the eigenvectors are equivalent to the graph Fourier basis. Now the SGWT framework, discussed in Sec. 2.3.3, is applied after the computation of the graph frequency and the graph Fourier basis.

Given a kernel function $g : \mathbb{R}^+ \to \mathbb{R}^+$ selected as a band-pass filter, a spectral graph wavelet $\psi^{\text{M}}{}_{\text{t,n}} \in \mathbb{R}^{\text{AB}}$ at scale t localized around vertex i can be written as follows [50]:

$$\psi^{\text{M}}{}_{\text{n}}(m) = \sum_{l=0}^{AB-1} g(\lambda^{\text{M}}{}_{\text{l}}) u^{\text{M}}{}_{\text{l}}(n) u^{\text{M}}{}_{\text{l}}(m) \tag{5.5}$$

For finding the wavelet scales t_j, J values are taken as logarithmically equispaced in the range $[0, \lambda^{\text{M}}{}_{\text{max}}]$. In addition, for the optimum choice of g, we have selected the specific filters from the different type of filterbank e.g. Meyer wavelet. As demonstrated in [93], [94] regarding the utility of wavelet transform esp. Meyer wavelet for the edge detection, the Meyer wavelet is selected for the multi scale decomposition of the graph signal. The plot of the transfer function of the Mexican

Hat filterbank [95] with four filters is shown in Fig. 5.1.

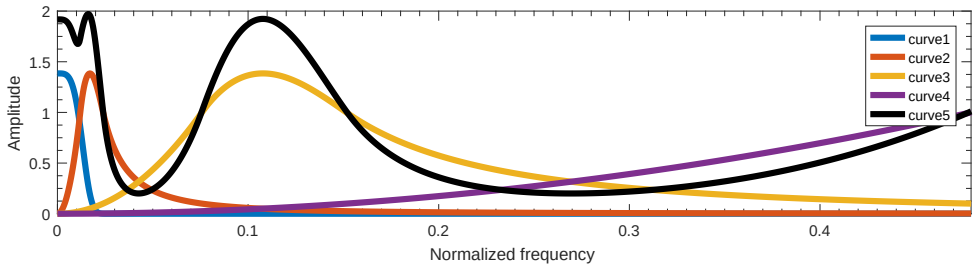

Figure 5.1: Amplitude frequency response of the Mexican hat filterbank[95]. Five curves indicate the different frequency responses of the filters of the filterbank.

Then, for considering the mouth region as a graph signal $\mathbf{x}^{\mathrm{M}}$, SGWT coefficients are obtained by the vector matrix multiplication $({\psi_{t_{\mathrm{j}}}}^{\mathrm{M}})^{\intercal}\mathbf{x}_{\mathrm{M}}$ where ${\psi_{t_{\mathrm{j}}}}^{\mathrm{M}} = [{\psi^{\mathrm{M}}}_{t_{\mathrm{j}}1}.....{\psi^{\mathrm{M}}}_{t_{\mathrm{j}}AB}]$. In order to decrease the computational complexity, approximate SGWT is used as follows [50]:

$$({\Psi^{\mathrm{M}}}_{t_{\mathrm{j}}})^{\intercal}\mathbf{x}^{\mathrm{M}} \approx \frac{1}{2}c_{\mathrm{j},0}\mathbf{x} + \sum_{k=1}^{M_{\mathrm{j}}} c_{\mathrm{j,k}}\overline{T_{\mathrm{k}}}(L^{\mathrm{M}})\mathbf{x}^{\mathrm{M}} \tag{5.6}$$

$${\Phi^{\mathrm{M}}}_{t_{\mathrm{j}}}{}^{\intercal}\mathbf{x}^{\mathrm{M}} \approx \frac{1}{2}c_{0,0}\mathbf{x} + \sum_{k=1}^{M_{\mathrm{j}}} c_{\mathrm{j,k}}\overline{T_{\mathrm{k}}}(L^{\mathrm{M}})\mathbf{x}^{\mathrm{M}} \tag{5.7}$$

where ${\Phi^{\mathrm{M}}}_{t_{\mathrm{j}}}$ corresponds to the low frequency of the graph signals. Thus, there are $J+1$ coefficients per axis for each vertex. Collectively, these coefficients represented as ${\mathbf{r}^{\mathrm{M}}}_{\mathrm{i}}$ also extract information about the localized frequencies of the graph signal. The coefficients belonging to the specific channel of the filterbank can be extracted. Let ${\mathbf{r}^{\mathrm{M}}}_{\mathrm{i,n}}$ represents the output at the '$n-th$' channel. In order to evaluate the

impact of the different channels of the graph filterbanks for a mouth based graph signal , the weight assigned to the '$n-th$' channel is α_{n}. In this way, the output coefficients for the eyes of the i^{th} face with n^{th} channel filterbank will be as follows [50]:

$$\mathbf{r}^{\mathrm{M}}{}_{\mathrm{i}} = \alpha_1 \mathbf{r}^{\mathrm{M}}{}_{\mathrm{i,1}} + \alpha_2 \mathbf{r}^{\mathrm{M}}{}_{\mathrm{i,2}} + ... + \alpha_{\mathrm{n}} \mathbf{r}^{\mathrm{M}}{}_{\mathrm{i,n}} \tag{5.8}$$

For the computation of these coefficients, we have used the Matlab version of the GSP toolbox [95]. Similiarly, SGWT coefficients of the eyes using the approximate SGWT are obtained as follows [50]:

$$(\Psi^{\mathrm{E}}{}_{\mathrm{t_j}})^{\intercal} \mathbf{x}^{\mathrm{E}} \approx \frac{1}{2} c_{\mathrm{j,0}} \mathbf{x} + \sum_{k=1}^{M_{\mathrm{j}}} c_{\mathrm{j,k}} \overline{T_{\mathrm{k}}}(L^{\mathrm{E}}) \mathbf{f}^{\mathrm{E}} \tag{5.9}$$

$$\Phi^{\mathrm{E}}{}_{\mathrm{t_j}}{}^{\intercal} \mathbf{x}^{\mathrm{E}} \approx \frac{1}{2} c_{\mathrm{0,0}} \mathbf{x} + \sum_{k=1}^{M_{\mathrm{j}}} c_{\mathrm{j,k}} \overline{T_{\mathrm{k}}}(L^{\mathrm{E}}) \mathbf{x}^{\mathrm{E}} \tag{5.10}$$

The SGWT coefficent of the eyes is collectively represented as:

$$\mathbf{r}^{\mathrm{E}}{}_{\mathrm{i}} = [\Psi^{\mathrm{E}}{}_{\mathrm{t_j}}, \Phi^{\mathrm{E}}{}_{\mathrm{t_j}}]^{\intercal} \mathbf{x}^{\mathrm{E}} \tag{5.11}$$

Let $\mathbf{r}^{\mathrm{E}}{}_{\mathrm{i,n}}$ represents the output at the '$n-th$' channel. In order to evaluate the impact of the different channels of the graph filterbanks for a eyes based graph signal, the weight assigned to the '$n-th$' channel is β_{n}. In this way, the output coefficients for the eyes of the i^{th} face with n^{th} channel filterbank will be as follows:

$$\mathbf{r}^{\mathrm{E}}{}_{\mathrm{i}} = \beta_1 \mathbf{r}^{\mathrm{E}}{}_{\mathrm{i,1}} + \beta_2 \mathbf{r}^{\mathrm{E}}{}_{\mathrm{i,2}} + ... + \beta_{\mathrm{n}} \mathbf{r}^{\mathrm{E}}{}_{\mathrm{i,n}} \tag{5.12}$$

Finally, the feature vector $\mathbf{r}_{\mathrm{i}}$ is constructed by concatenating the output coefficients $\mathbf{r}^{\mathrm{M}}{}_{\mathrm{i}}$ of the mouth and $\mathbf{r}^{\mathrm{E}}{}_{\mathrm{i}}$ of the eyes for the i^{th} face. Then, it is passed to train the classifier and get the classified output from the trained classifier.

Fig. 5.2 illustrates the flowchart of the proposed FER method and algorithm for the same is given below.

Algorithm 2: Algorithm for the FER using the SGWT

1 Load the testing facial image and convert it into the grayscale image;
2 Normalize the size of the image to $A_0 \times B_0$ in the preprocessing;
3 Extract the mouth and the eye pair from the image using the Viola-Jones algorithm;
4 The regions of the mouth and the eyes are represented in the form of the graph signal by defining weight between the vertices as defined in (5.1);
5 On constructing the graph signal for these regions, filterbanks are selected to design specific wavelets. Then, the filterbank coefficients for the mouth and the eyes are computed;
6 Using the approximate SGWT, the Chebyshev polynomial coefficients are found from the filterbank coefficients and the SGWT coefficients of the mouth and the eyes region are obtained as $\mathbf{r}^{\mathrm{M}}{}_{\mathrm{i}}$ and $\mathbf{r}^{\mathrm{E}}{}_{\mathrm{i}}$;
7 The SGWT coefficients of the mouth and the eyes region are concatenated to form the final feature vector $\mathbf{r}_{\mathrm{i}}$ for the i^{th} facial image;
8 Feed the feature vector $\mathbf{r}_{\mathrm{i}}$ to the trained kNN/SVM classifier;
9 Classify the fed feature vector into the category of the facial expressions;

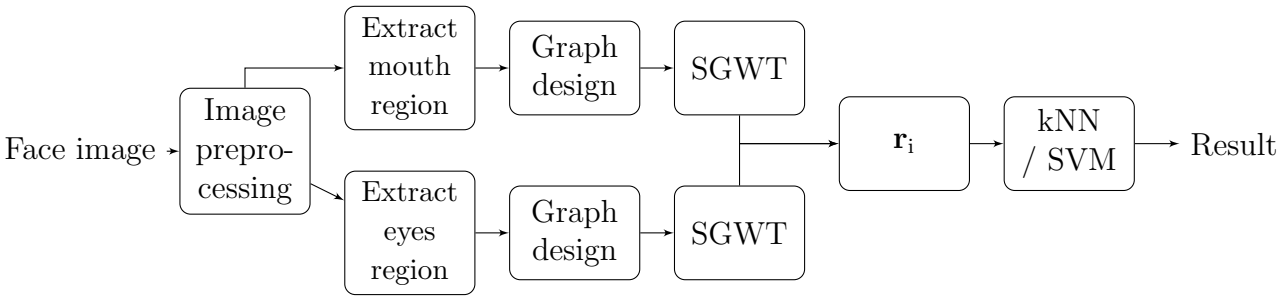

Figure 5.2: *The block diagram of the SGWT FER method*

5.3 Experimental Performance

Two standard facial databases have been used to study the performance of our method. The two databases are as follows: JAFFE database [66] and Extended Cohn-Kanade (CK+) database [67]. In JAFFE database, there are 213 images of 10 subjects with 7 expressions viz. "anger", "disgust", "fear", "happy", "neutral", "sad" and "surprise". CK+ database includes around 593 image sequences (posed expressions) of around 123 subjects, and consists of same expressions like JAFFE except "contempt" in place of the "neutral".

5.3.1 Experimental Data

(a) CK+ database: The 272 images have been selected from it. For each expression of a subject, the last two images from each sequence were selected as peak expression images. For the training, total samples around 190 have been randomly selected . The remaining samples were used for testing using leave-one-out-approach.

(b)JAFFE database: Unlike in the CK+ database, all the seven expressions (including neutral) have been considered to provide the sufficient number of images for the training and the testing. Every subject has 3-4 images for each of the expression. The training images (N=147) are selected using leave- one-out approach, i.e., for each expression one expression from the samples for every subject is left for testing and the remaining are taken for the training set.

As discussed in Section 5.2, the images are resized into the size of 256$\times$ 256 ($A_0 = B_0 = 256$). The face from the given image has been detected by the Viola-Jones Face Detection algorithm[87]. Further, the active face patches including the mouth and eye pair are extracted from the face using the Viola-Jones algorithm again. Then, the graph signals have been constructed with the weights mentioned in (5.1). Since the flexibility of choosing the weight allows to build the graph signal as per the requirement of the problem, we have selected the weight appropriately for both the datasets. For the CK+ dataset, the weight, as given in (5.1) and (5.2) is selected for building the graph signal of the eyes and the mouth respectively. However, in the case of the JAFFE dataset, the weight, as given in (5.1), has been selected for building the graph signal of the mouth and the eyes in view of the optimum performance. After defining the graph signal for the eyes and the mouth, the coefficients for all the channel filters contained in the filterbank (employing the Chebychev polynomial approximation algorithm [50]) are computed. Then, by assigning different weights to the channel filters of the filterbank, the outputs corresponding to the filterbank are evaluated for the mouth and the eyes individually. From combining their outputs, the feature vector as $\mathbf{r}_\mathrm{i}$ is obtained for the i^{th} facial image. This feature vector is then used to train the kNN classifier. The type of filterbank is selected and the performance of each filterbank is measured in the form of the accuracy of the classified expression. For the JAFFE dataset,

Table 5.1: Impact of the different weights on the **two** channel filterbanks on the accuracy for the CK+ and JAFFE datasets

	Mouth		Eyes			
Filterbank	α_1	α_2	β_1	β_2	CK+ (%)	JAFFE (%)
Abspline	0.5	0.5	0.5	0.5	96.15	72.72
Abspline	0.9	0.1	0.9	0.1	96.15	77.27
Abspline	0.1	0.9	0.1	0.9	96.15	66.66
Itersine	0.5	0.5	0.5	0.5	92.68	68.18
Itersine	0.9	0.1	0.9	0.1	92.68	78.78
Itersine	0.1	0.9	0.1	0.9	92.68	65.15
Meyer	0.5	0.5	0.5	0.5	92.68	66.66
Meyer	0.9	0.1	0.9	0.1	90.24	72.72
Meyer	0.1	0.9	0.1	0.9	92.68	71.21
Mexican hat	0.5	0.5	0.5	0.5	92.68	72.72
Mexican hat	0.9	0.1	0.9	0.1	90.24	80.30
Mexican hat	0.1	0.9	0.1	0.9	92.68	69.69

the SVM classifier was found to be better than the kNN classifier. Thus, the SVM classifier has been preferred for JAFFE. In the filterbank, we have the choice of selecting the specific channel output. Thus, initially beginning from the filterbank of two filters, the different channels of the filterbank have been assigned different weights. We have started with equal weight and the obtained performance is shown in Table 5.1.

In order to evaluate the impact of three filters in the filter-bank, the outputs at the three channels have been observed with different weights for the mouth and the eyes based graph signal. The filterbanks with the output at the three channel with their corresponding weights have been shown in Table 5.2 for both the graph signals along with the obtained accuracy for CK+ and JAFFE databases.

In the Table 5.1 and Table 5.2, the non-normalized graph Laplacian has been used. Moreover, the normalized graph Laplacian may also be used here. By considering the normalized graph Laplacian, the performance of the two channel filterbanks and the three channel filterbanks have been shown in Table 5.3 and Table 5.4 respectively. Here, in comparison to the performance of the filterbanks using non-normalized graph Laplacian, the performance of the filterbanks using normalized graph Laplacian have been improved for both the databases.

From the Table 5.3 and Table 5.4, it is observed that the performance of the Abspline filterbank is comparatively better than the other filterbanks for CK+ as

Table 5.2: Impact of the different weights on the **three** channel filterbanks on the accuracy for the CK+ and JAFFE datasets

	Mouth		Eyes					
Filterbank	α_1	α_2	α_3	β_1	β_2	β_3	CK+ (%)	JAFFE (%)
Abspline	0.3	0.4	0.3	0.3	0.4	0.3	91.46	80.3
Abspline	0.1	0.9	0	0.1	0.9	0	93.9	89.39
Abspline	0	1	0	0	1	0	92.68	90.9
Itersine	0.3	0.4	0.3	0.3	0.4	0.3	92.68	74.24
Itersine	0.1	0.9	0	0.1	0.9	0	93.9	78.78
Itersine	0	1	0	0	1	0	90.24	74.24
Meyer	0.3	0.4	0.3	0.3	0.4	0.3	92.68	72.72
Meyer	0.1	0.9	0	0.1	0.9	0	90.24	71.21
Meyer	0	1	0	0	1	0	81.7	66.67
Mexican hat	0.3	0.4	0.3	0.3	0.4	0.3	92.68	69.69
Mexican hat	0.1	0.9	0	0.1	0.9	0	91.46	74.24
Mexican hat	0	1	0	0	1	0	93.9	77.27

Table 5.3: Impact of the different weights on the **two** channel filterbanks on the accuracy for the CK+ and JAFFE datasets (using Normalized graph Laplacian)

	Mouth		Eyes			
Filterbank	α_1	α_2	β_1	β_2	CK+ (%)	JAFFE (%)
Abspline	0.5	0.5	0.5	0.5	95.12	93.93
Abspline	0.9	0.1	0.9	0.1	96.34	**94.28**
Abspline	0.1	0.9	0.1	0.9	91.46	92.42
Itersine	0.5	0.5	0.5	0.5	93.9	78.78
Itersine	0.9	0.1	0.9	0.1	93.9	77.27
Itersine	0.1	0.9	0.1	0.9	90.24	86.36
Meyer	0.5	0.5	0.5	0.5	92.68	68.18
Meyer	0.9	0.1	0.9	0.1	93.9	72.72
Meyer	0.1	0.9	0.1	0.9	91.46	83.33
Mexican hat	0.5	0.5	0.5	0.5	93.9	90.91
Mexican hat	0.9	0.1	0.9	0.1	96.34	**94.28**
Mexican hat	0.1	0.9	0.1	0.9	92.68	89.39

Table 5.4: Impact of the different weights on the **three** channel filterbanks on the accuracy for the CK+ and JAFFE datasets (using Normalized graph Laplacian)

	Mouth			Eyes				
Filterbank	α_1	α_2	α_3	β_1	β_2	β_3	CK+ (%)	JAFFE (%)
Abspline	0.3	0.4	0.3	0.3	0.4	0.3	**96.93**	89.4
Abspline	0.1	0.9	0	0.1	0.9	0	92.68	90.91
Abspline	0	1	0	0	1	0	93.9	92.42
Itersine	0.3	0.4	0.3	0.3	0.4	0.3	92.68	81.81
Itersine	0.1	0.9	0	0.1	0.9	0	95.12	89.4
Itersine	0	1	0	0	1	0	96.34	89.4
Meyer	0.3	0.4	0.3	0.3	0.4	0.3	93.9	75.75
Meyer	0.1	0.9	0	0.1	0.9	0	91.46	86.36
Meyer	0	1	0	0	1	0	90.24	81.81
Mexican hat	0.3	0.4	0.3	0.3	0.4	0.3	95.12	78.8
Mexican hat	0.1	0.9	0	0.1	0.9	0	96.34	89.4
Mexican hat	0	1	0	0	1	0	95.12	90.91

well as JAFFE databases in both the cases of the two and three channels. In case of CK+ dataset, the optimum performance is achieved with the three channels of the Abspline filterbank with different weights as shown in Table 5.1. While, for JAFFE dataset, the best result is obatined with the two channels of the Abspline filterbank and the Mexican hat filterbank. It is interesting to note that overall, the Abspline filterbank captures the significant characteristics in both the set of two and three channels for both the databases. Table 5.5 and Table 5.6 shows the confusion matrix for the CK+ dataset and the JAFFE dataset of the proposed method respectively.

Table 5.5: Confusion matrix for Facial Expression Recognition using the proposed method with kNN for the CK+ dataset(Average Recognition Rate=96.93%)

	Anger	Disgust	Fear	Happy	Sad	Surprise
Anger	93.75	0	0	0	6.25	0
Disgust	0	100	0	0	0	0
Fear	0	0	100	0	0	0
Happy	0	0	0	100	0	0
Sad	0	0	0	0	100	0
Surprise	9.1	0	0	0	0	90.9

Table 5.6: Confusion matrix for Facial Expression Recognition using the proposed method with linear SVM for the JAFFE dataset (Average Recognition Rate=94.28%)

	Anger	Disgust	Fear	Happy	Neutral	Sad	Surprise
Anger	90	10	0	0	0	0	0
Disgust	0	90	0	0	0	10	0
Fear	0	0	100	0	0	0	0
Happy	0	0	0	100	0	0	0
Neutral	0	0	0	0	100	0	0
Sad	10	0	0	0	0	90	0
Surprise	0	0	0	0	10	0	90

The results of the proposed method are compared with the present results of the existing approaches in Table 5.7. These approaches are selected because of using the similar testing strategy (leave-one-subject-out) and the similar databases. Moreover, the classifier in most of the selected approaches is restricted to be either from kNN or SVM only. The parentheses with the recognition rates indicate the number of testing expressions.

Table 5.7: Comparison of the proposed method with the present state-of-the-art methods

Method	Facial features	JAFFE Recognition rate(in %)	CK+ Recognition rate(in %)
Our method	SGWT	94.28 (7)	96.93(6)
[96], 2015	HDBF+ local FDA	-	91.3 (7)
[71], 2015	Histogram of LBP	91.8 (6)	94.1 (6)
[63], 2015	HOG+GA	87.82 (6)	95.64 (6)
[68], 2012	Patch based Gabor	92.93 (6)	94.4 (6)
[73], 2017	HOG-DWT+GSP	92.1 (6)	-
[7], 2009	Boosted-LBP	81.0 (7)	95.1 (6)
[97], 2007	Motion-units	-	93 (4)
[98], 2006	Pixel Intensity	-	90.7 (6)

As shown in Table 5.7, the proposed method outperforms all the other approaches in case of the JAFFE dataset and for CK+ dataset, the performance of our method is better than six of the eight methods. Overall, the performance of the proposed method is reasonably well on considering both the databases together.

5.4 Summary

This chapter proposes a novel method for the improved facial expression recognition using the SGWT framework. In this method, the graph signal used for the representation of the facial expression regions has the scalar value at the vertex and thus, weights are easily computed. Thereafter,the application of the different filterbanks assigning the different weights to the channels of the filterbank, the accuracy of the recognition has been analyzed. We find out that the Abspline filterbank esp. the second channel provides the best result with the two and three channel filterbanks. The obtained results on both the databases are significantly better than the present state-of-the-art methods. However, the dimension of the feature vector is around 1550. Since we have not addressed the dimension of the feature vector in the GSP independent schemes, we need to focus on the reduction of the dimension of the feature vector, which will be investigated in the next chapter.

Chapter 6

Dimensionality reduction of the feature vector and evaluation of different graph structures in FER using GFT

In the previous chapter, we notice that as the direct formulation of the graph structure from the facial image reduces the computational complexity to a large extent. But must still address the dimension of the graph feature vector in such cases. One option may be to treat each graph feature vector as the graph signal component but it will lead to an increase in computational complexity like the former composite schemes of GSP. Then we realize that by utilizing the GFT, the feature vector of the facial image can be computed. A novel method for the facial expression recognition is proposed using the GFT. In addition, the dimension of the feature vector is reduced by extracting the selective frequency components. The interesting observation is that the few eigenvectors corresponding to lower frequencies provide comparative accuracy in classification as using all the frequencies. Here, the spectral analysis of the facial graph signals suggests that the interrelationship of the graph signals is better captured in the lower frequencies. Next, the different graph structures are evaluated to find out the best results of GSP approach for the FER.

In this chapter, the contribution lies in proposing the scheme based on the

GFT which has not been probably appled in the context of FER. Some interesting observations have also been given about the role of the graph frequencies esp. lower order in the reduction of the feature vector for the FER.

6.1 Introduction

We propose to apply the GSP theory [17], particularly the GFT for finding out the feature vector and further investigate the selection of the different eigenvectors for the dimensionality reduction of the feature vector. Using the GSP, the activity recognition and the classification of neuroimaging data have been done in [99] and [16] respectively. The dimension of the brain imaging data based on the graph based filtering algorithm (GBF) has been reduced in [15] . While these studies provide promising results of using GSP on the dimensionality reduction and classification, they lack the analysis of finding the different combination of eigenvectors from the same graph 'pointsets'.

In this chapter, we analyze the different combination of eigenvectors for the facial graph signals and their effect on the accuracy as well as the dimensionality reduction in the facial expression recognition. The rest of the chapter is organized as follows. Section 6.2 introduces our proposed GFT- based method for the dimensionality reduction. In section 6.3, experimental performance is discussed. In next sections, the discussion regarding the methods to build the graphs are followed.

6.2 Proposed method using GFT

Let $\mathbf{X} \in \mathbb{R}^{\mathrm{M} \times \mathrm{N}} = [\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_{\mathrm{M}}]^\intercal$ be the matrix of a graph signal. The row, $\mathbf{x}_{\mathrm{i}} \in \mathbb{R}^{\mathrm{M}}$, (or column $\mathbf{c}_{\mathrm{i}} \in \mathbb{R}^{\mathrm{N}}$), of the matrix $\mathbf{X}$ represents the i-th component of the graph signal $\mathbf{X}$ lying on vertex v_{i} connected by the edges E=$\{e_{\mathrm{ij}}$:v_{i},$v_{\mathrm{j}} \in V\}$. Since most facial expressions are distinguishable using the information lying in the region of the mouth and the eye pair together, the edge information is extracted from these regions. This requires cropping of the the mouth and the eye pair in the face images. To that end, we have used the Viola-Jones algorithm [87]. In order to extract the edges information from the regions for the mouth and eyes of the i^{th} facial image, the graph signals $\mathbf{X}_{\mathrm{i}}{}^{\mathrm{M}}$ and $\mathbf{X}_{\mathrm{i}}{}^{\mathrm{E}}$ are constructed.Then, for

capturing the local neighborhood structure efficiently in the graph signals, the nearest neighbor based graph has been selected. Then, the rows $\mathbf{x}_{\mathrm{i}}{}^{\mathrm{M}}$ and $\mathbf{x}_{\mathrm{i}}{}^{\mathrm{E}}$ (or the column $\mathbf{c}_{\mathrm{i}}{}^{\mathrm{M}}$ and $\mathbf{c}_{\mathrm{i}}{}^{\mathrm{E}}$) of the $\mathbf{X}_{\mathrm{i}}{}^{\mathrm{M}}$ and the $\mathbf{X}_{\mathrm{i}}{}^{\mathrm{E}}$ are considered as the components of the mouth region based graph signal and the eyes region based graph signal respectively lying on their vertex v_{i}. The weights of the edges connecting the vertices of the nearest neighborhood graph [19] are selected as:

$$W_{\mathrm{ij}} = \begin{cases} exp(-\frac{\|\mathbf{x}_{\mathrm{i}}-\mathbf{x}_{\mathrm{j}}\|^2}{t}), \text{ if } \mathbf{x}_{\mathrm{j}} \in N_{\mathrm{i}} \\ 0, \text{ elsewhere} \end{cases} \tag{6.1}$$

where N_{i} represents the k nearest neighborhood of $\mathbf{x}_{\mathrm{i}}$.

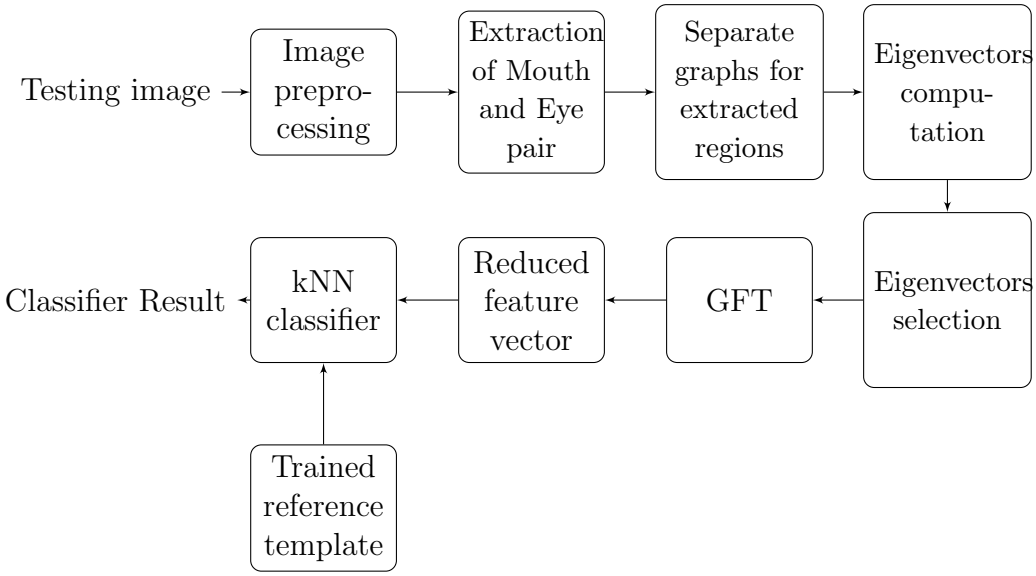

Figure 6.1: *The block diagram of GFT method for FER*

Seeing the better results in last chapter for the filterbanks, the normalized Laplacian has been used in place of the non-normalized Laplacian. Then the GFT of the graph signals $\mathbf{X}_{\mathrm{i}}{}^{\mathrm{M}}$ and $\mathbf{X}_{\mathrm{i}}{}^{\mathrm{E}}$ are computed as $\hat{\mathbf{X}}_{\mathrm{i}}{}^{\mathrm{M}}$ and $\hat{\mathbf{X}}_{\mathrm{i}}{}^{\mathrm{E}}$. By changing them into the corresponding column vectors, they are concatenated to form the initial feature vector $\hat{\mathbf{x}}_{\mathrm{G,i}}$ of the $i-th$ facial image. In order to reduce the dimension of the initial feature vector $\hat{\mathbf{x}}_{\mathrm{G,i}}$, the different eigenvectors (corresponding to different frequencies λ) are selected and the projection of the graph signals $\mathbf{X}_{\mathrm{i}}{}^{\mathrm{M}}$ and $\mathbf{X}_{\mathrm{i}}{}^{\mathrm{E}}$ are taken in the direction of these selected eigenvectors to find the new reduced feature vector as $\hat{\mathbf{Y}}_{\mathrm{i}}{}^{\mathrm{M}}$ and $\hat{\mathbf{Y}}_{\mathrm{i}}{}^{\mathrm{E}}$ respectively. Thereafter, both the new reduced feature vectors are concatenated to form the new reduced final feature vector $\hat{\mathbf{y}}_{\mathrm{G,i}}$

for the i-th facial image. At last, the kNN /SVM is trained and tested to classify each facial expression using the feature vector $\hat{\mathbf{y}}_{\mathrm{G,i}}$. The proposed GFT based algorithm is illustrated in Fig. 6.1.

6.3 Experimental Performance

In the preprocessing step, the images are normalized into the size of 256×256. The face is detected from the image through the Viola-Jones Face Detection algorithm[87]. Further, the active face regions including the mouth and eye pair are extracted from the faces using the same algorithm. The size of the extracted mouth and the eye regions on the basis of the optimum performance have been fixed to 16×32 and 32×32 respectively. Among the kNN, the value of k equal to 7 is found to be optimum and the weight as given in (6.1) was used to build the graphs. The value of 't' is chosen as 1. The GSP operations were implemented using the Matlab version of the GSP toolbox [95]. First, for the construction of the graphs, rows along the given pointsets of the extracted regions for the mouth as well as the eyes are considered as the vertices. In Table 6.1, the different set of eigenvectors (in an increasing order sequence associated with frequencies) for the mouth and the eyes have been considered to find the GFT and their impact on the dimension of the feature vector $\hat{\mathbf{y}}_{\mathrm{G,i}}$ and the accuracy are shown below:

Table 6.1: Impact of the GFT (Set I- Row as the vertices for the mouth as well as the eye regions) with the different set of eigenvectors for the CK+ and JAFFE datasets

Starting eigen vector of mouth	Last eigen vector of mouth	Starting eigen vector of eyes	Starting eigen vector of eyes	Dimension	Recognition rate CK+(in %)	Recognition rate JAFFE (in %)
1	16	1	32	1536	90.24	75.75
1	4	1	4	192	81.7	72.72
1	6	1	6	384	82.92	71.21
1	8	1	8	512	**86.58**	**77.27**
13	16	29	32	256	57.31	24.24
11	16	27	32	384	58.53	19.69

Later, the columns along the given pointsets are considered as the vertices of both the graph for the mouth and the eye regions. In Table 6.2, the different set of eigenvectors (in their increasing order sequence associated with frequencies) for the mouth and the eyes have been considered to find the GFT and their effect on the dimension of the feature vector $\hat{\mathbf{y}}_{G,i}$ and the accuracy are shown below:

Table 6.2: Impact of the GFT (Set II- Column as the vertices for the mouth as well as the eye regions) with the different set of eigenvectors for the CK+ and JAFFE datasets

Starting eigen vector of mouth	Last eigen vector of mouth	Starting eigen vector of eyes	Starting eigen vector of eyes	Dimension	Recognition rate CK+ (in %)	Recognition rate JAFFE (in %)
1	32	1	32	1536	93.9	69.69
1	4	1	4	192	91.46	62.12
1	6	1	6	288	89.02	68.18
1	8	1	8	384	**92.68**	**69.69**
29	32	29	32	192	39.02	16.66
27	32	27	32	288	39.02	22.72

Now, the rows along the given pointsets are considered as the vertices of the graph for the mouth while the column pattern has been selected for the eye region. Thereafter, the pattern along the graph for the mouth and the eyes have been reversed i.e. the column for the mouth while the rows for the eyes. The impact of the selected pattern of the eigenvectors on the dimension of the feature vector $\hat{\mathbf{y}}_{G,i}$ and the accuracy are shown below in Table 6.3 and Table 6.4 respectively.

Table 6.3: Impact of the GFT (Set III- Row as the vertices for the mouth and column for the eye regions) with the different set of eigenvectors for the CK+ and JAFFE datasets

Starting eigen vector of mouth	Last eigen vector of mouth	Starting eigen vector of eyes	Starting eigen vector of eyes	Dimension	Recognition rate CK+ (in %)	Recognition rate JAFFE (in %)
1	16	1	32	1536	86.58	66.67
1	4	1	4	192	78.04	**66.67**
1	6	1	6	384	**89.02**	63.63
1	8	1	8	512	86.58	63.63
13	16	29	32	256	52.43	24.24
11	16	27	32	384	50	25.75

Table 6.4: Impact of the GFT (Set IV- Column as the vertices for the mouth and row for the eye regions) with the different set of eigenvectors for the CK+ and JAFFE datasets

Starting eigen vector of mouth	Last eigen vector of mouth	Starting eigen vector of eyes	Starting eigen vector of eyes	Dimension	Recognition rate CK+ (in %)	Recognition rate JAFFE (in %)
1	32	1	32	1536	93.9	68.18
1	4	1	4	192	87.8	62.12
1	6	1	6	288	90.24	68.18
1	8	1	8	384	**92.68**	**68.18**
29	32	29	32	192	48.78	18.18
27	32	27	32	288	59.75	13.63

6.3.1 Discussion

The results in the previous tables reveal the following important findings:

i) The eigenvectors corresponding to the lower frequencies contribute significantly for the classification in both the databases in comparison to the higher frequencies. It suggests that the lower frequencies better capture the interrelationship and the structural pattern of the graph signals. That's why using only the lower frequency based eigenvectors yields better classification accuracy of classification than using all the eigenvectors for classification.

ii) The results of classification with respect to the CK+ database have been far better in terms of the accuracy. However, for the JAFFE database, the results are not likely to be among the best. This motivated us to do redesign our proposed method again. As the graph structure is at the core of the method, in the next section, we evaluate different ways to build the graph structures including the kNN to find the best performance.

6.4 Experimental Performance with different methods for building the graph

As we are interested in evaluating the geometrical, functional and the mixed connectivity based graphs for their performance, we have taken the kNN as the model of geometric structure. Next, under the functional connectivity, the models of absolute *correlations* and absolute *covariances* are considered. In addition, the Kalofolias's method [100] is taken which assumes the smoothness of the observed signals on the inferred graph. At last, we find the Fundis graph, defined in [15], apprporiate under the mixed connectivity because it mixes both the structure and connectivity of the signal. Their weights are given as follows (where t, β and σ are empirically determined parameters):

Geometric graph: *kNN*

$$\mathbf{W}_{\mathrm{ij}}{}^{\mathrm{knn}} = \begin{cases} exp(-\frac{\|\mathbf{x}_{\mathrm{i}}-\mathbf{x}_{\mathrm{j}}\|^2}{2t}), \text{ if } \mathbf{x}_{\mathrm{j}} \in N_{\mathrm{i}} \\ 0, \text{ elsewhere} \end{cases} \tag{6.2}$$

where N_{i} represents the k nearest neighborhood of $\mathbf{x}_{\mathrm{i}}$.

Functional graphs:

Absolute correlation

$$\mathbf{W}_{\mathrm{ij}}{}^{\mathrm{corr}} = \mid \mathrm{corr}(\mathbf{x}_{\mathrm{i}}, \mathbf{x}_{\mathrm{j}}) \mid \tag{6.3}$$

where $\mid \mathrm{corr}(\mathbf{r}_{\mathrm{i}}, \mathbf{r}_{\mathrm{j}}) \mid$ gives the absolute value of correlation coefficient between the vectors of $\mathbf{r}_{\mathrm{i}}$ and $\mathbf{r}_{\mathrm{j}}$.

Absolute covariance

$$\mathbf{W}_{\mathrm{ij}}{}^{\mathrm{cov}} = \mid \mathrm{cov}(\mathbf{x}_{\mathrm{i}}, \mathbf{x}_{\mathrm{j}}) \mid \tag{6.4}$$

where $\mid \mathrm{cov}(\mathbf{r}_{\mathrm{i}}, \mathbf{r}_{\mathrm{j}}) \mid$ gives the absolute determinant value of covariance matrix between the vectors of $\mathbf{r}_{\mathrm{i}}$ and $\mathbf{r}_{\mathrm{j}}$.

Kalofolias

$$\mathbf{L}^{\mathrm{kal}} = \arg\min_{\mathbf{L}\in\mathcal{L},\mathbf{H}} \sum_{k=1}^{N} (\|\mathbf{c}_{\mathrm{k}} - \mathbf{d}_{\mathrm{k}}\|_2)^2 + \beta(\mathbf{d}_{\mathrm{k}})^{\intercal}\mathbf{L}\mathbf{d}_{\mathrm{k}} \tag{6.5}$$

where $\mathcal{L}$ denotes the set of graph Laplacians, $\mathbf{H}$ is an optimization parameter with the same dimension of $\mathbf{G}$ and $\mathbf{c}_{\mathrm{k}}$ and $\mathbf{d}_{\mathrm{k}}$ represents the $k-th$ column of $\mathbf{G}$ and $\mathbf{H}$ respectively.

Mixed graph: *Fundis*

$$\mathbf{W}_{\mathrm{ij}}{}^{\mathrm{fundis}} = exp(-\frac{(1 - \mathbf{W}_{\mathrm{ij}}{}^{\mathrm{corr}})^2}{2\sigma}) \cdot exp(-\frac{\|\mathbf{x}_{\mathrm{i}} - \mathbf{x}_{\mathrm{j}}\|^2}{2t}) \tag{6.6}$$

The overall recognition rates using the weights of the different graph building methods are shown in Table 6.5. It is clear that the different ways to build the graph affect the recognition rate of GSP- based FER method for both the datasets.

As we have taken the normalized Laplacian also as the graph shift operator, we have computed the effect of the different types of the method to build the graph using the non-normalized Laplacian. Overall, the results indicate that the mixed graph- Fundis method (normalized graph Laplacian), using geometric structure and the functional connectivity, yields the best result.

6.4.1 Comparison of the different eigenvectors combination for the kNN and the Fundis structure

As the experimental performance of the different set of eigenvectors were carried out with kNN graph structure in Section (6.3), the same analysis is done with Fundis graph structure. The different sets (I, II, III and IV) with the rows and the columns for the mouth and the eyes regions have been shown in Tables 6.6,

Table 6.5: Effect of the different types of method to build the graph on the overall recognition rate of the facial expression

Type of graph structure	Feature dimension	Recognition rate CK+ (%) Normalized Laplacian	Recognition rate JAFFE (%) Normalized Laplacian	Recognition rate CK+ (%) Non-normalized Laplacian	Recognition rate JAFFE (%) Non-normalized Laplacian
kNN	1536	93.9	77.27	92.68	74.24
Absolute correlation	1536	86.58	80.3	75.61	60.61
Absolute covariance	1536	84.14	57.58	90.24	65.15
Kalofolias	1536	93.9	80.3	92.68	83.33
Fundis	1536	93.9	87.8	86.58	63.63

6.7, 6.8 and 6.9 respectively.

Table 6.6: Fundis method:The impact of the GFT (Set I- Row as the vertices for the mouth as well as the eye regions) with the different set of eigenvectors for the CK+ and JAFFE datasets

Starting eigen vector of mouth	Last eigen vector of mouth	Starting eigen vector of eyes	Starting eigen vector of eyes	Dimension	Recognition rate CK+(in %)	Recognition rate JAFFE (in %)
1	16	1	32	1536	93.9	87.87
1	4	1	4	256	93.9	**92.42**
1	6	1	6	384	93.9	90.91
1	8	1	8	512	93.9	90.91
13	16	29	32	256	82.92	77.27
11	16	27	32	384	84.14	74.24

Table 6.7: Fundis method: The impact of the GFT (Set II- Column as the vertices for the mouth as well as the eye regions) with the different set of eigenvectors for the CK+ and JAFFE datasets

Starting eigen vector of mouth	Last eigen vector of mouth	Starting eigen vector of eyes	Starting eigen vector of eyes	Dimension	Recognition rate CK+ (in %)	Recognition rate JAFFE (in %)
1	32	1	32	1536	91.46	77.27
1	4	1	4	192	92.68	71.21
1	6	1	6	288	93.9	78.78
1	8	1	8	384	93.9	72.72
29	32	29	32	192	88.92	60.6
27	32	27	32	288	85.36	59.09

Table 6.8: Fundis method: The impact of the GFT (Set III- Row as the vertices for the mouth and column for the eye regions) with the different set of eigenvectors for the CK+ and JAFFE datasets

Starting eigen vector of mouth	Last eigen vector of mouth	Starting eigen vector of eyes	Starting eigen vector of eyes	Dimension	Recognition rate CK+ (in %)	Recognition rate JAFFE (in %)
1	16	1	32	1536	86.58	81.81
1	4	1	4	256	92.68	71.21
1	6	1	6	384	92.68	72.72
1	8	1	8	512	91.46	71.21
13	16	29	32	256	85.36	56.06
11	16	27	32	384	89.02	59.09

Table 6.9: Fundis method: The impact of the GFT (Set IV- Column as the vertices for the mouth and row for the eye regions) with the different set of eigenvectors for the CK+ and JAFFE datasets

Starting eigen vector of mouth	Last eigen vector of mouth	Starting eigen vector of eyes	Starting eigen vector of eyes	Dimension	Recognition rate CK+ (in %)	Recognition rate JAFFE (in %)
1	32	1	32	1536	92.68	77.27
1	4	1	4	192	93.9	84.84
1	6	1	6	288	**95.12**	83.33
1	8	1	8	384	95.12	83.33
29	32	29	32	192	85.36	69.69
27	32	27	32	288	87.8	71.21

Table 6.10: Comparison of the kNN and the Fundis for the different set of eigenvectors

Set	kNN- Accuracy CK+ (%)	Fundis- Accuracy CK+ (%)	kNN- Accuracy JAFFE (%)	Fundis- Accuracy JAFFE (%)
I	86.58	93.9	77.27	92.42
II	92.68	93.9	69.69	78.78
III	89.02	92.68	66.67	72.72
IV	92.68	95.12	68.18	84.84

For complete anaysis, all the Sets-I, II, III and IV (as discussed in Tables 6.1, 6.2, 6.3 and 6.4 respectively) are taken for the comparison of the kNN with the Fundis method in Table 6.10. Set-wise comparison reveals that the method of Fundis structure is always better than the performance of the kNN based graph structure in both the databases.

Finally, on the dimensionality reduction of the feature vector, the Fundis method is compared with the classical approach of PCA and some FER methods, shown in Table 6.11, in terms of the accuracy and the feature dimension. It is clear from the Table 6.11 that the Fundis graph performs better than the other methods in both datasets.

Table 6.11: Comparison of the appropriate graph (Fundis) with PCA and some existing FER methods

Method	Feature dimension	Accuracy CK+ (%)	Accuracy JAFFE (%)
PCA	1536	91.46	68.18
Fundis graph	1536	93.9	87.8
PCA - four components	256	84.14	53.03
Fundis graph-four components	256	93.9	92.42
Labeled graph [101]	-	-	77
FRR-CNN [102]	4096	92.06	-

6.5 Summary

Our proposed method, based on GFT, evaluates the different combination of eigenvectors of the graph signals on the dimension and accuracy of the facial expression recognition. The role of the lower graph frequencies in capturing the structural pattern of the graph is relevant and this relationship may be useful for the classification purpose. Experimental results on the datasets of CK+ and JAFFE illustrates the effective performance of the proposed method by increasing the accuracy with the reduced dimension of the feature vector.

In order to find the suitable graph structure, we evaluate the different types of the graph for solving FER problem. Experimental results on CK+ and JAFFE dataset demonstrate that the Fundis graph outperforms other graph structures on the basis of the recognition rate. In addition, the performance of the Fundis graph after the dimensionality reduction (by using the selective components of the graph frequencies) also exceeds than the classical approach of PCA and some existing FER methods.

Chapter 7

Conclusion and future scope

GSP has been emerging as a promising approach to deal with multidimensional signals. In our case, we used the GSP to analyze the 'face' for the recognition of the facial expression. The main problem to apply the GSP requires the selection of the vertex as well as the weight. The obvious choice of every pixel selection from the facial image leads to the $N^2 \times N^2$ size of weight matrix where the size of the image is $N \times N$ (normally, N=256). That may lead to huge computational complexity while ignoring the part of the facial image results in capturing the less information of the face. In this work, the selection of the vertex and the weight has been done to represent the 'face' in the form of the graph signal with the balance between the optimum accuracy and the computational complexity. The concept of GSP helped to improve the existing FER methods (including HOG, CT and FRFT) by using GSP in their combination. The combined GSP based techniques led to the improved accuracy with the significant dimensionality reduction.

Apart from using the GSP in combination with the existing methods, the GSP itself has been used directly on the facial image. Such direct GSP approach provide very good results for the FER in comparison to the existing state- of-the-art methods. Here, the SGWT with the different filter banks was used to find the feature vector.

The notion of the graph frequency helped to represent the graph signal of the face into the transformed domain. By using the GFT, the classification of the facial expression was done with less computational complexity. As performed in PCA, some of the components are sufficient to extract out the distinctive information

for the classification purpose and it lead to dimensionality reduction. Similarly, by selecting the few graph frequency components, the dimension of the feature vector (used for the FER) was reduced without any compromise in the accuracy. Moreover, the evaluation of the different graph structure was carried out and the Fundis graph, based on the geometric and structural connectivity, was found to be optimum.

In this work, the facial expressions have been considered with the main focus on the well stated public databases of CK+ and JAFFE, which contains posed expressions. The objective is to demonstrate how to apply the graph based different concepts for the facial image and analyze the utility of the facial image based graphs for the FER. Nowadays, the field of the facial expression has been moving towards spontaneous and unconstrained datasets. The methods that work well under the conditions of the posed expression are not guaranteed to perform as well under realistic conditions. The GSP based approaches which have given better results are required to extend for the wild datasets.

We observe that unlike the face recognition, facial expression recognition involves the common universal pattern of the expressions irrespective of the person. The interesting question arises that how to design the graph such that minimum samples are required for the training. Further, as these samples will be used to train, whether the results of FER will be invariant to the different datasets. It implies that the obtained results can be applied across the different datasets. That will tend to recognize the facial expression with minimum dependence on the facial datasets like as we actually identify in our daily life.

The intensity of the facial expression conveys complementary information of the related expression and hence, the complete interpretation requires classification as well as the intensity of the identified expression. This work may be further extended to infer the intensity of the expression, which is likely to give better result than the present methods.

Regarding the extensive development in the GSP field esp. about computing the GFT, the fast graph Fourier transform can be implemented rapidly and efficiently. These fast GFT approaches may be extended in inferring the facial expressions and their intensity in the wild datasets.

Our work on the application of the GSP in the FER presents the composite methods of the GSP for the dimension reduction of the existing FER techniques. Thereafter, to address the increase in computational complexity, the methods independently based on the GSP exploiting the intrinsic relationship from the facial image are proposed. Overall, the GSP framework is provided to improve the FER.

Bibliography

[1] W. Freiwald, D. Tsao, and M.S. Livingstone. "A face feature space in the macaque temporal lobe". In: *Nature Neuroscience* 12.9 (2009), pp. 1187–1196.

[2] E. Sariyanidi, H. Gunes, and A. Cavallaro. "Automatic analysis of facial affect: a survey of registration, representation, and recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.6 (2015), pp. 1113–1133.

[3] M. Lyons, J. Budynek, and S. Akamatsu. "Automatic Classification of Single Facial Images". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21.12 (1999), pp. 1357–1362.

[4] M. A. Turk and A. P. Pentland. "Face recognition using eigenfaces". In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 39 (1991), pp. 586–591.

[5] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent component analysis*. Vol. 46. John Wiley & Sons, 2004.

[6] S. Nikitidis et al. "Facial expression recognition using clustering discriminant non-negative matrix factorization". In: IEEE. 2011, pp. 3001–3004.

[7] C. Shan et al. "Facial expression recognition based on local binary patterns: A comprehensive study". In: *Image and Vision Computing* 27.6 (2009), pp. 803–816.

[8] F. Y. Shih, C.-F. Chuang, and P. S. Wang. "Performance comparisons of facial expression recognition in JAFFE database". In: *International Journal of Pattern Recognition and Artificial Intelligence* 22.03 (2008), pp. 445–459.

[9] S. Liao et al. “Facial expression recognition using advanced local binary patterns, tsallis entropies and global appearance features”. In: IEEE. 2006, pp. 665–668.

[10] N. Dalal and B. Triggs. “Histograms of oriented gradients for human detection”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* 1 (2005), pp. 886–893.

[11] C. Cortes and V. Vapnik. “Support-vector networks”. In: *Machine learning* 20.3 (1995), pp. 273–297.

[12] P. Burkert et al. “Dexpression: Deep convolutional neural network for expression recognition”. In: *arXiv preprint arXiv:1509.05371* (2015).

[13] Y. Lv, Z. Feng, and C. Xu. “Facial expression recognition via deep learning”. In: *Smart Computing (SMARTCOMP), 2014 International Conference on*. IEEE. 2014, pp. 303–308.

[14] W. Huang et al. “Graph frequency analysis of brain signals”. In: *IEEE Journal of Selected Topics in Signal Processing* 10.7 (2016), pp. 1189–1203.

[15] L. Rui, H. Nejati, and N.-M. Cheung. “Dimensionality reduction of brain imaging data using graph signal processing”. In: *IEEE International Conference on Image Processing (ICIP)* (2016), pp. 1329–1333.

[16] M. Ménoret et al. “Evaluating Graph Signal Processing for Neuroimaging Through Classification and Dimensionality Reduction”. In: *IEEE Global Conference on Signal and Information Processing (GlobalSIP)* (2017), pp. 618–622.

[17] D. Shuman et al. “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains”. In: *IEEE Signal Processing Magazine* 30.3 (2013), pp. 83–98.

[18] P. Niyogi and X. He. “Locality preserving projections”. In: *In Neural Information Processing Systems, MIT* 16 (2003), pp. 153–160.

[19] X. He et al. “Face recognition using Laplacianfaces”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.3 (2005), pp. 328–340.

[20] D. G. Lowe. “Distinctive image features from scale-invariant keypoints”. In: *International journal of computer vision* 60.2 (2004), pp. 91–110.

[21] A. Vinciarelli, M. Pantic, and H. Bourlard. “Social signal processing: Survey of an emerging domain”. In: *Image and vision computing* 27.12 (2009), pp. 1743–1759.

[22] A. Kapoor, W. Burleson, and R. W. Picard. “Automatic prediction of frustration”. In: *International journal of human-computer studies* 65.8 (2007), pp. 724–736.

[23] D. Tran et al. “A driver assistance framework based on driver drowsiness detection”. In: *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2016 IEEE International Conference on*. IEEE. 2016, pp. 173–178.

[24] B. Fasel and J. Luettin. “Automatic facial expression analysis: a survey”. In: *Pattern recognition* 36.1 (2003), pp. 259–275.

[25] T. F. Cootes, G. J. Edwards, and C. J. Taylor. “Active appearance models”. In: *IEEE Transactions on pattern analysis and machine intelligence* 23.6 (2001), pp. 681–685.

[26] G. Tzimiropoulos et al. “Robust FFT-based scale-invariant image registration with image gradients”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.10 (2010), pp. 1899–1906.

[27] S. Baker and I. Matthews. “Lucas-kanade 20 years on: A unifying framework”. In: *International journal of computer vision* 56.3 (2004), pp. 221–255.

[28] H. Kobayashi and F. Hara. “Recognition of six basic facial expression and their strength by neural network”. In: *Robot and Human Communication, 1992. Proceedings., IEEE International Workshop on*. IEEE. 1992, pp. 381–386.

[29] M. Valstar et al. “Facial point detection using boosted regression and graph models”. In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE. 2010, pp. 2729–2736.

[30] F. Tang and B. Deng. “Facial expression recognition using AAM and local facial features”. In: *Natural Computation, 2007. ICNC 2007. Third International Conference on*. Vol. 2. IEEE. 2007, pp. 632–635.

[31] J. Sung and D. Kim. “Pose-Robust Facial Expression Recognition Using View-Based 2D + 3D AAM”. In: *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 38.4 (2008), pp. 852–866.

[32] K.-l. Zuo and W.-y. Liu. “Facial Expression Recogniton Using Active AppearanceModels”. In: *Journal of Optoelectronics Laser* 15.7 (2004), pp. 853–857.

[33] H.-S. Lee and D. Kim. “Expression-invariant face recognition by facial expression transformations”. In: *Pattern recognition letters* 29.13 (2008), pp. 1797–1805.

[34] C.-L. Huang and Y.-M. Huang. “Facial expression recognition using model-based feature extraction and action parameters classification”. In: *Journal of Visual Communication and Image Representation* 8.3 (1997), pp. 278–290.

[35] J. G. Daugman. “Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 36.7 (1988), pp. 1169–1179.

[36] J. Yu and B. Bhanu. “Evolutionary feature synthesis for facial expression recognition”. In: *Pattern Recognition Letters* 27.11 (2006), pp. 1289–1298.

[37] L. Xiao and Y. Zhang. “Facial expression recognition based on Gabor histogram features and MVBoost”. In: (2007).

[38] S. G. Mallat. “A theory for multiresolution signal decomposition: the wavelet representation”. In: *IEEE transactions on pattern analysis and machine intelligence* 11.7 (1989), pp. 674–693.

[39] G. Van de Wouwer, P. Scheunders, and D. Van Dyck. “Statistical texture characterization from discrete wavelet representations”. In: *IEEE transactions on image processing* 8.4 (1999), pp. 592–598.

[40] E. J. Candes and D. L. Donoho. “Curvelets: A surprisingly effective non-adaptive representation for objects with edges”. In: (2000).

[41] E. Candes et al. “Fast discrete curvelet transforms”. In: *Multiscale Modeling & Simulation* 5.3 (2006), pp. 861–899.

[42] I. W. Selesnick, R. G. Baraniuk, and N. C. Kingsbury. “The dual-tree complex wavelet transform”. In: *IEEE Signal Processing Magazine* 22.6 (2005), pp. 123–151.

[43] S.-C. Pei and M.-H. Yeh. “Two dimensional discrete fractional Fourier transform”. In: *Signal Processing* 67.1 (1998), pp. 99–108.

[44] M. S. Bartlett. “Independent component representations for face recognition”. In: *Face Image Analysis by Unsupervised Learning*. Springer, 2001, pp. 39–67.

[45] T. Jabid, M. H. Kabir, and O. Chae. “Robust facial expression recognition based on local directional pattern”. In: *ETRI journal* 32.5 (2010), pp. 784–794.

[46] L. Le Magoarou, R. Gribonval, and N. Tremblay. “Approximate Fast Graph Fourier Transforms via Multilayer Sparse Approximations”. In: *IEEE transactions on Signal and Information Processing over Networks* 4.2 (2018), pp. 407–420.

[47] F. Gama et al. “Rethinking sketching as sampling: A graph signal processing approach”. In: *arXiv preprint arXiv:1611.00119* (2016).

[48] M. Crovella and E. Kolaczyk. “Graph wavelets for spatial traffic analysis”. In: *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications* 3 (2003), pp. 1848–1857.

[49] R. R. Coifman and M. Maggioni. “Diffusion wavelets”. In: *Applied and Computational Harmonic Analysis* 21.1 (2006), pp. 53–94.

[50] D. K. Hammond, P. Vandergheynst, and R. Gribonval. “Wavelets on graphs via spectral graph theory”. In: *Applied and Computational Harmonic Analysis* 30.2 (2011), pp. 129–150.

[51] J. Yang et al. “Linear spatial pyramid matching using sparse coding for image classification”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)* (2009), pp. 1794–1801.

[52] D. Thanou, D. I. Shuman, and P. Frossard. “Parametric dictionary learning for graph signals”. In: *IEEE Global Conference on Signal and Information Processing (GlobalSIP)* (2013), pp. 487–490.

[53] C. Phillips. “Interpolation and Approximation by Polynomials”. In: *Springer Science and Business Media* (2003).

[54] G. Cheung et al. “Graph spectral image processing”. In: *Proceedings of the IEEE* 106.5 (2018), pp. 907–930.

[55] N. Perraudin and P. Vandergheynst. “Stationary Signal Processing on Graphs”. In: *IEEE Transactions on Signal Processing* 65 (2017), pp. 3462–3477.

[56] D. Tian et al. “Chebyshev and conjugate gradient filters for graph image denoising”. In: *Multimedia and Expo Workshops (ICMEW), 2014 IEEE International Conference on* (2014), pp. 1–6.

[57] F. Zhang and E. R. Hancock. “Graph spectral image smoothing using the heat kernel”. In: *Pattern Recognition* 41.11 (2008), pp. 3328–3342.

[58] A. Gadde, S. K. Narang, and A. Ortega. “Bilateral filter: Graph spectral interpretation and extensions”. In: *Image Processing (ICIP), 2013 20th IEEE International Conference on* (2013), pp. 1222–1226.

[59] Y. Boykov and G. Funka-Lea. “Graph cuts and efficient ND image segmentation”. In: *International journal of computer vision* 70.2 (2006), pp. 109–131.

[60] S. Dongcheng and J. Jieqing. “The method of facial expression recognition based on DWT-PCA/LDA”. In: *Image and Signal Processing (CISP), 2010 3rd International Congress on* 4 (2010), pp. 1970–1974.

[61] Z. Chen F.and Wang Z.and Xu and D. Wang. “Research on a Method of Facial Expression Recognition”. In: *International Conference on Electronic Measurements and Instruments (ICEMI)* 9 (2009), pp. 225–229.

[62] P. Carcagnì et al. “Facial expression recognition and histograms of oriented gradients: a comprehensive study”. In: *SpringerPlus* 4.1 (2015), p. 645.

[63] U. Mlakar and B. Potočnik. “Automated facial expression recognition based on histograms of oriented gradient feature vector differences”. In: *Signal, Image and Video Processing* 9.1 (2015), pp. 245–253.

[64] X. He et al. “Face recognition using Laplacianfaces”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.3 (2005), pp. 328–340.

[65] E. Rosten and T. Drummond. “Fusing points and lines for high performance tracking”. In: *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on* 2 (2005), pp. 1508–1515.

[66] M. Lyons et al. “Coding Facial Expressions with Gabor Wavelets”. In: *IEEE International Conference on Face and Gesture Recognition* (1998), pp. 200–205.

[67] P. Lucey et al. “The extended cohn-kanade dataset (ck+) : A complete dataset for action unit and emotion-specified expression”. In: *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on* (2010), pp. 94–101.

[68] L. Zhang, Tjondronegoro, and Dian. “Facial expression recognition using facial movement features”. In: *IEEE Transactions on Affective Computing* 2.4 (2011), pp. 219–229.

[69] A. Poursaberi et al. “Gauss–Laguerre wavelet textural feature fusion with geometrical information for facial expression identification”. In: *EURASIP Journal on Image and Video Processing* 2012.1 (2012), pp. 1–13.

[70] L. Zhong et al. “Learning active facial patches for expression analysis”. In: (2012), pp. 2562–2569.

[71] S. Happy and A. Routray. “Automatic facial expression recognition using features of salient facial patches”. In: *IEEE Transactions on Affective Computing* 6.1 (2015), pp. 1–12.

[72] S. Happy and A. Routray. "Robust facial expression classification using shape and appearance features". In: *Advances in Pattern Recognition (ICAPR), 2015 Eighth International Conference on* (2015), pp. 1–5.

[73] H. K. Meena, K. K. Sharma, and S. D. Joshi. "Improved facial expression recognition using graph signal processing". In: *Electronic Letters* 53.11 (2017), pp. 718–720.

[74] J. Chen et al. "Facial expression recognition based on facial components detection and hog features". In: *International Workshops on Electrical and Computer Engineering Subfields* (2014), pp. 884–888.

[75] A. Saha and Q. J. Wu. "Curvelet entropy for facial expression recognition". In: *Pacific-Rim Conference on Multimedia* (2010), pp. 617–628.

[76] M. N.Do and M. Vetterli. "The Finite Ridgelet Transform for Image Representation". In: *Trans. Img. Proc.* 12.1 (Jan. 2003), pp. 16–28.

[77] J. Starck, E. J. Candes, and D. L. Donoho. "The Curvelet Transform for Image Denoising". In: *Trans. Img. Proc.* (2002), pp. 670–684. ISSN: 1057-7149.

[78] M Manikandan, A Saravanan, and K. B. Bagan. "Curvelet transform based embedded lossy image compression". In: *2007 International Conference on Signal Processing, Communications and Networking* (2007), pp. 274–276.

[79] J. Starck, D. L. Donoho, and E. J. Candes. "Very high quality image restoration by combining wavelets and curvelets". In: *International Symposium on Optical Science and Technology* (2001), pp. 9–19.

[80] T. Mandal, A. Majumdar, and Q. M. J. Wu. "Face Recognition by Curvelet Based Feature Extraction". In: *Proceedings of the 4th International Conference on Image Analysis and Recognition*. ICIAR'07 (2007), pp. 806–817.

[81] A Majumdar and A Bhattacharya. "Face recognition by multi-resolution curvelet transform on bit quantized facial images". In: *Conference on Computational Intelligence and Multimedia Applications, 2007. International Conference on* 2 (2007), pp. 209–213.

[82] T. Mandal, Q. M. Jonathan Wu, and Y Yuan. "Curvelet Based Face Recognition via Dimension Reduction". In: *Signal Processing* 89.12 (Dec. 2009), pp. 2345–2353.

[83] X. Wu and J. Zhao. "Curvelet feature extraction for face recognition and facial expression recognition". In: *2010 Sixth International Conference on Natural Computation* 3 (2010), pp. 1212–1216.

[84] Chang, Chih-Chung, and C. Lin. "LIBSVM: a library for support vector machines". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 2.3 (2011), p. 27.

[85] J. L. Starck. "Image Processing by the Curvelet Transform". In: *http://jstarck.free.fr* (2002), p. 4.

[86] E. J. Candès. "What is... a curvelet?" In: *Notices of the American Mathematical Society* 50.11 (2003), pp. 1402–1403.

[87] P. Viola and M. Jones. "Rapid object detection using a boosted cascade of simple features". In: *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on* 1 (2001), pp. I–511.

[88] M. Tang and F. Chen. "Facial expression recognition and its application based on curvelet transform and PSO-SVM". In: *Optik-International Journal for Light and Electron Optics* 124.22 (2013), pp. 5401–5406.

[89] H. M. Ozaktas, M. A. Kutay, and D. Mendlovic. "Introduction to the fractional Fourier transform and its applications". In: *Advances in imaging and electron physics* 106 (1999), pp. 239–291.

[90] L. Gao et al. "Recognizing human emotional state based on the phase information of the two dimensional fractional Fourier transform". In: *Pacific-Rim Conference on Multimedia*. Springer. 2010, pp. 694–704.

[91] L. Zhang et al. "Recognizing smile emotion based on Fractional Fourier Transform". In: *Image and Signal Processing (CISP), 2011 4th International Congress on*. Vol. 2. IEEE. 2011, pp. 940–944.

[92] B. Ren, D. Liu, and L. Qi. "Emotion recognition based on multiple order features using fractional Fourier transform". In: *Proc.SPIE* 10420 (2017), pp. 10420 –10420 –6.

[93] A Grossmann. "Wavelet transforms and edge detection". In: *Stochastic Processes in Physics and Engineering* (1988), pp. 149–157.

[94] Y.-H. Baek, O.-S. Byun, and S.-R. Moon. "Image edge detection using adaptive morphology Meyer wavelet-CNN". In: *Proceedings of the International Joint Conference on Neural Networks* 2 (2003), pp. 1219–1222.

[95] N. Perraudin et al. "GSPBOX: A toolbox for signal processing on graphs". In: *arXiv preprint arXiv:1408.5781* (2014).

[96] S. E. Kahou, P. Froumenty, and C. Pal. "Facial expression analysis based on high dimensional binary features". In: *European Conference on Computer Vision* (2014), pp. 135–147.

[97] N. Sebe et al. "Authentic facial expression analysis". In: *Image and Vision Computing* 25.12 (2007), pp. 1856–1863.

[98] M. Yeasin, B. Bullot, and R. Sharma. "Recognition of facial expressions and measurement of levels of interest from video". In: *IEEE Transactions on Multimedia* 8.3 (2006), pp. 500–508.

[99] T. Batabyal, A. Vaccari, and S. T. Acton. "Ugrasp: A unified framework for activity recognition and person identification using graph signal processing". In: *IEEE International Conference on Image Processing (ICIP)* (2015), pp. 3270–3274.

[100] V. Kalofolias. "How to learn a graph from smooth signals". In: *Artificial Intelligence and Statistics*. 2016, pp. 920–929.

[101] W. Zheng et al. "Facial expression recognition using kernel canonical correlation analysis (KCCA)". In: *IEEE transactions on neural networks* 17.1 (2006), pp. 233–238.

[102] S. Xie and H. Hu. "Facial expression recognition with FRR-CNN". In: *Electronics Letters* 53.4 (2017), pp. 235–237.

Publications from the thesis

Journals

1. Hemant Kumar Meena, Kamalesh Kumar Sharma, Shiv Dutt Joshi, 'Improved facial expression recognition using graph signal processing', *Electronic Letters-IET*, Vol. 53, No.11, pp. 718-720, May 2017.

2. Hemant Kumar Meena, Kamalesh Kumar Sharma, Shiv Dutt Joshi, 'Automatic facial expression recognition using the spectral graph wavelet',*IET Signal Processing.*,Oct 2018.

3. Hemant Kumar Meena, Kamalesh Kumar Sharma,Shiv Dutt Joshi, 'Facial expression recognition using graph signal processing on HOG', *IETE- Technical Review-Taylor and Francis.*, Sep 2018.

4. Hemant Kumar Meena, Kamalesh Kumar Sharma, Shiv Dutt Joshi,'Effective Curvelet based facial expression recognition using graph signal processing',(Under review)

International Conferences

1. Hemant Kumar Meena, Kamalesh Kumar Sharma, Shiv Dutt Joshi 'Feature Fusion of HOG and GSP for Smile Recognition', Proc. of the 2017 Augmented Reality, Virtual Reality, and Computer Graphics (SALENTO AVR-2017), Leece, Italy,Springer, June 2017.

2. Hemant Kumar Meena, Kamalesh Kumar Sharma, Shiv Dutt Joshi , 'Low Dimensional Feature Vector based on the Combination of Fractional Fourier

Transform and Graph signal Processing for Facial Expression Recognition', Proc. of the IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI-2017), Chennai, India, September-2017.

3. Hemant Kumar Meena, Kamalesh Kumar Sharma, Shiv Dutt Joshi , 'Facial expression recognition with enhanced feature extraction using Graph Fourier Transform', Proc. of the IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI-2017), Chennai, India, September-2017.

4. Hemant Kumar Meena, Kamalesh Kumar Sharma, Shiv Dutt Joshi, 'Graph building methods for the application of graph Fourier transform in the facial expression recognition', 3rd IEEE International Conference on Image Processing, Applications and Systems (Under review)

Bio data of the Author

Hemant Kumar Meena was born in Alwar, Rajasthan on 28 May,1982. He received the B.Tech in Electrical Engineering and M.Tech in Information and Communication Technology from the Indian Institute of Technology,Delhi in 2005 under the Dual Degree programme.

He began his professional career as a Member of Research Staff in the Central Research Lab of Bharat Electronics Limited, Ghaziabad. Presently, he is pursuing his PhD from Electronics and Communication Engineering from Malaviya National Institute of Technology, Jaipur. His research interests include graph signal processing and image processing.