TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF ABREVIATIONS

ETS	École de Technologie Supérieure
ASC	Agence Spatiale Canadienne
RBF	Radial Basis Function
AU	Action Units
BDBN	Boosted Deep Belief Network
BN	Bayesian Networks
BoW	Bag of Words
C3D	3D Convolutional Network
CLM	Constrained Local Model
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CRF	Conditional Random Field
CS	Compressed Sensing
DBN	Dynamic Bayesian Network
DCNN	Dynamic Convolutional Neural Network
DTW	Dynamic Time Warping
FACS	Facial Action Coding System
FFT	Fast Fourier Transform
GAN	Generative Adversarial Network

HMM	Hidden Markov Models
HOF	Histogram of Optical Flow
HOG	Histogram of Oriented Gradients
HOF	Histogram of Optical Flow
ICP	Iterative Closest Point
PCA	Principal Component Analysis
RAM	Random Access Memory
RCNN	Convolutional Neural Network
LBP	LBP Local Binary Pattern
LDCRF	Latent Dynamic Conditional Random Fields
LSTM	Long-Short Term Memory
MBH	Motion Boundary Histograms
MoCap	Motion Capture
PCA	Principal Component Analysis
RAM	Random Access Memory
RCNN	Region Convolutional Neural Network
RNN	Recurrent Neural Network.
ROC	Receiver Operating Characteristic
SIFT	Scale Invariant Feature Transform
SO-SVM	Structural Output Support Vector Machines

STM	Selective Transfer Machine
SVM	Support Vector Machines

INTRODUCTION

Studying facial expression recognition is a challenging task and has drawn increasing attention from computer vision researchers due to its variety of applications in interacting human with computers, medical and psychological assistance and marketing. In fact, facial expression is one of the most meaningful manners for human beings to express their feelings, emotions and intentions and in the process of communication plays a significant role in human interactions. In addition, Facial expression recognition has vital applications in a variety of fields including security purposes which can reduce crime, enhance safety, assist psychologists and behavioural traits analysts, improve the advertisement techniques and enhance the human-robot interactions. In spite of the fact that facial expression recognition is an actively researched topic in computer vision society, it is still a challenging problem and has been significantly investigated among computer vision researcher communities over the past few decades.

The problem of facial expression recognition firstly addressed by studies and experiments of Darwin in (1872), demonstrate that movements of components of face and the tone of the speech are the two major ways for expressing the common emotions of human beings when communicating. In addition, Mehrabian (1968) indicated that the facial expression of the speaker contributes 55% to the effect of the spoken message, which is more than the verbal part (7%) and the vocal part (38%). Therefore, the face tends to be the most visible form of emotion communication. Those facts make facial expression recognition a widely used scheme for measuring the emotional state of human beings.

The first attempt to define the problem of facial expressions by computer vision community, naturally refers from the psychology theories and then adopt some of their theories, conventions and apply those concepts and theories to design the system. In spite of the fact that human beings make use of a range of ways to express their emotions for everyday communication than the six basic expressions with some expressions for everyday communication, Darwin was the

first scientist who theorized and defined the basic expressions. Humans have a universal way of expressing and understanding a set of feelings and emotions. The set of basic emotions are divided in to six : anger, disgust, fear, happiness, sadness, and surprise.

Facial expressions can be coded and defined using facial Action Units (AU) and the Facial Action Coding System (FACS), which was first introduced by (Ekman & Rosenberg, 1997). Typically, facial AU analysis should be accomplished by taking several steps: (i) face detection and tracking; (ii) alignment and registration; (iii) feature extraction and representation; and (iv) AU detection and expression analysis. Due to the recent advances that have been made in the face tracking and alignment steps, most approaches focus on feature extraction and classification methods. (Abbasnejad, Sridharan, Denman, Fookes & Lucey, 2015; Martinez & Du, 2012)

Although expression recognition is a well studied topic both in academia and industry and stunning progresses have been made over time, the problem of expression recognition has not been fully addressed in all its aspects. Expressions are complex movements of muscles and are correlated with the other objects and actions in videos. This issue makes the existing models fail in many scenarios due to the insufficiency of the extracted features from the video frames and also lack of robustness. For examples, many existing techniques fail to truly classify two expressions of "happiness" and "surprise" due to complexity of determining of starting and ending points in video frames. In addition, due to the complexity of video frames, most of the current classifiers in the field fail to model the temporal dynamic among video frames adequately. Furthermore, recognition of expression heavily rely on data, this needs human labour for labeling the videos and generating data for better event analysis. These challenges call for the development of novel methods to address these issues in FER systems.

On the other hand, discriminating of most of emotion states in the same person is also another complicated task. In addition, the external factors play an important role to increase the difficulty of the recognition process, in terms of illumination, environment and cameras.

The primary purpose of facial expression recognition is to identify the human emotional state (e.g., anger, disgust, fear, happiness, sadness, and surprise) based on the given facial images. Facial expressions are naturally occur over time and the state of each expression varies over time. The Dynamic process of facial expressions is important for the recognition process and also for making better distinction among facial expression categories. Our research and experiments presented in this thesis focus on 2D facial expression images. To address the problem of dynamic expression recognition we plan to deploy spatial-temporal to take full advantage of the motion information.

Our ideal goal in this project is to design a novel framework for facial expression recognition to automatically distinguish the expressions with a satisfactory recognition accuracy. We also aim to demonstrates the feasibility and effectiveness of our approach by conducting extensive experiments by several well-known datasets.

The Main Contributions of This Thesis

Although there have been various algorithms in the literature for solving the expression recognition (from Statistical algorithms such as using SIFT features and SVM to recently developed end to end trainable structures such as CNNs, RNNs and combination of CNNs and RNNs), the problem of expression analysis is not fully addressed. The purpose of this project is to utilise a 3D architecture neural network in conjunction of LSTM for efficient and accurate expression recognition. 3D neural networks have been used due to their ability to capture temporal features and they proved in the literature that they have superior performance for video analysis in compare to the conventional CNN based methods. In addition, LSTM models are able to learn temporal dependencies in an observed sequence. However, one problem with training neural network from scratch is that, training neural networks usually needs a large scale dataset.

In this work we address the following questions:

- How reasonable is to use 3D CNN in the problem of facial expression and what are the pros and cons?
- Can we overcome the problem of data limitation by using syntactic data?
- Is it suitable to train the neural networks for the expression recognition task on the other large-scale datasets from different domains?
- Conducting experiment with one of the most well known statistical methods and make a comparison with 3D CNN

This thesis is organized as follows: we review the literature in chapter 1. Chapter 2 discusses our conventional facial expression recognition model. In chapter 3 we will explain how a synthetic facial expression videos can be generates. Finally in chapter 4, we explains our proposed deep neural network architecture and provide comparisons, conclusions and suggestions.

CHAPTER 1

LITERATURE REVIEW AND APPLICATIONS OF FACIAL EXPRESSION RECOGNITION

So far, a variety of statistical approaches for representing expressions and classification of features have been applied to tackle our problem. The most popular algorithms can be named as: Local Binary Patterns (LBP) Guo, Zhang & Zhang (2010), Scale Invariant Feature Transformation (SIFT) Lindeberg (2012) and Gabor filters Zhang, Shan, Gao, Chen & Zhang (2005). Particularly, recently the histogram of oriented gradient (HOG) Déniz, Bueno, Salido & De la Torre (2011) as a good image representation of the texture of images and also the structures and shape of objects, has been widely used for expression analysis. To devise an accurate system, firstly the position of dense facial landmarks should be located with face alignment method. After this step, the feature vector will be constructed and formed by concatenating all descriptors which extracted form critical points on the faces.

The task of automated Facial Expression Recognition could be refereed to recognize expressions of persons based on their pictures Kleinsmith & Bianchi-Berthouze (2012); Zhao & Pietikainen (2007). This task can be performed on still image or sequences (videos). In this chapter various methods of computing and interpreting related to facial expression recognition will be explored. Existing work will be presented to understand what has been researched, utilised and applied to address dynamic expression recognition problem.

In spite of the fact that the area of Facial Expression Recognition has been under intensive investigation from several different perspectives since 1960s, during last few years a plenty of algorithms have been devised to address this problem from different perspectives. Specifically, in recent years facial expression recognition has gained considerably popularity among researchers who are mostly involved in solving problem regarding machine vision. The attraction of this

areas among researchers might be related to emerge of a number of novel methods which contribute to both performance and accuracy.

In addition, the increasing demand in applications of numerous domains such as commercials, industrial and governmental purposes. Recently, with growing the existence of effective and rapid computational processing units that process large blocks of data in parallel, we have witnessed significant improvements in performance and accuracy of novel improved algorithms. The goal of this chapter is to summarize the state of the art methods and discuss problems and potential contributions in this area of research.

To shed a light on our plan to tackle this problem, we initially review thoroughly the most related articles in this domain and then we will dive in to algorithms and the basic theories behind them. Then we describe the standard algorithms pipeline in statistical and deep learning methodologies which are common for most facial expression analysis. Finally, we make a comparison about each algorithm and explore their advantages and disadvantages regarding performance, accuracy and computational complexity.

Modeling a system capable of exploiting discriminant features from video-based expression is challenging since the deformations and movements of facial components considerably vary according to the specific situations, subjects and data acquisition. Therefore extracting discriminant spatio-temporal features while suppressing these conditions is a challenging task.

1.1 Automatic Facial Expression Recognition System

In the following section we provide information about how to recognise facial expression. To solve the problem of facial expression recognition, generally two procedures can be followed, (i) statistical FER Systems, (ii) deep learning model. Statistical method consists of three stages, face acquisition, facial expression extraction and representation, and classification (see Figure 1.1). We briefly summarize the major aims and challenges of each stages in the next section.

In order to achieve a robust FER system model, an understating is required for how a typical system is designed, what steps should be taken to reach from raw data (image sequences) to expression (desired output). The general architecture for these typical systems consist of some stages that are widely used in almost all computer vision systems.

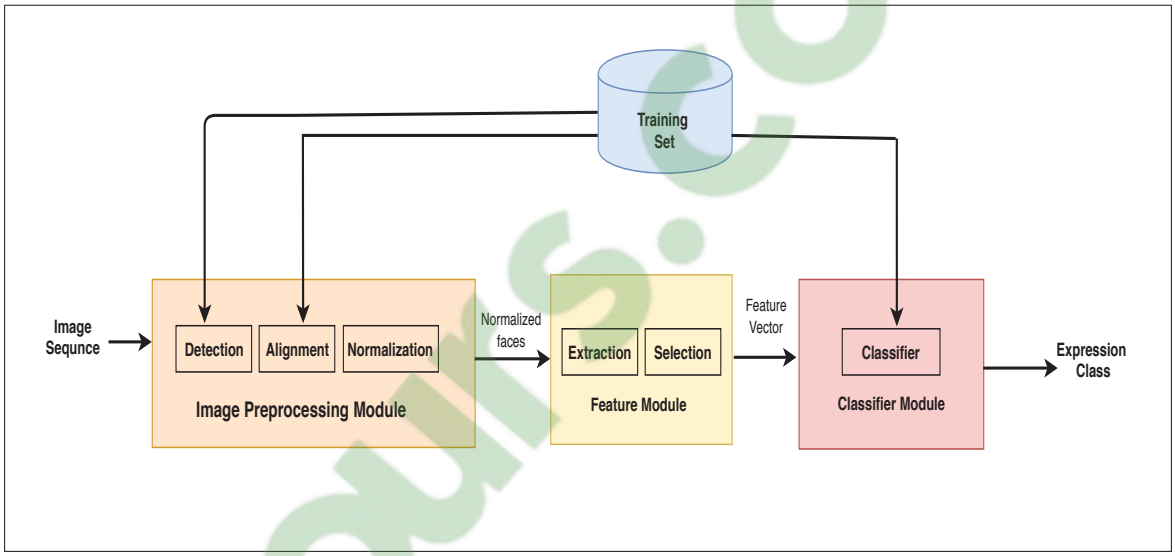

Figure 1.1 The standard Facial Expression Recognition System, Image preprocessing , facial expression extraction and classification

Darwin was the first person who commence to make assumptions related issues to expressions and ways of communication of humans, he figured out that humans have the same way of expressing and understanding a set of basic or prototypical emotions. After that, Ekman Ekman & Rosenberg (1997) extended the set of basic emotions to six expressions: anger, disgust, fear, happiness, sadness, and surprise. In computer vision community, the majority of researchers model the facial expression by either categorical approaches or Facial Action Coding System (FACS). Before reviewing researches related to FER, the most important terminology can be briefly summarized as follows:

1. The facial action coding system (FACS) Ekman & Rosenberg (1997) : This system was developed by scientists to encode some crucial parts of face according to facial muscle movements

and is able to characterize facial actions to show individual human emotions. FACS are able to encode the micro movements of specific facial muscles called action units (AUs). In figure 1.2 illustrates some AUs which;

2. Basic expressions: Human expression is categorized in seven classes: happiness, surprise, anger, sadness, fear, disgust, and neutral (see Figure 1.3 for examples);

3. Facial Landmarks (FLs) Koestinger, Wohlhart, Roth & Bischof (2011): Facial Landmarks are some visually critical points in facial regions such as the starting and end points nose, eye brows, and the mouth, (Figure 1.4).

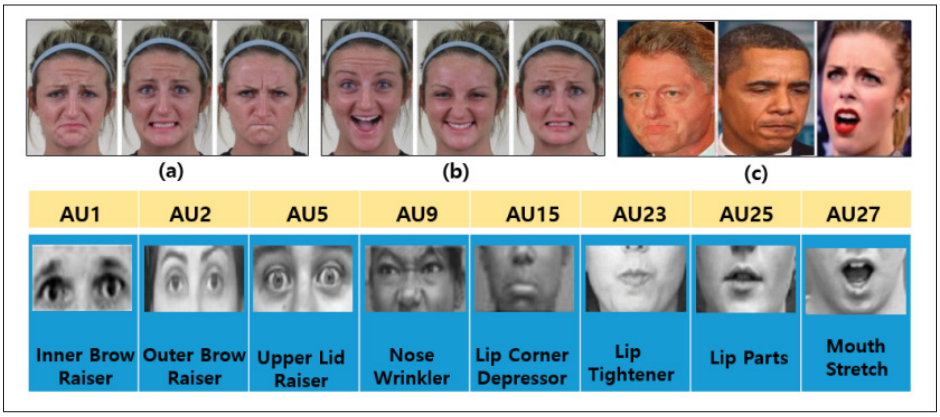

Figure 1.2 This figure shows eight different facial action units (AUs).

1.2 Conventional FER Systems

The most specific attribute of the conventional FER method is that the whole system is highly dependent on manual feature engineering Huang, Chen, Lv & Wang (2019). To obtain our desirable output sequences shall go through some pre-processing steps and then it is time for making the decision about the choices of feature extraction and classification method for the target dataset. Typically, the conventional FER procedure can be categorized into three major

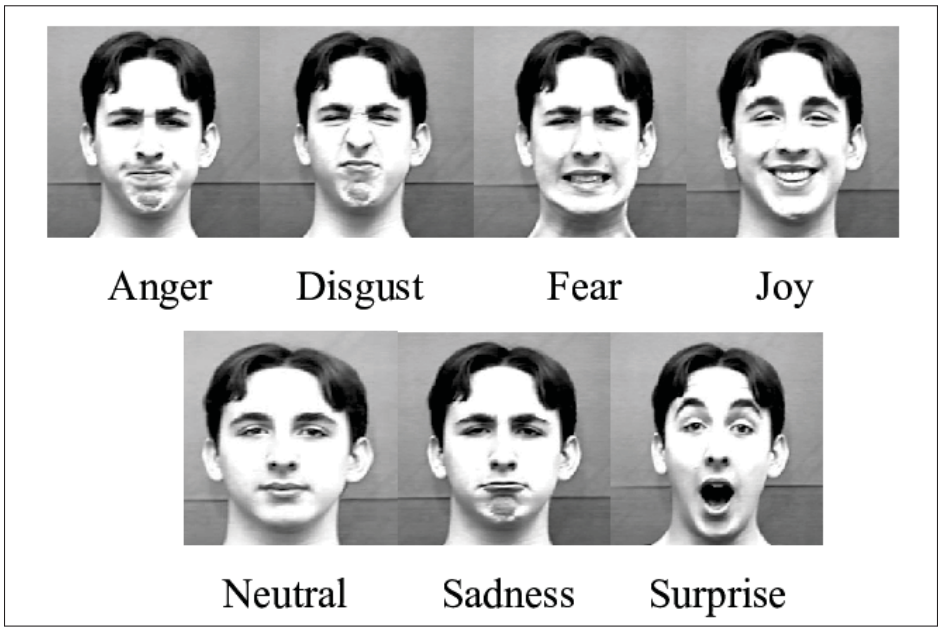

Figure 1.3 Basic expressions. Human expression is categorized in seven classes.

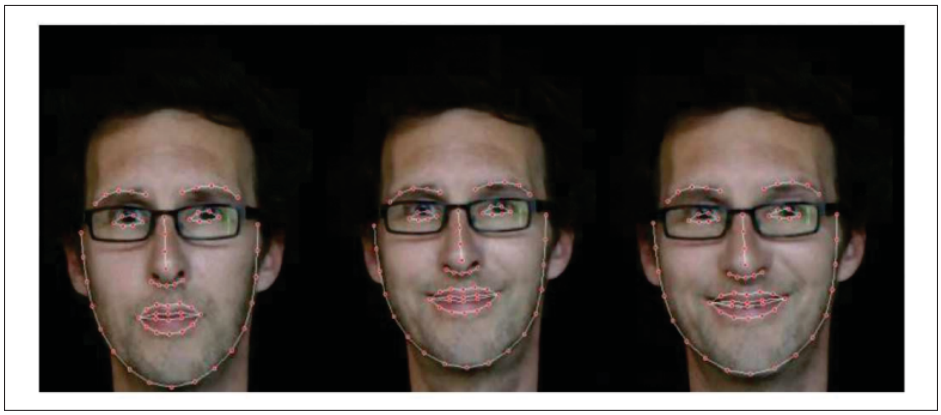

Figure 1.4 Facial Landmarks. Mainly salient regions on face are pointed as key features.

steps: image pre-processing, feature extraction, and expression classification as is depicts in Figure 1.1. In this section those steps will be discussed more in details.

1.2.1 Pre-processing

The purpose of this step is to remove unrelated information of each sequence such as variations that are not related to facial expressions, such as backgrounds, various poses, illuminations, and overall enhance the detection ability of relevant information. Pre-processing of sequences can directly influence the extraction of features and the performance of expression classification Huang *et al.* (2019). In addition, many datasets are different in the number of high quality images, and some are encompassed colour images, while some are include grayscale images. The main steps in process of sequence pre-processing are introduced as follows:

- **Face detection:** The initial step in expression recognition task is face detection. Face detector tries to find the position of the faces in an image and even returns the coordinates of a bounding box for each one of them. Some time we apply some algorithms to detect and extract only special regions of face, for instance finding components of face which playing significant role in expression such as mouth and eyes;
- **Face alignment:** During the face alignment process, faces should be scaled, cropped and most of the times compared with some template reference points located at fixed locations in the image. Typically this process requires finding a set of facial landmarks using a landmark detector algorithms, determining the best transformation that fits the reference points. For instance, changing the pose of a face to frontal.

1.2.2 Feature extraction and representation

The most vital step in modeling the system is the process of face representation. Feature extraction is a process of representing desirable information from region of interests. These extracted representations or descriptions are our desirable regions of the image, etc. Feature extraction is directly influence the performance of the algorithms, which is usually the backbone of the FER system. In fact the purpose of this step is to transform the desire regions at the face

(pixel values of a face image) into a compact and discriminative feature vector. Since one of the contributions in this work is to compare a conventional feature extraction method with a more recent developed approach, we will discuss more about it in this section.

Effective feature extraction is a crucial stage for facial expression recognition. In general, existing feature expression features can be categorized into two groups: appearance features Fasel & Luettin (2003); Zhang, Lyons, Schuster & Akamatsu (1998) and geometric features Baraniuk & Wakin (2009); Shan, Gong & McOwan (2009). The appearance features model the appearance changes of faces, such as wrinkles and furrows, by directly utilizing and calculating pixel values. On the other hand, geometric features exploit structure of shapes and locaations of facial components (e.g. eyes and mouth) to represent the face geometry.

In this section we are going to widely review the most well-known feature extraction methods in FER such as: Optical Flow method, Haar-like feature extraction, Gabor feature extraction and Local Binary Pattern (LBP).

1.2.2.1 Optical Flow Method

Optical flow is able to capture the motion of patterns on objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and a scene two consecutive frames caused by the movement of object or camera Kass, Witkin & Terzopoulos (1988). In Horn & Schunck (1981) scientists combine the two-dimensional velocity field and the gray scale to gather the maximum temporal dependencies. An efficient procedure for analysing temporal facial variations is presented in Yacoob & Davis (1996). Typically, optical flow caused by facial expressions to identify the direction of motions. Then a classifier is used for expression recognition. In Cohn, Zlochower, Lien & Kanade (1998), an optical flow-based approach is designed and implemented to capture emotional expression by automatically recognising subtle

changes in facial expressions. In Sánchez, Ruiz, Moreno, Montemayor, Hernández & Pantrigo (2011) authors made a comparison between two optical flow-based facial recognition methods.

1.2.2.2 Haar-like Feature Extraction

The Haar-like features are digital image computing used in object recognition Viola & Jones (2001) is the combination of edge, linear, centre and diagonal features. Each feature template will be segmented into two rectangle (or other shapes) regions, white and black, and the template's feature values are defined as the differences between intensities of pixels. The Haar-eigenvalue signifies the gray scale variation of the image regions. In Yang, Liu & Metaxas (2009), in order to represent the temporal variations in appearance of human face, dynamic Haar-like features are defined and encoded into binary pattern features. When the illumination of region is stable, Haar has capacity to represent the local gray scale variation of the face.

1.2.2.3 Gabor Feature Extraction

The gabor feature extraction method is able to exploit the Gabor Features of the images based on wavelet image Transform-based kernel function. In Lyons, Akamatsu, Kamachi & Gyoba (1998), series of Gabor filters have been applied for representing the multi-orientation and multi-resolution of image. Authors in Yu & Bhanu (2006) used linear and nonlinear synthesis of new algorithms on the basis of Gabor feature. One of the advantages of the gabor filters is existing of Discrete Wavelet Transform function which are able to provide a more compact feature vector compared to existing Gabor-based expression classification to alleviate the problem of dimensionality Mattela & Gupta (2018). Gabor wavelets have a great robustness to multi-scale and multi-directional texture feature transformation, and are not sensitive to variations of illumination intensity. The demerit of Gabor wavelets can be mentioned as their memory consumption since they usually work on global features.

1.2.2.4 Local Binary Pattern Family

In spite of the fact that Gabor filters are widely used and are popular among computer vision communities due to their ability to accurately describe appearance, they are computationally complex and costly. One interesting procedure to overcome this issue is to apply only a one computational measure (such as the mean or variance) computed from the entire feature space, it is possible to mitigate the overall dimensionality of the Gabor feature space and as a result computation time will be deceased. To name the downside of them we can mention that by reducing the spaces usually it lacks sufficient descriptive power to model the spatial structure of images.

Local binary pattern (LBP) operators are one of the most practical and efficient feature representations commonly-used in FER. The merit of LBP operators is that they are computationally fast without any significant decrease in accuracy. The LBP Ahonen, Hadid & Pietikäinen (2004) calculates the brightness relationship between each pixel contained in the image and its local neighbourhood. The binary sequence is then encoded to form a local binary pattern. Finally, it uses a multi-region histogram as a feature description of the image, as shown in Figure 1.5.

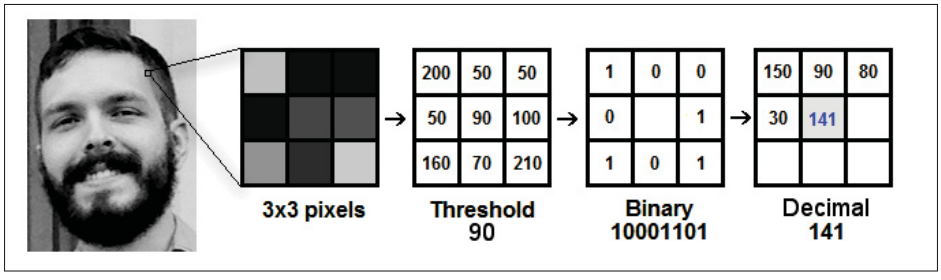

Figure 1.5 Extraction of LBP features from a facial image.

There exist different types of LBP that have been applied in the task of facial expression feature extraction. We will conduct intensive experiments on one of its varients in our work since it has robust feature representation. We used LBP-Three Orthogonal Planes (TOP).

LBP-Three Orthogonal planes (TOP) operator is coming from extension of the LBP algorithm. The idea of LBP-TOP is a simple approach based on LBP but the scientists extends it in three dimensions to exploit describing the spatio-temporal domain. LBP-TOP concatenate the feature distributions from each individual plane and then combine them together, making the feature vector much shorter when the number of neighboring points increases.

We dedicated the next chapter for this method and will explain all details related to this operator and carry out some experiments from CK+ dataset and report the results.

Here to make a comparison between Gabor wavelet Zhang *et al.* (1998), the LBP operator requires less storage space and has proved its computational efficiency advantage. However, the LBP operator as lack enough effectiveness on the images with noise. It may lose some useful feature information since it only considers the pixel features of the picture centre and its neighbourhood ignoring the difference in amplitude. Since LBP-TOP representing 2D features from three orthogonal planes, the dimensionality of the final vector is usually has a noticeable low level of dimensionality (typically three times that of static features) compared to Gabor or Haar feature vectors.

1.2.3 Classification

After doing some reprocessing steps and representing our desirable region of interests in to real values (feature vector) it is time to obtain required classes. This stage is know as classification task which is a crucial step in FER task to determine how to select the appropriate classifier that can successfully predict the face expressions. In literature the most commonly used and widely applied classifiers in FER systems include K-Nearest Neighbours (KNN), Support Vector Machine (SVM), Adaptive Boosting (Adaboost), Bayesian, Sparse Representation-models, and Probabilistic and Bayesian Neural Network. In the following sections, we will explain in more details.

1.2.3.1 Support Vector Machine Hearst *et al.* (1998)

Support Vector Machines (SVMs) Hearst *et al.* (1998) are well-known for their robustness and accuracy in classifying similar patterns. SVMs are discriminative classifiers which are defined by separating supporting hyperplanes. In other words, given labeled training data, the output of algorithm is an optimal hyperplane which categorizes new examples. For given a training set of N data points $\{y_k, x_k\}_{k=1}^{N}$, where $x_k \in R_n$ is the $k-$th input and $y_k \in R$ is the $k-$th output pattern, the objective function for SVM can be written as Hearst *et al.* (1998):

$$y(x) = sign[\sum_{k=1}^{N} \alpha_k y_k \phi_k(x, x_k) + b] \tag{1.1}$$

where α_k are positive real constants, b is a real constant and ϕ_k is a kernel. Then the classifier is built as:

$$\omega^T \phi(x_k) + b \geq 1 \text{ , if } y_k = 1 \text{ and, } \omega^T \phi(x_k) + b \leq -1 \text{ , if } y_k = -1 \tag{1.2}$$

The SVM Tsai & Chang (2018) can find a good solution on sophisticated models by providing limited sample data information to obtain generalisation ability. It is also possible to map linearly indivisible data to higher dimensions by kernel functions to convert the data into linear separable. By introducing a kernel function, the computer can effectively process high-dimensional data, and avoid dimension disasters to some extent. We will review and discuss this algorithm in depth in next chapter.

1.2.3.2 Naive Bayes Classifier

The Naive Bayes Classifier technique Moghaddam, Jebara & Pentland (2000) is based on the so-called Bayesian theorem and is specially adaptable when the dimensionality of the inputs is

high. In spite of its simplicity, Naive Bayes may often outperform more complex classification methods. Naive Bayes classifier is highly scalable, requiring linear parameters for the number of variables in learning problems. One of the advantages of this classifier is that only a small amount of training data is needed to estimate the parameters required for classification.

1.3 Deep Learning-Based FER Systems

Most of our study has been dedicated for reviewing and experimenting on deep learning techniques and issues since they have been considered as a breakthrough and demonstrated outstanding performance.

Deep learning algorithms including CNNs and recurrent neural network (RNN), have been applied to various fields of computer vision identification, feature extraction, classification, segmentation, and target detection and tracking.

A based CNN model contains three types of layers: convolution layers which are followed by max pooling layers and usually at the end the fully connected layers (please see Figure 2.2 for visualization). Mostly the input which is usually images or feature maps are feed into convolution layers. Then the convolution applies on the inputs and set of convolution filters in a sliding-window manner to output feature maps that represent a spatial arrangement of the input image or feature maps LeCun, Boser, Denker, Henderson, Howard, Hubbard & Jackel (1989).

In the second step, max pooling layers apply on the extracted input features from the conv layers and reduce their dimensions LeCun *et al.* (1989). Finally, last fully connected layers of the model calculates the scores of each classed on the entire input image.

However, since CNN-based methods do not capture temporal dependencies between the facial parts, hybrid methods introduced. These methods, usually combine a CNN model as the spatial feature extractor, and RNNs or long short-term memory (LSTM) as the temporal feature extractor.

In the next sections other convolutional neural networks which have the capability of exploiting the temporal features are introduced.

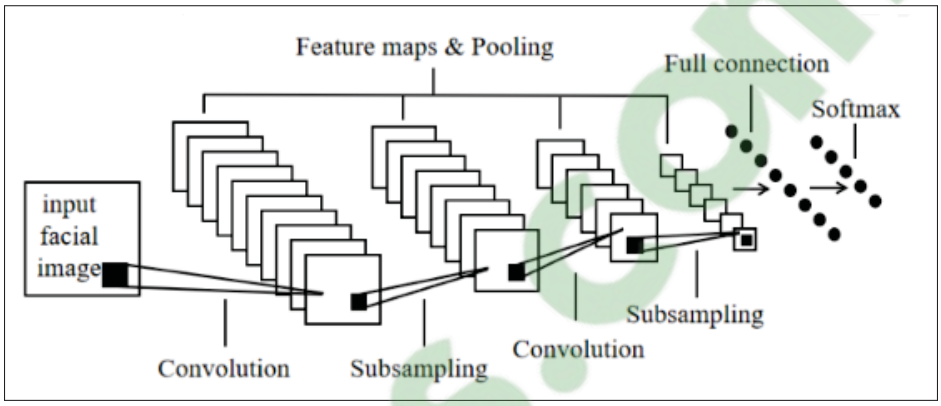

Figure 1.6 Procedure of CNN-based FER approach Li & Deng (2018)

1.3.1 Spatio-temporal Neural Network

These networks are designed to capture both texture and temporal information among images and input frames. RNNs, LSTMs, IRNNs, BRNNs and C3D networks are common examples for learning spatio-temporal features.

1.3.2 Hybrid Models

In the hybrid models the is to integrate two or more diffrent architecture to extract useful information that are more related to the task. For example, a two-stream CNN models was used for recognizing actions in videos Chen, Konrad & Ishwar (2018). In this work, one stream is trained on the optical flow features to extract temporal information and the other stream is trained on the still images to learn the appearance features. At the end the learned features are fused as the output of the model. Similar to Chen *et al.* (2018), Yu, Liu, Liu & Deng (2018) proposed a multi-channel network that extracts the appearance information from faces

and temporal information from the optical flow features and investigates three feature fusion strategies including SVM-based fusion, score average fusion and neural network-based fusion.

1.3.3 3D CNN

In conventional 2D CNNs, convolutions are applied on the 2D appearance features and mostly the spatial features can be extracted. On the other hand, for temporal analysis, we are interested to capture the temporal information between frames. To this end, 3D convolution has been proposed to compute features from both spatial and temporal dimensions. The 3D convolution is calculated by convolving a set of cube formed of convolutions.

In other words, the 3D convolution is just an extension of 2D convolution where instead of applying 2D convolutions on the input sequence, it applies a 3D set of convolutions.

1.3.4 GAN-Based Models

Recently, Generative Adversarial Networks (GANs)-based methods have been successfully used in image synthesis to generate impressively realistic faces, numbers, and a variety of other image types, which are beneficial to training data augmentation and the corresponding recognition tasks. Zhang *et al.* (2018) proposed a GAN-based model that can generate images with different expressions under arbitrary poses for multi-view FER.

Another advantage of GANs is that the identity variations can be explicitly disentangled through generating the corresponding neutral face image Yang, Ciftci & Yin (2018a) or synthesizing different expressions while preserving the identity information for identity invariant FER Yang, Zhang & Yin (2018b). Moreover, GANs can help augment the training data on both size and diversity. The main drawback of GAN is the training instability and the trade-off between visual quality and image diversity.

1.4 Datasets

In this thesis we test our model on the following datasets:

- **CK+ Database:** The CK+ dataset contains 593 facial expression sequences from 123 participants. The duration of the expression sequences varies between 4 and 71 frames and the location of 68 facial landmarks are provided along with database. Facial poses are frontal with slight head motions;

- **BU-4DFE:** This facial expression database consists both 3D and 2D expression videos. This dataset consists of six different facial expressions including *Happiness. Anger, Fear, Sadness, Disgust*, and *Surprise*. The database contains 606 three dimensional facial expression sequences captured from 101 subjects, with a total of approximately $60,600$ frames.

1.5 Performance Metrics

Selecting a decent metric is vital in evaluating machine learning (ML) models. In practical tasks, a variety of learning algorithms can be selected, and even for the same learning algorithm method, different parameters lead to different results. Evaluation metrics are critical for understanding the merits and demerits of each method, because it provides a standard to measure comparisons. In this section, we present the evaluation methods and evaluation metrics that are publicly available in the FER domain. The recognition rate of different methods is also compared with the FER typical classification method introduced in the previous section.

1.5.1 Evaluation Methods

Evaluating machine learning models or algorithms is essential for any projects. The difference among different types of evaluation methods is the ratio of correct prediction and total samples in training sets and test sets. In multi-classification tasks like FER, each category of emotion should be divided into training sets and test sets in the same way, and the model is evaluated according to the average performance of each category of emotion. Commonly used evaluation methods include the hold-out method, K-fold cross-validation, leave-one-out cross-validation (LOOCV), and bootstrapping method Lucey, Cohn, Kanade, Saragih, Ambadar & Matthews (2010).

In hold-out method the dataset is split up in test set and train set and it helps with avoiding over-fitting. The training set is what the model is trained on and the parameters are tuned, and the test set is used to see how well our model performs on unseen data. Therefore, in hold-out method the evaluation result is sensitive to the ratio of the training set and the validation set partitions.

The k-fold cross-validation or Cross-validation is when the dataset is randomly split up into k groups. One of the groups is used as the test set and the rest are used as the training set. The model is trained on the training set and scored on the test set. Then the process is repeated until each unique group as been used as the test set. For example, for 10-fold cross validation, the dataset would be split into 10 groups, and the model would be trained and tested 10 separate times so each group would get a chance to be the test set. This method avoids over-fitting and under-fitting effectively, but is computationally expensive due to k parameter, as it needs to be trained k times and tested k times.

LOOCV is a special case of K-fold cross-validation when the value of parameter k is equal to the number of the samples. LOOCV is suitable for small samples because the sample utilisation rate

is the highest. Nevertheless, high utilisation will lead to high computational complexity when dealing with large sample problems. The bootstrapping method is useful when the sample size is small and it is difficult to partition the training set and the test set effectively. However, the bootstrapping method changes the distribution of the initial data, which introduces an estimated bias.

1.5.2 Evaluation Metrics

The evaluation metric plays a vital role during the training process and the selection of which is an important key for discriminating and obtaining the optimal classifier. FER is naturally a multi-class classification problem, Acc (accuracy), i.e., the proportion of the samples that are correctly classified is a direct performance evaluation metric. In order to comprehensively take the recognition effect for each category of expression into consideration, the final accuracy can also be defined as the average of the recognition accuracy of each category of expression. These two methods of accuracy calculation are called overall accuracy and average accuracy, respectively. In general, higher accuracy stands for better classification performance. The definition of Acc is given below:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{1.3}$$

where TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative, respectively. The maximum F_1-score. The F_1-score is defined as:

$$F_1 = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{1.4}$$

F_1-score conveys the balance between the precision and recall. The F_1-score is a better performance measure than accuracy because accuracy is designed to measure the binary classification rather than detection and fails to reflect the effect of the proportion of the positive to negative samples.

1.6 Chapter Summary

Reviewing the current literature on FER systems, we identify that designing an accurate system which are able to exploit spatio-temporal dynamics of this task to improve the overall performance still remain a challenging task, and we identify the feature representation as being a key factor limiting the performance of such systems. Literature on expression recognition can be categorized in two main approaches, conventional systems and deep learning based systems. This chapter reviewed some previous work on each system. We discussed that the conventional approaches mostly extract information from local spatio-temporal patches from the video frames, however they are sensitive to illumination and occlusion of the objects. On the other hand, deep systems represent a high dimensional representation of input data, but they are not able to fully capture the temporal and appearance features of the input frames. Deep learning methods are able to handle a huge amount of data, however in the area of expression recognition, the amount of data is limited. On the other hand. conventional approaches, are simple and efficient methods for facial expression classification.

CHAPTER 2

STATISTICAL MODEL

2.1 Introduction

In previous chapter we have demonstrated how our desirable system looks like. The problem of expression recognition can be categorized in two classes: Statistical systems and deep-based systems. Since one of the our aims in the work is to compare these two methods, in this chapter we will discuss our conventional method for expression recognition. In the proposed method we use a local feature extraction method and a linear classifier for expression recognition. We will explain the proposed approach in the next sections.

Recently the expression recognition has been drawing enormous attention among machine learning and computer vision community due to its considerable influence and functionality in different areas included different domains in security and surveillance camera, the interactions between humans and machines and robotics. In this chapter, an efficient method will be introduced for facial expression recognition task which has already proved its robustness. We will review and use one of the most well-known algorithms to tackle this problem.

The following sections of this chapter are organized as follows: section 2.2 discusses the details of our method, section 2.2.1 and section 2.2.2 explain the pre-processing and feature extraction methods we use, section 2.2.3 gives details regarding our classifier. The implementation and results are presented in section 2.3 and section 2.4 respectively. Finally we summarize this work in section 2.5.

2.2 Approach

As was discussed in the previous chapter designing a conventional facial expression recognition system can be achieved by taking three important steps, (i) frames pre-processing, (ii) extraction of features from each sequence and (iii) determine the desired classes. For preprocessing the video frames and tracking faces we use Viola and Jones Viola & Jones (2004) method. For feature extraction part we used Local Binary Patterns and for classification we used Support Vector Machines, we believe that both approaches are very efficient and help us to gain insight of our system. During the following sections each step and also our baseline will be discussed in details. Figure 2.1 is a flowchart diagram to show the pipeline of our proposed method.

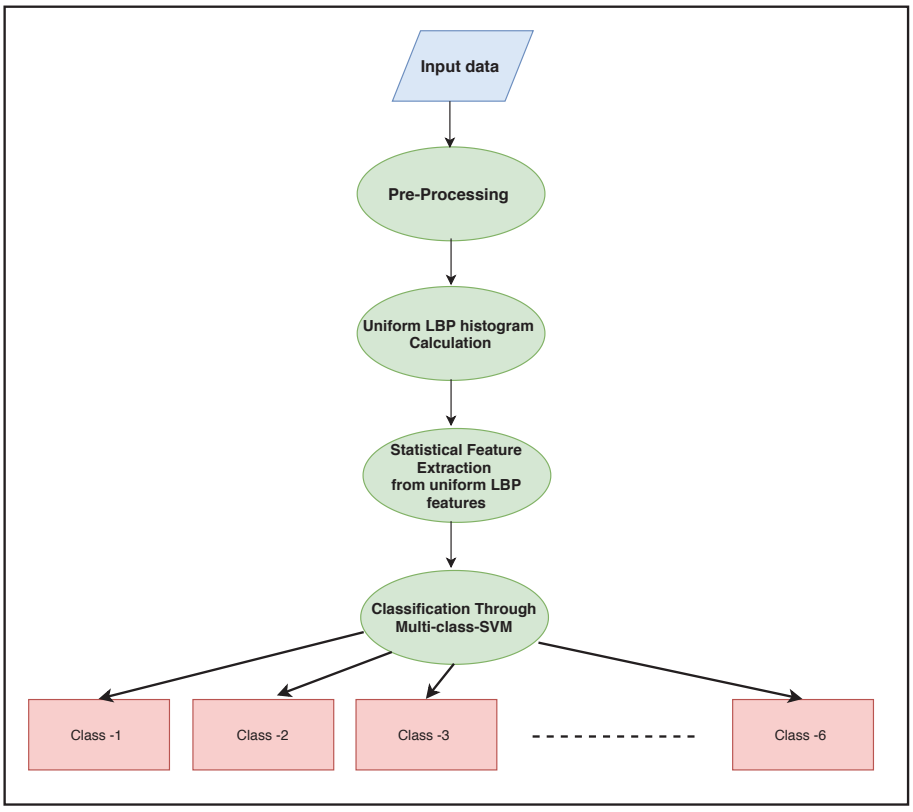

Figure 2.1 This diagram shows important steps in our research.

2.2.1 Image Pre-processing

The first step of our method is the pre-processing. Given video frames, we first detect face in each video frame using Viola and Jones Viola & Jones (2004) method. Viola & Jones (2004) proposed a robust near-frontal face detector using Haar features and AdaBoost classifiers. After finding the bounding box we crop the face in order to remove the background information before feature extraction and classification.

2.2.2 Feature Extraction

After performing pre-processing and detecting faces in video frames, the next stage is to exploit appropriate characteristics that can fully represent the facial movements and facial expression information. Applying Local Binary Patterns operator by extending it on Three Orthogonal Planes (LBP-TOP) is able to exploit space and temporal information into a single descriptor. In the next section each part of the algorithm will be explained.

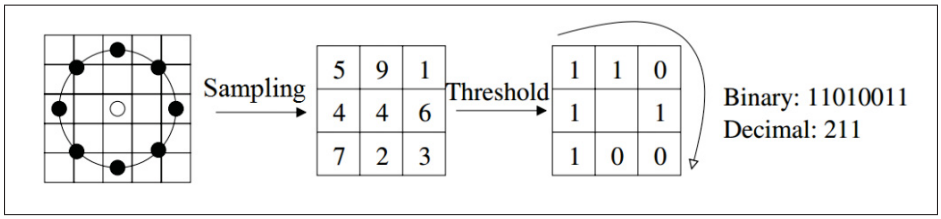

Figure 2.2 The sampling, threshold and creation of the central pixel decimal value

This operator generates some binary numbers by thresholding over pixel values of the image and defining a circles over each region of the image. By computing the intensities and making comparison between each region and its neighboring region the binary numbers will be produced. We utilised 59 bin $\sum_{8,2}^{u2}$ Operator.

2.2.2.1 LBP in time domain

The input of our desired system is video and we know that the videos contain motions of frames in temporal domain, therefore we tacking with the problem of motion we opt for using a dynamic texture descriptor that encodes this information. Thus, this fact inspired scientists to extend LBP in the way to cope with temporal information. LBP-TOP works exactly like LBP but it is able to capture information on three dimensions, which proved to be highly practical and efficient. Although the idea behind this algorithm seems simple, this method has showed promising performance in dealing with image sequences. By capturing distribution of features on each plane and concatenate those information together, we able to construct a Comprehensive feature vector which encompass both temporal and spacial information.

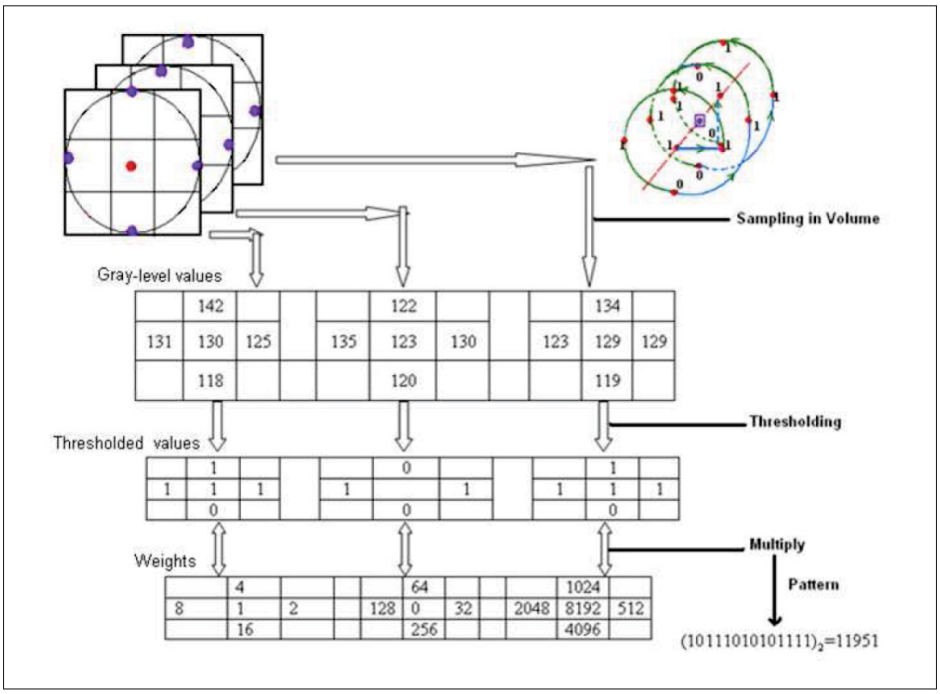

Figure 2.3 Generating binary samples by LBP - Top

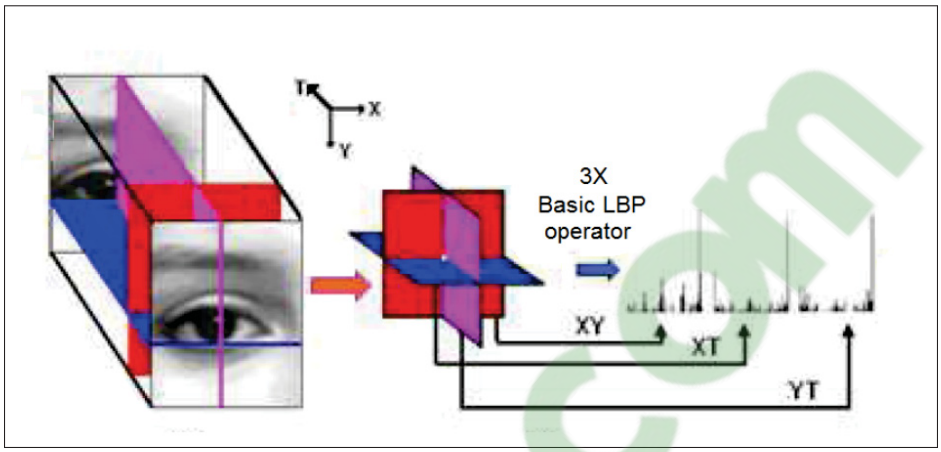

Figure 2.4 The procedure for the LBP-TOP

2.2.3 Classification

Classification of the given data is a common task in machine learning. In a classification problem, the goal is to approximate a function which can map a given vector data into one of the various class labels. Selection of fitting function according to output-input plays an important role the performance and accuracy of the classification. The finite input-output example data which is used for learning the classification function is called the training data.

Support vector machines (SVMs) is a powerful classification method that has achieved outstanding performance in many classification tasks. Therefore, we adopted SVMs as the classifier for facial expression recognition in this part. The main purpose of SVM is to find the hyperplane that maximizes the margin between the positive and negative observations for a specified class when given a collection of data points.

2.2.3.1 Support Vector Machines Hearst *et al.* (1998)

Support Vector Machines (SVM) Hearst *et al.* (1998) is one of the most suitable methods for our classification part which is a supervised learning problem.

They have strong theoretical foundations and have shown excellent empirical success in various fields. Support Vector Machines are trained so that the decision function would classify the unseen example data accurately. This ability to classify unseen example data accurately is referred to as generalization. High generalization capability is one of the main reasons for the success of SVMs.

Let us begin with the basic idea behind the SVMs. Given a set of a d-dimensional vectors, a linear classifier tries to separate the observed vectors with a $d-1$ dimensional hyperplane. There are many hyperplanes that are able to classify the data. If we define "margin" as the distance between the nearest samples on both sides of the hyperplane, SVMs are designed to choose the hyperplane that has the largest margin between the two classes. If such a hyperplane exists, it is known as the maximum-margin hyperplane and the linear classifier is known as a maximum margin classifier. Figure 2.5 illustrates how SVM works.

SVM is a widely used discriminative classifier that finds the optimal hyperplane to separate data patterns into two classes. It requires a small number of training patterns to correctly model the boundary. Consider a training dataset $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_l, y_l)\}$ a 2-class classification problem, where $x_i \in R^n$ and $y_i \in \{-1, +1\}$ represent an n-dimensional data pattern and the classes of these data, respectively, for $i = 1, 2, ..., l$. These data patterns are typically mapped into a higher dimensional feature space using a mapping function ϕ to find the best separation of classes. Therefore, the soft-margin optimization problem is formulated as the following expression:

$$\begin{aligned} &\min_{\mathbf{w},b,\xi} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{l}\xi_i \\ &\text{subject to: } y_i\left(\boldsymbol{w}^T\phi\left(x_i\right) + b\right) \geq 1 - \xi_i \\ &\xi_i \geq 0 \end{aligned} \tag{2.1}$$

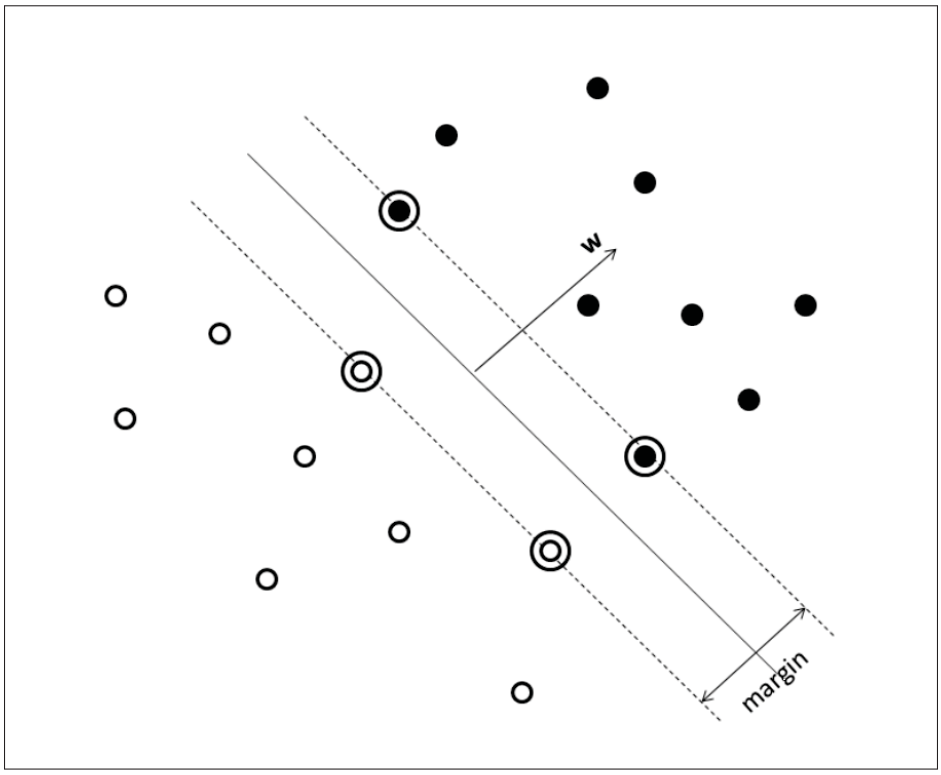

Figure 2.5 Example showing the margin and support vectors in the case of linearly separable data

where ξ_i variable is introduced to account for misclassified examples. Thus, ξ_i it can be considered as a misclassification amount, **b** is the bias, and **W** is the weight vector. Constant **C** is a misclassification cost of a training example, where it controls the trade-off between maximizing the margin, as well as, minimizing the number of misclassification.

2.3 Implementation

As illustrated in 2.1 at the first step we apply the pre-processing and face tracking on our input data. The face detection and extraction is done by using the builtin library from OpenCV package. Then cropped faces are resized to a fixed size 350×350 and also converted to grayscale in order to feed them to the feature extraction step.

The feature extraction from the videos was achieved using a dynamic texture descriptor, Local Binary Patterns on Tree Orthogonal Planes of the video. To apply this descriptor the implementation of Zhao & Pietikainen (2007) was adapted to our task. Their implementation was created to output three LBP histograms from a block of video, one for each plane.

In our approach, the video is not divided in a grid of pre-defined blocks; instead, all the pixels of a video could be chosen randomly to serve the central pixel of a block. Once the size of these blocks is fixed, this pixel is enough to reference a block. It is too computationally expensive and unnecessary to compute the LBP-TOP feature histogram for each possible block. As an alternative, the LBP-TOP value of each pixel is calculated for the whole video and stored beforehand ,one for each video plane.

Finally, for classifying facial expressions including, happiness, sadness, anger, fear, disgust, surprise, we use the LIBSVM Chang & Lin (2011) library. In order to allow for a fair comparison, the parameters are optimized empirically during a series of preliminary experiments. The effects of the parameter settings will be discussed in the following sections.

As we mentioned in previous section SVM was originally developed for binary classification. As our problem is multi-clsss classification problem, in order to extend SVM for our problem, we used the One-Versus-All approach, which trains a binary classifier to classify one class of interest (positive) versus all other classes (negative). These independent SVM classifiers are used to provide seven predictions of the presence or absence of the facial expression in unseen face images and the class with the greatest class-membership probability estimation value is output as the recognized facial expression. Also we set the kernel as linear. In our experiment, the dataset is randomly divided into 6 partitions of roughly equal number of subjects belonging to each facial expression class.

We used 4 partitions for training and 1 partition for estimating the parameters of the SVM classifier. After the parameters are fixed, the SVM classifier is applied to the last partition which is unseen during the training process of the classifier. The process is repeated, and the average recognition performance on the test sets are reported as the final result.

2.4 Results

In this section we report our results on the CK+ dataset (section 1.4). As was presented in the previous sections, our method leverages the benefits of local features that extract the spatio-temporal features from the video frames. Then the features are feed to an SVM for multi-class classification. The results of this experiment can be seen in Table 2.1.

We have also compared our method with the state-of-the-art. We compared with Wu, Bartlett & Movellan (2010) whom used spatio-temporal Gabor filter and Lorincz, Attila Jeni, Szabo, Cohn & Kanade (2013) that used dynamic time warping for feature extraction.

Table 2.1 Results on the CK+ dataset.

Expressions	**Our method (LBP-TOP + SVM)**	**Wu *et al.* (2010)**	**Lorincz *et al.* (2013)**
Happy	**97.53**	87.70	89.20
Surprise	**94.97**	87.90	90.90
Sadness	**92.56**	78.40	84.30
Anger	79.94	82.90	**87.30**
Fear	**87.61**	66.70	79.30
Disgust	**92.39**	67.70	89.30
Mean	**90.80**	78.60	86.70

2.5 Conclusion

This chapter presented our method for expression recognition. The proposed algorithm uses LBP-TOP feature extraction method for extracting spatio-temporal features from a sequence of

raw data. Then the feature vectors are fed into a linear SVM classifier for expression recognition. We evaluated our model on CK+ dataset and we showed that this approach is effective for expression recognition. We also compared our method with other conventional approaches and we observed an improvement in classification accuracy.

CHAPTER 3

DEEP LEARNING FER MODEL WITH SYNTACTIC DATA

3.1 Introduction

A large scale of data has become increasingly available due to advancement in tools such as the imaging devices and the explosion of data on the Internet. In addition, the power of modern graphics processors (GPUs) which are very efficient in image processing and mass calculation motivated computer scientists to extend their work in deep learning and carry out experiments with different frameworks. Due to efficiency of Neural networks in many tasks such as locating, classifying, detecting and segmenting objects. The breakthrough of CNNs is that features are learned automatically from training examples.

In this chapter we will start by introducing the convolutional neural network and describe the tools and basic operations used to construct an architecture for such a network. In addition, we will present a review of the most important components that have improved the accuracy of these models in recent years.

3.2 Convolutional Neural Network

Those networks are originally model of statistical learning inspired by biological neural networks of the human nervous system, shown in Figure 3.1. Generally a biological neuron has a set of entrance dendrites that receive electrical signals, then the signal crosses a threshold, the neuron is activated and the signal is transmitted to other neurons by dendrites.

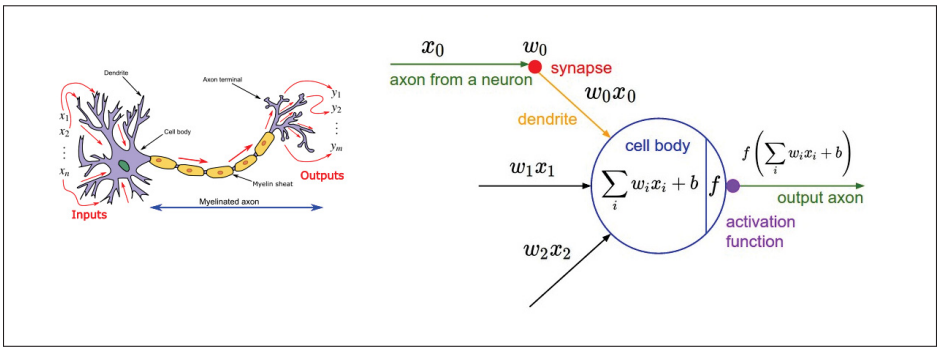

Figure 3.1 Examples of Neural Network.

They can be classified based on their number of hidden layers and their connection. Neural networks with more than two hidden layers can be thought of as a Deep Neural Network. The advantage of using deeper neural networks is that more complex models can be recognized. These networks require more data to avoid over-fitting during training.

Neurons read an input, process it, and generate an output. Neurons in between adjacent layers are fully connected. Each connection has a weight that controls the signal between the two neurons. Each artificial neuron calculates the sum of the products between the weights and inputs that came to it, then added a bias.

Although Convolutional neural networks and ordinary neural networks have some similarities about their basics, their architecture are completely different from regular neural networks. Generally, convolutional neural networks are made up of three main types of layers; the convolutional layers and activation function. These layers consider the context and spatial information of neighboring pixels, which leads to learn more information from the entrance. The configuration and the number of these layers in the network architecture depends on the type and complexity of the problem. In the next sections we will discuss about these stages in more details.

3.2.1 Convolution layer and activation function

A convolutional layer encompass a series of filters. Each filter is convolved with a number of pixels to compute an activation map made of neurons. In other words, the filter is slid across the width and height of the input image and the dot products between the input and filter are computed at every spatial position.

Convolution is a linear operation and the purpose of the activation function is to introduce the nonlinearity. It is an additional operation used after each convolution operation. The hyperbolic tangent function and the sigmoid function have been widely used in machine learning and in some implementations of basic neural networks. The rectified linear unit (ReLU), introduced by Nair & Hinton (2010), becomes the most commonly used non-linearity function in convolutional neural networks, this function puts just negative entries at zero. All the positive, the spatial information and the depth remain unchanged. When all negative values become zero, it causes to reduce the model's ability to train or adjust to the data correctly. The problem here is that the output will always be zero if the input is negative as well as the gradient. It may basically deactivate (kills) neurons and prevent them from learning. Figure 3.2 visualizes how the ReLU function works.

After convolution and the non-linearity function, most CNNs add a layer pooling between the convolutional layers. It is used continuously to reduce the dimensionality and the number of parameters. This shortens the calculation and the time learning in the network and controls over-learning.

3.2.2 Downsampling

The feature map is downsampled in such a way that the maximum feature response within a given sample size is retained. Subsampling, also known as pooling is performed to retain the

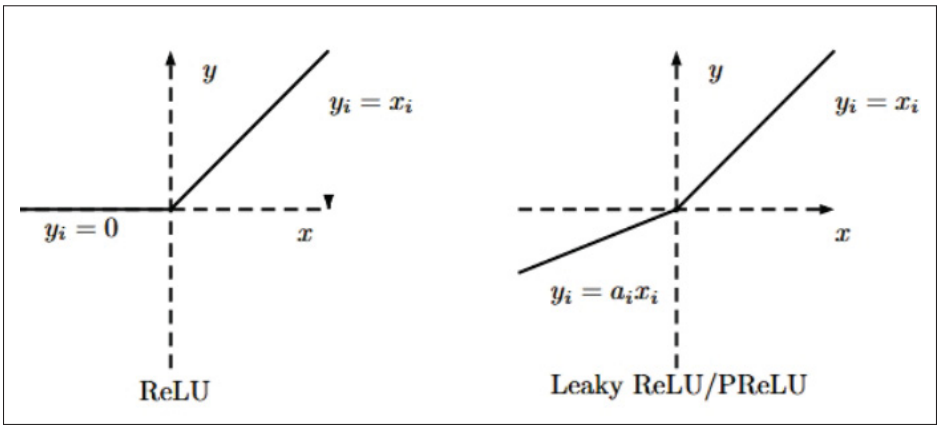

Figure 3.2 ReLU, Leaky ReLU and PReLU. For PReLU, a_i is learned in the back propagation drive and for Leaky ReLU a_i is fixed He *et al.* (2015)

most important information. Like convolutional layers, pooling is to drag a window onto the input data. It works independently on each spatial slice of the entrance and resizes its size. There are several types of subsampling, the most popular are the maximum pool and the average pool as is illustrated in Figure 3.3.

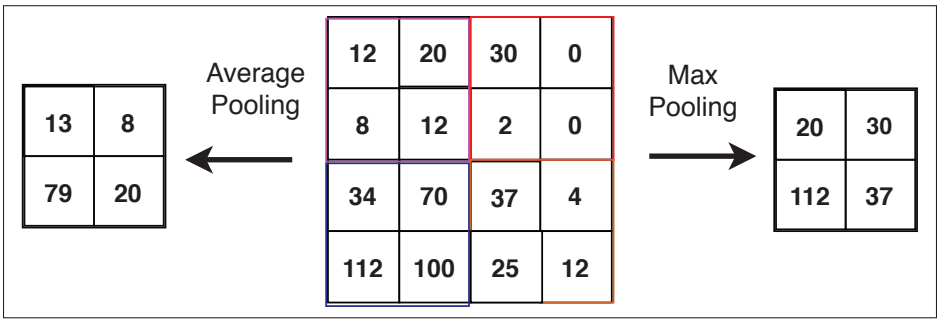

Figure 3.3 Different types of 2 × 2 pooling with a step of 2. For the maximum pool, the output is the maximum value in each 2 × 2 size window. For pooling mean, the output is the mean of the values.

In the most of Convolution Neural Network architectures, the convolutional block of the network is followed by one or more fully connected layers. These layers take the flattened characteristic maps as input and transmit them through the neural network. They calculate the score of

each class from the high level extracted features from the convolutional layers. The last fully connected layer is used as the classification layer. It contains only one node for each target class in the model.

There are various activation functions that could be applied to the last layer of fully connected. Generally activation function differs from task to task and is linked to the result of desired output. The appropriate activation function for the pixel classification task multi-class is the softmax function. This function is the most widespread and the most used for image classification tasks. It normalizes the actual output values of the last fully connected layer in target class probabilities, where each value is between 0 and 1 and the sum of all values is 1.

3.2.3 Recurrent Neural Networks

CNNs have been proved powerful in image related tasks like computer vision and image classification. In spite of those capabilities in extracting spatial information, they lack the ability to exploit temporal information. On the other hand, RNNs are useful for dealing with sequence information, capturing both short and long-term relationships in the data Schuster & Paliwal (1997).

RNNs are a popular class of Neural Networks specialized in sequential processing which have shown promising results in many temporal tasks such as NLP and video analysis Donahue, Anne Hendricks, Guadarrama, Rohrbach, Venugopalan, Saenko & Darrell (2015). Unlike conventional neural networks that assume all the inputs and outputs independent from each other, RNNs' inputs and outputs can be selected in arbitrary size and depend on previous observations. In other words, the training step t depends also on step $t - 1$. This enables RNNs to capture temporal correlations in the form of using the weights.

One common problem with the previous 2D-CNN based methods for facial expression analysis is that, the Convolution layers do not learn the temporal information between frames. This

information is vital for expression recognition problem Tran, Bourdev, Fergus, Torresani & Paluri (2015). In addition, due to the lack of data for the task of expression classification, previous CNN based methods usually utilised pre-trained models which have been trained on large object classification datasets and fine-tune the pre-trained models on the expression videos Walecki, Rudovic, Pavlovic, Schuller & Pantic (2017). The problem with these approaches is, they are trained on the object data and cannot fully capture the facial expression features.

3.2.4 Transfer learning

Recently, the invention of deep neural network with its ability to learn from large labeled training data has resulted in dramatic improvement in the performance of many computer vision tasks. Significantly, outperforming the classical statistical machine learning techniques, notably in areas such as object and event recognition and event analysis. Deep learning based methods while showing great promise in many feature extraction and classification tasks, unfortunately suffer from the disadvantage of requiring large amounts of training data to provide effective solutions.

3.3 Generating Synthetic Method

In this section we briefly demonstrate the approach of generating new samples. This task is accomplished in two stages: initially *Modeling of Faces*, that stands for the face template and the process of generating different faces with various textures; secondly *Modeling of Expressions*, that includes the process of face fitting and expression generation.

3.3.1 Modeling of Faces and Expressions

As is presented in Abbasnejad, Sridharan, Nguyen, Denman, Fookes & Lucey (2017b) to attain different subjects by different expressions two parametric models are required: the *shape* and

texture models. Therefore, different subjects by different expressions can be generated by changing the shape and texture parameters. In the following sections we briefly explain the theoretical details of this approach.

To build a $3D$ facial shape model, initially a set of $3D$ training meshes should be transferred into dense correspondence. Once the correspondence between the vertices of all scans and the corresponding meshes is established, they then should be brought into a shape space by applying some transformations (included, Generalized Procrustes Analysis and then PCA).

$$\boldsymbol{S}(\boldsymbol{p}_i) = \bar{\boldsymbol{s}} + \boldsymbol{U}_s \boldsymbol{p}_i, \tag{3.1}$$

where $\boldsymbol{p}_i = [p_1, ..., p_{n_s}]^T$ are the first n_s shape parameters.

where the texture vector contains the R, G, B color values of N corresponding vertices. The $3D$ texture model is then constructed using the set of training examples. Texture is extracted by applying PCA. Then the new texture example will be established using the functions $\mathcal{T} : \boldsymbol{R}^{n_t} \rightarrow \boldsymbol{R}^{3N}$ as,

$$\mathcal{T}(\mathbf{b}_i) = \bar{\boldsymbol{t}} + \boldsymbol{V}_t \boldsymbol{b}_i, \tag{3.2}$$

where $\boldsymbol{b} = [b_1, ..., b_{n_t}]^T$ are the first n_t texture parameters.

3.3.2 Expression model

It should be considered that the facial expression space is dependent on the face space. Facial expression can be generated by changing the shape and texture parameters in the facial domain. Therefore in order to generate different facial expressions, we should manipulate the weights of the PCA components. In order to define the facial expression sequence we utilize a $3D$ template mesh of a face, the shape parameters, and the animation sequence. The mesh topology uses a $3D$ mesh of a face explained in Section 3.3.1, and to estimate the shape parameters six scanned facial expressions were fitted to the face template in order to generate synthetic facial expression data.

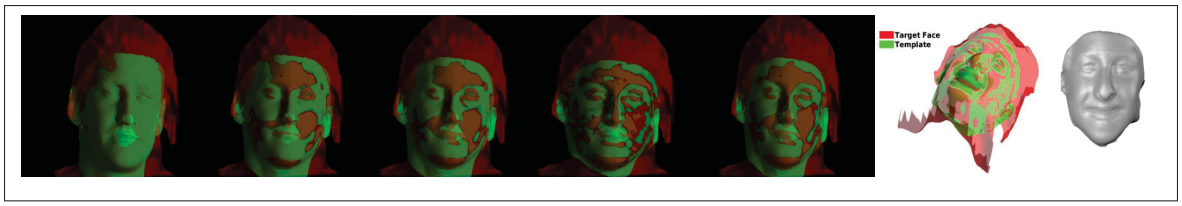

Figure 3.4 The first five faces show the fitting process from template (green) to scanned face (red).

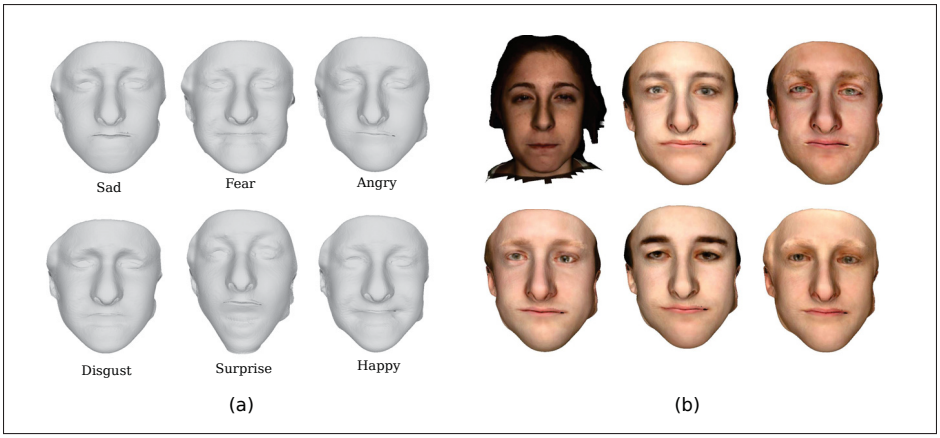

Figure 3.5 (a), Examples of scanned faced fitted to the template. (b), Different synthetic subjects that are performing, *anger* expression, Zhang *et al.* (2018)

Different subjects can be created by changing the texture parameters in Eq. 3.2. Five examples of the generated subjects are shown in Figure 3.5. This figure illustrates five different subjects of expression *Anger*.

3.4 Conclusion

In this chapter we explained about the basic of neural networks, their structure and how they work. We reviewed some recent methods in the area of expression recognition and we explained what could be the main limitation of deep neural network for expression task. As was discussed deep neural networks need a large scale data for training. However, in the area of expression recognition, the amount of data is limited. In addition, it is hard to create a large amount of expression data. In this chapter we present a simple and efficient method for generating a large scale of synthetic data. We will use this data for training a deep neural network model. We will discuss regarding our model and the training in the next chapter.

CHAPTER 4

PROPOSED MODELS AND ARCHITECTURES

4.1 Introduction

We went through literature regarding the process of expression recognition using statistical methods which have demonstrated that can be categorized in three stages: (i), data pre-processing, (ii), feature extraction and (iii), classification. However, one of the main limitations of the conventional approaches is that parameters in each stage has to be tuned separately under different strategies. On the other hand the desirable FER system should ideally consist of: (i) spatial feature representation: which must be efficient and be able to generalize to any arbitrary subject regardless of the environment and identity of each face and (ii) temporal modeling: that should be able to learn all the temporal correlations and variation dynamics among the video frames. Therefore, our desired model should be encompass both structure.

As we discussed earlier, one of our primary purposes of this thesis is to tackle the problem of expression recognition more efficiently by designing an end-to-end model. Therefore, in this chapter we are going to introduce and deploy a model which is able to bypass the intermediate tuning process to obtain our required output. This framework allows us to use a unique optimization technique to enhance our model. End-to-end automatic recognition models have shown superior performance and have been successfully applied in many tasks to solve many complex problems. End-to-end models refer to train a possibly complex learning system represented by a compact platform, in fact the model is learned given only the input data.

There are numerous machine learning methods in the literature that perform well on temporal classification tasks An & Liu (2019); Ji, Xu, Yang & Yu (2013); Ma, Sigal & Sclaroff (2016);

Tran *et al.* (2015), after a review of backbone models and reported results, we limited our experiments to the hybrid model with three Dimensional Convolutional Neural Networks and a Long-term Recurrent Neural Network (LSTMs).

Concerning the choice of the model, the hybrid model has been selected. 3D*ConvNet* architecture was selected based on Tran *et al.* (2015) which was initially introduced for action and activity recognition. Since there are some similarities between action recognition and facial expression classification tasks, Tran *et al.* (2015) these facts motivated us to examine whether such a model can be applied for expression analysis. In addition, LSTMs have shown promising results in many sequential data analysis, those models inspired us to construct a hybrid model which integrates two promising architectures. It is been proved that this model has capacity in representing variations of expressions for classification, since 3D*ConvNet* consists of three dimensional convolution which the third dimension signifies the temporal variations over time axes and also three dimensional pooling, which are are able to observe the appearance of the faces and learns the temporal dependency among frames and LSTMs and initially devised for capturing maximum temporal dependencies.

In addition, as was discussed in the previous chapter, another novelty of our project is utilising syntactic data for training neural networks. It has been proven that while the network architecture can strongly impact the performance of an AI system, in most situations the amount of training data have an enormous influence on the performance since adding more examples, adds diversity and it decreases the generalization error. As shown in Figure 4.1, we use the facial expression model we explained in the previous chapter to generate synthetic expression sequences. We will discuss regarding our approaches in the next sections.

In this chapter we will first start with a presentation of our proposed model in section 4.2. Then we will discuss the evaluation and the results of our model in comparison with the state-of-the-art in section 4.3. Finally we will summarize our results in conclusions and recommendations.

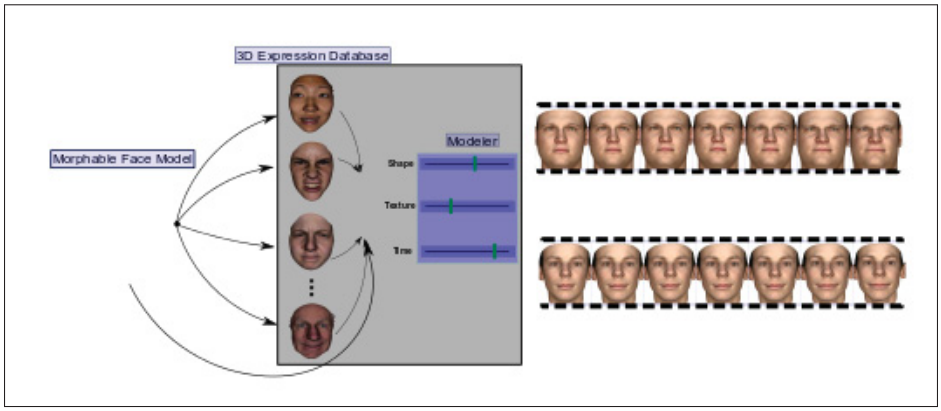

Figure 4.1 Our synthetic facial expression module.

4.2 Network architectures and training process

As discussed in the introduction, one of our novelty of this thesis is to learn the spatio-temporal features using a deep neural network. In this section, we will explain about the deep neural network structure we used in this research. Figure 4.2 shows the pipeline of our approach. We will divide our method into two stages. (i), a C3D neural network that extract the spatial features that are correlated with temporal information and (ii), a LSTM model that learns the temporal information based on the features extracted from C3D.

C3D:

After studying different models and statures we concluded to deploy a model which is able to efficiently work with image sequences and to extract temporal dependencies. We conducted several experiments on different networks and the final architecture is depicted in Figure 4.3, our three dimensional convolutional neural network has 8 convolution, 5 max-pooling, and 2 fully connected layers, followed by a softmax output layer. The configuration is the same as Abbasnejad, Sridharan, Denman, Clinton & Lucey (2017a). All 3D convolution kernels are $3 \times 3 \times 3$, with stride 1 in both spatial and temporal dimensions. The convolution layers consist

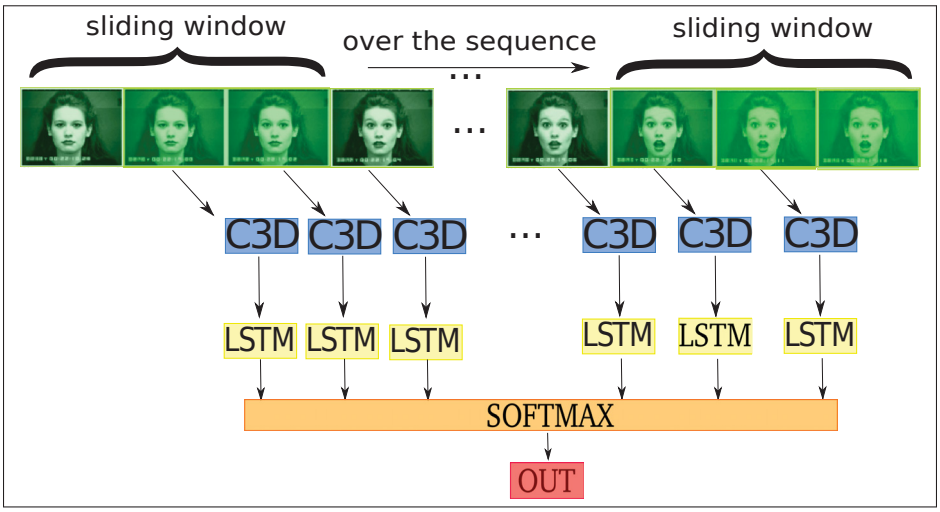

Figure 4.2 Our proposed model of FER system.

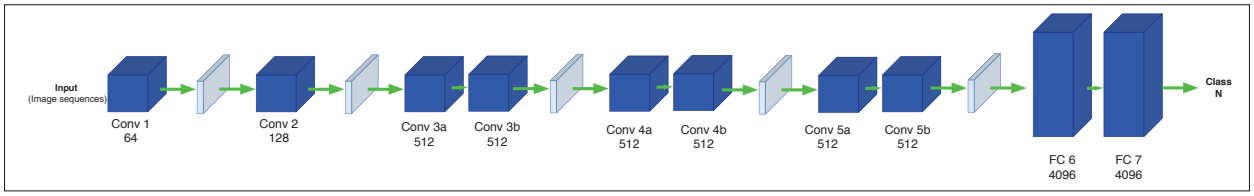

Figure 4.3 3DCNN architecture

of, 64, 128, 256, 256, 512, 512, 512 and 512 filters respectively and the last two fully connected layers have 4096 outputs.The input of the C3D is a sliding window that we can change its size from 2 frames to max 16 frames.

LSTM:

As we anticipated the integration of LSTM and three dimensional convolutional neural network may boost the over performance of our network, we tried to extend our network by a devising recurrent neural networks which are well-known for their ability with working with sequences.

Our LSTM model consists of one layer. The output of the C3D model is the input for our LSTM model. The output of the LSTM layer is fed to a softmax layer to transform the output codes to probability values of class labels.

Training:

In the training phase we first resize all the input images to $3 \times 350 \times 350$. Then, we start with pre-training the C3D network on the synthetic expression data which was introduced in section 3.3.2. We train for 50 epochs and we use the Adam optimizer Kingma & Ba (2014) for training the model. We batch size of 30 for different window sizes shown in Figure 4.2 and a learning rate of 10^{-3}. After the pre-training step, we fine-tuned our network on the real datasets. We use Pytorch Paszke, Gross, Massa, Lerer, Bradbury, Chanan, Killeen, Lin, Gimelshein, Antiga, Desmaison, Kopf, Yang, DeVito, Raison, Tejani, Chilamkurthy, Steiner, Fang, Bai & Chintala (2019) framework for implementing the neural network. In order to avoid exploding gradients in our LSTM model we use the Pytorch implementation of clipping gradient Pascanu, Mikolov & Bengio (2013). Figure 4.4 shows the learning curve of training C3D+LSTM on the synthetic data. In this experiment we set the value of window size as 4.

4.3 Evaluation

This section provides details regarding our evaluation settings and the results of our proposed models on the expression datasets. Similar to Abbasnejad *et al.* (2017b) we evaluate our method with generic and alternative approaches using two scenarios for facial expression recognition: *Within-dataset, Cross-dataset, Synthetic model*, and *Real dataset*. We report results separately for each scenario Chu, De la Torre & Cohn (2017).

4.3.1 Dataset

For evaluation we use the following datasets:

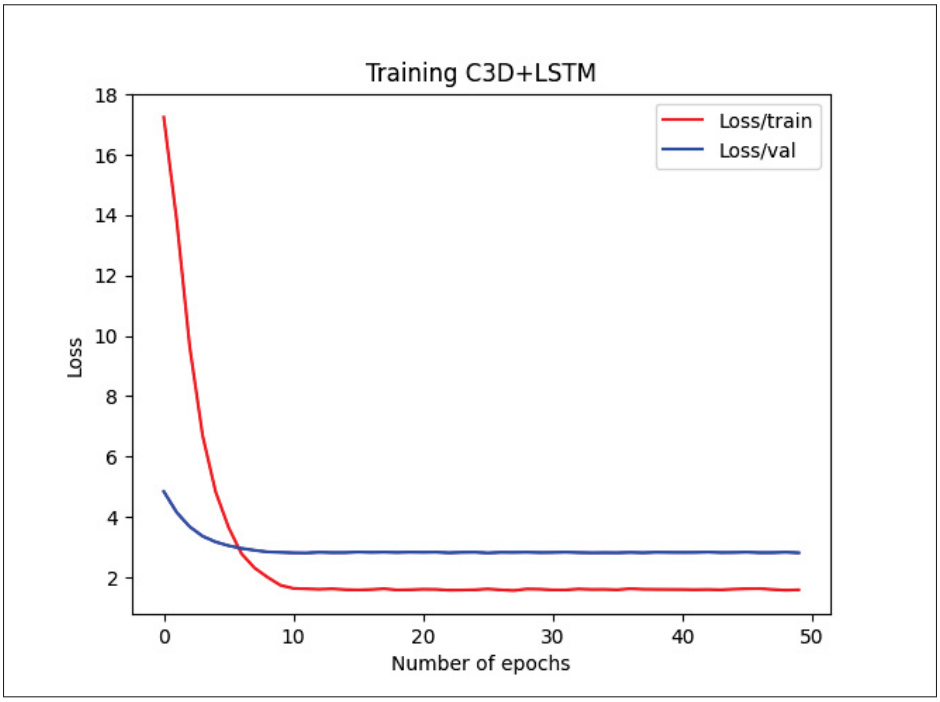

Figure 4.4 Training and validation loss for every epoch

- **Synthetic facial expression:**

  To meet our goal to tackle the problem of data shortage, we used synthetic expression dataset. According to Abbasnejad *et al.* (2017b) researchers in the artcile explained how to generate 12, 000 facial expression sequences where each expression sequence contains 16 frames.

- **Cohn-Kanade dataset:** See section 1.4 for details;

- **BU-4DFE dataset:** See section 1.4 for details.

4.3.2 Evaluation Setting

- **Images pre-Preprocessing:** The length of each sequence in our dataset is not consistent therefore we did some pre-processing steps to fix each sequences in the equal sizes. For those sequences which are less than our fixed size we repeat the last frame. In the sequences which are longer we dropped the frames with respect to the following frame ratio,

- **Train/Test split:** Initially we pre-train the network on the unreal dataset faces. Then, we fine-tune our network on the real datasets. 10-fold cross validation has been used for evaluation.

- **Face tracking registration:** To make sure all the faces contain the same bounding box, we pre-processed all videos with the same method presented in section 2.2.1. The tracked faces are then cropped into 350350.

- **Evaluation metrics:** To evaluate the performance, we used the area under ROC curve, and the F_1-score.

4.3.3 Results

In this section we report the performance of our model on the presented datasets. Our evaluation, consists of four scenarios:, (i) within-dataset, (ii) cross-dataset, (iii) synthetic evaluation, (iv) real data evaluation. We will explain each scenario in details in the following sections.

4.3.4 Within-dataset Evaluation

The goal in this evaluation procedure is to conduct our experiments by the same training and testing dataset. Initially we pre-train our architecture by the unreal dataset and then we fine-tune and test it on the same real-world dataset. A number of size of windows have been examined as is shown in Figure 4.2. Table 4.1 and Table 4.2 show the accuracy and F_1-score results on the CK+ and BU-4DFE datasets. As can be seen when we set the window size as 4 we gain the highest classification results.

4.3.5 Cross-dataset Evaluation

At first we carry out our experiments on the unreal data 3.3.2. After that we train the network on Bu-4DFE dataset and extract features from the last layer of C3D model for CK+ dataset. Then the extracted features from C3D network are fed into LSTM model. We follow the same process for Bu-4DFE dataset. Since from the previous experiment we observed setting

Table 4.1 Experiments on CK+ dataset. In this table ω defines the window size.

Class	ROC Curve			F_1-score		
	C3D+LSTM	**C3D+LSTM,** $\omega = 4$	**C3D+LSTM,** $\omega = 8$	**C3D+LSTM**	**C3D+LSTM,** $\omega = 4$	**C3D+LSTM,** $\omega = 8$
Happy	98.55	98.77	97.24	85.68	87.19	84.46
Surprise	98.05	98.16	97.48	78.36	79.86	77.77
Sadness	98.21	98.45	97.35	82.79	83.47	81.30
Anger	98.75	98.63	97.58	87.07	89.34	86.69
Fear	97.02	97.18	96.57	87.79	89.55	86.65
Disgust	98.87	98.87	96.74	88.09	89.09	85.63
Mean	98.24	**98.34**	97.16	84.96	**86.42**	83.75

Table 4.2 Experiments on the Bu-4DFE+ dataset. In this table ω defines the window size.

Class	ROC Curve			F_1-score		
	C3D+LSTM	**C3D+LSTM,** $\omega = 4$	**C3D+LSTM,** $\omega = 8$	**C3D+LSTM**	**C3D+LSTM,** $\omega = 4$	**C3D+LSTM,** $\omega = 8$
Happy	89.78	90.78	90.47	81.89	84.46	80.46
Surprise	92.69	94.27	91.48	82.98	85.87	81.37
Sadness	92.01	93.66	92.40	82.84	85.38	82.76
Anger	90.77	91.87	88.39	81.48	824.17	81.08
Fear	89.64	91.85	90.74	80.58	83.84	80.26
Disgust	88.47	89.56	87.67	78.13	81.55	78.04
Mean	90.56	**91.99**	90.19	81.31	**84.21**	80.66

window-size of $\omega = 4$ is performing better than window size of $\omega = 8$, we will set it as $\omega = 4$ for this experiment. The results can be observed on Table 4.3.

4.3.6 Synthetic Model

Our idea in this experiment is to see if we it is possible to use the unreal data as a data for training a network. In this experiment we only train the network on synthetic data. Then we used the trained model and extract features from the last fully connected layer of the C3D network on the real datasets. We then feed the extracted features to the LSTM layer for classification.

Table 4.3 Experiments on the CK+ and BU-4DFE datasets.

Class	ROC Curve		F_1-score	
	CK+	**BU-4DFE**	**CK+**	**BU-4DFE**
Happy	94.07	84.48	79.45	70.79
Surprise	94.18	87.87	72.79	70.57
Sadness	93.43	88.83	77.04	71.16
Anger	93.38	87.01	81.81	70.65
Fear	92.66	85.41	82.42	69.79
Disgust	94.69	83.76	84.01	68.87
Mean	93.74	86.22	79.59	70.31

Table 4.4 Experiments on unreal dataset the CK+ and BU-4DFE datasets.

Class	ROC Curve		F_1-score	
	CK+	**BU-4DFE**	**CK+**	**BU-4DFE**
Happy	87.84	77.84	70.63	65.84
Surprise	87.69	81.02	68.37	65.71
Sadness	85.48	81.31	70.56	60.85
Anger	85.98	81.74	77.35	60.79
Fear	86.43	79.17	76.47	61.53
Disgust	86.41	78.48	77.48	61.48
Mean	86.64	79.92	73.48	62.70

4.3.7 Real dataset

In this experiment we ask this question that, do we really need to use synthetic data or a large scale data for training a neural networks? To answer this question, we eliminate the training on synthetic data step and we only train and test the network on the real datasets.

Table 4.5 Real dataset experiment on the CK+ and BU-4DFE datasets.

Class	ROC Curve		F_1-score	
	CK+	**BU-4DFE**	**CK+**	**BU-4DFE**
Mean	65.21	56.14	46.71	40.02

4.3.8 Comparisons and discussions

As can be observed we have done a comprehensive study and conducted plenty of expeiments on diffrent datasets and made comparisons according to different scenarios. and have showed our results comparison of different strategies for the evaluation of our methods. As was empirically shown in previous sections, deep features can improve the classification performance significantly. However, one problem with the deep networks is their dependency on large scale datasets. To cope with this problem We used syntactic dataset to enrich our dataset which enables us to train a deep network in the context of expression analysis.

We compared different scenarios to intensely examine our approach in four scenarios: (i), within-dataset 4.3.4, (ii), cross-dataset 4.3.5, (iii), synthetic 4.3.6 (iv) real 4.3.7 experiments. For comparison, we compared the classification performance in two scenarios:

- when we use synthetic data for pre-training our proposed model and fine-tuning using real datasets;
- when we don't use synthetic data and for training we use only the real datasets.

As can be seen from the table 4.6, using synthetic data can enormously enhance the classification performance.

Table 4.6 Comparisons according to different scenarios

Method	ROC Curve		F_1-score	
	CK+	**BU-4DFE**	**CK+**	**BU-4DFE**
Training on unreal data	98.34	91.99	86.42	84.21
Training on CK+	65.21	-	46.71	-
Training on BU-4DFE	-	56.14	-	40.02

Comparing the results indicate that how a large scale dataset plays a vital role for training a deep network. Furthermore, it can be seen that, how using relevant data for training network is important. In addition, comparing Tables, show that using synthetic data for pre-training improves the classification performance.

Table 4.7 Comparing with the state-of-the-art.

Method	ROC Curve		F_1-score	
	CK+	**BU-4DFE**	**CK+**	**BU-4DFE**
Liu et al.Liu, Han, Meng & Tong (2014)	96.70	-	-	-
Mollahosseini et al.Mollahosseini, Chan & Mahoor (2016)	93.20	-	-	-
Abbasnejad *et al.* (2017b)	97.87	91.22	**86.59**	83.62
LBP-TOP + SVM 2	90.80	-	-	-
C3D+LSTM (ours)	**98.34**	**91.99**	86.42	**84.21**

On the other hand, the traditional method we presented in this project does not need a huge amount of data for training, the model we utilized is LBP-TOP. Although the model is not performing as good as a high capacity network, it is easy to generalize on any arbitrary dataset with any amount of training data (in contrast to the C3D+LSTM model which has to train on a relevant large scale dataset). In addition, our deep learning method is computationally expensive and needs a considerable amount of resources like time, processing power and memory.

CONCLUSION AND RECOMMENDATIONS

In this thesis we have focused on statistical machine learning and deep learning approaches for addressing the problem of facial expression analysis. Our work begins from scratch, building a dataset and training both conventional and newly developed networks frameworks. We also provided a comprehensive survey of FER models and the importance of its applications in both industry and government has been discussed in details.

We developed our model by comparing our architecture by other state of the art methods which have been proved by conducting intensive experiments. In experimental parts, to design a robust and accurate facial expression framework, abundant realistic training data is required and obviously obtaining such data is a tedious and time consuming task. To address this issue syntactic data has been used and extensive experiments have been performed to empirically evaluate deep learning models including 3 dimensional Convolutional Neural Networks and LSTMs. The evaluation process focuses on the task of expression recognition through facial expression in image sequences on a number of datasets including Ck+ Lucey *et al.* (2010) and BU-4DFE Zhang, Yin, Cohn, Canavan, Reale, Horowitz & Liu (2013).

In spite of the fact that C3D+LSTM network has been shown as a efficient architecture in learning spatio-temporal features for temporal analysis, as is proved in the literature (Tran *et al.*, 2015) having better performance over C3D networks is considerably correlated with the existence of plenty of training data.

For our future work, we are interested in fostering our research and expanding our project in deploying recently developed networks and data generation methods. In particular, with the existence of the generative adversarial networks (GAN) (Goodfellow, Pouget-Abadie, Mirza, Xu, Warde-Farley, Ozair, Courville & Bengio, 2014) which are able to be used for augmenting data by generating a variety of appearances in poses and expressions. In addition, with the progress of the 3D rendering models and frameworks, researchers recently have generated realistic synthetic dataset using the state-of-the-art techniques. We could also

extend our project in integrating our statistical features by deep learning representation and then select the best features to acquire high quality feature vector.

Another interesting idea which has gained a lot of popularity is recognition of facial expression by multi-modal approaches. Different approaches can be deployed in multi-modality, some by utilizing visual modalities obtained from the face or some using another source of information such as speech. It is anticipated that, by providing complementary information stream and deploying a robust system capable of exploiting information from different modalities can bring about more accurate result and higher performance .

REFERENCES

Abbasnejad, I., Sridharan, S., Denman, S., Fookes, C. & Lucey, S. (2015). Learning Temporal Alignment Uncertainty for Efficient Event Detection. *Digital Image Computing: Techniques and Applications (DICTA), 2015 International Conference on*, pp. 1–8.

Abbasnejad, I., Sridharan, S., Denman, S., Clinton, F. & Lucey, S. (2017a). Joint Max Margin and Semantic Features for Continuous Event Detection in Complex Scenes. *arXiv preprint arXiv:1706.04122*.

Abbasnejad, I., Sridharan, S., Nguyen, D., Denman, S., Fookes, C. & Lucey, S. (2017b). Using synthetic data to improve facial expression analysis with 3d convolutional networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1609–1618.

Ahonen, T., Hadid, A. & Pietikäinen, M. (2004). Face recognition with local binary patterns. *European conference on computer vision*, pp. 469–481.

An, F. & Liu, Z. (2019). Facial expression recognition algorithm based on parameter adaptive initialization of CNN and LSTM. *The Visual Computer*, 1–16.

Baraniuk, R. G. & Wakin, M. B. (2009). Random projections of smooth manifolds. *Foundations of computational mathematics*, 9(1), 51–77.

Chang, C.-C. & Lin, C.-J. (2011). LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3), 27.

Chen, J., Konrad, J. & Ishwar, P. (2018). Vgan-based image representation learning for privacy-preserving facial expression recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1570–1579.

Chu, W.-S., De la Torre, F. & Cohn, J. F. (2017). Selective transfer machine for personalized facial expression analysis. *IEEE transactions on pattern analysis and machine intelligence*, 39(3), 529–545.

Cohn, J. F., Zlochower, A. J., Lien, J. J. & Kanade, T. (1998). Feature-point tracking by optical flow discriminates subtle differences in facial expression. *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 396–401.

Déniz, O., Bueno, G., Salido, J. & De la Torre, F. (2011). Face recognition using histograms of oriented gradients. *Pattern recognition letters*, 32(12), 1598–1603.

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K. & Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2625–2634.

Ekman, P. & Rosenberg, E. L. (1997). *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, USA.

Fasel, B. & Luettin, J. (2003). Automatic facial expression analysis: a survey. *Pattern recognition*, 36(1), 259–275.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, pp. 2672–2680.

Guo, Z., Zhang, L. & Zhang, D. (2010). A completed modeling of local binary pattern operator for texture classification. *IEEE transactions on image processing*, 19(6), 1657–1663.

He, K., Zhang, X., Ren, S. & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034.

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J. & Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4), 18–28.

Horn, B. K. & Schunck, B. G. (1981). Determining optical flow. *Artificial intelligence*, 17(1-3), 185–203.

Huang, Y., Chen, F., Lv, S. & Wang, X. (2019). Facial expression recognition: A survey. *Symmetry*, 11(10), 1189.

Ji, S., Xu, W., Yang, M. & Yu, K. (2013). 3D convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1), 221–231.

Kass, M., Witkin, A. & Terzopoulos, D. (1988). Snakes: Active contour models. *International journal of computer vision*, 1(4), 321–331.

Kingma, D. P. & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kleinsmith, A. & Bianchi-Berthouze, N. (2012). Affective body expression perception and recognition: A survey. *IEEE Transactions on Affective Computing*, 4(1), 15–33.

Koestinger, M., Wohlhart, P., Roth, P. M. & Bischof, H. (2011). Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, pp. 2144–2151.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4), 541–551.

Li, S. & Deng, W. (2018). Deep facial expression recognition: A survey. *arXiv preprint arXiv:1804.08348*.

Lindeberg, T. (2012). Scale invariant feature transform.

Liu, P., Han, S., Meng, Z. & Tong, Y. (2014). Facial expression recognition via a boosted deep belief network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1805–1812.

Lorincz, A., Attila Jeni, L., Szabo, Z., Cohn, J. F. & Kanade, T. (2013, June). Emotional Expression Classification Using Time-Series Kernels. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

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

Lyons, M., Akamatsu, S., Kamachi, M. & Gyoba, J. (1998). Coding facial expressions with gabor wavelets. *Proceedings Third IEEE international conference on automatic face and gesture recognition*, pp. 200–205.

Ma, S., Sigal, L. & Sclaroff, S. (2016). Learning activity progression in lstms for activity detection and early detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1942–1950.

Martinez, A. & Du, S. (2012). A model of the perception of facial expressions of emotion by humans: Research overview and perspectives. *Journal of Machine Learning Research*, 13(May), 1589–1608.

Mattela, G. & Gupta, S. K. (2018). Facial Expression Recognition Using Gabor-Mean-DWT Feature Extraction Technique. *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 575–580.

Moghaddam, B., Jebara, T. & Pentland, A. (2000). Bayesian face recognition. *Pattern recognition*, 33(11), 1771–1782.

Mollahosseini, A., Chan, D. & Mahoor, M. H. (2016). Going deeper in facial expression recognition using deep neural networks. *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pp. 1–10.

Nair, V. & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814.

Pascanu, R., Mikolov, T. & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. *International conference on machine learning*, pp. 1310–1318.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J. & Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. & Garnett, R. (Eds.), *Advances in Neural Information Processing Systems 32* (pp. 8024–8035). Curran Associates, Inc. Consulted at http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Sánchez, A., Ruiz, J. V., Moreno, A. B., Montemayor, A. S., Hernández, J. & Pantrigo, J. J. (2011). Differential optical flow applied to automatic facial expression recognition. *Neurocomputing*, 74(8), 1272–1282.

Schuster, M. & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on*, 45(11), 2673–2681.

Shan, C., Gong, S. & McOwan, P. W. (2009). Facial expression recognition based on local binary patterns: A comprehensive study. *Image and vision Computing*, 27(6), 803–816.

Tran, D., Bourdev, L., Fergus, R., Torresani, L. & Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. *Proceedings of the IEEE international conference on computer vision*, pp. 4489–4497.

Tsai, H.-H. & Chang, Y.-C. (2018). Facial expression recognition using a combination of multiple facial features and support vector machine. *Soft Computing*, 22(13), 4389–4405.

Viola, P. & Jones, M. (2001). 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*, 1, I–511.

Viola, P. & Jones, M. J. (2004). Robust real-time face detection. *International journal of computer vision*, 57(2), 137–154.

Walecki, R., Rudovic, O., Pavlovic, V., Schuller, B. & Pantic, M. (2017). Deep Structured Learning for Facial Expression Intensity Estimation. *IMAVIS (article in press)*, 259, 143–154.

Wu, T., Bartlett, M. S. & Movellan, J. R. (2010). Facial expression recognition using gabor motion energy filters. *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pp. 42–47.

Yacoob, Y. & Davis, L. S. (1996). Recognizing human facial expressions from long image sequences using optical flow. *IEEE Transactions on pattern analysis and machine intelligence*, 18(6), 636–642.

Yang, H., Ciftci, U. & Yin, L. (2018a). Facial expression recognition by de-expression residue learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2168–2177.

Yang, H., Zhang, Z. & Yin, L. (2018b). Identity-adaptive facial expression recognition through expression regeneration using conditional generative adversarial networks. *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 294–301.

Yang, P., Liu, Q. & Metaxas, D. N. (2009). Boosting encoded dynamic features for facial expression recognition. *Pattern Recognition Letters*, 30(2), 132–139.

Yu, J. & Bhanu, B. (2006). Evolutionary feature synthesis for facial expression recognition. *Pattern Recognition Letters*, 27(11), 1289–1298.

Yu, Z., Liu, G., Liu, Q. & Deng, J. (2018). Spatio-temporal convolutional features with nested LSTM for facial expression recognition. *Neurocomputing*, 317, 50–57.

Zhang, F., Zhang, T., Mao, Q. & Xu, C. (2018). Joint pose and expression modeling for facial expression recognition. *Proceedings of the IEEE Conference on Computer Vision and*

Pattern Recognition, pp. 3359–3368.

Zhang, W., Shan, S., Gao, W., Chen, X. & Zhang, H. (2005). Local gabor binary pattern histogram sequence (lgbphs): A novel non-statistical model for face representation and recognition. *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, 1, 786–791.

Zhang, X., Yin, L., Cohn, J. F., Canavan, S., Reale, M., Horowitz, A. & Liu, P. (2013). A high-resolution spontaneous 3d dynamic facial expression database. *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, pp. 1–6.

Zhang, Z., Lyons, M., Schuster, M. & Akamatsu, S. (1998). Comparison between geometry-based and gabor-wavelets-based facial expression recognition using multi-layer perceptron. *Proceedings Third IEEE International Conference on Automatic face and gesture recognition*, pp. 454–459.

Zhao, G. & Pietikainen, M. (2007). Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6), 915–928.