FACIAL EXPRESSION RECOGNITION IN THE WILD USING IMPROVED TRAJECTORIES AND FISHER VECTOR ENCODING

by

Sadaf Afsharsavojbolaghi

B.S., Computer Engineering, Islamic Azad University (South Tehran Branch), 2010

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computational Science and Engineering
Boğaziçi University

2016

ACKNOWLEDGEMENTS

First I would like to sincerely thank my graduate advisor Assist. Prof. Albert Ali Salah for his support, patience, professional supervision and encouragement. He gave me the freedom to plan, schedule and manage the work myself, while supporting me closely to implement this study.

I also extend my appreciation to the members of my thesis committee, Prof. Lale Akarun and Prof. Zehra Çataltepe for allocating their time to evaluate this work and their valuable comments.

Words would never say how appreciative I am from deepest part of my heart to my husband Amin who was a great source of support, encouragement and love, which was in the end what made this dissertation possible. Mahin and Reza, my parents and my sisters Sara and Sanaz receive my deepest gratitude and love for their dedication and the many years of support. Even though we are far away their supports always encouraged me to keep on each step of my life.

I would like to thank my friends Gül Varol, Heysem Kaya, Altay Brusan, Behnaz Dehghan and Sahand Bagheri for their help and friendship. I would also like to appreciate the staff of the computational science and engineering program at Boğaziçi University for their help.

ABSTRACT

FACIAL EXPRESSION RECOGNITION IN THE WILD USING IMPROVED TRAJECTORIES AND FISHER VECTOR ENCODING

Automatic video data analysis has been a growing interest in order to improve human computer interaction. One of the most challenging parts in video analysis is the ability of evaluating human emotion robustly. Vast applications of human facial expression recognition can be seen everywhere from educational systems to treatment of Asperger's and surveillance. In this thesis, we explore facial expression recognition on both laboratory and realistic videos. After studying recent works about face detection, facial alignment, video description and classification, we present our novel approach in, which our proposed pipeline including facial alignment in combination with improved dense trajectory, geometric, encoded with Fisher vector encoding and LGBP-TOP features are fed to extreme learning machine. It is the first time that improved dense trajectory features are used in facial expression recognition. Furthermore, we extensively study each step of our pipeline in a comparative manner. We evaluate our approach on CK+ and EmotiW 2015 challenge datasets. Videos in first dataset are captured in laboratory settings and start from neutral state and end with peak expression while the second one is selected from movies with realistic conditions, spontaneous emotions, complicated background and challenging illumination variations. On Ck+ dataset, we obtained 94.80% and 95.79% (without contempt) accuracy, which is among the best results obtained on the CK+. On EmotiW 2015 challenge dataset, we got 43.39% accuracy, which is higher than the baseline of the challenge considerably. In both datasets we were able to obtain the state-of-the-art results. Our results show that using appropriate pipeline of face alignment combined with efficient visual descriptors can result in a robust system with high ability of recognition.

ÖZET

İYİLEŞTİRMİŞ İZLEK VE FİSHER VEKTÖRÜ KODLAMASI İLE ZOR ŞARTLAR ALTINDA YÜZ İFADESİ TANIMA

Otomatik video görüntüsü işleme yöntemleri özellikle insan bilgisayar etkileşimini iyileştirme amacı ile öncem kazanmıştır. Video görüntülerinin analizinde özellikle zor bir problem görüntüdeki kişilerin duygu durumunu kestirebilmektir. Yüz ifadesi sınıflandırmanın uzaktan eğitim sistemlerinden Asperger sendromlu kişilerin kullanacağı uygulamalara ve güvenlik uygulamalarına uzanan geniş uygulama alanı mevcuttur. Bu tez çalışması kapsamında kontrollü ve gerçekçi koşullar altında toplanmış video görüntülerinden yüz ifadesi tanıma problemini ele alıyoruz. Yakın zamanda yapılan yüz bulma, hizalama, video öznitelik çıkartma ve sınıflandırma yaklaşımlarını inceledikten sonra yeni bir yöntem öneriyoruz. Bu yöntemde iyileştirilmiş yoğun izlekler yaklaşımını yüz hizalama sonrası uyguluyor, geometrik öznitelikler ve LGBT-TOP özniteliklerini Fisher vektörleri ile kodlayarak ekstrem öğrenme makineleri sınıflandırıcılarına veriyoruz. İyileştirilmiş yoğun izlekler yaklaşımı bu çalışma ile ilk defa yüz ifadesi tanıma problemine uygulanmıştır. Yaklaşımın her aşamasını karşılaştırmalı deneylerle, CK+ ve EmotiW 2015 veritabanları üzerinde sınıyoruz. Bu veritabanlarından birincisi kontrollü kayıt koşullarında toplanmış, nötr yüzden ifadeli yüzlere geçişleri içermektedir. İkinci veritabanı ise gerçekçi koşullarda, doğal ifadeler, zor ışıklandırma ve karmaşık arkaplan görüntüleri içeren film klipleridir. CK+ veritabanında 94.80% (aşağılama ifadesi olmadan 95.79%) ile en iyi sonuçlardan birini elde ediyoruz. EmotiW 2015 veritabanında elde ettiğimiz 43.39% sınıflandırma başarısı ise yarışma temel sonucundan oldukça yüksektir. İki veritabanında da elde ettiğimiz iyi sonuçlar kullandığımız hizalama ve öznitelik çıkartma yöntemlerinin başarılı bir sistem ortaya koyduğunu göstermiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

C_{ELM}	Regularization parameter of ELM
C_{SVM}	Cost parameter of SVM
D	Dimensionality
$g(x; u, \Sigma)$	Gaussian component
H	Hidden layer matrix
$\mathrm{H}^{\dagger}$	Inverse of H
I	Identity matrix
$\mathrm{I(x, y, t)}$	Optical flow
K	Number of cluster components
$\mathrm{K}\left(\mathrm{x_i}, \mathrm{x_j}\right)$	Kernel function
L	Log-likelihood
n_σ	Number of spatial divisions
n_T	Number of temporal divisions
N	Number of instances
$p\left(x \mid \lambda\right)$	Probability of x given λ
P_t	Point at frame t
T	Training label matrix
y_i	Class label of instance i
β	Hidden layer output weight matrix
μ	Mean
π	Pi number
σ	Variance
Σ	Covariance matrix
Ω	Kernel matrix
∇_λ	Gradient vector with respect to parameter λ

LIST OF ACRONYMS/ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
BOF	Bag of Features
CK+	Extended Cohn and Kanade
DRMF	Discriminative Response Map Fitting
ELM	Extreme Learning Machine
EM	Expectation-Maximization
EmotiW	Emotion recognition in the Wild
FV	Fisher Vector
GEO	Geometric Features
GMM	Gaussian Mixture Model
GPA	Generalized Procrustes Alignment
HOF	Histogram of Optical Flow
HOG	Histogram of Oriented Gradients
HOG3D	Histograms of 3D Gradient Orientations
KLT	Kanade-Lucas-Tomasi
LBP	Local Binary Pattern
LGBP	Local Gabor Binary Pattern
LPQ	Local Phase Quantization
MBH	Motion Boundary Histograms
PCA	Principal Component Analysis
RANSAC	Random Sample Consensus
RBF	Radial-basis Function
SDM	Supervised Descent Method
SIFT	Scale-Invariant Feature Transform
SLFN	Single-hidden-layer Feed-forward Network
SURF	Speeded Up Robust Features
SVM	Support Vector Machine

1. INTRODUCTION

1.1. Motivation

Automatic video data analysis has attracted a lot of attention due to the fast development of video data over the recent decades. Among different field of video analysis, evaluating human emotion is one of the most challenging parts. Massive application of human emotion recognition has made it an important field of research. Some of these application are as follows: human-computer interaction (HCI), medical analysis [1], educational systems and surveillance [2].

Positions and movement of the muscles of the face lead to different facial expressions. These movements express the emotional state of a person to observers. In other words a facial expression is a form of nonverbal communication. It is a crucial mean of transmission of social information between humans. The facial expressions are typically classified into seven basic emotions such as neutral, happiness, anger, disgust, sadness, fear, and surprise [3]. In Figure 1.1. we see examples of universal emotions.

A lot of research have been done in facial expression recognition from static images like [4–6] but motion information, which plays an important role in facial expression recognition is discarded in static images. In order to solve this problem, datasets consisting of image sequences (i.e. videos) has been developed recently. Most of the video datasets are collected in laboratory settings and are not compatible with the real world. A robust emotion recognition system should be applicable in realistic conditions and for achieving this purpose developing video datasets in realist condition is the first step. Facial expression recognition in the wild is the name, which has been given to the research that is done on these datasets.

Even with the intensive works that computer vision scientists have done in the emotion recognition task, designing a robust pipeline remains a challenge. Most of the previous researches are done on videos collected under controlled conditions [7–9], such

as the CK dataset [10].

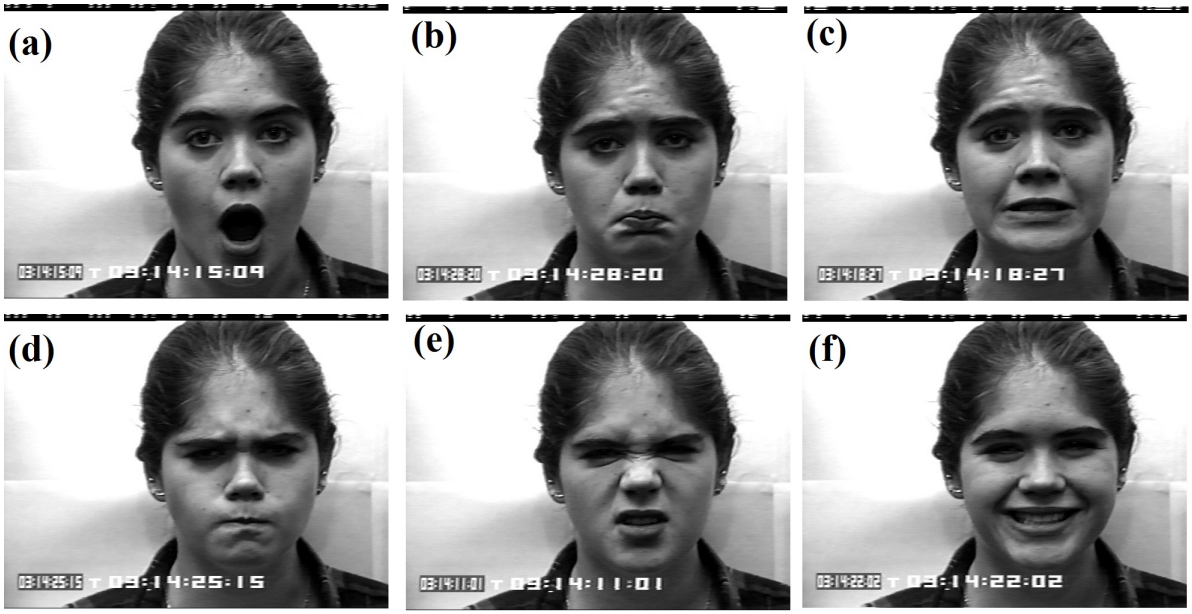

Figure 1.1. Six universal emotions selected from CK+ dataset. a) surprise, b) sad, c) fear, d) angry, e) disgust, f) happy.

Evaluating human emotion in real-world videos like EmotiW 2015 challenge dataset [1] , is still an open challenge. Complexity of facial expression recognition in the real-world videos is due to many reasons such as different illumination conditions, various head poses, unspecified apex of emotion, scaling, occlusion of face and complicated background.

Designing robust facial alignment and discriminative features for video representation is critical to overcome the complexities listed above. Video representation plays a significant part in emotion recognition task since well describing of the appearance, structure and motion of facial parts are dependent on informative features. Actually, facial features can be used to efficiently describe facial motion in static or dynamic facial images. They usually describe shape, color and texture characteristics. Otherwise, they can also minimize within-class differences of facial expressions while maximizing between-class dissimilarities. Among face descriptors, facial geometric and appearance

[1]The Third Emotion Recognition in the Wild Challenge was held at the ACM International Conference on Multimodal Interaction, 2015, Seattle.

descriptors are usually used to extract features in facial expression recognition task.

Besides the importance of feature extraction part in facial expression classification, there exists the facial alignment task, which is an open challenge in the research community. Actually automatic facial alignment is an essential requirement for human computer interaction. In this study, we propose a novel approach for emotion recognition, which is able to tackle the challenges of real-world setting.

1.2. Related Work

The development in video datasets causes progress in emotion recognition. Happy et al. [19] recently published a dataset for spontaneous facial expression. Table 1.1 summarizes some benchmark datasets for facial expression recognition. Kanade et al. [10] introduced the Cohn-Kanade (CK) dataset in 2000 for classifying emotion from videos. This database contains grayscale videos of seven human emotions, namely surprise, sad, angry, fear, happy, disgust and neutral. Videos are recorded in laboratory conditions and have uniform backgrounds. In each video, there is one performer. Videos start from neutral state and end in apex time.

The datasets issued each year are developing in terms of number of video content and conditions of video recording. Recently published datasets, such as EmotiW 2014 [2] and EmotiW 2015, have videos, which are selected from movies with real-world conditions. These videos have challenging illumination condition, background complexity and a lot of numbers of face occlusion. Therefore, they are very hard in terms of emotion recognition.

Below, we briefly review the well-known techniques in the literature that are used for face detection, facial alignment and describing facial expressions. These techniques will be detailed in Chapter 2.

[2]The Second Emotion Recognition in the Wild Challenge was held at the ACM International Conference on Multimodal Interaction, 2014, Istanbul.

Table 1.1. Facial expression recognition datasets and benchmarks.

Database	Database information	Expression description	Posed vs Spontaneous
CK [10]	•100 Subjects,(multi-ethnicity),486 videos • 69% female, 31% male (age 18-50,yrs) • Frontal and 30 degree imaging	• AU-coded face database • 23 series of facial display • Single and combinations of AUs	Posed
(CK+) [11]	• Extension of Cohn-Kanade • 123 Subjects,(multi-ethnicity),593 videos	• Onset to peak coded • Spontaneous smiles (66 subjects)	Posed Spontaneous
MMI [12]	• 25 subjects (multi-ethnicity) • 12 female , 13 male (age 20-32,yrs)	• Single and combinations of AUs • Temporal analysis (e.g.,onset,apex,offset)	Posed Spontaneous
JAFFE [13]	• 10 female Japanese models • Grayscale images	• Neutral+6 Basic expressions • 2 to 4 samples per expression	Posed
Bosphorus [14]	• 105 subjects • 44 female , 61 male	• AU-coded (2D/3D data) • Pose and illumination variations	Posed
NVIE [15]	• 215 Student (age 17-31 yrs),118 videos • Visible and infrared imaging	• Basic facial expressions • Temporal analysis for posed data	Posed Spontaneous
DISFA [16]	• 27 Participants (15 male,12,female) • 130,000 video frames	• Intensity of 12 AUs coded	Spontaneous
EmotiW 2015 [17]	• 723 train videos • 383 validation videos • 539 test videos	• Neutral+6 Basic expressions • selected from realistic movies	Spontaneous
Peng et al. [18]	• 2000 colored images • 350 images for each expressions • different races, countries, ages	• 6 Basic expressions • selected from web	Spontaneous

In computer vision, the facial expression recognition pipeline consists of face detection, landmark localization, alignment, feature extraction and classification steps. Face detection is one of the most studied topics in the computer vision literature. Various algorithms were developed to solve this fundamental computer vision problem. Viola & Jones [20] proposed a face detection method, which is practically feasible in real world applications such as digital cameras and photo organization software. Afterwards Lienhart & Maydt [21] generalized the feature set of [20]. Jones & Viola [22] proposed a feature called diagonal filter, which is essential for detecting non-upright faces and non-frontal faces. Jones et al. [23] further extended the Haar-like feature for video-based pedestrian detection. Zhu & Ramanan [24] proposed a model based on a mixtures of trees, which outperforms Viola & Jones's method in [25].

The second important step in facial expression recognition pipeline, which also affects alignment part is landmark localization. If landmark points on faces can be located efficiently and accurately, these will guide the registration. Zhu & Ramanan [24] presented a unified tree-structured model for face detection and landmark localization in the wild, which is shown to be efficient for capturing deformation. Dibeklioglu et al. [26] offered a method for 2-D facial landmarking, which is based on the combination of a mixture model of Gabor wavelet features and a shape prior, estimated with a multivariate Gaussian mixture model. Xiong & De La Torre [27] proposed a supervised descent method for minimizing a Non-linear Least Squares (NLS) function, which achieved state-of-the-art performance in facial features detection. Asthana et al. [28] introduced Discriminative Response Map Fitting method for landmark localization, which uses Zhu & Ramanan [24] face detector and outperforms state of art algorithms.

Alignment (or registration) is an important step, since removing rotation, scale and translation can improve the recognition system considerably. Different alignment methods have been proposed for face registration. In the field of morphometrics, Gower [29] proposed the Generalized Procrustes Analysis (GPA), which can be used for aligning any number of shapes represented by point sets, to a reference model. This approach requires accurate landmarks to produce good results. If this condition is met, GPA will provide a very good registration for 2D or 3D faces. Zhu & Li [30] proposed

a novel method for pose and expression normalization in the wild. Hassner et al. [31] introduced a new alignment method, which unlike the recent methods estimates the shape of all input faces by using a single 3D surface. Recently Kim et al. [32] proposed an approach, which deals with the problem of registration in real-world conditions by fusing alignable faces with the non-alignable facial images where facial landmarks can not be detected.

The next crucial step in this pipeline is feature extraction. An efficient visual descriptor should be able to extract meaningful information from video. In the literature visual descriptors, which are mainly used for video modeling consist of LBP-TOP [33], LGBP-TOP [34], which is the Gabor extension of LBP-TOP, SIFT descriptor [35] that is a 3-D spatial histogram of the image gradients, HOG [36] descriptor, which filter the image with point discrete derivative mask. Other well-known methods are Local Phase Quantization from three orthogonal planes (LPQ-TOP) [37], which benefits from 2-D Discrete Fourier Transform (DFT), Geometric features, which have shown good performance in facial expression recognition [38,39] with precise located and tracked facial landmarks. Recently Mollahosseini et al. [40] proposed an approach based on training deep neural network on both well-labeled and combination of noisy and well-labeled facial images, collected from the web. Furthermore histograms of 3D gradient orientations (HOG3D) [41, 42], oriented histograms of flow (HOF) [43] and motion boundary histograms (MBH) [44, 45] has shown promising results. Wang and Schmid [46] recently proposed an approach based on MBH, HOG, HOF descriptors sampled along improved dense trajectories. They used this method for action recognition, but since in emotion recognition we need to track changes in facial dynamics, this method can also be useful for emotion recognition.

In the literature video representation by using bag of words (BOW) has been vastly used [47–49] The pipeline for traditional BOW consists of feature detection, feature description, codebook generation with clustering algorithms like k-means and feature encoding in an accumulated way. A recent research shows that a better encoding, namely Fisher vector (FV) representation, considerably increases recognition performance [50].

The last step in recognition pipeline is classification. In the literature, support vector machines (SVM) have been frequently used for classification of facial expressions [39, 51–53]. Recently, extreme learning machines (ELM) introduced by [54] is shown as a viable alternative to the SVM, which is slow to train. In our study, we use ELM and show that it has good generalization performance for multi-class classification and needs shorter time for the learning phase, compared to SVM.

Deep methods have attained more attention in the context of facial expression recognition recently. The convolutional neural network (CNN) is one of the popular deep learning structure. In a recent study by Li et al. [55], 10,595 external images were used for training CNN models and 83% mean recognition rate was reported on CK+. Lv et al. [56] proposed a method based on face parsing detectors trained via deep belief networks and obtained 91.11% mean recognition rate. Liu et al. [9] proposed a new Boosted Deep Belief Network (BDBN), which yields 96.70% mean recognition rate, but it should be stated that in that work, the contempt emotion was not considered.

1.3. Contributions

Facial alignment is one the most challenging part in the emotion recognition system. Considerable numbers of researches have been done in this field but since in realistic videos detecting face and landmarks are challenging, accuracy of these methods drop. This problem affects the overall performance of the system dramatically and thus the presence of an efficient alignment method can improve the performance of the system.

In this thesis, we present an alignment pipeline, which is able to detect face and landmarks under challenging conditions. Our registration pipeline align all faces to a reference model and removes scale, in plane rotation and translation. We compare the result of our alignment with the provided aligned faces by the EmotiW 2015 challenge organizers. We show that our registration method outperforms the prepared set of aligned faces by EmotiW 2015 challenge organizers. We improve the alignment in terms of detecting more faces with low number of false positive. Also our registration

benefits from a precise landmarking method, which is able to detect a consistent number of landmarks while removing scale, rotation and translation.

In the literature, many appearance based features have been used to extract information from important parts of the face. Across the appearance of the face, motions of the face parts have valuable information about changing the emotion in a subject. In this study, we propose to extract improved dense trajectory features along with geometric and LGBP-TOP features. To the best of our knowledge, we are the first to apply improved dense trajectories in facial expression recognition task.

The approach proposed in this thesis is published as [57]. Other publications, which appeared during the course of the thesis are [58] and [59]. [59] was the first runner up of the EMOTIW AFEW 2015 Challenge [17].

1.4. Organization of the Thesis

The rest of this thesis is organized as follows. We review technical background of facial expression recognition in videos in Chapters 2, 3 and 4. In Chapter 2, we present popular techniques for face detection landmark localization and facial alignment. In Chapter 3 we present popular techniques for video description with local descriptors. The concepts of appearance and local motion descriptors in space-time volumes are discussed in detail. The development of trajectory-based methods is described, and finally, the local feature aggregation methods are explained. In Chapter 4, classification methods, extreme learning machines and support vector machines are presented. We detail our proposed methodology in terms of facial alignment, feature extraction and classification in Chapter 5. We report and discuss our results in Chapter 6, where we extensively study the effects of certain parameters. Finally, we conclude our work and state some possible future work in Chapter 7.

2. FACE DETECTION, ALIGNMENT AND LANDMARK LOCALIZATION

For designing a robust emotion recognition system in the wild, face detection, landmark localization and registration are important pre-processing steps.

2.1. Face Detection

Face detection is the first step in automated face recognition. Its performance has the great influence on the accuracy of the whole facial expression recognition system. An ideal face detector should be able to identify and locate all the existing faces regardless of their position, scale, orientation, age, and expression in an image or a video. Face detection can be performed based on several factors: skin color (for faces in color images and videos), motion (for faces in videos), facial/head shape, facial appearance, or a combination of these parameters. Most successful face detection algorithms are appearance-based without using other cues [60]. The procedure is as follows: An input image is skimmed at all possible locations and scales by a subwindow. Face detection is performed by classifying the pattern in the subwindow as either face or nonface while The face/nonface classier is learned from face and nonface training samples.

The face detection algorithm, which proposed by Viola and Jones [20] was the most impressive method in the 2000's. Success and real time detection of the Viola-Jones face detector results from three main properties, namely the integral image representation, classifier learning with AdaBoost, and the attentional cascade structure. Integral image is an algorithm for computing the sum of values in a rectangle subset of a grid. Viola and Jones used the integral image for the computation of Haar-like features. Integral image is calculated by taking the integral of the original image at each pixel location and computing sum of pixels in a rectangular region, which is shown

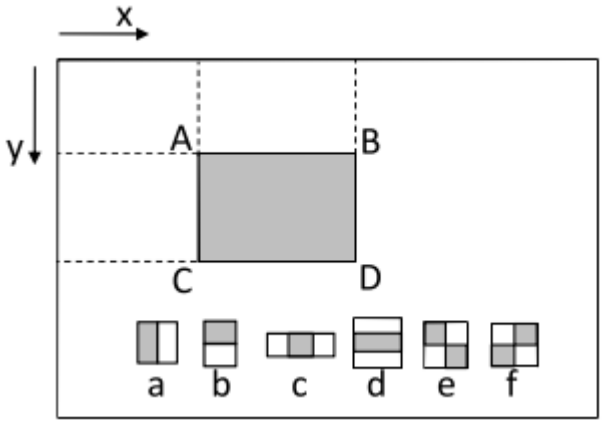

Figure 2.1. Illustration of the integral image and Haar-like rectangle features (a-f) [61].

in Figure 2.1 and in Equation 2.4.

$$\sum_{(x,y)\in ABCD} i\left(x,y\right) = ii\left(D\right) + ii\left(A\right) - ii\left(B\right) - ii\left(C\right) \tag{2.1}$$

The method of training a boosted classifier is called AdaBoost. A boosted classifier is defined by Equation 2.2.

$$F_T\left(x\right) = \sum_{t=1}^{T} f_t\left(x\right) \tag{2.2}$$

where each f_t is a weak learner (like one-level decision tree) that takes an object as input and outputs a real value as the class of the object. The sign of output indicates the predicted object class and the absolute value gives the confidence in that classification. Attentional cascade is an important part in the Viola and Jones detector. Actually by using attentional cascade, the boosted classifier will be more effective, and will discard most of the negative sub-windows while keeping more or less all the positive instances. This idea makes the detection process really efficient by excluding great portion of the sub-windows in the initial stages of the detector.

Lienhart and Maydt [21] improved the feature set of [20] by proposing 45 degree rotated Haar-like features. Afterwards Jones and Viola generalized these features further by adding diagonal filters in order to handle profile views and rotated faces [22]. Haar-like features were further enhanced in Jones et al. study [23]. Their proposed method is the first detector, which benefits from both motion and appearance information for video based pedestrian detection in small scale.

In 2012, Zhu and Ramanan [24] presented a unified tree-structured model for face detection and landmark localization in the wild, which is shown to be efficient for capturing deformation due to viewpoint variation. This method works based on a model formed by appearance and shape information. For example, for an image I and the pixel location of landmark i, li = (xi, yi), the shape model is defined as:

$$S = App_m\left(I, L\right) + Shape_m\left(L\right) + \alpha^m \tag{2.3}$$

where App_m stands for appearance features (HOG) and $Shape_m$ represents the displacement of the one parts of the face relative to another part and finally the last term α^m is a scalar bias. To learn the model, a fully-supervised scenario is used where positive images with landmark and mixture labels, as well as negative images without faces are provided. Both shape and appearance parameters are learned discriminatively using a structured prediction framework.

2.2. Landmark Localization

Facial feature localization is an important component of a facial expression system, which is also important for robust facial feature tracking and facial modeling. Due to the illumination, pose and expression variation, it can be said that efficient and automatic detection of facial features is one the most challenging parts of a recognition system.

2.2.1. Supervised Descent Method

Xiong & de la Torre [27] proposed a Supervised Descent Method (SDM), which tries to learn a sequence of descent directions for minimizing the error between estimated and true landmark positions. Manually annotated landmarks are used in the training phase as true landmarks and SIFT features [35] are extracted at those landmark positions X_*. Then initial guess of landmark positions is obtained by using Viola & Jones face detection namely, X_0. Afterwards, an error function, which is defined as:

$$f\left(X_0 + \triangle X\right) = \| \; SIFT\left(d\left(X_0 + \triangle X\right)\right) - SIFT\left(d\left(X_*\right)\right) \|^2 \tag{2.4}$$

is updated over $\triangle$X until initial guess converges to true landmark positions.

2.2.2. Discriminative Response Map Fitting

Asthana et al. [28] proposed a discriminative regression based incremental face alignment method, which constructs a person-specific model automatically by incremental updating of the generic model. DRMF is a part based model, which unlike holistic based methodologies that use texture based features, uses local image patches around the landmark points. Actually the method works based on the finding a mapping from the response estimate of shape perturbations to shape parameter updates. For this purpose, a perturbation Δp is defined in the training set and from a $w \times w$ window centered around each point of the perturbed shape, responses are estimated,here HOG features,$A_i\left(\Delta p\right) = \left[p\left(l_i = 1 \mid x + x_i(\Delta p)\right)\right]$. Then a function f such that $f\left(\{A_i\left(\Delta p\right)\}_{i=1}^{n}\right) = \Delta p$ is learned from the response maps around the perturbed shape $\{A_i\left(\Delta p\right)\}_{i=1}^{n} = 1$. The optimization of the parameters are done in a way that the positions of the created model correspond to well-aligned facial parts. This method has shown promising results for reconstructing unseen response maps.

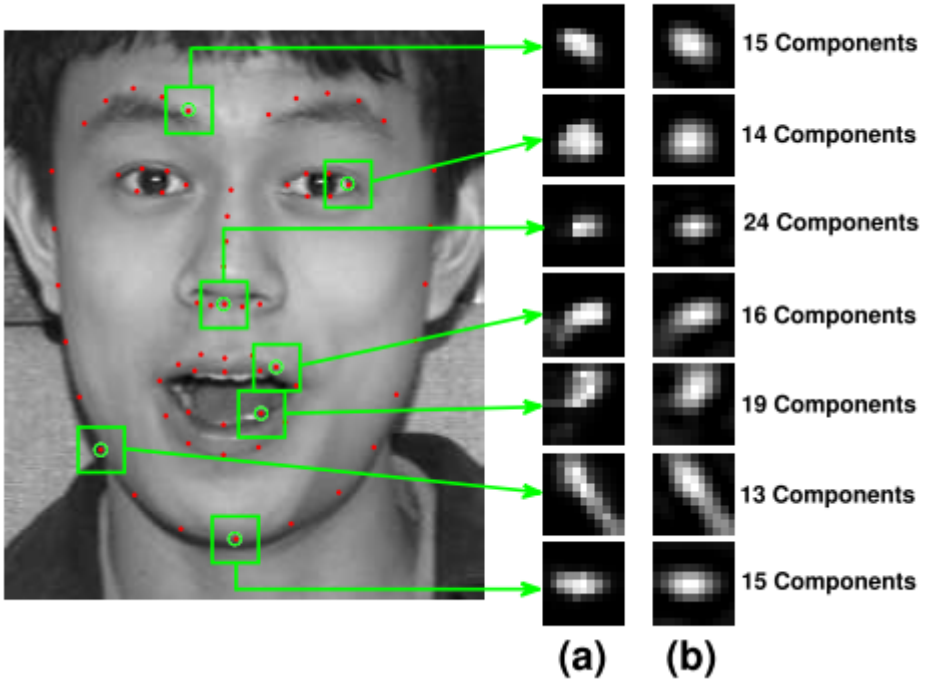

Figure 2.2. Overview of the response patch model: (a) Original HOG based response patches. (b) Reconstructed response patches using the response patch model [28].

2.3. Facial Registration

Purpose of face alignment is transforming different sets of faces into a common coordinate system. In the registration pipeline one face is chosen as the reference face and the rest are considered as target faces. Usually a transformation model consisting of linear transformations such as rotation, scaling and translation is used to align the target face to the reference face.

There are two different approaches for face alignment, namely, intensity-based and feature-based methods. Intensity based methods compare intensity patterns in faces via correlation metrics, whereas feature-based methods benefit from correspondence between facial landmarks. Intensity-based methods use either the whole face, or sub-faces. In sub-face methods, centers of corresponding sub faces are treated as matching feature points. But in feature-based methods, a correspondence between facial landmarks is established and the target face is mapped to the reference face by

using geometrical transformation [62].

2.3.1. Generalized Procrustes Alignment (GPA)

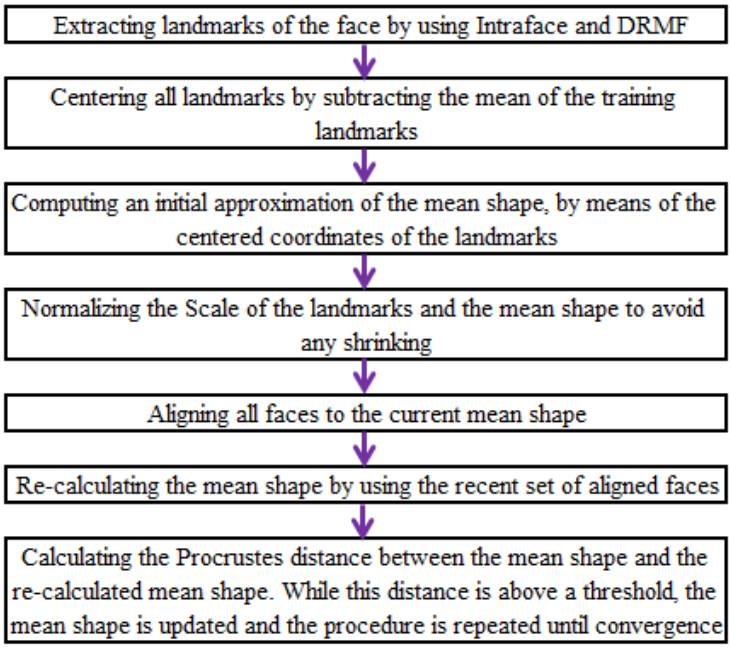

Figure 2.3. Procrustes alignment.

In order to remove translation, rotation and scale effects from a set of faces, an alignment system with a reference face model can be used. To obtain this goal, a procedure is used known as the generalized Procrustes analysis, proposed by Gower [29]. Faces are represented by landmarks and an iterative approach is employed to obtain the reference model and registered set of faces, at the same time. The steps are explained in Figure 2.3.

2.3.2. Face Frontalization

Recent researches have suggested that frontalization may noticeably improve facial recognition systems. Zhu et al. [30] proposed a novel method for pose and expression normalization in the wild. They made a pose adaptive 3DMM fitting algorithm by using landmark marching. Finally they used depth information of 3D meshed face image to find the 3D transformation, which normalizes pose and expression while preserving identity information for face recognition. In Figure 2.4 an overview of the

method is shown.

Hassner et al. [31] introduced a new method which doesn't make a 3D facial shape for each image. Instead, they estimated the shape of all input faces by using a single 3D model. They obtained the initial frontal face by using back-projecting the appearance of the image to the reference model. After that they used the symmetry of the face to finalize the procedure. Pipeline of the system is illustrated in Figure 2.5.

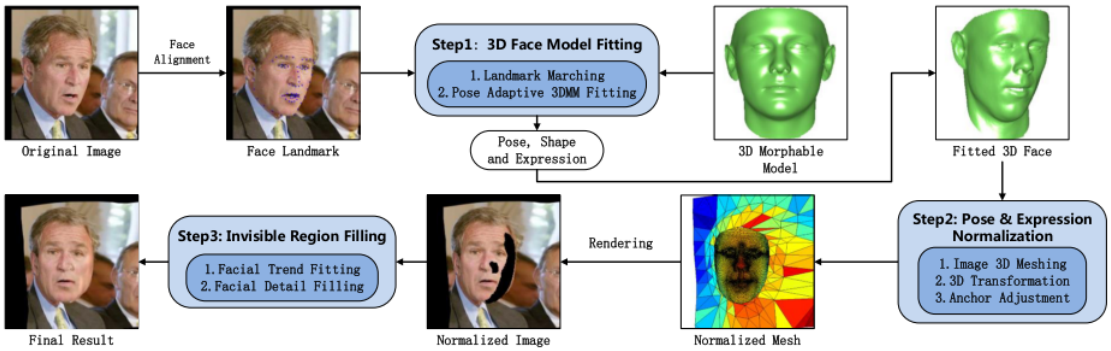

Figure 2.4. Overview of the High-Fidelity Pose and Expression Normalization (HPEN) method [30]

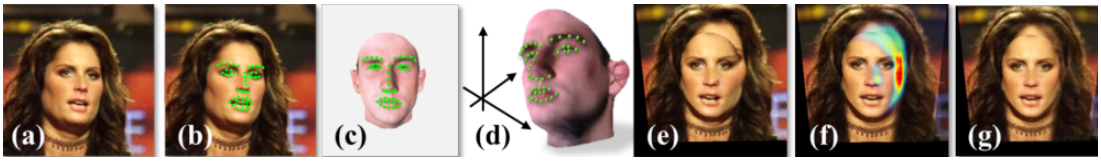

Figure 2.5. Frontalization process overview. (a) query photo (b) facial features (c) the same facial features on the reference face model, (d) reference face model(e) estimated projection matrix (f) estimated visibility (g) final frontalized result [31].

3. VIDEO DESCRIPTION WITH LOCAL DESCRIPTORS

A video sequence is a group of static images; so static-image-based emotion recognition algorithms can always be applied on videos. An important feature of a video sequence is its temporal continuity. In this chapter, we investigate how temporal continuity can be incorporated for video-based emotion recognition. Given a video clip, we need to represent the raw data of pixels into a compact form, which is informative in terms of emotion content. This representation needs to be distinctive, low-dimensional, interpretable and with a fixed size. Distinctive representations are more successfully recognized by classifiers. The advantage of low-dimensional representations of vectors is that they require less storage and do not suffer from the curse of dimensionality. Interpretability can be help us to know which information is encoded in which part of the feature vector. Finally, for using most classification algorithms, a global representation for a video clip should have a fixed size for each sample.

In this chapter, we study various methods for extracting and encoding local motion and appearance information.

3.1. Detectors

To decide which feature points should be selected for describing image contents, different sampling methods are developed. There are sparse [63], dense [64] and random sampling strategies [65]. Sparse methods assume that sampling only the interest points is adequate for describing the video content. Interest points, known also as keypoints have valuable local information, because they have high variation in space and/or time. Dense methods simply sample points from a regular grid with or without a multiscale pyramid. Multiscale pyramid is a method in which an image is subjected to repeated smoothing and subsampling. The motivation is that dense coverage of the video domain ensures that the context information is also captured. In this work we will discuss the methods which use dense sampling.

3.1.1. Dense Sampling

The dense trajectory features proposed by Wang et al. [44] shows that dense sampling is more successful than sparse sampling for action recognition. Actually for dense sampling a multi-scale regular grid is used for extracting feature points. The video blocks are sampled from 5 dimensions $(\mathrm{x}, \mathrm{y}, \mathrm{t}, \sigma, \tau)$. This notation means a feature point p is positioned at $(\mathrm{x}, \mathrm{y}, \mathrm{t})$ in the space-time volume and the video patch centered at p has size determined by the scale (σ, τ).

3.2. Descriptors

Face representation has been studied intensively for automatic emotion recognition. Different approaches have been proposed based on static and dynamic facial images. Facial feature representations are categorized into geometric, appearance-based approaches, hybrid feature and motion based methods. In this thesis we will study the contribution of motion based feature extraction methods when they are combined with geometric and appearance features.

3.2.1. Appearance-Based Approaches

Different characteristics of an image such as shape, color, texture and motion can be described by visual descriptors. Appearance-based features are representing the facial characteristics such as texture and seem to be more stable to image spatial transforms than the geometric features, especially under inaccurate alignment and low-resolution.

3.2.1.1. LBP-TOP. The principle of local binary pattern from three orthogonal planes (LBP-TOP) is such that it applies LBP [66, 67] separately on three orthogonal planes (XY, XT and YT), which intersect in the center pixel. The simplest form of LBP feature vector is created by dividing the examined image into blocks and for each pixel in a block, comparing the pixel to each of its 8 neighbors. If the center pixel's value is greater

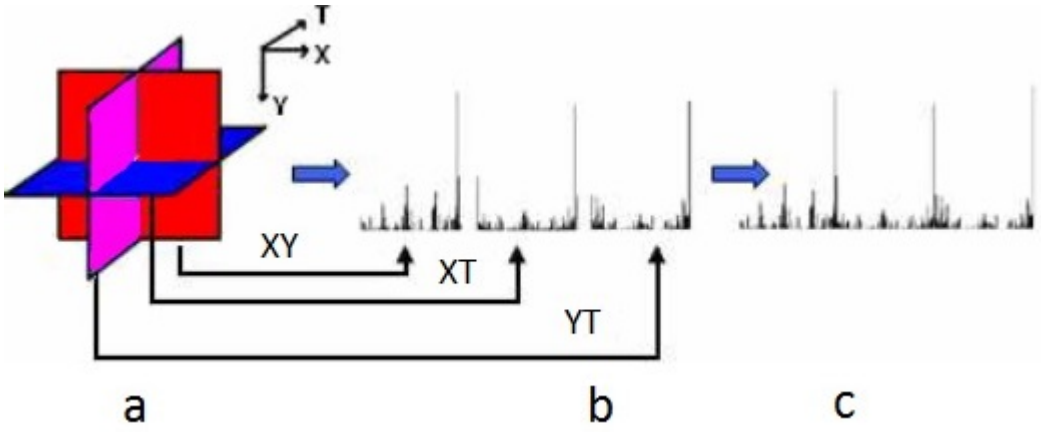

Figure 3.1. (a) three planes in dynamic texture (b) LBP histogram from each plane (c) concatenated feature histogram [33].

than the neighbor's value, write "0" Otherwise, write "1". In the last step histograms of all cells are concatenated to form a feature vector for the entire image. Actually LBP-TOP labels the pixels of an image on three orthogonal planes by thresholding the neighborhood of each pixel and considers the result as a binary number. The final descriptor is a histogram of these binary numbers. This histogram feature vector can describe effectively appearance, horizontal motion and vertical motion from an image sequence [33]. Overview of the method is represented in Figure 3.1.

3.2.1.2. LGBP-TOP. In realistic videos frontal-view facial images may not be available. Consequently, it is vital to investigate methods that recognize facial emotion from random views. According to the study of Moore & Bowden [68], it is observed that local Gabor binary pattern (LGBP) feature operators outperforms other variants of LBP in multi-view emotion recognition. For computing LGBP [34], Gabor-pictures are obtained by convolving images with a set of 2D complex Gabor filters, and then LBP-TOP is extracted from each Gabor-picture. Illustration of the method is represented in Figure 3.2.

3.2.1.3. SIFT. Scale-invariant feature transform (SIFT) descriptor is a three dimensional histogram of the image gradients in describing the appearance of a keypoint. A three-dimensional feature vector, which is formed by the pixel location and the gradient

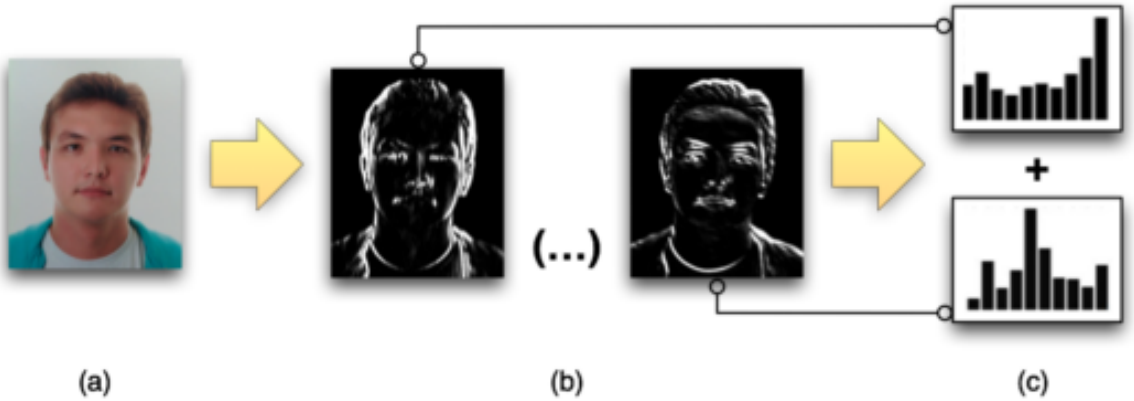

Figure 3.2. LGBP features: a) original image, b) Gabor pictures, c) concatenation of resulting histograms after applying LBP [34].

orientation is considered as the gradient at each pixel. Gradient norm is used to weigh samples and then samples are gathered in a 3-D histogram h, which forms the SIFT descriptor of the region. In order to give less significance to gradients more distant from the keypoint, a Gaussian weighting function is also applied to center. Summary of the method is illustrated in Figure 3.3.

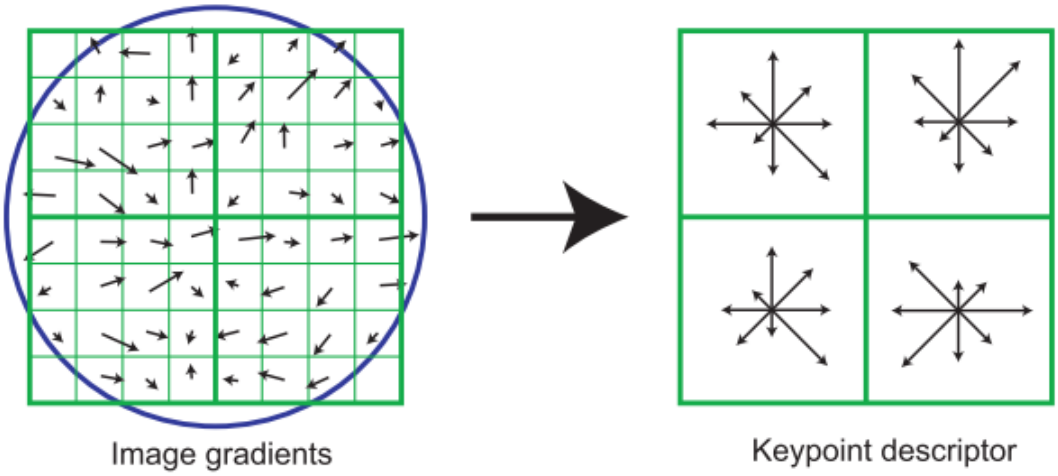

Figure 3.3. Computing the gradient magnitude and orientation as shown on the left. These Samples are then accumulated into orientation histograms as shown on the right [35].

3.2.1.4. HOG. The histogram of oriented gradients (HOG) is a feature descriptor used in computer vision for object detection. The advantage of HOG to other descriptors is that it is invariant to geometric and photometric transformations due to operation on local cells. In this method, the image is filtered with $[-1, 0, 1]$ and $[-1, 0, 1]^\tau$ point discrete derivative masks in one or both of the horizontal and vertical directions and then occurrences of gradient orientation in localized portions of an image is counted.

3.2.1.5. LPQ-TOP. Local phase quantization is computed by taking 2-D Discrete Fourier Transform (DFT) of M-by-M neighborhood of each pixel in the gray scale image. 2D-DFT is computed at four frequencies $\{[a; 0]^\tau; [0; a]^\tau; [a; a]^\tau; [a; -a]^\tau\}$with $a = 1/M$, which correspond to four of eight neighboring frequency bins centered at the pixel of interest. After calculating LPQ, LPQ-TOP is calculated by extracting LPQ features independently from three sets of orthogonal planes: XY, XT and YT, considering only the co-occurrence statistics in these three directions, and stacking them into a single histogram [37].

3.2.2. Geometry Based Approaches

Geometric features are efficient methods, most of which have promising performance in emotion recognition, but they depend on how accurate fiducial points are detected and tracked. Geometric features (GEO) can describe important information like shape, which can be defined by using the position of landmarks, movement of facial landmarks and shape of each facial component. This information can help us to understand facial structure of various facial expressions. Geometric features have the advantage of low dimensionality. But accuracy of face registration, changes in lighting and non-rigid motion can affect the accuracy of geometric methods considerably.

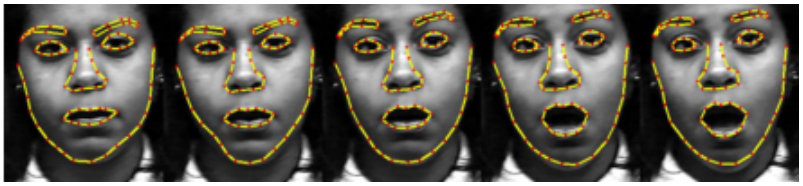

Figure 3.4. Displacement of fiducial points in case of surprise [38].

3.2.3. Improved Dense Trajectory Features

3.2.3.1. Histogram of Oriented Gradients. Dalal & Triggs [36] first proposed to use a histogram of gradient orientation (HOG) as a feature vector. After that Laptev et al. [43] extended the application of this method to spatio temporal space.

In this method, after filtering the image with a point discrete derivative mask, the space-time volume is divided into $n_\sigma \times n_\sigma \times n_\tau$ cells and for each cell, the number of quantized gradient orientations is counted and formed in to a histogram. In the case of [43], four orientations are considered, whereas in [44], all eight orientations are used. These histograms are normalized and concatenated to form the HOG descriptor.

HOG is an appearance encoding feature. Since it uses structure information, it is more robust than geometric features in presence of inaccurate alignment and low-resolution. More precisely, an appearance based feature can characterize variation of low-level features like pixel intensity in the face. Different classes of emotions that are recognized better with appearance information benefit from this descriptor.

3.2.3.2. Histogram of Optical Flow. For computing the histogram of optical flow similar to HOG, spatio temporal space is divided into $n_\sigma \times n_\sigma \times n_\tau$ cells and for each of these cuboids histograms are calculated. But instead of gradient vectors, the optical flow vectors are counted. These motion vectors are quantized into five bins in [43] and nine bins (eight orientations + zero bin) in [44]. The zero bin is related to the pixels, whose optical flow magnitudes are lower than a threshold. Normalized histograms for each cell are concatenated to form the HOF descriptor. HOF captures first order local motion information, since it uses absolute motion. Emotions that involve more specific motion information like surprise benefit from this descriptor.

3.2.3.3. Motion Boundary Histogram. Dalal et al. [45] first introduced motion boundary histograms (MBH) for human detection from videos, and after that, Wang et al. [44] used it for action recognition. This descriptor is formed by calculating the derivatives

of the optical flow vectors and making a histogram according to the number of optical flow derivatives in each spatio-temporal cell. The spatial derivatives of optical flow $I_w = (I_x, I_y)$ are computed for both x and y components. The orientations are quantized into eight bins for each component and the normalized histograms are concatenated. Since MBH uses the derivative of the motion, it captures second order local motion information.

3.2.3.4. Trajectories. *Point Tracking and Optical Flow.* Various computer vision applications need point tracking as an important component. Given a video and the initial position of a certain point, tracking algorithms try to estimate the position of this point in the next frame(s). Point tracking is directly related to distribution of velocities of objects in an image, which is determined as optical flow. Optical flow is described in terms of intensity $I(x; y; t)$, by following brightness constancy constraint:

$$I(x, y, t) = I(x + \triangle x, y + \triangle y, t + \triangle t) \tag{3.1}$$

where $\triangle x, \triangle y$ and $\triangle t$ denote the displacement of point (x, y, t)between two frames.

For a feature-based system to be able to work properly, identifying appropriate features and then tracking them is a vital step. Lucas & Kanade [69] proposed a system, which is a famous method for optical flow estimation. They used the assumption of constant flow under local neighborhood in their method. Afterwards, the Lucas and Kanade method was further developed by Shi and Tomasi [70]. They called their method KLT, which is today among the best tracking techniques. KLT is a sparse tracking technique and it is based on spatial intensity information in order to find the best match position.

Farneback [71] proposed an optical flow method and OpenCV [72] has an open source implementation of it. Unlike the KLT tracker, the Farneback algorithm yields dense optical flow. It uses quadratic polynomials to approximate each neighborhood of two frames and makes use of the these polynomial coefficients for computing global

displacement vector.

Dense Trajectories. Wang et al. [44] proposed a method for action recognition based on dense trajectories and motion boundary histograms. They showed that dense sampling is more successful than sparse sampling methods. In this method, points are sampled densely from multiscale pyramid from each frame of the video. These points are then tracked for a certain time window. The tracking is based on dense optical field computation [71] and is applied on each spatial scale separately. Local motion information is encoded using local descriptors, which are extracted along the trajectories. Dense trajectories method is shown in Figure 3.5.

A trajectory is defined as the concatenation of $P_t, P_{(t+1)}, P_{(t+2)}, \ldots$ in [44], where

$$P_{t+1} = (x_{t+1}, y_{t+1}) = (x_t, y_t) + (M \times \omega)\,|_{(\hat{x}_t, \hat{y}_t)} \tag{3.2}$$

Here, $(\hat{x}_t, \hat{y}_t)$ signifies the rounded position of (x_t, y_t) The optical flow field $\omega = (u_t, v_t)$ is convolved with median filtering kernel M to track point P_t to the next frame.

In order to remove false trajectories, points with displacement larger than a threshold are deleted in order to avoid large jumps between consecutive frames. The points that lie in uniform regions are unlikely to be tracked. These points are not tracked if their cornerness value (the eigenvalues of the auto-correlationmatrix) is below a certain threshold. To avoid drifting in tracking, the points are tracked up to a certain time window of L frames. When tracking is done, trajectories that do not contain any motion are removed. Finally, to guarantee that there are tracked points in each $W \times W$ neighborhood, in every frame, if there aren't any points, a new feature point is sampled. Table 3.1 shows the parameters and their default values, which are used in the original implementation [44].

3.2.3.5. Improved Trajectories. Wang et al. [46] improved the dense trajectory features by compensating for the camera motion by calculating the homography matrix between

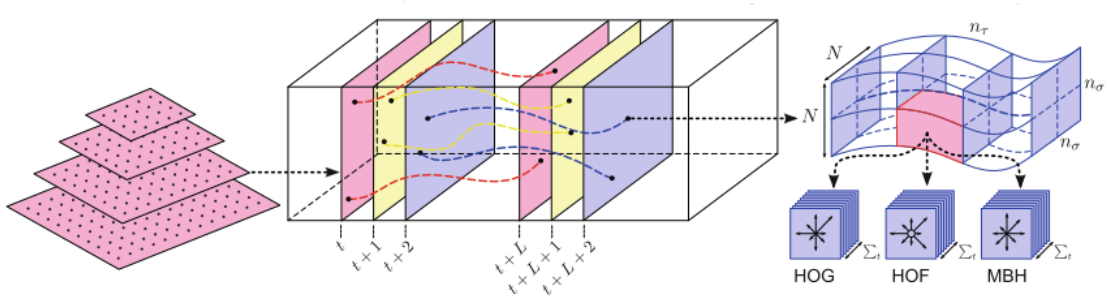

Figure 3.5. (Left) Feature points are densely sampled from multiple scale; (Middle) Tracking is done for each scale; (Right) $N \times N$ trajectory neighborhood is divided into $n_\sigma \times n_\sigma \times n_\tau$ cells and the descriptors (HOG, HOF, MBH) are extracted [44].

successive frames. The inspiration for this improvement was to get rid of the false trajectories, which are created by camera motion in realistic videos. These improved features have yielded promising results on action recognition datasets, but since in emotion recognition we are trying to capture partial facial motions this descriptor can be useful for tracking changes in facial dynamics as well. Figure 3.6 visualizes the trajectories on some videos from the EmotiW Challenge and from the Cohn-Kanade Dataset. This visualization prepared with the software provided by [46]. The tracked points in the current frame are given as red dots, and the motion of each such point is indicated with a green line. In the following we will discuss finding camera motion by means of the homography matrix.

Table 3.1. Dense trajectory parameters.

Parameter	Definition	Default
W	sampling step of the regular grid	5 pixels
Σ	height of the multiscale pyramid for sampling	8
s	factor between spatial scales	$1/\sqrt{2}$
L	tracking time	15 frames
N	size of the volume for computing descriptors	32 pixels
n_σ	number of divisions of the descriptor volume in spatial space	2
n_τ	number of divisions of the descriptor volume in temporal space	3

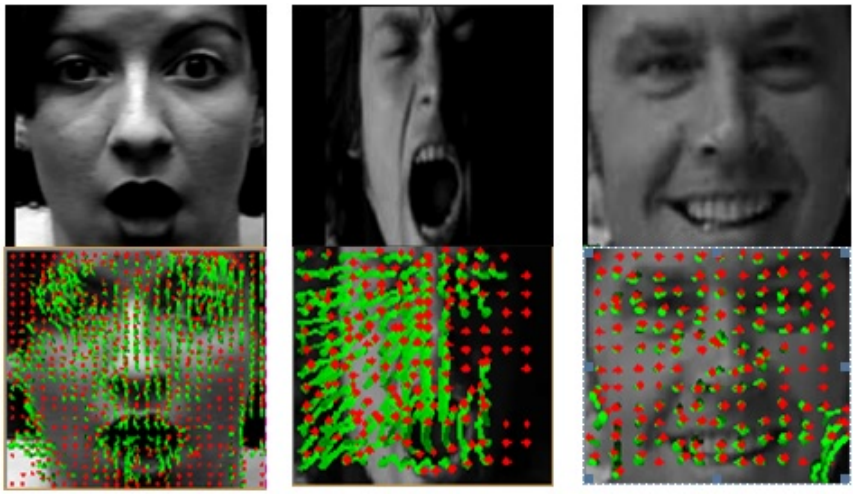

Figure 3.6. Original video frames (first row) and their visualized improved dense trajectories (second row). Images are selected from the CK+ and EmotiW 2015 Challenge datasets. Best viewed in color.

Homography Matrix. Projective transformations have been described with 3×3 homography matrices. If two images like x and $x^{'}$ are on the same planar surface, they can be related by homography matrix. It is the same for the two successive frames of video, since there is a tiny motion from the current frame to the next one; it is assumed that they have a homography relation. If R_1 and R_2 show rotation matrix of camera, the relation between $3D$ world coordinates X and the $2D$ image plane defines as follows:

$$x = K\left[R_1 0\right] X \tag{3.3}$$

$$x^{'} = K\left[R_2 0\right] X \tag{3.4}$$

$$x^{'} = K R_2 R_1^{-1} K x \tag{3.5}$$

$$x^{'} = H x \tag{3.6}$$

where H is the homography and K is the camera intrinsic matrix, which depends on the focal length and camera center. x and $x^{'}$ are in homogeneous coordinates.

Feature Matching. For computing transformation we should first find the correspondence between two images. A unique descriptor should be extracted from both images and then a similarity measure will be used to match them according to the matching algorithm. In improved trajectories [46], both Speeded Up Robust Features (SURF) [73] matches and motion vector matches have been used.

RANSAC. A famous algorithm for efficiently estimating homography matrix is Random Sample Consensus (RANSAC) [74]. It is an iterative method, which has the ability to remove the effect of outlier matches. A random sample of feature correspondences is selected in every iteration. Afterward each correspondence is consigned as being inlier or outlier according to the current estimation of homography matrix. At the end of iterations, the homography with the largest number of inliers is selected as the approximation.

3.3. Local Feature Aggregation

Two popular approaches that produce a compact vector representations with a fixed sized from a set of local descriptors are bag of features [47–49] and Fisher vectors [75]. For making this compact form, old methods train local descriptors and learn a codebook called visual vocabulary by means of clustering. Then each local descriptor is encoded according to its distance to centroids, which are the means of codebook clusters.

3.3.1. Unsupervised Clustering Methods

Categorizing unlabeled data into similarity groups is called clustering. A cluster is a group of similar data samples, which are different from samples in other clusters. In machine learning, visual vocabularies are typically learned by using unsupervised clustering methods. In this section, we discuss two popular clustering methods, namely k-means and Gaussian mixture models. According to the encoding method, the appropriate clustering technique is used.

3.3.1.1. K-means. K-means is a centroid clustering algorithm. It proposed by Macqueen in 1967 [76]. K-means clustering intends to divide the samples into k clusters in a way that each sample belongs to the cluster with the minimized squared error function, which is defined as follows:

$$\arg\min_{C} \sum_{i=1}^{K} \sum_{x \in c_i} \parallel x - \mu_i \parallel^2 \tag{3.7}$$

where $C = \{c_1, \cdots, c_K\}$ are clusters and μ_i is the mean of cluster c_i and x are data points in cluster c_i.
The algorithm for optimizing the error function is as follows:

(i) Choose cluster centers randomly.
(ii) Find the distance of each data sample to the cluster centers.
(iii) For each data point, find the cluster center, which has the minimum distance to it.
(iv) Update the current cluster center by:

$$\mu_i = (1/N_{c_i}) \sum_{j=1}^{N_{c_i}} x_j \tag{3.8}$$

where N_{c_i} signifies the number of data points in i^{th} cluster.
(v) Recalculate the distance between each data point and new cluster centers.

(vi) Stop if there is no reallocation, otherwise repeat from step 3.

K-means is fast and robust and gives the best result when data points are well separated from each other. It has also some disadvantages, for example the number of data groups may be unknown, and for finding the best K, multiple candidates should be tested experimentally. The other drawback is that K-means provides a local optimum of the error function and so the procedure should be repeated multiple times.

3.3.1.2. Gaussian Mixture Model. A Gaussian mixture model (GMM) is a more general version of k-means clustering method. K-means use spherical covariances and equal priors and the cluster memberships are hard, while GMM clustering is probabilistic. GMM is a generative model which tries to fit a number of Gaussian components on the data. The covariance matrices Σ_i can be full rank or limited to be diagonal. The amount of data accessible for estimating the GMM parameters determine number of components, full or diagonal covariance matrices. The iterative Expectation-Maximization (EM) algorithm is usually used to estimate the GMM parameters from training data [77]. GMM works well when data groups have dissimilar sizes and different covariance structure between clusters. Different from k-means, data are clustered according to the probability and not by hard assignment to the cluster centers.

A Gaussian mixture model consists of three parameters $\lambda = \{\omega_i, \mu_i, \Sigma_i\}$: mixing weights, means and variances, respectively. For D-dimensional vectors a GMM model is defined as:

$$f(x) = \sum_{i=1}^{K} \omega_i g\left(x \mid \mu_i, \Sigma_i\right) \tag{3.9}$$

where x is a D-dimensional feature vector and $g\left(x \mid \mu_i, \Sigma_i\right)$ is a Gaussian component of the form:

$$g\left(x \mid \mu, \Sigma\right) = (2)^{-D/2} \det\left(\Sigma\right)^{-1/2} \exp\left\{-\frac{1}{2}\left(x-\mu\right)^T \Sigma^{-1}\left(x-\mu\right)\right\} \tag{3.10}$$

Numerous techniques can be used for estimating the parameters of a GMM. The most prominent method is maximum likelihood (ML) estimation [78]. The goal of ML estimation is to find the model parameters in a way that maximizes the likelihood of the GMM given the training data. Since the function is a non-linear function, ML parameters can be estimated by using the expectation-maximization (EM) algorithm. It is an iterative algorithm alternating between expectation and maximization steps. In the expectation step, current values of parameters are used to calculate the log-likelihood. In the maximization step, the parameters are re-computed maximizing the expected log-likelihood and the procedure is repeated until reaching to some convergence threshold.

In the EM algorithm, the succeeding updates are applied on each component i to learn the parameters of a mixture, given data x.

$$P\left(i|x_j\right) = \omega_i g\left(x_i \mid \mu_i, \Sigma_i\right) / f\left(x_i\right) \tag{3.11}$$

$$\omega_i = \sum_{j=1}^{N} P\left(i|x_j\right) / N \tag{3.12}$$

$$\mu_i = \sum_{j=1}^{N} P\left(i \mid x_j\right) x_j / \left(N\omega_i\right) \tag{3.13}$$

$$\Sigma_i = \sum_{j=1}^{N} P\left(i|x_j\right) \left(x_j - _i\right) \left(x_j - _i\right)^T / \left(N\omega_i\right) \tag{3.14}$$

On each EM iteration, the above re-estimation formulas guarantee a monotonic increase in the model's likelihood value However, EM catches a local peak, so the initialization is a vital step to find the global peak [79].

3.3.2. Bag of Features

Bag of features (BOF) model (or bag of words) is a technique that works by treating image features as words and was originally developed for text retrieval problems [80]. This method is extensively used in computer vision [81, 82].

Local descriptors are grouped by the BOF model. A clustering method, which is typically k-means, is used to construct a codebook of k visual words from the training set. For a new instance of data, local descriptors are extracted and assigned to the closest cluster centroid. Then, all image descriptors are assigned to visual words and the BOF representation is obtained by occurrence counts of words in a histogram format. The result is a k-dimensional vector, which should be normalized for further usage in classification. There are various methods for normalizing the histogram, such as the Manhattan distance-based normalization and Euclidean normalization, which are also known as L1 and L2 normalization, respectively. The formal definitions of L1 and L2 norms are represented in Equations 3.15 and 3.16, respectively. Once the L1 and L2 normalization factor are obtained normalization is done as 3.17. A number of variations have been offered to improve the quality of BOF representation [83, 84] by using soft quantization techniques instead of k-means. the overview of the BOF method is illustrated in Figure 3.7.

$$\| z \|_1 = \sum_{i=1}^{n} | z_i | \tag{3.15}$$

$$\| z \|_2 = \sqrt{\sum_{i=1}^{n} | z_i |^2} \tag{3.16}$$

$$z_i = \frac{z_i}{\| z \|} \tag{3.17}$$

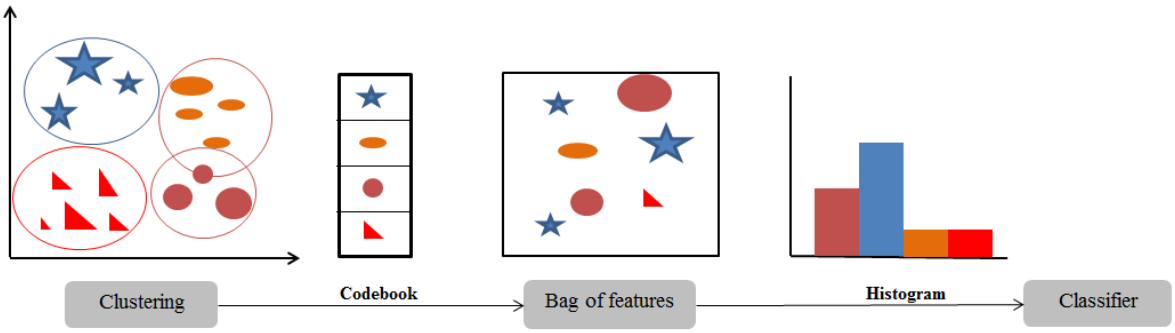

Figure 3.7. Bag of features.

3.3.3. Fisher Vector Encoding

Fisher vector (FV) representation can be named as an extension of BOV. Perronnin and Dance [75] proposed the usage of GMM and Fisher kernels for making visual vocabularies. Since then FV has attracted more attention. The Fisher vector benefits from characteristics of both generative statistical models like hidden Markov model (HMM) and discriminative methods like support vector machines (SVM). Unlike BOF, fisher vector uses both 0-order statistics (counting) and second order statistics. This specification enables the Fisher vector to find the best direction in which parameters of GMM model are modified in order to fit to the data efficiently [82]. For this goal, the gradient of the likelihood is encoded by applying derivative operations with respect to the distribution parameters of the vocabulary. Then the differences between pooled local features and dictionary items are kept.

FV has some advantages over other aggregation methods. One advantage is that FV benefits from both generative and discriminative methodologies. This property causes appreciable performance by just using a simple linear classifier. The other key superiority of FV over BOF is that acceptable performance is obtained by using much fewer vocabulary components.

In this work, in order to encode the descriptors with FV, prior to building the dictionary, first of all principal component analysis (PCA) should be applied in order to both reduce the dimensionality of descriptors and make them decorrelated in order

to support diagonal covariance matrices assumption which is considered here.

In this study, in the first step vocabularies are made by means of a K component GMM, which is trained over training features. Then, the learned parameters in Equation 3.18 have been used for query instances.

$$\lambda = \{\omega_i, \mu_i, \Sigma_i\} \mid_{i=1}^{k} \tag{3.18}$$

Given a set of descriptors $X = \{x_1, x_2, \cdots, x_N\}$, the following gradient vector can be defined in terms of a distribution with parameters λ,

$$\nabla_\lambda \log p\left(X \mid \lambda\right) \tag{3.19}$$

where the distribution p is a GMM and its components keep information about frequency, mean and variation of a visual word. Under an independence assumption,

$$L\left(X \mid \lambda\right) = \sum_{j=1}^{N} \log p\left(x_j \mid \lambda\right) \tag{3.20}$$

$$= \sum_{j=1}^{N} \log \left(\sum_{i=1}^{N} \omega_i p_i \left(x_j | \lambda\right) \right) \tag{3.21}$$

where $p_i\left(x_j \mid \lambda\right)$ is calculated by Equation 3.10.

The Fisher vector encoding is obtained by taking the gradient of L with respect to μ and Σ parameters. The gradient with respect to ω carries little extra information, so it is discarded. Normalized partial derivatives of means and deviations are estimated as follows:

$$u_i = \frac{1}{N\sqrt{\omega_i}} \sum_{j=1}^{N} \gamma_{ij} \left(\frac{x_j - \mu_i}{\sigma_i} \right) \tag{3.22}$$

$$v_i = \frac{1}{N\sqrt{2\omega_i}} \sum_{j=1}^{N} \gamma_{ij} \left[\left(\frac{x_j - \mu_i}{\sigma_i} \right)^2 - 1 \right] \tag{3.23}$$

where γ_{ij} represents the posterior probability correlating each vector x_j with a component i in the GMM and $\sigma_i^2 = diag(\Sigma_i)$. Finally, concatenation of the vectors u_i and v_i makes the FV.

After that, a visual vocabulary is created by means of GMM. In our experiments we used the Fisher vector, which is normalized firstly by the signed square root function and secondly by L2 normalization [85]. The final dimensionality of FV is $2 \times D \times K$ where D is the dimensionality of the descriptor and K is the number of GMM components. In this work, we used FV representations of HOG, HOF, MBH, Trajectories and Geometric features to model our videos. We used various dimensionalities and different combinations of Fisher representations of descriptors, which will be discussed in detail in chapter 6.

4. CLASSIFICATION

After obtaining video features by using the given emotion labels for the training set, classification can be done by running a machine learning method. There are several classification algorithms, among which support vector machines (SVM) are commonly used in the context of video based emotion recognition. The other powerful method for emotion recognition is Extreme Learning Machine (ELM), which is not used as frequently as SVM. Actually ELM is a special case of LS-SVM where the bias term is set to zero. In this chapter, a summary of SVM and the basics of ELM will be explored.

4.1. Support Vector Machine

Cortes and Vapnik proposed a supervised learning algorithm called support vector machines in 1995 [86], which has been used for both classification and regression. This method is based on finding a discriminant without modeling densities generatively. Actually support vector machines are an extension to nonlinear models of the algorithm developed by Vladimir Vapnik [87].

The purpose of SVM is to find a higher dimensional space in which feature vectors are able to be discriminated linearly. This is due to the reason that usually features from two different classes are not linearly separable in their original finite dimensional space. In the SVM context, training features which lie close to the class boundaries are called "support vectors", which are used as SVM model parameters.

Finding the best hyperplane is a goal for SVM. This hyperplane should maximize the margin between two classes. SVM is applied on two-class classification problems, however recent versions are applicable on multi-class classification problems as well.

There are many kernel-based algorithms in machine learning, and SVM is one of them. By means of the kernel trick, SVM is able to map data points to a higher dimensional space. Kernel-based algorithms are formulated as convex optimization

problems; therefore there is a single optimum to be found [88].

The ideal hyperplane, which is able to separate two classes, can be represented by the following formula:

$$\begin{cases} wx_i + \omega_0 \geq +1, & \text{if } y_i = 1 \\ wx_i + \omega_0 \leq -1, & \text{if } y_i = -1 \end{cases} \tag{4.1}$$

where x_i represents the training features and y_i is related to the class label and takes the value $\{-1, +1\}$. Parameters w and ω_0 should be calculated in a way that Equation 4.1 becomes true for all $X\ |_{i=1}^{N}$. This formula can also be written as follows:

$$y_i\,(wx_i + \omega_0) \geq 1 \tag{4.2}$$

In order to maximize the margin, which is defined by $\frac{y_i(wx_i+\omega_0)}{\|w\|}$, we should minimize $\parallel w \parallel$. The optimization process is as follows:

$$min\frac{1}{2} \parallel w \parallel^2 \quad \text{s.t.} \quad y_i\,(wx_i + \omega_0) \geq 1 \quad \forall i \tag{4.3}$$

A standard quadratic optimization solution is sufficient for calculating w and ω_0 and the complexity depends on the data point dimension D.

Most of the time data points are not linearly separable; in this case we need a soft margin, and some support vectors may fall in it. So we need to add a penalty term to Equation 4.2 as below:

$$y_i\,(wx_i + \omega_0) \geq 1 - \epsilon_i \tag{4.4}$$

$$\min \frac{1}{2} \parallel w \parallel^2 + C_{SVM} \sum_i \epsilon_i \tag{4.5}$$

where C_{SVM} is a complexity parameter and $\sum_i \epsilon_i$ is the penalty term.

The dual of this problem can be obtained by using Lagrange multipliers α_i as follows:

$$\begin{aligned} &\max_{\alpha_i} \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j {X_i}^T X_j \\ &\text{subject to} \quad \sum_i \alpha_i y_i = 0 \quad and \quad 0 \leq \alpha_i \leq C_{SVM}, \quad \forall i \end{aligned} \tag{4.6}$$

In this case complexity depends on the number of instances N.

According to the Mercer conditions every (semi) positive definite, symmetric function is a kernel: i.e. there exists a mapping $\phi(x)$ such that it is possible to write: $K = \phi(x_i)^T \phi(x_j)$. Therefore here the factor ${X_i}^T X_j$ in Equation 4.6 is replaced by $\phi(x_i)^T \phi(x_j)$. This dot product can be defined as a kernel function $K(x_i, x_j)$ in the D-dimensional space without actually going to the mapped space of ϕ. Thus, non-linear kernel functions can be integrated to solve more complex problems. Usually, kernel functions are considered to measure similarity; therefore, any valid kernel function can be defined specific to the application.

Some popular kernels are:

(i) Linear kernel

$$K(x_i, x_j) = X_i^T X_j \tag{4.7}$$

(ii) Gaussian radial basis function (RBF) kernel

$$K(x_i, x_j) = \exp\left(- \parallel x_i - x_j \parallel^2 \sigma^2\right) \tag{4.8}$$

(iii) X^2 kernel

$$K\left(x_i, x_j\right) = 1 - \sum_{d=1}^{D} \frac{\left(x_i^{(d)} - x_j^{(d)}\right)^2}{\frac{1}{2}\left(x_i^{(d)} + x_j^{(d)}\right)} \tag{4.9}$$

Libsvm [89] is a popular software, which is created for solving the optimization in Equation 4.6. It is an iterative algorithm and has the ability of using different kernels. Liblinear [90] is also an efficient software in case of linear classification.

4.2. Extreme Learning Machine

Huang et al. [54, 91] proposed a method called extreme learning machine (ELM) for both classification and regression purposes. Recently in the literature, [59] and [92] used ELM for emotion recognition in the wild and obtained promising results.

In this part, we present methods based on ELM. The proposed algorithm works for the generalized single-hidden-layer feed-forward networks (SLFN), but the main difference is that the hidden layer in ELM need not be tuned, but is assumed to be known.

ELM can be used in both classification and regression tasks. It is actually a feedforward neural network with single layer of hidden nodes, in which weights from the input layer to the hidden nodes are initialized randomly, and unlike neural networks, will not be updated with backpropagation. This specification of ELM is the reason of its short training time. In ELM, the mapping from input layer to the hidden nodes is known. Therefore, the weights β, which map hidden nodes to the output nodes can be solved analytically by a least-squares solution.

In order to obtain the formulation for multiclass ELM classifier an optimization problem should be solved, based on two objectives. The first one is minimizing the training error, and the second one is minimizing the norm of the output weights for better generalization. Let $h\left(x\right)$ be the feature mapping from the D-dimensional input

node x to the L-dimensional hidden-layer feature space, and $\beta_{(L\times C)}$ be the output weight matrix between the hidden layer of Lnodes and the output nodes. Then, the optimization takes the form:

$$\min \parallel H\beta - T \parallel^2 and \parallel \beta \parallel \tag{4.10}$$

Where $T_{(N\times C)}$ stands for the training label matrix of N training instances and $H_{(N\times L)}$ is the hidden layer output matrix defined as below:

$$\begin{bmatrix} h(x_1) \\ \dots \\ h(x_N) \end{bmatrix} = \begin{bmatrix} h_1(x_1) & \dots & h_L(x_1) \\ \vdots & \ddots & \vdots \\ h_1(x_N) & \dots & h_L(x_N) \end{bmatrix} \tag{4.11}$$

Assuming H is known, the least square solution of the linear equation $H\beta = T$ becomes:

$$\beta = H^{\dagger}T \tag{4.12}$$

where $H^{\dagger}$ is the Moore-Penrose generalized inverse.

4.2.1. Kernels

We can integrate kernels in extreme learning machines similar to what we did in support vector machines. A kernel function after applying Mercer conditions takes the form in Equation 4.13:

$$\Omega = HH^T \tag{4.13}$$

where $\Omega_{ij} = h\left(x_i\right).h\left(x_j\right) = K\left(x_i, x_j\right)$.

In the kernel space, the hidden-layer output computation becomes a kernel operation. During training, we add a regularization coefficient C_{ELM} to the kernel matrix

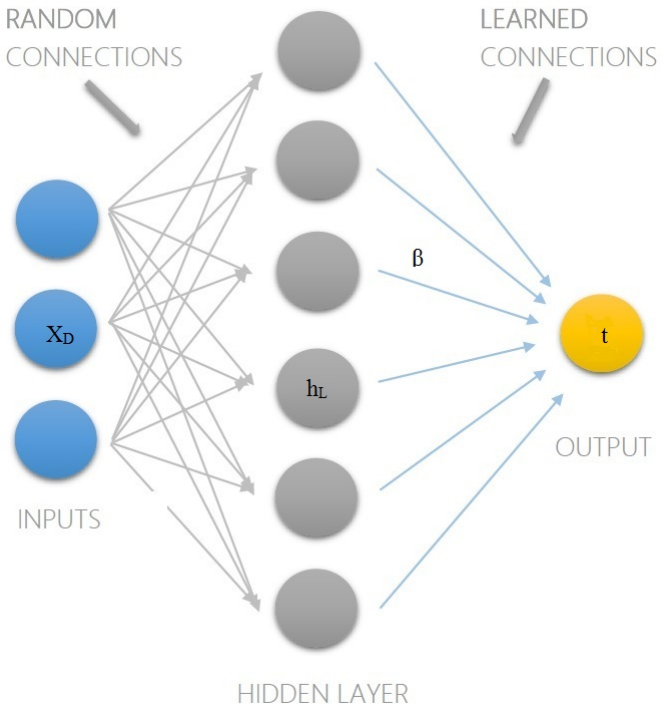

Figure 4.1. ELM architecture, a single-hidden-layer feed-forward network.

calculated with training data $X_{(N\times D)} = [x_1 \dots x_N]^T$ as follows:

$$H = \left(\frac{1}{C_{ELM}} + \Omega\right) \tag{4.14}$$

Any valid kernel can be used for Ω. In the case of a linear kernel, the input features X themselves are feature mapping weights. Therefore, the output of the kernel ELM classifier becomes:

$$f(x) = K(x, X)^T \beta = K(x, X)^T \left(\frac{1}{C_{ELM}} + \Omega\right)^{\dagger} T \tag{4.15}$$

5. PROPOSED METHODOLOGY

5.1. Face and Landmark Detection

Our face detection, facial registration, feature extraction, encoding and classification pipeline is illustrated in Figure 5.2. For landmark localization, we propose to combine two different methods in order to make a robust system. We use the supervised descent method (SDM) [27] in conjunction with the Discriminative Response Map Fitting (DRMF) method [28]. The Intraface (SDM) method is fast (about 0.51 second per frame) and can detect facial features precisely, but since its face detector is based on the Viola & Jones face detector [25], sometimes it fails to find faces. This is a problem in many cases of realistic videos, where different illumination, pose and complicated background conditions are present. The DRMF method, on the other hand, benefits from a tree-based face detector, which is proposed by Zhu & Ramanan [24]. Although the DRMF method works well in the wild conditions, its current implementation is very slow. It takes about 23 seconds to find the face and its corresponding landmarks in a frame. In order to preserve both accuracy and speed, we combine these two methods. The video is first processed by Intraface, using the Viola & Jones face detector and for each frame, faces and their landmarks with high confidence scores are selected. If a given video has less than three frames with detected faces, we use the combination of Zhu & Ramanan and DRMF for face and landmark localization. If this approach also fails to find faces, the video is tagged as not having any faces.

5.2. Generalized Procrustes Alignment

In our proposed system, we use a single reference model and align all faces to it in order to remove translation, rotation, and scale effects. For this purpose, we use the generalized Procrustes analysis (GPA) proposed by [29]. A set of faces are represented by their landmarks and an iterative approach is employed to obtain the reference model. This procedure automatically produces the registered set of faces from the training set at the same time. Given a new facial image, GPA will find the affine transformation

Figure 5.1. Sample aligned faces taken from EmotiW 2015 dataset.

that aligns the face to the reference face, minimizing a distance functions that equally weights each landmark. For details, we refer the reader to [93].

5.3. Improved Dense Trajectory Features

Improved dense trajectory features reach state-of-the-art in the action recognition problem [46]. These features are based on image descriptors (HOG, HOF and MBH descriptors) computed along tracked trajectories. We use these features to capture the changes in facial dynamics. Wang et al. illustrated that tracking densely sampled feature points from a multi-scale pyramid (built from each frame of the video) can outperform sparse sampling. Tracking is based on dense optical flow [71] and is done for a certain time window. In realistic videos, camera motion should be filtered out to prevent generating trajectories which correspond to the background. For solving this problem, Wang et al. proposed using the homography matrix of the points from continuous frames, which is extracted by the RANSAC approach. After filtering out the camera motion, 96-dimensional HOG, 108-dimensional HOF, 192-dimensional MBH and 30 dimensional trajectory features are computed to describe the appearance, shape and motion information.

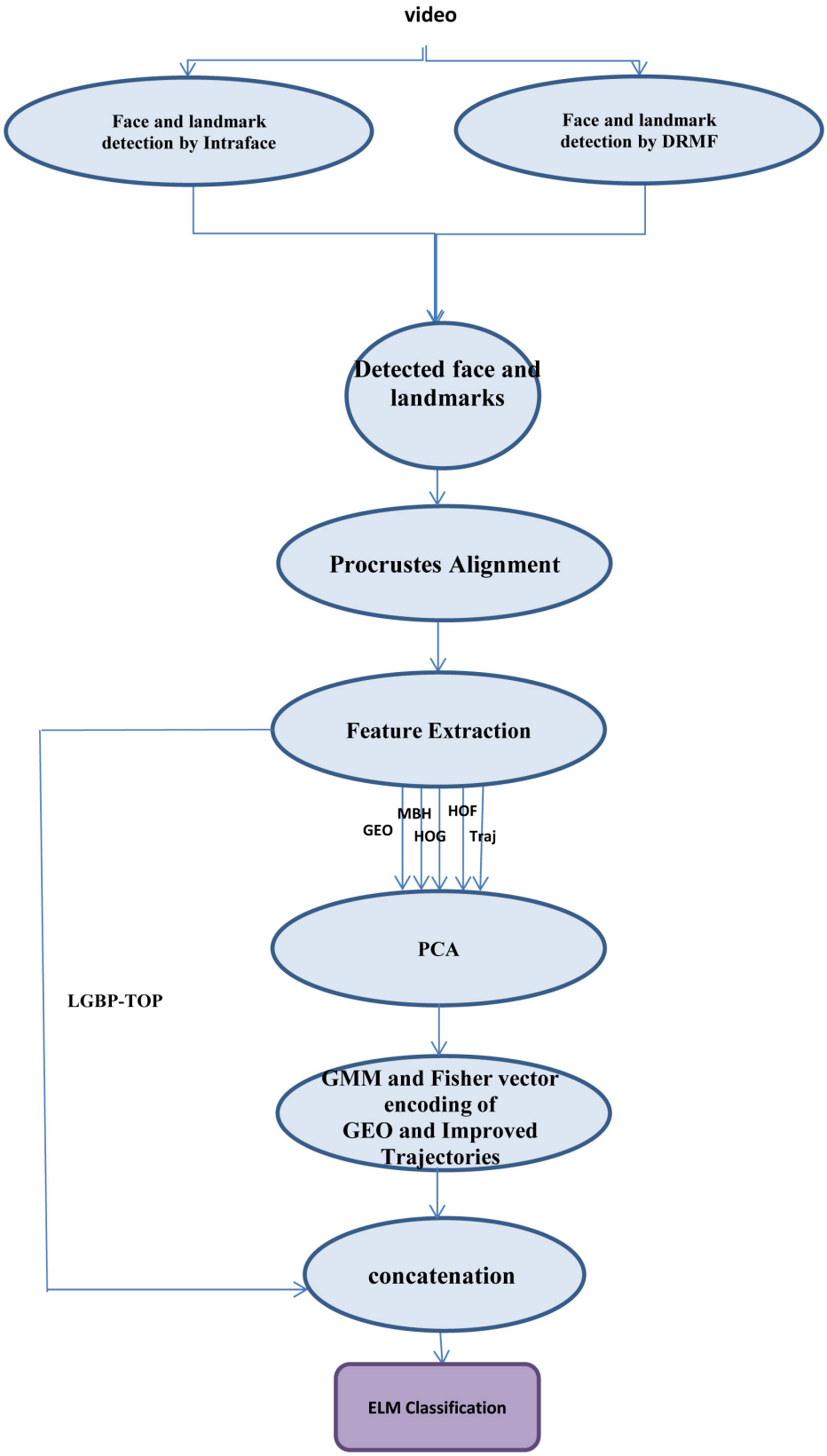

Figure 5.2. Proposed pipeline.

Using improved dense trajectory features has some important advantages over simple tracking of interest points. First, removing camera motion from optical flow improves HOF descriptors as discussed in [44]. Second, canceling out camera motion also removes trajectories that are produced by camera motion. Therefore, only trajectories related to face movements are kept. This specification is very important in real-world videos, where there are lots of pan and tilt camera motions.

We use the software developed by Wang et al. [46] to extract the improved trajectory features with the default parameters. Afterward a random subset of local descriptors is extracted from training videos for making the codebook in the training phase. Since making a codebook by means of all the descriptors needs large amount of memory random selection is used in the literature. Actually, a random uniform sampling will not change the distribution of the descriptors, which are extracted from a video.

Four different GMM models are created for each descriptor (trajectory, HOG, HOF and MBH), separately. Before encoding the feature vector, PCA is applied in order to both reduce the dimensionality of descriptors and also make them de-correlated, which is pivotal to be able to use GMM with diagonal covariance matrix.

In addition to improved dense trajectory features, geometric features are also extracted by using localized landmarks on the face. These geometric features are explained in detail in the next section.

5.4. Geometric Features

The Shape of the face can be captured by its landmarks, and interpreting the movement of the landmarks can improve the performance of an emotion recognition system. In this study, we used the landmarks of the face that are detected by both Intrafce and DRMF methods [27], [28]. Geometric features are mostly the same as the features introduced in [59] by Kaya et al. We have included three more features to enhance this set. Indices of extracted landmarks can be seen in Figure 5.3 and details

of geometric features are tabulated in Table 5.1. After extracting geometric features, PCA is applied to decorrelate the features and a GMM model is learned from the training part.

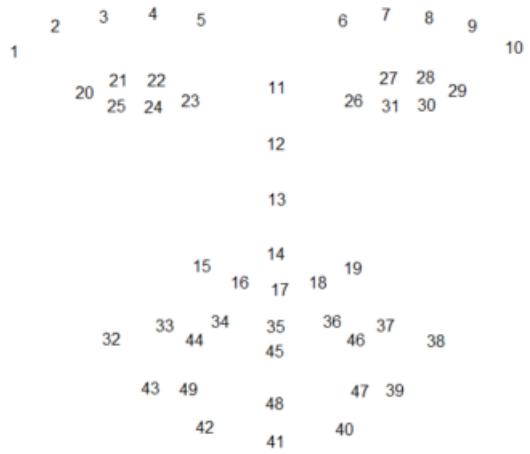

Figure 5.3. Order of the located landmarks.

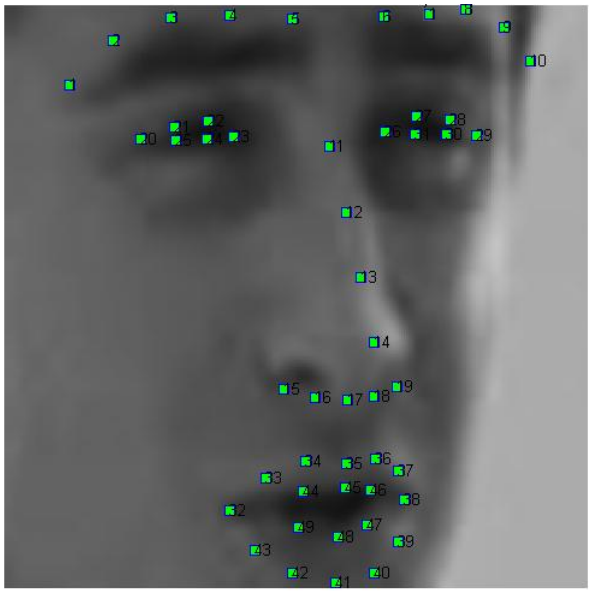

Figure 5.4. Landmarks extracted from the face (EmotiW 2015).

5.5. Local Gabor Binary Patterns From Three Orthogonal Planes

The final feature we incorporate is LGBP-TOP [34]. For this feature, images are convolved with a set of 2D complex Gabor filters to obtain Gabor-pictures, and then LBP-TOP is applied to each Gabor-picture. A 2D complex Gabor filter is defined

Table 5.1. Explanation of geometric features.

#	**Features**	**Type and Explanation of feature**
1	Eye aspect ratio	Distance, averaged over left and right parts of the face
2	Mouth aspect ratio	Distance
3	Upper lip angles	Angle, averaged over left and right parts of the face
4	Nose tip - mouth corner angles	Angle, averaged over left and right parts of the face
5	Lower lip angles	Angle, averaged over left and right parts of the face
6	Eyebrow slope	Angle, averaged over left and right parts of the face
7,8	Lower eye angles	Angle, averaged over left and right parts of the face
9	Mouth corner - mouth bottom angles	Angle
10	Upper mouth angles	Angle, averaged over left and right parts of the face
11	Curvature of lower-outer lips	Curvature, averaged over left and right parts of the face
12	Curvature of lower-inner lips	Curvature, averaged over left and right parts of the face
13	Bottom lip curvature	Curvature
14	Mouth opening	Distance
15	Mouth up/low	Distance
16	Eye - middle eyebrow distance	Distance, averaged over left and right parts of the face
17	Eye - inner eyebrow distance	Distance, averaged over left and right parts of the face
18	Inner eye - eyebrow center	Distance, averaged over left and right parts of the face
19	Inner eye - mouth top distance	Distance
20	Mouth width	Distance
21	Mouth height	Distance
22	Upper mouth height	Distance
23	Lower mouth height	Distance
24	Inner eye - mouth corner distance	Distance
25	Mouth center-left mouth corner	Distance
26	Mouth center-right mouth corner	Distance

as the convolution of a complex sinusoid $s(x, y)$ (carrier) with a 2-D Gaussian kernel $\omega_\tau(x, y)$ (envelope):

$$g(x, y) = s(x, y)\,\omega_\tau(x, y) \tag{5.1}$$

$$s(x, y) = \exp\left(j\left(2\pi\left(u_0 x + v_0 y\right) + p\right)\right) \tag{5.2}$$

where (u_0, v_0) stands for spatial frequency and p defines the phase of the sinusoid.

$$\omega_\tau(x, y) = K \exp\left(-\pi\left(a^2 (x - x_0)_r^2 + b^2 (y - y_0)_r^2\right)\right) \tag{5.3}$$

where a, b are scaling parameters of the Gaussian, K is amplitude and r subscript stands for a clockwise rotation around point (x_0, y_0) such that:

$$(x - x_0)_r = (x - x_0)\cos\theta + (y - y_0)\sin\theta \tag{5.4}$$

$$(y - y_0)_r = -(x - x_0)\sin\theta + (y - y_0)\cos\theta \tag{5.5}$$

According to [34] we take $a = b = \sigma$, $u_0 = v_0 = \phi$ and K = 1. It should be stated that we only use the magnitude response of the filter.

For this dissertation, LGBP histograms from three orthogonal planes (XY, XT, and YT, respectively, with X and Y representing the image plane, and T representing time) are extracted from two equal length volumes of the video, which are obtained by dividing the video over the time axis. The resulting features are concatenated in order to form the final feature vector. We take the idea of dividing the video for improving temporal modeling from Kaya et al. [59], and the Gabor pictures were obtained using an open source script [94]. Three scales and six orientations are used to prepare the Gabor filter bank, and each Gabor picture is divided into blocks. We have used 4

blocks for experiments on CK+, and 16 blocks for the EmotiW 2015 Challenge Dataset, respectively. Our approach is robust to operational parameters, and since CK+ has a smaller number of samples to train, reducing the number of blocks slightly improves generalization.

5.6. Fisher Vector Encoding

In this work, before making GMM models, we used PCA in order to reduce the dimensionality of descriptors and for decorrelating them. Empirically, we got the best results on the training set by reducing the dimension of each trajectory to 25 and that of the other three descriptors (HOG, HOF, and MBH, respectively) to 64. The geometric features are projected to a decorrelated space by PCA, while their full dimensionality is kept.

We used GMMs with diagonal covariance matrix to produce the FV. The GMM clustering produces a visual vocabulary, where the number of clusters is a parameter of the method optimized on the training set. 20 to 64 clusters work well for each of the feature categories. In our experiments, the Fisher vectors are normalized firstly by the signed square root function, and secondly by L2 normalization. By this normalization process the usage of linear classifier is more successful. The final dimensionality of FV is $2 \times D \times K$, where D is the dimensionality of the descriptor, and $K$is the number of GMM components. In this work we used FV representation of HOG, HOF, MBH, Trajectories, Geometric features combined with LGBP-TOP features to obtain a global video description. The final feature vector is given to ELM, and emotion label for each video is predicted. We have tried various dimensionality and different combination of descriptors, which will be discussed in detail in the experiment part.

5.7. Classification

For a video with extracted feature vector, the multi-class ELM yields the confidence score for each emotion class depending on whether the video belongs to that class. The video is assigned to the emotion class with the highest confidence score.

There are different versions of ELM, in this study we used ELM with a linear kernel, which is simple and has good performance when combined with Fisher vectors. The hyper parameter C_{ELM} is optimized during the training part in order to obtain the best set of parameters. Since the training process of ELM takes a short time, a lot of possibilities for C_{ELM} $\left(2^{\{-16:1:17\}}\right)$have been tested. In the experiment part, we show that using Kernel ELM is more efficient than SVM.

6. EXPERIMENTS

6.1. Datasets

In this section, we describe the datasets, which we have used in our experiments. Possessing the adequate number of labeled facial emotion datasets is an essential precondition to form a successful automatic emotion recognition system. Most of the existing researches on emotion recognition have been based on the data sets of purposely expressed emotions, obtained by asking the participants to perform a sequence of emotional expressions in front of a camera. More recent approaches consider data collected "in-the-wild", with naturally occurring facial expressions and uncontrolled pose and illumination conditions.

6.1.1. The Extended Cohn-Kanade Dataset

The Cohn-Kanade dataset includes 100 university students as subjects, ranging in age from 18 to 30 years. Sixty-five percent were female, fifteen percent African-American, and three percent Asian or Latino. The dataset consists of 23 facial displays including single action units (AUs) and combinations of AUs. Action units (AUs) represent the muscular activity that produces facial appearance changes defined in Facial Coding System by Ekman and Friesen [95]. The dataset contains six universal facial expressions, namely anger, disgust, fear, happiness, sadness, and surprise [10].

The CK+ dataset is the extension of the Cohn-Kanade dataset [11]. It has been further enlarged to include 593 sequences from 123 subjects for seven expressions (additional 107 sequences, 26 subjects, and the contempt expression), which makes it more challenging than the original dataset. The sequences are recorded in laboratory conditions and coded at the peak frame with the facial action coding system (FACS) [95]. All the videos start from the neutral face and end with the apex expression. Among these, only 327 samples have emotion labels, which are used in our experiments [11]. In order to be able to compare our results with the state-of-the-art, Leave-One-Subject-

Out protocol (LOSO). In this protocol, in each iteration samples of one subject are used as the test set and the rest are used in the training part. Some sample images from CK+ dataset are illustrated in Figure 6.1.

Table 6.1 compares several state-of-the-art approaches on the CK+ dataset, which are obtained with the same standard protocol. As it can be seen in the Table 6.1, the baseline recognition rate of CK+ dataset is 88.38% and the best state-of-the-art result is 99.2%.

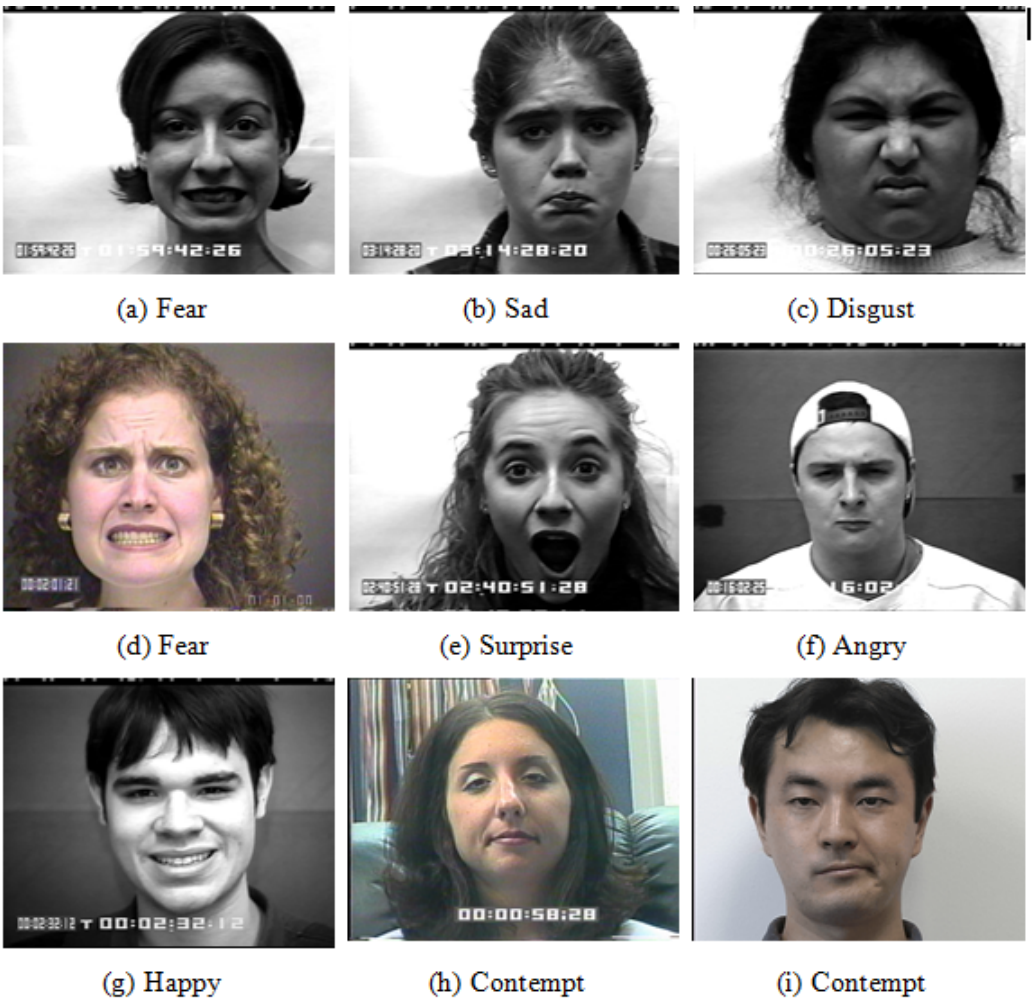

Figure 6.1. Overview of facial expression in the CK+ dataset.

6.1.2. EmotiW 2015 Challenge Dataset

The audio-video Emotion Recognition In The Wild Challenge (EmotiW) took place in order to investigate the power of emotion recognition systems when applied on

Table 6.1. State of the art results on the CK+

Algorithm	Protocol	Mean Rec. R
STPS+CAPP (baseline, Lucey et al., 2010 [11])	LOSO	88.38%
STLMBP (Huang et al., 2012 [96])	LOSO	92.62%
Cov3D (Sanin et al., 2013 [97])	LOSO	92.30%
RCC (Huang et al., 2014 [7])	LOSO	95.38%
LCRF (Walecki et al., 2015 [98])	10-fold	93.90%
PPDN (Zhao et al.,2016 [99])	10-fold	99.2%

real word videos for simulating the real-world conditions. The term "in the wild" implies inconsistency in environments/scenes and backgrounds, illumination conditions, head pose, occlusion etc. EmotiW 2015 challenge dataset contains videos that are taken from movies and mimic the real world condition. In the past, all the emotion recognition tasks have been done on datasets captured in constrained laboratory environments. Although these laboratory-controlled datasets played an important role in improving the performance of the facial expression recognition systems, they were not able to represent different environments and the conditions of real-world situations. By an exponential growth in the number of videos which are accessible online, it is valuable to investigate the performance of emotion recognition techniques that work 'in the wild'. The aim of EmotiW 2015 Challenge was to expand the data defined during EmotiW 2014 for evaluation of emotion recognition methods in real-world conditions. EmotiW 2015 has been collected from movies with close-to-real-world conditions.

The EmotiW 2015 Challenge Dataset consists of 723 training, 383 validation and 539 test videos. Videos are collected via a semi-automatic approach with a video clip recommender system, which is based on subtitle parsing. The labelers did not scan the full movie manually but used the recommender system, which suggests only those video clips, which have a high probability of a subject showing a meaningful expression. The mission is to give a particular emotion label to the video clip. Emotion labels are from the six universal emotions (Anger, Disgust, Fear, Happiness, Sad and Surprise) and Neutral. Video clips of the dataset are divided into three partitions, namely, train, validation, and test sets. The labels of the test videos are sequestered, and the number

Table 6.2. Numbers of samples for each emotion class (EmotiW 2015).

	Angry	Disgust	Fear	Happy	Neutral	Sad	Surprised
Train	118	72	77	145	131	107	73
Validation	64	40	46	63	63	61	46
Test	79	29	66	108	159	71	27

Table 6.3. State of the art results on the validation partition of the EmotiW 2015.

Algorithm	Accuracy
LBP-TOP (Baseline) (Dhall et al., 2015 [17])	36.08%
LPQ+LBP-TOP+OpenSmile (Kayaoglu et al., 2015 [100])	40.70%*
AU-AFF (winner of the challenge) (Yao et al., 2015 [101])	49.09%*
RNN (Ebrahimi et al., 2015 [102])	39.60%

of evaluations are limited.

The organizers of the challenge have provided aligned set of faces for each video clip. In the given alignment, the face is localized using the Zhu & Ramanan [24] method and tracking is achieved by means of the IntraFace library [27]. Faces are aligned by the landmarks produced through IntraFace.

Since emotion labels of the test set are sequestered, we use cross validation on the training set to find the best parameters and then test the proposed method on the validation set. Table 6.3 compares several approaches on the validation set of EmotiW 2015 dataset.

The video-only 36.08% baseline accuracy reported by the organizers is achieved by extracting LBP-TOP features from non-overlapping spatial 4×4 blocks. Then these local features are concatenated to form a general feature vector and the final feature vector is classified by Chi square kernel based SVM [17].

During our experiments, we analyze different combinations of features and also compare each block of our pipeline with other methods. We test each step of our proposed methodology to see how much each block contributes to the final result.

Figure 6.2. Illustration of sample frames taken from EmotiW 2015 dataset.

6.2. Comparison of Descriptor Types

Several experiments have been done in order to find the best combination of descriptors. We investigate the contribution of each descriptor both individually and in combination with others. We learned PCA and GMM models from each descriptor (HOG, HOF, MBH, improved trajectories, and geometric features), separately.

Results on the CK+ dataset are shown in Table 6.4. Concatenation of LGBP-TOP (after dimensionality reduction and power-L2 normalization) and Fisher-encoded HOG, HOF and GEO yields 94.80% and 95.79% (without contempt) accuracy on the

CK+ dataset, which is among the best results obtained for this dataset so far. HOG is the most successful individual feature type in discriminating the emotion classes on this dataset. As expected, the combination of an appearance based feature (HOG) and a motion based feature (HOF) produces higher accuracy than combining two motion based (HOF and MBH) features. Joining only two descriptors (HOG and HOF) already gives a very promising result (93.58%) compared to baseline method [11]. Among the seven classes, "sad" is the most challenging emotion to recognize and "happy" is the easiest one. Table 6.1 compares several state-of-the-art approaches on the CK+ dataset. The confusion matrix of the final system is shown in Figure 6.3.

Table 6.4. Contribution of different descriptors (CK+).

Descriptor	Dimension	Mean Accuracy
Trajectory	1250	71.56%
HOG	8192	90.52%
HOF	8192	87.77%
MBH	8192	89.91%
HOG+HOF	8192+8192	93.58%
HOG+MBH	8192+8192	92.05%
HOF+MBH	8192+8192	90.52%
HOG+HOF+MBH	8192+8192+8192	93.27%
Traj+HOG+HOF+MBH	1250+8192+8192+8192	91.13%
GEO	1352	69.42%
LGBP-TOP	75168	86.24%
GEO+HOG+HOF+ LGBP-TOP(RN)	1352+8192+8192+326	**94.80%**
GEO+HOG+HOF+ LGBP-TOP(RN) (without contempt)	1352+8192+8192+326	**95.79%**

We use the same procedure on the EmotiW 2015 dataset; results are shown in Table 6.5. Again, the combination of HOG and HOF yields the best performance among improved trajectory features and produces higher recognition rate compared to the baseline on the validation set. Our final approach achieves 43.39% accuracy on the

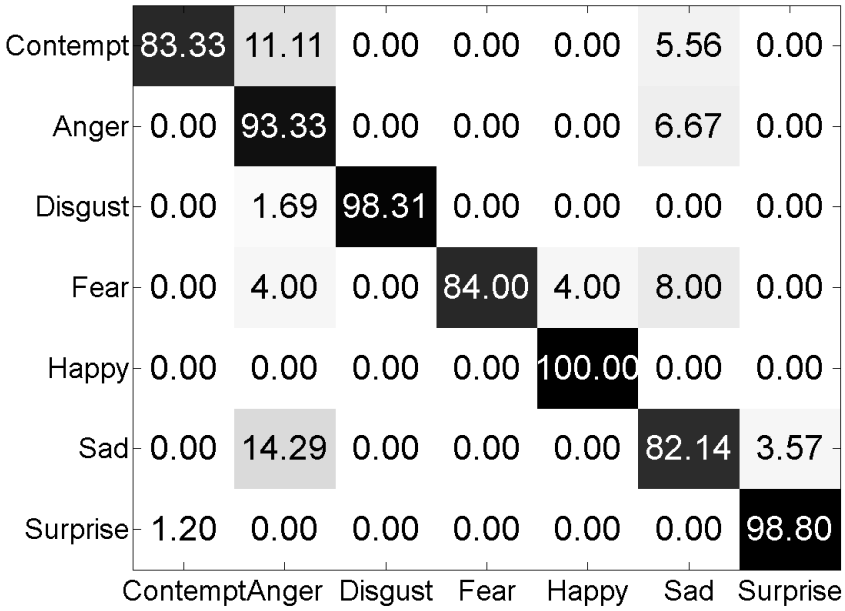

Figure 6.3. Confusion matrix of the final system (CK+).

Emotiw 2015 validation set, which is 7.31% higher than the baseline (36.08%). The best result is obtained by a combination of Fisher encoded geometric, HOG, HOF, MBH and LGBP-TOP (after dimensionality reduction and power-L2 normalization) features. Table 6.3 shows several approaches on the validation set of EmotiW 2015 dataset. The results that are marked with an asterisk are not completely comparable with the results reported here, since they do not follow the same protocol. The confusion matrix of our final system is shown in Figure 6.4.

As stated before, the combination of a motion-based feature (MBH or HOF) and a appearance based feature (HOG) resulted in better performance than joining two motion-based features. Furthermore, Geo and HOG are the most successful individual features in discriminating the emotion classes on the EmotiW 2015 dataset. Among the EmotiW 2015 emotion classes, Disgust and Surprise are the most difficult expressions and Anger is the easiest emotion to recognize.

Table 6.5. Contribution of different descriptors and standard deviations among classes (EmotiW 2015).

Descriptor	Dimension	Accuracy	s.d.
Trajectory	1250	26.72%	22.98%
HOG	8192	34.13%	26.83%
HOF	8192	32.28%	29.61%
MBH	8192	31.22%	20.49%
HOG+HOF	8192+8192	36.77%	28.68%
HOG+MBH	8192+8192	34.92%	25.73%
HOF+MBH	8192+8192	29.63%	20.64%
HOG+HOF+MBH	8192+8192+8192	34.92%	26.30%
Traj+HOG+HOF+MBH	1250+8192+8192+8192	33.86%	27.79%
GEO	1404	38.10%	27.17%
LGBP-TOP(RN)	712	32.28%	25.81%
GEO+HOG+HOF+ LGBP-TOP(RN)	1404+8192+8192+712	41.53%	27.20%
GEO+HOG+HOF+ MBH+LGBP-TOP(RN)	1404+8192+8192+8192+712	**43.39%**	33.03%

6.3. Effect of Facial Alignment

In the provided alignment by the organizers of the challenge, faces are detected only in 711 training and 371 validation videos. There are false positives due to challenging conditions of sequences. Our proposed alignment pipeline was able to detect 713 faces in the training set and 378 faces in the validation set, with a small amount of false positives, in a completely automatic manner. The given alignment by the challenge organizers is good for frontal faces, but the alignment was not very efficient in the case of rotations. With our proposed method, we were able to improve the alignment.

In order to investigate how our registration pipeline improves the recognition performance, we apply the same procedure on the registered images that are provided by the organizers of EmotiW 2015 challenge. We were not able to extract geometric

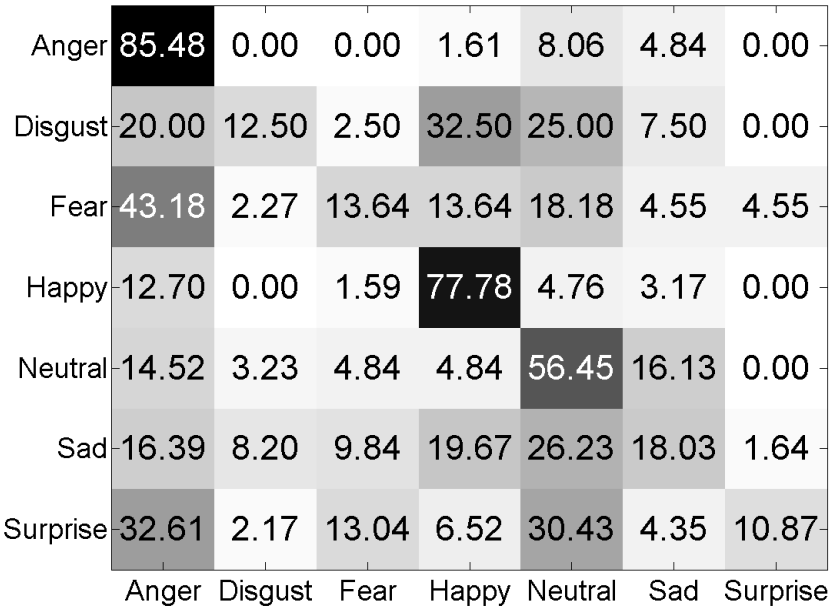

Figure 6.4. Confusion matrix of the final system (Emotiw 2015).

features from the provided alignment, since for a considerable number of frames, there are no landmarks, and for the rest, the number of detected landmarks is not consistent. Therefore, Fisher vector encoding of HOG, HOF, MBH concatenated with LGBP-TOP are used as feature vector. We obtain 38.54% accuracy on the validation set with the default alignment. Using the improved landmark detector and the generalized Procrustes alignment improves this result by 4.85%. Figure 6.5 shows some of the cases where the given alignment by the challenge organizers fails while our proposed pipeline works robustly.

6.4. Comparison of Different Encodings

We further made experiments with the BOW encoding and compared the results with FV representation. For BOW encoding, we used K-means in order to obtain vocabularies. For each test video the most similar vocabulary to its descriptor is selected and finally the histogram based feature vector is calculated according to the occurrence of the vocabularies. We used 4000 and 2000 cluster centers for the CK+ and EmotiW 2105 datasets, respectively. The lower number of cluster centers for the

Figure 6.5. Our alignment(first row), given alignment by challenge organizers(second row)

Emotiw 2015 dataset is due to the memory restriction.

In order to compare Fisher vector and BoW, we prepared vocabularies for each modality (i.e. HOG, HOF, MBH and improved trajectory), separately. Each BoW vector is separately normalized with L1 normalization. The concatenation of the BoW vectors is used in ELM with a linear kernel.

With BoW representation, the best result obtained on the CK+ using improved dense trajectory features (with a combination of HOG, HOF and MBH) is 88.99%, by leave one subject out protocol. FV encoding with considerable fewer number of visual words (64 words) outperforms BoW encoding (4000 words) by 4.59%. The results are shown in Table 6.6 and Figure 6.6. For the EmotiW 2015 dataset similarly, the best result of BOW encoding by MBH descriptor (with 2000 words) is 16.4% lower than the best result of Fisher vector encoding. The accuracy of each descriptor is reported in Table 6.7 and Figure 6.7.

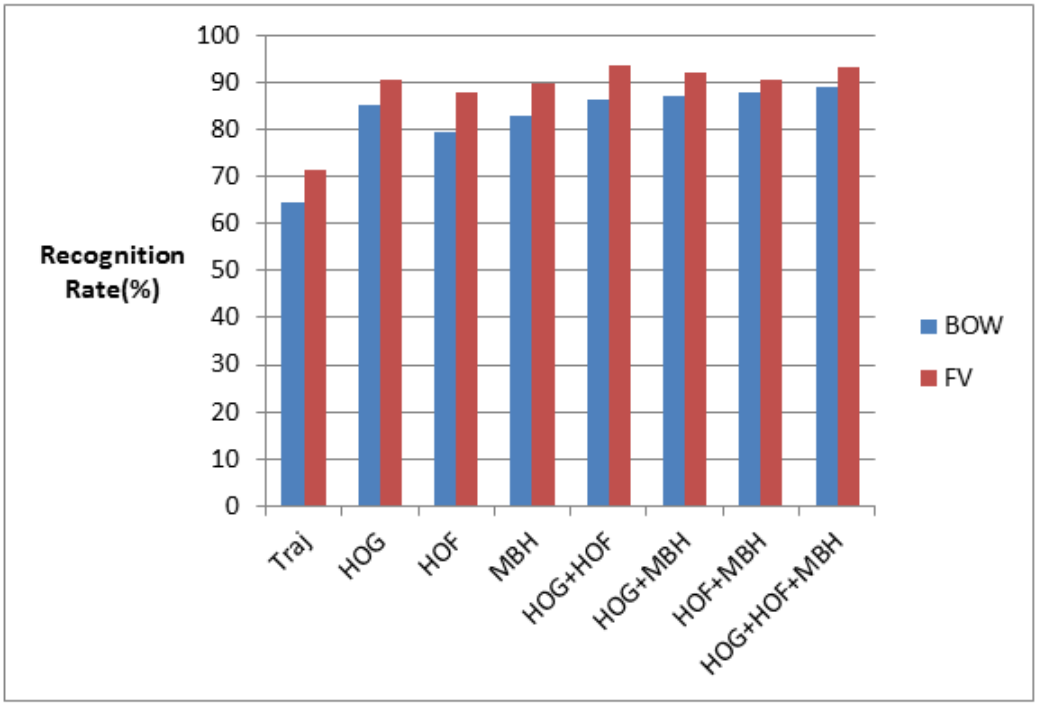

Figure 6.6. Contribution of different combinations of improved dense trajectory features with BOW and FV (CK+).

Table 6.6. Contribution of different descriptors with BOW and FV encodings (CK+).

Descriptor	Accuracy(%) BOW	Accuracy(%) FV
Trajectory	64.53	71.56
HOG	85.32	90.52
HOF	79.51	87.77
MBH	82.87	89.91
HOG+HOF	86.54	93.58
HOG+MBH	87.16	92.05
HOF+MBH	87.77	90.52
HOG+HOF+MBH	88.99	93.27

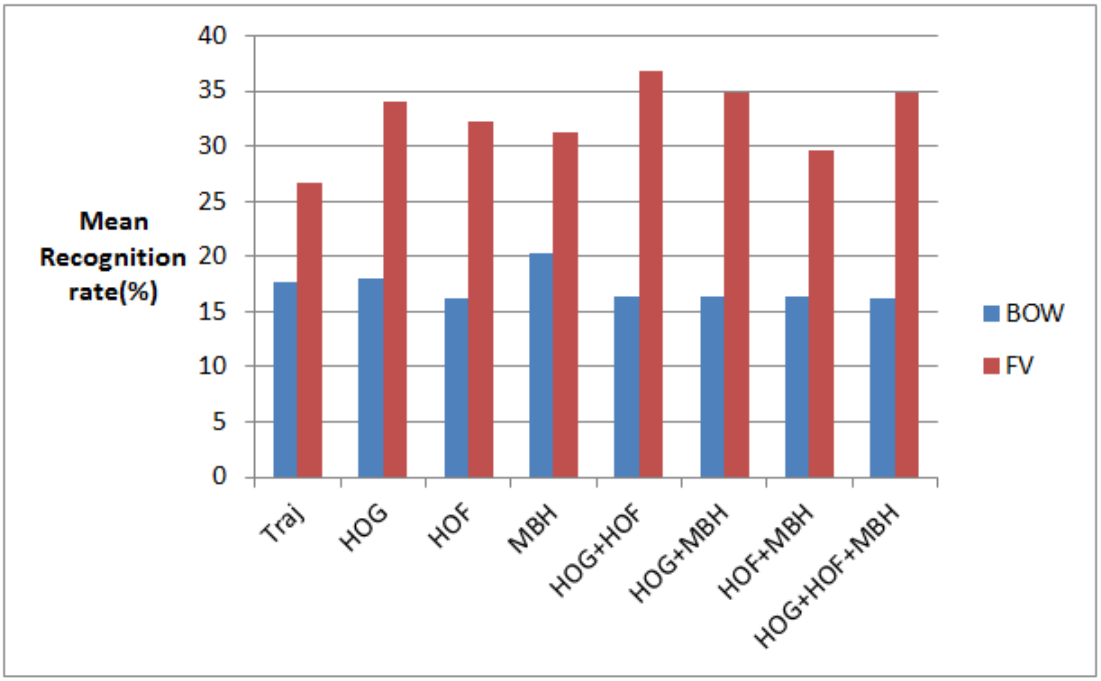

Figure 6.7. Contribution of different combinations of improved dense trajectory features with BOW and FV (EmotiW 2015).

Table 6.7. Contribution of different descriptors with BOW and FV encodings (EmotiW 2015).

Descriptor	Accuracy(%) BOW	Accuracy(%) FV
Trajectory	17.72	26.72
HOG	17.99	34.13
HOF	16.14	32.28
MBH	20.37	31.22
HOG+HOF	16.40	36.77
HOG+MBH	16.40	34.92
HOF+MBH	16.31	29.63
HOG+HOF+MBH	16.14	34.92

6.5. Comparison of ELM with SVM

We contrast ELM and SVM in terms of training time and accuracy on the CK+ dataset. We have found that ELM is faster and more accurate than SVM. For this test, concatenation of Fisher vector encoding of HOF and HOG features are used. Experiments are done on a machine with an Intel (R) core i5 CPU 2.50 GHz and 6 GB of RAM. ELM reaches 93.58% accuracy, while LIBSVM [89] and LIBLINEAR [90] achieve 80.73% and 91.66% accuracy, respectively. Table 6.8 shows the results on the CK+ dataset. We note here that it is possible to get faster computation times with more optimized SVM implementations like liblinear, but the difference remains significant.

Table 6.8. ELM and SVM comparison in terms of time and performance.

Classifier	Training time	Testing time (one subject)	Accuracy(%)
ELM	0.45 s	0.0627 s	93.58
libSVM	27.79 s	0.24 s	80.73
liblinear	2.03 s	0.0014 s	91.66

6.6. Cross Database Results

Typically, models trained on one imaging condition do not generalize well to other conditions, and the variance in a novel setting needs to be learned for proper generalization. We tested the conditions between CK+ and EmotiW 2015 datasets by running the trained models in a cross-database experiment. Training with CK+ and testing on EmotiW 2015 gives 16.14% percent accuracy (as opposed to 43.39% for training on EmotiW 2015). Conversely, training on EmotiW 2015 and testing on CK+ reduces the accuracy from 94.80% to 21.04%.

6.7. The Effect of Dimensionality and Fisher Vector Parameters

We explored the efficiency of the FV representation by using different parameters for both D descriptor dimensionality and K the number of GMM components. Since extracting features from EmotiW 2015 dataset takes a long time, we conducted these

experiments only on the CK+ dataset.

Three different configurations of dimensionality reduction and GMM components are used for this part of experiments. Each descriptor is reduced to half size (MBH=96, HOF=54, HOG=48), to 64, or the original dimensionality has been kept. In all cases we used PCA in order to project the descriptors to the de-correlated space. For the GMM components, we examined 32, 64 and 128 components for each dimensionality. For this test, we used the combination of HOG and HOF features.

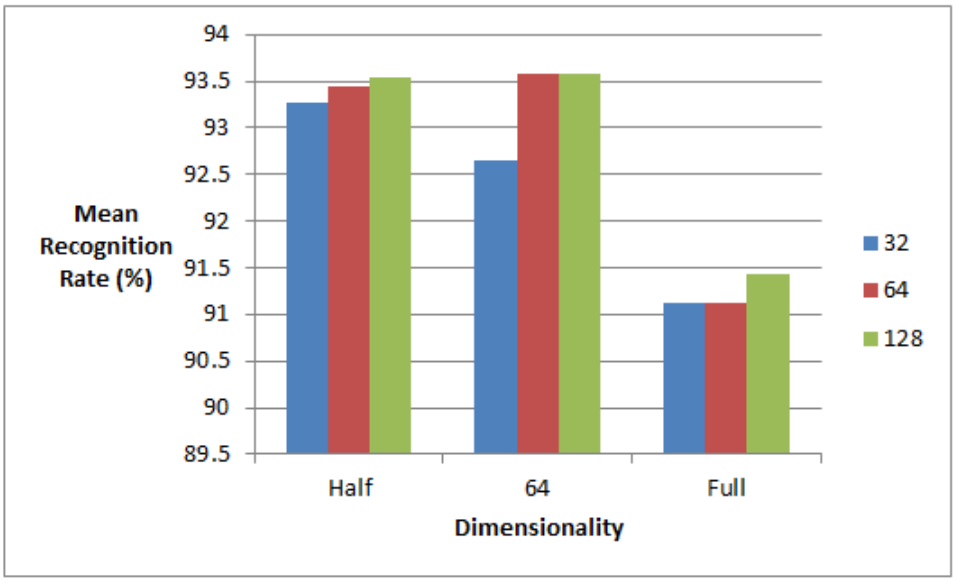

Figure 6.8. The effect of dimensionality reduction and the number of GMM components (CK+).

Table 6.9. The effect of dimensionality reduction and the number of GMM components (CK+).

GMM components \ Dimensionality	Half	64	Full
32	93.27%	92.66%	91.13%
64	93.45%	93.58%	91.13%
128	93.55%	93.58%	91.44%

Figure 6.8 and Table 6.9 illustrate how the dimensionality D and GMM components K have effect on the recognition rate of FV. By looking at the results, we can see that increasing K up to a point will improve the performance, but further increase will decrease the accuracy due to the high complexity. From the results it can be concluded that reducing the dimensionality of descriptors to 64 and using 64 GMM components outperforms the other combinations.

6.8. Deep Learning

Deep learning is becoming popular in the context of facial expression recognition. One of the widely used deep learning structures is the convolutional neural network (CNN). Training deep learning approaches requires very large datasets and longer training times. Here we briefly discuss the results of some deep learning methods applied on the CK+ dataset. In a recent study by Li et al. [55], 10,595 external images were used for training CNN models and 83% mean recognition rate was reported on CK+. Lv et al. [56] proposed a method based on face parsing detectors trained via deep belief networks and obtained 91.11% mean recognition rate. Liu et al. [9] proposed a new Boosted Deep Belief Network (BDBN), which yields 96.70% mean recognition rate, but it should be stated that in that work, the contempt emotion was not considered. By excluding the contempt expression, we were able to obtain 95.79% mean recognition rate. By comparing these results with the ones reported here, it can be seen that our approach is advantageous in terms of high accuracy and low complexity, as well as low training time. Recently different deep learning approaches have been proposed in order to deal with the problem of facial expression recognition from realistic videos. For instance Kim et al. [32] proposed a deep learning approach, which tries to solve the problem of registration in real-world conditions by fusing alignable faces with the non-alignable facial images where facial landmarks can not be detected. Mollahosseini et al. [40] proposed an approach based on training deep neural networks on both well-labeled and combination of noisy and well-labeled facial images, collected from the web.

Our results prove that our pipeline yields state-of-the-art results in both datasets. In the next chapter, we conclude our work and state possible improvements that would help increase the performance in emotion recognition from realistic videos.

7. CONCLUSION

Despite intensive work in facial expression recognition, the real problem, which is facial expression recognition in realistic conditions, is not solved yet [103]. Actually, the emotion recognition problem is beyond the finding six universal emotion labels on out-dated datasets. We need to find ways which are able to understand true emotional displays in the wild or in noisy environments.

In this thesis, we presented a new approach for facial expression recognition in the wild that uses a combination of different static and dynamic features. We located faces on video frames and registered them using our proposed registration pipeline. Then we extracted improved dense trajectory features (MBH, HOG, HOF, Trajectory), geometric features and LGBP-TOP. Afterwards we encoded improved dense trajectory and geometric features by Fisher vector encoding. In the last step, we classified different emotions using extreme learning machines with linear kernels.

We tested the proposed approach on the well-known CK+ and the challenging EmotiW 2015 Challenge datasets. We obtained 94.80%, 95.79%(without contempt) and 43.39% accuracy by using our proposed pipeline on CK+ and EmotiW 2015 datasets, respectively. The results show that our method yields state-of-the-art results in both databases. The main contribution of this dissertation is that this is the first time that improved dense trajectory features are used for facial expression recognition. In the original improved dense trajectory features, a human bounding box is used to remove camera motion, in case of facial expression recognition, an accurate face detection can be used instead of human detection, as what we have done in this work.

During our experiments, we analyzed different methods and parameters. We also compared each block of our pipeline with well-known methods. We evaluated each step of our proposed methodology to see how much each part contributes to the final results.

Different descriptors are examined in our study. The results show that for improved dense trajectory features, combination of motion-based features (HOF, MBH) and an appearance-based feature (HOG) is more successful. Furthermore we saw that fusion of Geo and LGBP-top features with the improved trajectories improves the results further.

Geometric features are calculated from the shape, angle and distance between different facial parts like mouth, eye, lips and eyebrows. Its dimension (26) is small, and it has shown promising results when encoded by FV.

Our experiments show that Fisher vector encoding outperforms bag of words encoding for all improved dense trajectory descriptors. The difference comes from the fact that FV benefits from the Gaussian mixture model, which encodes both first and second order statistics. However BOW uses zero-order statistics, which is provided by the K-means clustering algorithm.

Complexity of the system is the outcome of two factors; dimensionality of local descriptors and the numbers of Gaussian components, respectively. Increasing these factors can improve the performance, but after a certain value, the accuracy of them system starts dropping. However, it can be seen from our experiments that changing the parameters will not change the result dramatically and this can be a positive point for using Fisher vectors instead of the bag of words method.

We have shown that extreme learning machines outperform support vector machines in our study. ELM is faster than SVM in terms of required training time, which facilitate the evaluation process of possible feature combinations in a very short time comparing to SVM optimization process.

In the case of the Emotiw dataset, the recall of surprise, disgust and fear classes are low, which can be due to the low number of training samples in these classes. Also, a lot of training samples can be considered to contain a mixture of two or more facial expressions (such as surprise, fear and happy), which makes the recognition

more challenging. Therefore improving the annotation can be considered as a possible way of improving the results. Recently Benitez-Quiroz et al. [104] proposed a novel algorithm to annotate facial expression datasets. This method can recognize AUs and their intensities reliably in realistic videos, and can be substitue by former problamatic annotating approch.

The proposed method is sensitive to small facial changes and works successfully in some of the difficult cases, which contain very small facial changes and are hard to distinguish even for a human annotator. An example is shown in Figure 7.1. On the other hand, the sensitivity of method to small changes and the aforementioned problem of mixing expressions in the training set, sometimes cause the failure of the method in simple cases. For instance Figure 7.2 illustrates such an example.

The other problem in the Emotiw dataset is that class distribution is not balanced. For example, we have many more videos in happy and anger classes compared to fear, surprise and disgust. Collecting more samples for under-sampled classes or making the dataset balanced by combining SMOTE and random under-sampling is a possible improvement. We can under-sample the majority to different percentages of the original majority class and then apply SMOTE to minority samples. Also fusion of visual features with audio features, which contain significant emotional information, may improve the performance of the system.

Figure 7.1. A correctly classified sample from the disgust class.

Figure 7.2. A misclassified sample from happy class.

Frontalization of the face can probably improve the overall performance of the facial expression recognition system, but in order to obtain reliable results, frontalization should be able to preserve facial texture and emotion.

One possible improvement can be using a more precise facial tracker and landmarking method, which improves both geometrical and appearance based features. The other possible improvement can be the fusion of convolutional neural networks features as appearance features with already extracted features. Using additional datasets with the similar imaging conditions with the target dataset may increase the accuracy of the recognition system.

REFERENCES

1. Kaltwang, S., O. Rudovic, and M. Pantic, "Continuous pain intensity estimation from facial expressions", *International Symposium on Visual Computing*, pp. 368–377, Springer, 2012.

2. Clavel, C., I. Vasilescu, L. Devillers, G. Richard, and T. Ehrette, "Fear-type emotion recognition for future audio-based surveillance systems", *Speech Communication*, Vol. 50, No. 6, pp. 487–503, 2008.

3. Ekman, P., "Cross-cultural studies of facial expression", *Darwin and facial expression: A century of research in review*, pp. 169–222, 1973.

4. Tariq, U., *Image-based facial expression recognition*, Ph.D. thesis, University of Illinois at Urbana-Champaign, 2013.

5. Feng, X., B. Lv, Z. Li, and J. Zhang, "A Novel Feature Extraction Method for Facial Expression Recognition.", *JCIS*, 2006.

6. Rivera, A. R., J. A. R. Castillo, and O. Chae, "Recognition of face expressions using local principal texture pattern", *2012 19th IEEE International Conference on Image Processing*, pp. 2609–2612, IEEE, 2012.

7. Huang, X., G. Zhao, M. Pietikäinen, and W. Zheng, "Robust Facial Expression Recognition Using Revised Canonical Correlation.", *ICPR*, pp. 1734–1739, 2014.

8. Jabid, T., M. H. Kabir, and O. Chae, "Facial expression recognition using local directional pattern (LDP)", *2010 IEEE International Conference on Image Processing*, pp. 1605–1608, IEEE, 2010.

9. Liu, P., S. Han, Z. Meng, and Y. Tong, "Facial expression recognition via a boosted deep belief network", *Proceedings of the IEEE Conference on Computer*

Vision and Pattern Recognition, pp. 1805–1812, 2014.

10. Kanade, T., J. F. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis", *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pp. 46–53, IEEE, 2000.

11. Lucey, P., J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression", *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, pp. 94–101, IEEE, 2010.

12. Valstar, M. and M. Pantic, "Induced disgust, happiness and surprise: an addition to the mmi facial expression database", *Proc. 3rd Intern. Workshop on EMOTION (satellite of LREC): Corpora for Research on Emotion and Affect*, p. 65, 2010.

13. Lyons, M., S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with gabor wavelets", *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pp. 200–205, IEEE, 1998.

14. Alyüz, N., B. Gökberk, H. Dibeklioğlu, A. Savran, A. A. Salah, L. Akarun, and B. Sankur, "3D face recognition benchmarks on the Bosphorus database with focus on facial expressions", *European Workshop on Biometrics and Identity Management*, pp. 57–66, Springer, 2008.

15. Wang, S., Z. Liu, S. Lv, Y. Lv, G. Wu, P. Peng, F. Chen, and X. Wang, "A natural visible and infrared facial expression database for expression recognition and emotion inference", *IEEE Transactions on Multimedia*, Vol. 12, No. 7, pp. 682–691, 2010.

16. Mavadati, S. M., M. H. Mahoor, K. Bartlett, P. Trinh, and J. F. Cohn, "Disfa: A spontaneous facial action intensity database", *IEEE Transactions on Affective Computing*, Vol. 4, No. 2, pp. 151–160, 2013.

17. Dhall, A., O. Ramana Murthy, R. Goecke, J. Joshi, and T. Gedeon, "Video and image based emotion recognition challenges in the wild: Emotiw 2015", *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pp. 423–426, ACM, 2015.

18. Peng, X., Z. Xia, L. Li, and X. Feng, "Towards Facial Expression Recognition in the Wild: A New Database and Deep Recognition System", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 93–99, 2016.

19. Happy, S., P. Patnaik, A. Routray, and R. Guha, "The Indian Spontaneous Expression Database for Emotion Recognition", 2015.

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

21. Lienhart, R. and J. Maydt, "An extended set of haar-like features for rapid object detection", *Image Processing. 2002. Proceedings. 2002 International Conference on*, Vol. 1, pp. I–900, IEEE, 2002.

22. Viola, M., M. J. Jones, and P. Viola, "Fast multi-view face detection", *Proc. of Computer Vision and Pattern Recognition*, Citeseer, 2003.

23. Viola, P., M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance", *International Journal of Computer Vision*, Vol. 63, No. 2, pp. 153–161, 2005.

24. Zhu, X. and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild", *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 2879–2886, IEEE, 2012.

25. Viola, P. and M. J. Jones, "Robust real-time face detection", *International journal of computer vision*, Vol. 57, No. 2, pp. 137–154, 2004.

26. Dibeklioğlu, H., A. Salah, and T. Gevers, "A Statistical Method for 2-D Facial Landmarking", *IEEE Transactions on Image Processing*, Vol. 21, No. 2, pp. 844–858, Feb 2012.

27. Xiong, X. and F. De la Torre, "Supervised descent method and its applications to face alignment", *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 532–539, 2013.

28. Asthana, A., S. Zafeiriou, S. Cheng, and M. Pantic, "Robust discriminative response map fitting with constrained local models", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3444–3451, 2013.

29. Gower, J. C., "Generalized procrustes analysis", *Psychometrika*, Vol. 40, No. 1, pp. 33–51, 1975.

30. Zhu, X., Z. Lei, J. Yan, D. Yi, and S. Z. Li, "High-fidelity pose and expression normalization for face recognition in the wild", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 787–796, 2015.

31. Hassner, T., S. Harel, E. Paz, and R. Enbar, "Effective face frontalization in unconstrained images", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4295–4304, 2015.

32. Kim, B.-K., S.-Y. Dong, J. Roh, G. Kim, and S.-Y. Lee, "Fusing Aligned and Non-Aligned Face Information for Automatic Affect Recognition in the Wild:A Deep Learning Approach", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016.

33. Zhao, G. and M. Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions", *IEEE transactions on pattern*

analysis and machine intelligence, Vol. 29, No. 6, pp. 915–928, 2007.

34. Almaev, T. R. and M. F. Valstar, "Local gabor binary patterns from three orthogonal planes for automatic facial expression recognition", *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*, pp. 356–361, IEEE, 2013.

35. Lowe, D. G., "Distinctive image features from scale-invariant keypoints", *International journal of computer vision*, Vol. 60, No. 2, pp. 91–110, 2004.

36. Dalal, N. and B. Triggs, "Histograms of oriented gradients for human detection", *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 1, pp. 886–893, IEEE, 2005.

37. Jiang, B., M. Valstar, B. Martinez, and M. Pantic, "A dynamic appearance descriptor approach to facial actions temporal modeling", *IEEE transactions on cybernetics*, Vol. 44, No. 2, pp. 161–174, 2014.

38. Kanaujia, A. and D. Metaxas, "Recognizing facial expressions by tracking feature shapes", *18th International Conference on Pattern Recognition (ICPR'06)*, Vol. 2, pp. 33–38, IEEE, 2006.

39. Jain, S., C. Hu, and J. K. Aggarwal, "Facial expression recognition with temporal modeling of shapes", *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pp. 1642–1649, IEEE, 2011.

40. Mollahosseini, A., B. Hassani, M. J. Salvador, H. Abdollahi, D. Chan, and M. H. Mahoor, "Facial Expression Recognition from World Wild Web", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016.

41. Klaser, A., M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients", *BMVC 2008-19th British Machine Vision Conference*, pp. 275–1,

British Machine Vision Association, 2008.

42. Scovanner, P., S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition", *Proceedings of the 15th ACM international conference on Multimedia*, pp. 357–360, ACM, 2007.

43. Laptev, I., M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies", *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.

44. Wang, H., A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition", *International journal of computer vision*, Vol. 103, No. 1, pp. 60–79, 2013.

45. Dalal, N., B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance", *European conference on computer vision*, pp. 428–441, Springer, 2006.

46. Wang, H. and C. Schmid, "Action recognition with improved trajectories", *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3551–3558, 2013.

47. Li, Z., J.-i. Imai, and M. Kaneko, "Facial-component-based Bag of Words and PHOG Descriptor for Facial Expression Recognition.", *SMC*, pp. 1353–1358, 2009.

48. Li, Z., J.-i. Imai, and M. Kaneko, "Robust face recognition using block-based bag of words", *Pattern Recognition (ICPR), 2010 20th International Conference on*, pp. 1285–1288, IEEE, 2010.

49. Sikka, K., T. Wu, J. Susskind, and M. Bartlett, "Exploring bag of words architectures in the facial expression domain", *European Conference on Computer Vision*, pp. 250–259, Springer, 2012.

50. Krapac, J., J. Verbeek, and F. Jurie, "Modeling spatial layout with fisher vectors for image categorization", *2011 International Conference on Computer Vision*, pp. 1487–1494, IEEE, 2011.

51. Rudovic, O., I. Patras, and M. Pantic, "Coupled gaussian process regression for pose-invariant facial expression recognition", *European Conference on Computer Vision*, pp. 350–363, Springer, 2010.

52. Liu, M., S. Shan, R. Wang, and X. Chen, "Learning expressionlets on spatio-temporal manifold for dynamic facial expression recognition", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1749–1756, 2014.

53. Dhall, A., R. Goecke, J. Joshi, K. Sikka, and T. Gedeon, "Emotion recognition in the wild challenge 2014: Baseline, data and protocol", *Proceedings of the 16th International Conference on Multimodal Interaction*, pp. 461–466, ACM, 2014.

54. Huang, G.-B., H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 42, No. 2, pp. 513–529, 2012.

55. Li, W., M. Li, Z. Su, and Z. Zhu, "A deep-learning approach to facial expression recognition with candid images", *Machine Vision Applications (MVA), 2015 14th IAPR International Conference on*, pp. 279–282, IEEE, 2015.

56. Lv, Y., Z. Feng, and C. Xu, "Facial expression recognition via deep learning", *Smart Computing (SMARTCOMP), 2014 International Conference on*, pp. 303–308, IEEE, 2014.

57. Afshar, S. and A. Ali Salah, "Facial Expression Recognition in the Wild Using Improved Dense Trajectories and Fisher Vector Encoding", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 66–74, 2016.

58. Gürpınar, F., H. Kaya, S. Afshar, H. Dibeklioğlu, and A. A. Salah, "KERNEL ELM BASED AGE ESTIMATION", *Deniz Bilimleri ve Mühendisliği Dergisi*, Vol. 11, No. 3, 2015.

59. Kaya, H., F. Gürpinar, S. Afshar, and A. A. Salah, "Contrasting and Combining Least Squares Based Learners for Emotion Recognition in the Wild", *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pp. 459–466, ACM, 2015.

60. Jain, A. K. and S. Z. Li, *Handbook of face recognition*, Springer, 2011.

61. Zhang, C. and Z. Zhang, "A survey of recent advances in face detection", Technical Report MSR-TR-2010-66, Microsoft Research, 2010.

62. Goshtasby, A. A., *2-D and 3-D image registration: for medical, remote sensing, and industrial applications*, John Wiley & Sons, 2005.

63. Laptev, I., "On space-time interest points", *International Journal of Computer Vision*, Vol. 64, No. 2-3, pp. 107–123, 2005.

64. Wang, H., M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition", *BMVC 2009-British Machine Vision Conference*, pp. 124–1, BMVA Press, 2009.

65. Shi, F., E. Petriu, and R. Laganiere, "Sampling strategies for real-time action recognition", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2595–2602, 2013.

66. Ojala, T., M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns", *IEEE Transactions on pattern analysis and machine intelligence*, Vol. 24, No. 7, pp. 971–987, 2002.

67. Ahonen, T., A. Hadid, and M. Pietikainen, "Face description with local binary

patterns: Application to face recognition", *IEEE transactions on pattern analysis and machine intelligence*, Vol. 28, No. 12, pp. 2037–2041, 2006.

68. Moore, S. and R. Bowden, "Local binary patterns for multi-view facial expression recognition", *Computer Vision and Image Understanding*, Vol. 115, No. 4, pp. 541–558, 2011.

69. Lucas, B. D., T. Kanade, *et al.*, "An iterative image registration technique with an application to stereo vision.", *IJCAI*, Vol. 81, pp. 674–679, 1981.

70. Shi, J. and C. Tomasi, "Good features to track", *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pp. 593–600, IEEE, 1994.

71. Farnebäck, G., "Two-frame motion estimation based on polynomial expansion", *Scandinavian conference on Image analysis*, pp. 363–370, Springer, 2003.

72. Bradski, G., "Dr. dobb's journal of software tools", *Dr Dobb's Journal of Software Tools*, 2000.

73. Bay, H., A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)", *Computer vision and image understanding*, Vol. 110, No. 3, pp. 346–359, 2008.

74. Fischler, M. A. and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography", *Communications of the ACM*, Vol. 24, No. 6, pp. 381–395, 1981.

75. Perronnin, F. and C. Dance, "Fisher kernels on visual vocabularies for image categorization", *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2007.

76. MacQueen, J. *et al.*, "Some methods for classification and analysis of multivariate observations", *Proceedings of the fifth Berkeley symposium on mathematical*

statistics and probability, Vol. 1, pp. 281–297, Oakland, CA, USA., 1967.

77. Moon, T. K., "The expectation-maximization algorithm", *IEEE Signal Processing Magazine*, Vol. 13, No. 6, pp. 47–60, Nov 1996.

78. Dempster, A. P., N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm", *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

79. Reynolds, D., "Gaussian Mixture Models", *Encyclopedia of Biometrics*, pp. 659–663, Springer, 2009.

80. Joachims, T., "Text categorization with support vector machines: Learning with many relevant features", *European conference on machine learning*, pp. 137–142, Springer, 1998.

81. Csurka, G., C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints", *Workshop on statistical learning in computer vision, ECCV*, Vol. 1, pp. 1–2, Prague, 2004.

82. Jégou, H., M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation", *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 3304–3311, IEEE, 2010.

83. Van De Sande, K., T. Gevers, and C. Snoek, "Evaluating color descriptors for object and scene recognition", *IEEE transactions on pattern analysis and machine intelligence*, Vol. 32, No. 9, pp. 1582–1596, 2010.

84. Philbin, J., O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases", *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.

85. Perronnin, F., J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-

scale image classification", *European conference on computer vision*, pp. 143–156, Springer, 2010.

86. Cortes, C. and V. Vapnik, "Support-vector networks", *Machine learning*, Vol. 20, No. 3, pp. 273–297, 1995.

87. Vapnik, V., *The nature of statistical learning theory*, Springer Science & Business Media, 2013.

88. Alpaydin, E., *Introduction to machine learning*, MIT press, 2014.

89. Chang, C.-C. and C.-J. Lin, "LIBSVM: a library for support vector machines", *ACM Transactions on Intelligent Systems and Technology (TIST)*, Vol. 2, No. 3, p. 27, 2011.

90. Fan, R.-E., K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification", *Journal of machine learning research*, Vol. 9, No. Aug, pp. 1871–1874, 2008.

91. Huang, G.-B., Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications", *Neurocomputing*, Vol. 70, No. 1, pp. 489–501, 2006.

92. Kaya, H. and A. A. Salah, "Combining modality-specific extreme learning machines for emotion recognition in the wild", *Proceedings of the 16th International Conference on Multimodal Interaction*, pp. 487–493, ACM, 2014.

93. Salah, A. A., N. Alyüz, and L. Akarun, "Registration of three-dimensional face scans with average face models", *Journal of Electronic Imaging*, Vol. 17, No. 1, pp. 011006–011006, 2008.

94. Haghighat, M., S. Zonouz, and M. Abdel-Mottaleb, "Identification using encrypted biometrics", *Computer analysis of images and patterns*, pp. 440–448, Springer, 2013.

95. Ekman, P. and W. V. Friesen, "Facial action coding system", 1977.

96. Huang, X., G. Zhao, W. Zheng, and M. Pietikainen, "Spatiotemporal local monogenic binary patterns for facial expression recognition", *IEEE Signal Processing Letters*, Vol. 19, No. 5, pp. 243–246, 2012.

97. Sanin, A., C. Sanderson, M. T. Harandi, and B. C. Lovell, "Spatio-temporal covariance descriptors for action and gesture recognition", *Applications of Computer Vision (WACV), 2013 IEEE Workshop on*, pp. 103–110, IEEE, 2013.

98. Walecki, R., O. Rudovic, V. Pavlovic, and M. Pantic, "Variable-state latent conditional random fields for facial expression recognition and action unit detection", *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, Vol. 1, pp. 1–8, IEEE, 2015.

99. Zhao, X., X. Liang, L. Liu, T. Li, N. Vasconcelos, and S. Yan, "Peak-Piloted Deep Network for Facial Expression Recognition", *arXiv preprint arXiv:1607.06997*, 2016.

100. Kayaoglu, M. and C. Eroglu Erdem, "Affect Recognition using Key Frame Selection based on Minimum Sparse Reconstruction", *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pp. 519–524, ACM, 2015.

101. Yao, A., J. Shao, N. Ma, and Y. Chen, "Capturing AU-aware facial features and their latent relations for emotion recognition in the wild", *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pp. 451–458, ACM, 2015.

102. Ebrahimi Kahou, S., V. Michalski, K. Konda, R. Memisevic, and C. Pal, "Recurrent neural networks for emotion recognition in video", *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pp. 467–474, ACM, 2015.

103. Gunes, H. and H. Hung, "Is automatic facial expression recognition of emotions coming to a dead end? The rise of the new kids on the block", *Image and Vision Computing*, 2016.

104. Fabian Benitez-Quiroz, C., R. Srinivasan, and A. M. Martinez, "EmotioNet: An Accurate, Real-Time Algorithm for the Automatic Annotation of a Million Facial Expressions in the Wild", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5562–5570, 2016.