Research on Facial Expressions Recognition based on Deep Learning Methods

馮 鐸

A Thesis submitted to Tokushima University in partial fulfillment of the requirements for the degree of Doctor of Philosophy

2021

Department of Information Science and Intelligent Systems
Graduate School of Advanced Technology and Science
Tokushima University, Japan

Contents

List of Tables

List of Figures

Abstract

Facial expression recognition (FER) is a process of automatically recognizing and inferring the performance of human emotional states on the face through artificial intelligence technology. As the most important part of recognizing human emotion, FER technology crosses and integrates physiology, psychology, image processing, machine vision, pattern recognition, and other research fields. It has received extensive attention in the fields of human-computer interaction, information security, robotics, automation, medical care, communication technology, autonomous driving, etc. Although decades of research work on FER have been carried out, in actual situations, realizing accurate and effective FER is still a challenging problem. In recent years, with the success of deep learning technology in various fields, more and more deep neural networks are used to learn the discriminative representation of automatic FER. This thesis studies FER by combining traditional machine learning methods and constructing efficient deep model architecture. The main contributions of this thesis are summarized as follows:

(1) This thesis first reviewed and summarized the currently widely used methods and existing problems in FER. After fully understanding the limitations of the traditional handcrafted features, a multi-stream neural network model combining the manually extracted LBP-TOP features and the deep learning model is proposed to recognize the dynamic process of facial expression changes. In the multi-stream neural network proposed in this thesis, to recognize dynamic facial expressions, the cascaded CNN-RNN model is used to extract the features of the input facial expression image sequence from space expand to time series. At the same time, the handcrafted features LBP-TOP is used to directly extract the spatiotemporal features of the image sequence, and then the CNN and RNN networks are used to process the spatiotemporal features. Finally, through the fusion of the two streams of features, and through experiments on the public database, it is proved that the handcrafted

spatiotemporal features can effectively supplement the CNN-RNN model and improve the results of FER.

(2) Application-oriented FER faces two challenges. One is the transition of FER from laboratory control to challenging in-the-wild conditions, and the other is the recent challenge of decentralizing deep network application technology to mobile platforms. Simply using larger and deeper neural network models for recognition tasks can no longer cope with this problem. In recent years, FER has been proved to be more natural and effective from consecutive frames. The motivation of this thesis becomes to create a lightweight network that processes dynamic facial expression sequences. After studying the amount of calculation of the model architecture, the MobileNet series with a deep separable convolution architecture is chosen as the basic model of the CNN part and used GRU as the frame-to-sequence approach part to construct a lightweight CNN-RNN cascade network. The performance improvement is demonstrated by using the proposed technique on both the laboratory control and in-the-wild conditions databases.

(3) Through previous research, first the supplementary ability of handcrafted features extraction for deep learning methods is verified. Then the application of the lightweight depth model in FER is discussed. Further used the updated technology to combine the advantages of local binary convolution (LBC) and deep separable networks and proposed a new model architecture. Inspired by model pruning and SE optimization, using the feature that the convolution kernel parameters in LBC are not trainable, this thesis proposes a pruning method on depthwise LBC and SE optimization model architecture. Experiments were not only conducted on the general image classification database, but also on the in-the-wild conditions facial expression databases. The experimental results prove the effectiveness of our proposed model and pruning method.

Chapter 1

Introduction

1.1 Background and Significant

Create Human-Machine Interaction (HMI) systems that able to reach the full emotional and social capabilities for rich and robust interaction with human beings will be a long and arduous but important task. Humanoid robots are required not only to have a Human-like appearance but also to possess the ability of emotional interaction [1,2]. Facial expression recognition (FER) is a vital research field to reach this goal since facial expression is one of the most powerful, natural, and universal signals for human beings to convey their emotional states and intentions [3,4]. In fact, it is not limited to HMI systems. Due to the actual importance of automatic facial expression analysis in the fields of social robots, medical care, driver fatigue monitoring, etc., a lot of research on automatic facial expression analysis has been carried out. As early as the twentieth century, Ekman and Friesen [5] defined six basic emotions based on a cross-culture study [6], which indicated that humans perceive certain basic emotions in the same way regardless of culture. These prototypical facial expressions are anger, disgust, fear, happiness, sadness, and surprise. Contempt was subsequently added as one of the basic emotions [7]. Recently, advanced research on neuroscience and psychology argued that the model of six basic emotions is culture-specific and not universal [8].

Research of FER is widely considered to have started with psychologists decomposed facial expressions as a combination of facial muscle activations, which

called Facial Action Units (AUs) under the form of the Facial Action Coding System (FACS) [9]. However, this emotional description model and a continuous model using emotional dimensions [10] are considered to represent a wider range of emotions. Due to the groundbreaking research on discrete basic emotions and the direct and intuitive definition of facial expressions, the classification model for describing emotions based on basic emotions is still the most popular view of FER. And in this thesis, I will limit my research on FER based on classification models with basic emotions.

In the process of researching FER, according to its characteristics, it can be divided into two categories: static image FER and dynamic sequence FER. In static-based methods, the feature representation is encoded with only spatial information from the current single image, whereas dynamic-based methods consider the temporal relation among contiguous frames in the input facial expression sequence. In addition, other modalities, such as audio and physiological channels, have also been used in multi-modal systems to aid expression recognition.

In the traditional method of FER, handcrafted features are mainly used to extract geometric-based and appearance-based feature of the target face region that needs to be recognized. The former refers to the FER systems, which extract local facial features including shape, positions, and angles between various facial elements, i.e., ear, eye, mouth, and nose, and the feature vector is illustrated based on the geometrical relationship. The latter refers to the FER systems, which describe the appearance and employ texture information of the face as a feature vector. The appearance-based approaches can obtain a higher recognition rate and are more popular than geometric-based methods since it is a complicated task to find the efficient and proper geometric features in unconstrained environments and real-world applications. However, research in recent years, especially the collection of relatively sufficient training data from challenging realistic scenarios, has promoted the transformation of FER from laboratory control to in-the-wild environments. This brings many challenges such as FER based on traditional methods. It is widely recognized that although the laboratory-controlled FER system achieves a very high accuracy of about 97%, the technology transfer from the laboratory to the actual application faces a huge obstacle of extremely low accuracy, approximately 50%.

At the same time, due to the rapid development of computer technology, especially chip processing technology (such as GPU units), deep learning technology has

developed rapidly. Thanks to the well-designed network architecture, research in various fields began to transfer to deep learning methods, achieving the most advanced recognition accuracy and greatly exceeding previous results. With more effective facial expression training data, deep learning techniques are increasingly being used to deal with the challenging factors of emotion recognition. In such a situation, FER's research methods have changed from handcrafted features to deep learning, and the research goal has changed from laboratory control to in-the-wild conditions.

1.2 Motivation & Research Contents

Although deep learning has powerful feature learning capabilities, problems still exist when applied to FER.

First, deep neural networks need a lot of training data to avoid overfitting. However, the existing facial expression databases are not enough to train well-known neural networks with deep architecture, which have achieved the most promising results in object recognition tasks.

Second, due to different personal attributes, such as age, gender, ethnic background, and expression level, there are high levels of inter-subject differences [11]. In addition to the subject's identity bias, changes in posture, lighting, and occlusion are also common in unconstrained facial expression scenes. These factors are nonlinearly coupled with facial expressions, thus strengthening the requirements for deep networks to resolve large intra-class variability and learn effective expression-specific representations.

Third, compared to the image classification task, FER is a subtask of face recognition, and the features that need to be extracted are concentrated in the range of the face. The granularity of its features is also much smaller than that of image classification tasks. This also requires the features extracted by deep neural networks to be more targeted.

Fourth, with the development of smart devices and mobile platforms, in the absence

of specific hardware (e.g. GPU units), the computing speed of deep neural networks does not dominate. In tasks including FER, some scenarios require real-time performance, which also means that there are further requirements in the deep network architecture and other processing including pre-processing.

In this thesis, we introduced some research progress in solving the above-mentioned deep FER problem. We try to combine the advantages of handcrafted features and deep learning technology to solve the problem of FER. At the same time, new deep learning network architectures were proposed, and solutions to existing problems are discussed based on ensuring lightweight and effectiveness.

This thesis revolves around the above challenges. The main research contents mainly include three parts.

(1) Multi-Stream Network for Dynamic FER

There are two main research objectives of FER, based on static images and based on dynamic image sequences. In recent years, facial expression recognition from consecutive frames has been proven to be more natural and effective. Under such a premise, it is very important and necessary to study the spatial and temporal characteristics of dynamic FER image sequences.

This chapter attempts to add the handcrafted LBP-TOP feature to directly extract the spatiotemporal features of the image sequence, and then use it as a stream of the deep network. At the same time, the general CNN-RNN cascade model with image sequence as input is used as another stream. Through the spatiotemporal features of multiple streams, using handcrafted features as a supplement to the deep network, we can get better results than using the deep model alone.

(2) Real-time FER System using CNN-GRU

For the FER task, many proposed CNN architectures are deeply influenced by image classification research. The CNN network has been used as a feature extractor in FER on a static level. To further improve the recognition effect of FER in the real world, the researchers introduced temporary information of dynamic FER. But the usual dynamic FER research does not consider the real-time capabilities of mobile devices.

To further solve this problem, this chapter attempts to analyze the impact of the

deep model architecture on the number of model parameters and calculations and proposes a cascade model based on the MobileNet series model and GRU. And based on this model, a real-time FER system is constructed to recognize dynamic facial expressions. And discussed the pre-processing of the whole system.

(3) Pruning LBC network for FER

In recent years, with the development and wide application of deep learning, due to higher requirements for model results, deep learning network architecture has become deeper and more complex. This is followed by exponential growth in model parameters and memory requirements. In addition to reducing the number of parameters and calculations required by changing the model architecture, methods such as model pruning and quantification have also been proposed to solve this problem.

In this chapter, we propose a basic network structure based on mobile inverted bottleneck convolutional layer and squeeze-and-excitation optimization. By changing its inverted bottleneck expand ratio, adjusting the number of LBC kernels in each LBC layer. By increasing the expansion ratio of the inverse bottleneck structure in the model, a large model with higher accuracy but a huge amount of parameters can be obtained through training. According to the SE optimization weight, we perform channel-based model pruning of the basic model and only retain the depthwise LBC convolution channel that contributes more to the result. Through the method of model pruning, only more reasonable LBC parameters are retained, so that the recognition rate of the model is higher and the model parameters are less. Using this method, we have obtained a model with good results on the image classification database and the FER database, while the number of parameters and calculations of the model is maintained at a low level.

1.3 Thesis Organization

This thesis mainly studies and analysis the research with the FER task. The whole thesis covers the basic theories, methodology, experiment, and discussions about FER,

which is organized in the rest chapters as follows:

Chapter 1: Talks about the motivation and significance of the FER task, and introduces the main research contents and organizational structure of this thesis.

Chapter 2: Introduces the background and related works of the FER task. This chapter first introduces some existing psychological emotion models and publicly FER database. Then the research status and progress of FER technology in recent years is reviewed.

Chapter 3: Introduces the multi-stream neural network that integrates deep learning and handcrafted feature extraction. By studying the influence of manually extracted LBP-TOP spatiotemporal features on the deep learning network, we try to merge the handcrafted features with the deep model.

Chapter 4: After studying the parameters and calculations of the deep learning model architecture, a lightweight cascaded network for processing dynamic facial expression sequences is proposed. The effects of different pretreatments on the results and speed of FER under different conditions are discussed.

Chapter 5: Further use of local binary convolution (LBC) and efficient network architecture, combined with the technical characteristics of model pruning and SE optimization, proposes a pruning method based on depthwise LBC and SE optimization model architecture.

Chapter 6: Concludes the whole thesis and discusses future works.

Chapter 2

Related Works

2.1 Introduction of Databases

The research object of this thesis is mainly for these two types of facial expression data. Laboratory-controlled database collects data from the photographed persons in a fixed light/angle. The person being photographed usually makes expressions corresponding to the basic emotions either actively or passively. On the other hand, in-the-wild conditions database is more to collect specific facial images or video clips from the Internet or movies and manually labeled by professional personnel. Such databases have relatively more complicated conditions such as illumination and posture, and the background is more cluttered, making it more difficult to recognize. This thesis has conducted FER research of these two kinds of datasets, which are listed below:

CK+ [12]: The Extended CohnKanade (CK+) database is the most extensively used laboratory-controlled database for evaluating FER systems. CK+ contains 593 video sequences from 123 subjects. The sequences vary in duration from 10 to 60 frames and show a shift from a neutral facial expression to the peak expression. Among these videos, 327 sequences from 118 subjects are labeled with seven basic expression labels (anger, contempt, disgust, fear, happiness, sadness, and surprise) based on the Facial Action Coding System (FACS). Because CK+ does not provide specified training, validation, and test sets, the algorithms evaluated on this database are not uniform. For static-based methods, the most common data selection method is to

extract the last one to three frames with peak formation and the first frame (neutral face) of each sequence. Then, the subjects are divided into n groups for person-independent n-fold cross-validation experiments, were commonly selected values of n are 5, 8, and 10.

Oulu-CASIA [13]: The Oulu-CASIA database includes 2,880 image sequences collected from 80 subjects labeled with six basic emotion labels: anger, disgust, fear, happiness, sadness, and surprise. Each of the videos is captured with one of two imaging systems, i.e., near-infrared (NIR) or visible light (VIS), under three different illumination conditions. Similar to CK+, the first frame is neutral and the last frame has the peak expression. Typically, only the last three peak frames and the first frame (neutral face) from the 480 videos collected by the VIS System under normal indoor illumination are employed for 10-fold cross-validation experiments.

Ren-VFEdb [14]: REN-VFEdb is a self-collection database composed of six basic facial expression classes (anger, disgust, fear, happiness, sadness, and surprise) for 40 people in the 20's to 40's (26 men and 14 women). The database consists of three sets of data per facial expression (with different expression intensity) per person, for a total of 18 data per person. The duration of the video from non-expression to a certain expression was five to seven seconds, incorporating the expression face as the onset to peak formation of a certain expression. In each video, we intercepted the same 10-frame-sequence to represent the expression change process from the non-expression frame. Among all 720 sequences, we excluded a part of the sequence that was not recorded correctly, and then deleted the sequence where the facial expression was always expressionless and there was no visible change. There are also some sequences where the landmarks are marked incorrectly due to the occlusion of the hat and hair. Finally, 407 sequences were selected for the experiment.

FER2013 [15]: The FER2013 database was introduced during the ICML 2013 Challenges in Representation Learning. FER2013 is a large-scale and unconstrained database collected automatically by the Google image search API. All images have been registered and resized to 48*48 pixels after rejecting wrongfully labeled frames and adjusting the cropped region. FER2013 contains 28,709 training images, 3,589 validation images, and 3,589 test images with seven expression labels (anger, disgust, fear, happiness, sadness, surprise, and neutral).

SFEW [16]: The Static Facial Expressions in the Wild (SFEW) was created by selecting static frames from the AFEW database by computing key frames based on facial point clustering. The most commonly used version, SFEW 2.0, was the benchmarking data for the SReco sub-challenge in EmotiW 2015 [17]. SFEW 2.0 has been divided into three sets: Train (958 samples), Val (436 samples), and Test (372 samples). Each of the images is assigned to one of seven expression categories, i.e., anger, disgust, fear, neutral, happiness, sadness, and surprise. The expression labels of the training and validation sets are publicly available, whereas those of the testing set are held back by the challenge organizer.

AFEW [18]: The Acted Facial Expressions in the Wild (AFEW) database was first established and introduced in [19] and has served as an evaluation platform for the annual Emotion Recognition In The Wild Challenge (EmotiW) since 2013. AFEW contains video clips collected from different movies with spontaneous expressions, various head poses, occlusions, and illuminations. AFEW is a temporal and multimodal database that provides vastly different environmental conditions in both audio and video. Samples are labeled with seven expressions: anger, disgust, fear, happiness, sadness, surprise, and neutral. The annotation of expressions has been continuously updated, and reality TV show data have been continuously added. The AFEW 7.0 in EmotiW 2017 [20] is independently divided into three data partitions in terms of subject and movie/TV source: Train (773 samples), Val (383 samples), and Test (653 samples), which ensures data in the three sets belong to mutually exclusive movies and actors.

Table 1.1 provides an overview of these datasets, including the main reference, number of subjects, number of image or video samples, collection environment, expression distribution, and additional information. Figure1.1 showing some sample images of facial expressions in a different database.

In addition to the above FER database, we also used the image classification database in the experiment of the Pruning LBC network:

CIFAR10 [21] is an image classification dataset containing a training set of 50K and a testing set of 10K 32 × 32 color images across the following 10 classes: airplanes, automobiles, birds, cats, deers, dogs, frogs, horses, ships, and trucks. Basically, all image classification models will be verified in the CIFAR10 database.

Table 2.1 An overview of the facial expression datasets.

P = Posed; S = Spontaneous; Condit. = Collection condition; Elicit. = Elicitation method.

Database	Samples	Subject	Condit.	Elicit.	Expression distribution
CK+ [12]	593 image sequences	123	Lab	P & S	6 basic expressions plus contempt and neutral
Oulu-CASIA [13]	2,880 image sequences	80	Lab	P	6 basic expressions
Ren-VFEdb [14]	407 image sequences	40	Lab	P	6 basic expressions
FER2013 [15]	35,887 images	N/A	Web	P & S	6 basic expressions plus neutral
SFEW [16]	1,766 images	N/A	Movie	P & S	6 basic expressions plus neutral
AFEW [18]	1809 videos	N/A	Movie	P & S	6 basic expressions plus neutral

Figure 2.1 Sample images of facial expressions in different databases.

From left to right are CK+(Neutral), CK+(Peak), Fer2013, and SFEW (Cropped image from AFEW).

2.2 FER Based on Handcrafted Features

Early FER research mostly focused on hand-crafted features, which can be broadly divided into two groups: static FER (frame-based) and dynamic FER (sequence-based). The static FER only focused on static facial features extracted by hand-crafted features from the selected expressional (peak) frames of expression sequences. The effectiveness of various types of appearance features (e.g., local binary patterns (LBPs) [22], local phase quantization (LPQ) [23], etc.) has been investigated. Also, to reduce the effect of feature confusion between facial identity and facial expression, some methods utilized either appearance difference [24] and geometrical difference [25], or even both differences [25] between the non-expression (neutral) face and the expressional face. However, the methods based on the static frame could be difficult to recognize in real-world situations, such as the situations when the expression intensity is small (i.e., subtle and micro-expressions) or the non-expression face frame of the subject is unavailable beforehand [27].

On the other hand, research works have utilized temporal relations in facial expression sequences, cause the dynamic process of expression change to give more information to been recognized. By utilizing a probabilistic graphical model (e.g., Hidden Markov Model [28], conditional random field [29,30], etc.) to process the static facial features extracted by each frame of expression sequence. Also, the spatiotemporal feature has been utilized to capture expression dynamics. By considering the space-time transition T, the spatiotemporal features, such as local binary patterns from three orthogonal planes (LBP-TOP) [31] and local phase quantization from three orthogonal planes (LPQ-TOP) [32,33], were obtained by extending LBP and LPQ into the three orthogonal planes (XY plane: appearance, XT planes: horizontal motion, and YT plane: vertical motion). Those methods have shown an improvement with the recognition performance compared to the static features; they also have the following disadvantages. The static feature also has a feature confusion problem by encoding facial identity. Moreover, those methods

adopt temporal normalization to obtain expression sequences into a specific number of frames, which could result in the loss of temporal scale information and could be negatively affected by the different characteristics of the facial expression.

2.3 Deep FER Networks

Deep learning has recently become a hot research topic and has achieved state-of-the-art performance for a variety of applications. Deep neural networks represented by CNN have been widely used in various computer vision applications, including FER. At the beginning of the 21st century, several studies in the FER literature [34,35] found that the CNN is robust to face location changes and scale variations and behaves better than the multilayer perceptron (MLP) in the case of previously unseen face pose variations. [36] employed the CNN to address the problems of subject independence as well as translation, rotation, and scale invariance in the recognition of facial expressions.

In [37], a CNN was utilized to learn a spatial feature representation. By only considering the spatial feature, the method did not utilize the facial expression dynamic feature, which could be limited by subtle and micro-expressions. In [38,39], an RNN was utilized with conventional CNN to encode dynamics in the sequence for the classification of facial expression. The methods showed that the architectures of CNN with RNN can improve recognition performances compared to conventional CNN. In [30], a Conditional Random Fields (CRF) module was replaced by the RNN, to recognize the sequence of the static features encoded by CNN for classification of facial expression. In [41], a 3D CNN was utilized to learn spatiotemporal features from facial action parts. In [42], a simplified 3D CNN was also utilized to learn the spatiotemporal appearance features of the sequence, and also utilized a temporal geometric feature to jointly learned to reduce the effect of the identity on the learned spatiotemporal appearance features. In [43], both a 3D CNN was utilized to learn the spatiotemporal appearance features and an RNN to encode the sequence of facial expressions. 3D CNN able to extract the dynamics of expression, but also need more computation with the 3D convolution. And a large number of parameters (weights and

biases) in 3D convolution layers because the learning processes harder.

As mentioned before, direct training of deep networks on relatively small facial expression datasets is prone to overfitting. To mitigate this problem, many studies used additional task-oriented data to pre-train their self-built networks from scratch or fine-tuned on well-known pre-trained models (e.g., AlexNet [44], VGG [45], VGG-face [46], and GoogleNet [47]). Further multi-stage fine-tuning strategy [48] can achieve better performance: after the first-stage fine-tuning using FER2013 on pre-trained models, a second-stage fine-tuning based on the training part of the target dataset is employed to refine the models to adapt to a more specific dataset.

Although handcrafted features such as LBP-TOP have gradually declined with the development of deep learning, there are still many studies looking for a method of fusion of manual features and deep learning. A novel mapped LBP feature [49] was proposed for illumination-invariant FER. Scale-invariant feature transform (SIFT) [50]) features that are robust against image scaling and rotation are employed [51] for multiview FER tasks. In [52], LBP-TOP is applied to part of the convolutional layer of CNN to extract spatiotemporal features. In [53], the LBP-TOP feature histogram extracted directly from the image sequence, and the features extracted by CNN are fused to identify emotions. At the same time, the ideas of LBP and LBP-TOP have also been transformed into non-trainable convolution kernels and have been applied to CNN [54,55]. These studies that incorporate deep learning and LBP-TOP features all prove the effectiveness.

2.4 Model Pruning

With the good results of deep neural networks in various fields, the computing resources required by the model also increase. Model size, memory footprint, number of calculation operations (FLOPs), and power consumption are the main aspects that hinder the use of deep neural networks in certain resource-constrained environments. Those large models may not be stored and cannot be run in real-time on embedded systems. To solve this problem, many methods have been proposed, such as low-rank

approximation of weights [56,57], weight quantization [58,59], knowledge distillation [60], and model pruning, where model pruning has attracted much attention due to its competitive performance and compatibility.

In the work of the early 1990s, when the weight is set to zero, the second-order Taylor approximation method with increased network loss function is used for pruning. In Optimal Brain Damage [61], the saliency for each parameter was computed using a diagonal Hessian approximation, and the low saliency parameters were pruned from the network, and the network is retrained. In Optimal Brain Surgeon [62], the saliency of each parameter was calculated using the inverse Hessian matrix, the low saliency weight was pruned, and all other weights in the network are updated with second-order information. More recently, paper [63] proposed to trim the weight of the network to a small extent, and further integrate this technology into the deep compression pipeline [64] to obtain a highly compressed model. Besides, many researchers have proposed various algorithms to iteratively remove redundant neurons, use Variational Dropout to trim excess weights [65], and learn sparse networks through regularization of the L0 paradigm based on random gates [66]. But, one disadvantage of these unstructured pruning methods is that the resulting weight matrix is sparse, and if there is no dedicated hardware/library, it cannot cause compression and acceleration [64].

In contrast, the structured pruning method is pruning at the channel or even at the level. Since the original convolution structure is still retained, no dedicated hardware/library is required to achieve these benefits. Among the structured pruning methods, channel pruning [67,68,69] is the most popular method because it operates at the most granular level while still being suitable for conventional deep learning frameworks. There are three classic ideas for the channel pruning algorithm. The first is based on the importance factor [69], that is, to evaluate the effectiveness of a channel, and to constrain some channels to make the model structure itself sparse, so that pruning is based on this. The second is to use reconstruction errors to guide pruning [67,68], indirectly measuring the impact of a channel on the output. The third is to measure the sensitivity of the channel based on the change of the optimization target. However, the paper [70] pointed out that, for structured pruning, after obtaining the compression model through the pruning algorithm, it is better to initialize and train the compression model randomly instead of using the weights of

the large network for fine-tuning. For the final compressed small model, the network architecture obtained by the pruning algorithm is more important than the "important" weight obtained by the pruning. The model through the pruning algorithm can provide design guidance for designing an effective network architecture. For most convolutional neural networks, fine-tuned convolution kernels from large networks are more likely to cause the model to fall into overfitting.

Chapter 3

Multi-Stream Network for Dynamic FER

3.1 Introduction

The previous chapter describes that it is generally believed that FER's research began when psychologists decomposed facial expressions into combinations of facial muscle activations, which are called facial action units (AU) in the form of the Facial Action Coding System (FACS). Further, the researchers tried to use other hand-made features to identify static and dynamic facial images. In recent work, deep learning methods have been used in FER, and the recognition accuracy has been improved to an acceptable level in a simple environment. Although research work has been done on FER, achieving accurate and effective FER is still a challenging problem in the real situation, where facial expression could be influenced widely in expression intensity, such as the micro-expression and exaggerated expressions [21,22,23], and the imaging conditions are also complex because of the illumination and angle conditions. Therefore, a robust FER method that expresses intensity variation is of paramount importance for practical FER.

Due to the convenience of data processing and the availability of relevant training and testing materials, a large number of existing research perform FER tasks based on static images without considering temporal information. But FER can benefit from the temporal correlation of consecutive frames in the sequence. How to extend spatial

information to spatiotemporal information is the key to dynamic FER. Under the existing deep learning framework, common methods include CNN-RNN cascaded network, 3D CNN, multi-stream network ensemble, and other methods. Among them, RNN and its variants (such as LSTM, GRU) and 3D CNN are the basic networks for learning spatiotemporal features. However, the performance of these networks is almost unsatisfactory. RNN cannot capture powerful convolution features. The 3D filter in 3D CNN is applied to very short video clips, ignoring long-distance dynamics. In addition, training such a large network is a computational problem, especially for dynamic FER with insufficient video data. The cascaded network is proposed to first extract the discriminative representation of facial expression images, and then input these features into the sequence network to enhance temporal information coding. However, this model introduces additional parameters to capture sequence information. The feature learning network (for example, CNN) and temporal information coding network (for example, LSTM) in the current work are not jointly trained, which may lead to unsatisfactory parameter settings.

In our research, we are inspired by action recognition research. Like FER, action recognition also studies the problem of sequence recognition. In paper [71], Two-stream-CNN architectures were utilized to extract both spatial and temporal features in two different streams, and then fuse the two-stream recognition probability distributions. The method is shown multi-stream architecture able to fuse the different features, which are extracted from focused information. Multi-stream network integration is used to train multiple networks for spatial and temporal information, and then merge the network output in the final stage. Optical flow and facial landmark trajectories can be used as temporal representations to coordinate spatial representations. One of the disadvantages of this framework is the pre-calculation and storage consumption of optical flow or landmark trajectory vectors.

Based on these studies, in this chapter, we propose a multi-stream architecture based on cascaded deep convolutional neural networks (CNN) and gated recurrent units (GRU). And use LBP-TOP as the input of another stream to replace the computational cost of optical flow. In this architecture, spatial features and temporal features were cascaded, and handcrafted spatiotemporal features are fused to extract the features of dynamic FER. The contributions of this chapter are summarized as follows:

(1) The first stream utilized CNN to extract the static feature and Gated Recurrent

Unit (GRU) to extract the sequence feature. About the static feature, we selected the apex expression frame of each sequence to train a CNN model. To avoid the overfitting problem since the number of samples of each database was small, we utilized the data augmentation and fine-tuning method. We utilized DTAN [42, 72] liked CNN as the static feature extractor.

(2) After extracting the static features from the static expression frames, we utilized the GRU network to extract the dynamic features of the sequence. After the static CNN to extract the static feature, we not only utilized extracted features as the input of the dynamic network but also utilized softmax outputs as another input. Our CNN-GRU cascade model utilized two GRU layers to process both two sets of inputs separately, and then extract the dynamic features of the expression sequence as two CNN-GRU streams.

(3) The second stream part aimed to extract the spatiotemporal feature of the expression sequence. As a supplement stream of the mainstream, we utilized the LBP-TOP feature to extract spatiotemporal information, which was utilized in the FER field commonly. Two of the most important properties of the LBP-TOP are its robustness to monotonic gray-scale changes caused, and its computational simplicity. With these properties, we utilized two CNN-GRU networks to process the XT & YT orthogonal planes of the LBP-TOP feature. We utilized Bi-GRU in our model. Each frame of the LBP-TOP feature map was extracted by the CNN part and combined into a feature sequence and processed by the GRU part. Both XT and YT as two streams of the overall model.

(4) After extracted the RGB feature and the LBP-TOP feature by four streams of the proposed method, we not only calculated the accuracy level of each stream but also calculated the accuracy of averaged outputs. The supplement stream helped the mainstream to recognize expressions with similar representation and improve recognition performances compared to each stream. The whole method considered computational speed, which makes it possible to analyze images in challenging real-time settings.

In this work, extensive and comprehensive experiments have been conducted on a self-collection database Ren-VFEdb [14] for deliberate expressions with different expression intensity, and also conducted on the two benchmark databases CK+

database [12] and Oulu-CASIA dataset [13] is comparable with state-of-the-art methods. Experimental results on the self-collection database got an accuracy of 79.0%, prove the effectiveness of the proposed method on variable expression intensity. Experimental results on benchmark datasets got the recognition accuracy with 98.7% on CK+ database and 88.7% on Oulu-CASIA database, showing that the expression class separability of the multi-stream method representation was improved in terms of the recognition rate with the state-of-the-art methods.

The remainder of this chapter is organized as follows. Section 3.2 detailed the proposed FER method with multi-stream architecture. Section 3.3 presented experimental results to verify the effectiveness of the proposed method. Section 3.4 discussed the details of differences with related research. Finally, Section 3.5 summarizes our works and outlines the direction of future work.

3.2 FER With Multi-Stream-CNN

3.2.1 Overview of the Proposed Method

Figure 3.1 shows an overview of the proposed multi-stream network method for FER. The proposed architecture consists of two main parts: 1) static feature extracted by a plain CNN model, then expanded into sequence level with GRU, and 2) spatiotemporal feature extracted by LBP-TOP to created XT & YT orthogonal planes feature, then processed by two CNN-GRU networks. In the first part, static image data were selected from the database and utilized the most expressional frame as the expression static image. From the static image, we trained DTAN [42,72] liked CNN to recognize the static expression image. In this part, [42] showing the FRE task could achieve good results on such a plain model. We created a plain CNN with 3 convolutions and pooling blocks, to extract the probability distribution of each expression image. Back to the sequence of expression, representative frames of different expression-states able to

extract different features. In our research, we tried to make CNN learn the unique representation of different expressions, we utilized GRU to find the most representative feature, as the probability of each frame in the whole expression process in our work. As we were not extracted the expression changing process feature, the second stream part was proposed to supplement the first stream part. In this part, two CNN-GRU networks were trained to extract the feature of the XT & YT orthogonal planes created by LBP-TOP. LBP-TOP is a very common spatiotemporal feature, especially in FER. By calculating the adjacent frames change the LBP-TOP able to create the spatiotemporal feature and expressed it as a grayscale image, which also able to process by CNN. After extract the two parts' features, we fused the four streams' features to recognize the expression. The details of each step of the proposed method are described in the following subsections.

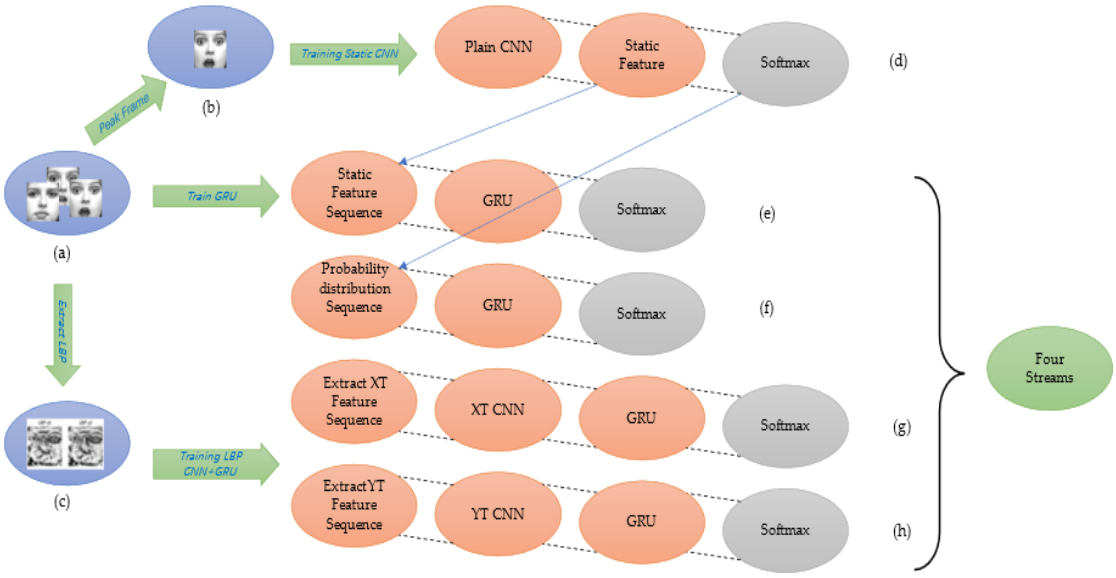


Figure 3.1 Overview of the proposed multi-stream model with CNN-GRU stream and LBP-TOP stream.

(a) is the input as expression sequence. (b) is the peak frame extracted from sequence. (c) was the LBP-TOP feature extracted from sequence. (d) is the static network trained by only peak expression frame. (e) is the CNN-GRU and utilized static feature sequence as input. (f) is the CNN-GRU but utilized probability distribution sequence as input. (g) is spatio-temporal network utilizing the LBP-TOP XT feature. (h) is spatio-temporal network utilizing the LBP-TOP YT feature. (e),(f),(g) and (h) are the four streams of the proposed model architecture.

3.2.2 Pretreatment and Data Augmentation

In this work, the facial expression sequence was pretreated to adapt our method. In this work, we register facial images in each of the databases utilizing research standard techniques. We utilized the Active Shape Model (ASM) [73] to extract facial landmarks. ASM utilizes the shape of the object to fit the facial image training set to extract 68 landmark points, especially the three landmark points of eyes and nose. We utilized these three points to register faces to an average face in an affine transformation, to make the preprocessed image maintain a relatively stable state when utilized as input to the model. Finally, a fixed rectangle around the average face is considered as the face region.

Once the face image has been registered, since [42,72] suggested that decreasing image resolution does not greatly impact the accuracy, the images were resized into 64×64 pixels for the next process.

For the static CNN part, we selected the expressional facial frame of the sequence as the dataset, which often is the last frame in most databases. After static CNN trained, we utilized transfer learning to expanded into a sequence. In this part, we utilized GRU and feed the model with the sequence of the whole expression changed from non-expression frame to apex expression-statue. The sequence also normalized to a specific number of frames, which utilized 10 frames in our work.

For the spatiotemporal stream, each sequence is calculated with the LBP-TOP feature, and each frame of the LBP-TOP feature is treated as a channel of the input sample. The sequence of each expression was considered as an image with multi-channels and process with the proposed spatiotemporal stream CNN.

To avoid the over-fitting problem, we utilized the data augmentation method. Although in the pretreatment part, we corrected the head rotation data in the data by the positions of the eyes and nose, this is the result of an affine transformation. Therefore, we artificially added some rotated and horizontally flipped face images through data augmentation, thereby increasing the generalization ability of our model. Data augmentation is only applied to the training data of the model, and it is also a common method to make up for the insufficient number of samples in the database. For dynamic image sequences, we applied the same rotation and horizontal flip parameters to the

same sequence.

3.2.3 RGB Sequence Stream Part

Since the AlexNet [44] got great success on the ILSVRC test, CNN also got great development. GoogLeNet [47] and ResNet [74] bring the CNN architecture going deeper. InceptionResNet [75] combined those two architectures got a better result. In [37] have shown the Inception architecture also has remarkable results in FER. In [42], a plain 3D CNN model was proved decreasing image resolution also worked on sequence-based FRE task, and in [72] showed that plain CNN architecture also worked

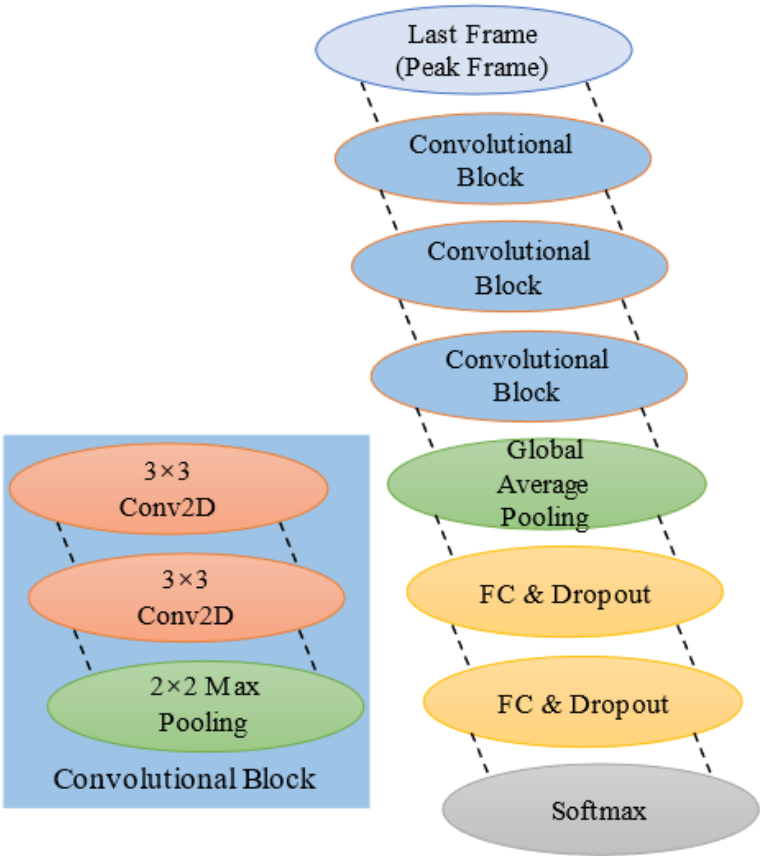


Figure 3.2 The spatial feature extraction model utilized peak expression frame as input.

Left part is a Convolutional Block with 2 convolutional layer and 1 max pooling layer. Right part is the network an architecture with 3 Convolutional Block. Output of the spatial feature extraction model is a probability distribution of emotional classes.

on the frame-based task. The CNN model in our work has only 6 convolutional layers, the plain model also avoids training a large number of parameters.

In our proposed architecture, the spatial part extracts the feature on the static frame of facial expression. Normally, the expression sequence in our database changed from non-expression frame to expression frame and showing the changing process. From the sequence of facial expressions, we selected the expression frame for our spatial stream CNN to extract, which is usually the final framework of the expression sequence.

Then we build the spatial feature extract model, the architecture has shown in Figure 3.2. After the convolutions and pooling blocks, we utilized the Global Average Pooling (GAP) layer and 2 fully connected layers. We also utilized ReLU as an activation function for each convolution layer and dropout was also applied after the fully connected layers to prevent overfitting. A softmax layer was utilized to calculate the probability distribution of static frame expression. The convolution kernel size is 3×3 in all layers. The filter number of 6 layers were 64, 64, 128, 128, 256, 256. The output dimension of 2 fully connected layers is 128, the dropout rate is set as 0.7.

After the spatial feature extract model is trained by the selected static expression frames, we utilize this trained model and transferred it into the sequence feature extract network. We not only utilized the feature extracted by pre-trained static CNN but also utilized probability distribution of the static classification as another feature

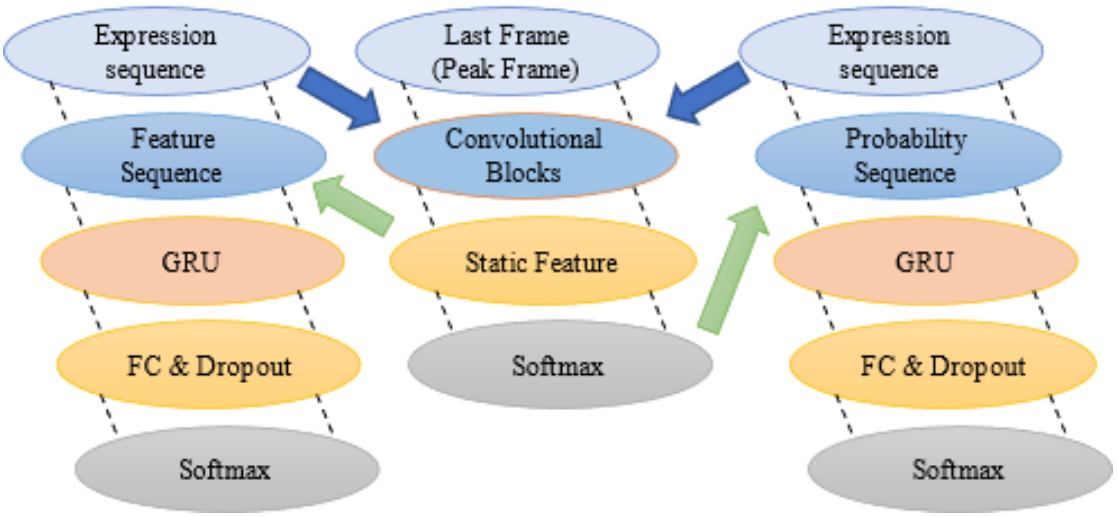


Figure 3.3 The CNN-GRU temporal feature extraction model.

Utilize trained static model to process each frame of sequence with fine-tuning. The left side is the static feature sequence network. The right side is the probability distribution sequence network.

representation, the sequences of the whole expression process were extracted to create two feature sequences. Two GRU layers with 128 hidden states were added to analyze those two feature sequences, next for 2 fully connected layers with 128 nodes and a dropout layer with a 0.7 rate for each part. We added softmax at the end of each stream to pre-train the end-to-end model. The architecture has shown in Figure 3.3. In this part, we attempt to extract the feature able to represent the facial expression and the change process mostly.

3.2.4 LBP-TOP Stream Part

The LBP-TOP stream part aims to extract auxiliary change information from the expression sequence. First, the traditional LBP-TOP method is utilized to extract spatiotemporal features from the expression image sequence and establish spatiotemporal feature maps. Then utilized CNN to process the extracted spatiotemporal features. Instead of utilizing RNN or 3D-CNN architecture network, we extracted the spatiotemporal feature from sequence and utilized plain CNN architecture which similar to a static stream to extract the temporal information. The optical flow

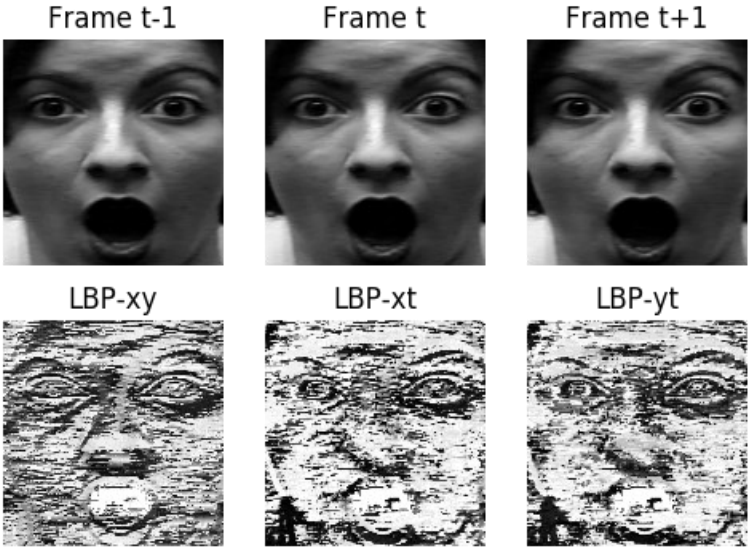


Figure 3.4. An example of LBP-TOP feature.

Treat the LBP-xt and LBP-yt as the input of LBP-TOP network.

was extracted as a spatiotemporal feature in action recognition research. But in our method, we selected LBP-TOP as the spatiotemporal feature.

By consider the space-time transition T with LBP, LBP-TOP could extract features on three orthogonal planes as XY, XT, YT. Set t as the length of expression sequence, LBP-TOP gets the XT and YT feature with the length of t-2. For each frame in sequence LBP-TOP needs one frame before and one frame after this frame, a total of three frames to extract the temporal information. We set t=10 for each sequence by downsampling or upsampling from the original dataset sequence. And extracted t-2=8 frames XT and YT LBP-TOP feature frames. The XY orthogonal plane feature extracted by LBP-TOP (also the normal LBP feature) shown the static feature of each frame. Since we extract the static feature by spatial stream part from the original expression frame, only XT, YT orthogonal planes LBP-TOP feature is needed and as the input of our temporal stream. Figure 3.4 shows the LBP-TOP feature image. With a frameset (t-1, t, t+1), LBP-TOP outputs three feature images. The XT and YT showing the X-direction and Y-direction variations with time information.

Each orthogonal plane was trained by a plain architecture CNN-GRU. The

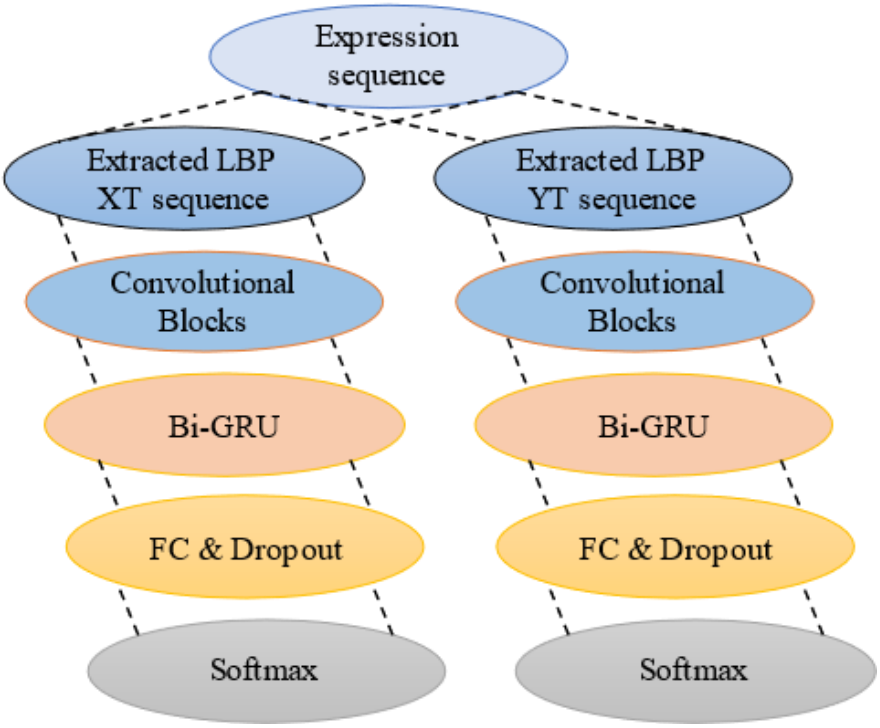


Figure 3.5 The LBP-TOP CNN network utilized LBP XT & YT feature extracted from whole expression sequence. We utilized 2 networks to process the LBP-TOP feature maps. Each network utilized 3 convolutional Blocks with same architecture in static network in CNN

architecture has shown in Figure 3.5. Both XT &YT have 3 convolutional blocks same as static stream CNN in the CNN part. After CNN performs feature extraction on each feature map, the entire 8-frame LBP-TOP feature maps constitute one feature sequence and utilized a layer of Bi-GRU with 128 hidden states to analyze it. The merge mode of the Bi-GRU is the summation. Next for 1 fully connected layer with 128 nodes and a dropout layer with a 0.7 rate. The input of each network with all frame's XT or YT orthogonal plane feature to create feature maps. We pre-trained two end-to-end nets for each orthogonal plane by calculated softmax as output.

In this part, we also trained the parameters in the network from a random start.

3.2.5 Fusion Model & Visualization of CNN Filters.

Through the CNN-GRU streams, we got two 128-dimensional feature vectors from feature sequence and probability distribution sequence as F1 & F2. From the LBP-TOP CNN stream, we got two 128-dimensional feature vectors from XT & YT as F3 & F4. We concatenate the F1~F4 of the four streams got a 512-dimensional feature vector as

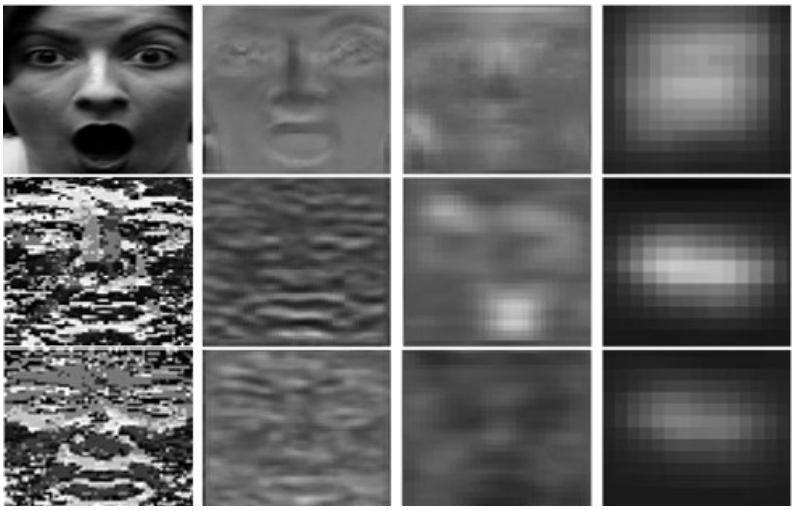


Figure 3.6. Visualization of two streams filter.

The first line was the plain CNN stream. From left to right were input image, and after 2, 4, 6 convolutional layers feature maps. The second line & third line were XT & YT LBP-TOP CNN stream. From left to right were LBP-TOP feature maps, and after 2, 4, 6 convolutional layers feature maps.

the fusion feature vector, and finally utilized softmax for classification.

In Figure 3.6, both plain CNN stream and LBP-TOP CNN-GRU stream filters' visualization were shown. In this part, we extract the features of specific layers of the trained network and make it more similar to the input image through several iterations. Figure 3.6 shows the visualization of the first convolution filter of different convolution layers. The second line & third line showing XT and YT feature maps of LBP-TOP CNN-GRU, and the visualization of different convolutional layers. Two streams of CNN have performed feature extraction on different face parts separately.

3.3 Experiments

3.3.1 Experimental Setup

This work mainly aimed with FER on sequence level, databases of independent, irrelevant still images containing only facial expressions cannot be examined by our method. By considering the openness and the comparability, we conducted experiments on a self-collection database Ren-VFEdb [14] for deliberate expressions with different expression intensity, and evaluate our proposed method on two benchmark databases CK+ database [12], and Oulu-CASIA dataset [13] for deliberate expressions.

All the experiments in this chapter were conducted in a subject-independent manner, such that the subjects in the training set were excluded from the test set.

The input of each database was pretreatment with resized the face frames to 64×64 pixels and then utilized data augmentation. The last frame of each sequence was extracted to train the static CNN network. And sequence was first utilized downsampling or upsampling into 10 frames to do the experiments in a temporal network. In each sequence, we also extracted the LBP-TOP feature into XT & YT parts. Each frame was resized into 64×64 in 10 frames and extracted LBP-TOP feature with

62×62 in 8 frames for the LBP-TOP network to do the experiment.

3.3.2 Experimental Result of Self-collection Database.

Table 3.1 shows the experimental results of our self-collection database Ren-VFEdb. The static network means the result of a static network trained by the last frame of each sequence. The CNN-GRU means the result of the expanded static network with GRU of sequence recognition. SF means Static Feature, and PD means Probability Distribution. The LBP-TOP shows results of both the XT & YT CNN-GRU networks. The final Multi-Stream Network shows the fusion result of the above four streams.

Table 3.1 Experimental results of REN-VFEdb.

Accuracy (%)	REN-VFEdb
Static Plain CNN	72.4
(a) CNN + GRU with SF sequence	**73.1**
(b) CNN + GRU with PD sequence	**72.7**
(c) CNN + GRU with LBP-TOP XT	**69.1**
(d) CNN + GRU with LBP-TOP YT	**72.1**
Fusion Network with (a) & (b)	74.6
LBP-TOP Fusion Network with (c) & (d)	72.9
Fusion Multi-Stream Network	**79.0**

In our database, the temporal network result was a little bit higher than the static network. The model accuracy of the static feature sequence got 73.1%, and the model accuracy of the probability distribution sequence got 72.7%. The fusion model of these two streams obtained a relatively high accuracy of 75.6%. The accuracy of the LBP-TOP network got 69.1% & 72.1% in XT and YT models. The YT model shows higher accuracy than the XT model. The fusion model of LBP-TOP with both XT and YT obtained higher accuracy of 72.9%. And the final fusion multi-stream network got much higher results than each stream network, got 79.0%.

Figure.3.7(a) shows the confusion matrices of each proposed network in

REN-VFEdb.

3.3.3 Experimental Result Of Benchmark Database

Table 3.2 showing the experimental results of the CK+ and Oulu-CASIA database.

Table 3.2 Experimental results of CK+ and OULU-CASIA database

Accuracy (%)	CK+	Oulu-CASIA
Static Plain CNN	97.1	83.4
(a) CNN + GRU with SF sequence	97.3	83.7
(b) CNN + GRU with PD sequence	96.8	81.6
(c) CNN + GRU with LBP-TOP XT	92.9	76.3
(d) CNN + GRU with LBP-TOP YT	94.4	78.1
Fusion Network with (a) & (b)	97.5	84.1
LBP-TOP Fusion Network with (c) & (d)	94.4	80.0
Fusion Multi-Stream Network	98.7	88.7

In the CK+ database, the static network got an accuracy of 97.1%, And after utilized the GRU of frames' static feature and probability distribution, the accuracy was 97.3% and 96.8%. The fusion model of those two streams got 97.5%.

And in the Oulu-CASIA database, the result of the static network got an accuracy of 83.4%, and the CNN-GRU network of static feature and probability distribution accuracy got 83.7% & 81.6%. The fusion model of those two streams got 84.1%.

Both the probability distribution part was lower than the static network. Considering that the static model and the dynamic model have different inputs, the amount of data that the model needs to process is not an order of magnitude at all. And the accuracy of the static model has a good level, the probability distribution of its output results is more inclined to the form of a one-hot vector. CNN-GRU with probability distribution sequence got lower accuracy also reasonable.

The LBP-TOP networks with XT & YT were not as good as the above networks. In CK+ database, XT &YT network got accuracy of 92.9% & 94.4%. The fusion model with both XT and YT got an accuracy of 94.4%. Compared with the YT stream, the results did not improve.

In Oulu-CASIA database, XT &YT network got accuracy of 76.3% & 78.1%. The results also showing that the YT feature got higher recognition accuracy. The fusion model with both XT and YT got an accuracy of 80.0%. Compared with the respective XT and YT streams, the result has improved.

Figure 3.7(b) shows the confusion matrices of each proposed network in the CK+ database. The accuracy rate of the final fusion stream network that fused the above four streams and reached 98.7%. In Table 3.3 we show the comparison of our results with the state-of-the-art in CK+ database.

Table 3.3 Comparison with state-of-the-art in the CK+ database.

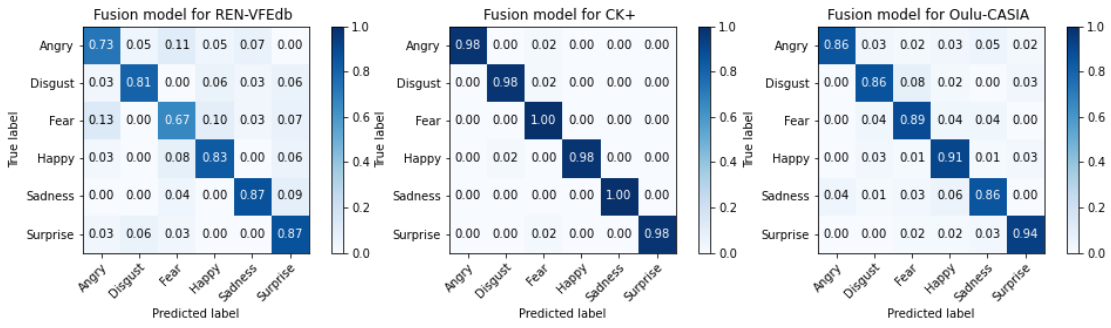
CK+ Database	**Accuracy (%)**
Sparsere presentation [27]	84.4
SPSD [77]	88.5
AUDN [78]	92.1
Inception [37]	93.2
3DCNN [41]	92.4
lp normMKL multiclass-SVM [79]	93.6
Inception-ResNet+CRF [30]	93.0
3DInception-ResNet+landmarks [43]	93.2
LBP-TOP+VLBP [31]	95.2
Two-Stream-CNN [40]	93.7
DTAGN [42]	97.3
Fusion Multi-Stream Network	**98.7**

Figure 3.7 (c) shows the confusion matrices of each proposed network in the Oulu-CASIA database. The accuracy rate of the final fusion stream network that fused the above four streams and reached 88.7%. Table 3.4 shows the comparison of our results with state-of-the-art in the Oulu-CASIA database.

Table 3.4 Comparison with state-of-the-art in the Oulu-CASIA database.

OULU-CAISA Database	**Accuracy (%)**
LBP-TOP [31]	68.1
HOG 3D [80]	70.6
AdaLBP [13]	73.5
Atlases [81]	75.5
STM-ExpLet [82]	74.6
DTAGN [42]	81.4
Peak-Piloted [83]	84.6
Fusion Multi-Stream Network	88.7

The results in the two standard databases show that the results of the four-stream fusion model are better than the results of any single stream model. In particular, the accuracy in two models of the LBP-TOP streams is lower than the CNN-GRU models of the RGB image sequence. However, the result of the fusion model has increased substantially. This also further proves the effectiveness of our fusion model method.



(a) REN-VFEdb. (b) CK+. (c) Oulu-CASIA.

Figure 3.7 The confusion matrices of proposed Fusion Multi-Stream Network in (a) REN-VFEdb, (b) CK+, and (c) Oulu-CASIA database.

3.4 Discussion

3.4.1 RNN vs. MaxPooling

In this work, we utilized GRU in the first stream to extract the dynamic feature of the sequence. But in our experiments, we also tried to utilize the MaxPooling layer to find out the impatience of each frame's static feature.

In [76], Pooling layers were proposed to process the feature vectors extracted by CNN to do the action recognition. In the early stages, we utilized fine-tuning the well-known Inception-ResNet-v2 architecture as the base static CNN architecture. Compared to the proposed plain CNN architecture, Inception-ResNet-v2 has many more parameters to train. Pooling layers were helpful in avoided to train a large number of parameters in RNN or 3DCNN networks. But with further experiments, the plain CNN architecture reduced parameters and get higher accuracy performance than Inception-ResNet-v2. Since the pooling layer has no trainable parameters, we replace it with GRU to improve accuracy performance. Although the pooling layer has similar accuracy performance and lower parameter numbers, we finally choose GRU for better performance. Table 3.5 showing the accuracy of Inception-ResNet-v2+MaxPooling comparing with plain CNN-GRU in the CK+ database.

Table 3.5 Comparison of GRU and Pooling on the BENCHMARK database.

Accuracy (%)	**CK+**	**Oulu-CASIA**
Inception-ResNet-v2	91.4	83.3
Inception-ResNet-v2+GRU with PD	90.0	76.5
Inception-ResNet-v2+GRU with SF	91.9	82.4
Inception-ResNet-v2+Pooling with SF	93.9	79.2
Plain CNN	97.1	83.4
Plain CNN+GRU with PD	96.8	81.6
Plain CNN+GRU with SF	97.3	83.7
Plain CNN+Pooling with SF	95.5	80.6

3.4.2 LBP-TOP vs. Optical Flow

In our proposed spatiotemporal stream, we utilized the LBP-TOP feature to replace optical flow, which is utilized in the action recognition field normally.

Optical flow is the pattern of apparent motion of objects between two consecutive frames caused by the movement of the object and showing the movement of points between frames by a 2D vector field. Optical flow works on several assumptions: the pixel intensities of an object do not change between consecutive frames, and neighboring pixels have similar motions. Different from action recognition the body part or object's motion works like rigid motion in consecutive frames, facial part (eyes, mouth) changing between consecutive frames with tiny, non-rigid, and complex. Also, the illumination changes are a problem of FER in the real environment. The illumination changes have a great influence on the optical flow.

However, same with LBP, one of the most important properties of the LBP-TOP is its robustness to monotonic gray-scale changes caused. LBP-TOP compares pixels with neighboring pixels, and this relation was less affected by illumination variations. Another important property is its computational simplicity, which makes it possible to analyze images in challenging real-time settings. LBP-TOP has been proved to be very effective on FER.

3.5 Summary

In this chapter, we presented a Multi-Stream Network for the task of facial expression recognition in sequence. Proposed Multi-Stream architecture which extends from well-known Two-Stream-CNN for action recognition field. By utilizing plain CNN architecture module to extract static expression frame and extended into sequence level with GRU by both the static feature and the probability distribution of frames, and extract LBP-TOP feature with CNN-GRU networks as spatial-temporal streams part.

Compared with the RGB sequence CNN-GRU streams' accuracy performance, the LBP-TOP streams got lower accuracy performance. However, even if the accuracy of

the LBP-TOP stream networks is not as high as the first two streams, the multi-stream architecture shows that the LBP-TOP network could still provide information between frames and improve the overall accuracy.

We also evaluated our method on a self-collection database Ren-VFEdb, which got 79.0% recognition accuracy. We also evaluated our proposed method on two benchmark databases CK+ and Oulu- CASIA, and comparison the experiments result with many of the state-of-the-art methods. On the CK+ database, the result got 98.7% recognition accuracy and 88.7% accuracy on the Oulu-CASIA database. The results showing that our proposed method was effective.

In future work, we will try other architectures of DNN (3DCNN, etc.) to improve the supplemental stream part. And we will do more tests on other facial expression databases, especially in the wild environment. We also consider utilizing other stream features to improve the recognition performance.

Chapter 4

CNN-GRU for Real-time FER System

4.1 Introduction

The AlexNet [44] had convolution layers with kernel sizes 11, 5, and 3 followed by fully connected layers. With the success of AlexNet, researchers tried many experiments to improve their performance. By fixed the kernel size to 3 and increased the depth of the network, VGG [45] has gained widespread acceptance due to its simplicity. Recently there has been lots of progress in algorithmic architecture exploration included Inception [47,85], ResNet [74], where the computational cost was also much lower than VGG or its higher-performing successors. Also, network architectures like MobileNet [86], ShuffleNet [87] keep the computational cost at a low level to build lightweight networks. By utilized Inverted residuals block with Depthwise Separable Convolutions and Linear Bottlenecks, MobileNetV2 [88] got higher performance than MobileNetV1. And MobileNetV3 [84] improved recognition performance again and accelerated calculation speed by adding Squeeze-and-Excite (SE) block [89] and hswish [84]. These lightweight model architectures make it possible to perform real-time recognition tasks on mobile devices.

The motivation of this chapter was to create a lightweight network deal with dynamic facial expression sequences. Based on this idea, we did not choose 3D-CNN to avoid large numbers of training parameters but chose the cascade networks of CNN-RNN. In the CNN part, we selected MobileNetV3[84] architecture through experiments. And reduce the problem of overfitting through transfer learning. At the same time, as the

input of CNN, we kept the size of the image small. Although the smaller image size cannot reduce the training parameters of the model, it can reduce the amount of calculation required. In the RNN part, we did not choose the feature vector sequence extracted by CNN but chose the probability distribution of each expression class after the softmax layer was processed. We combine the probability distribution of each frame in videos into a sequence, as the input of the GRU. Compared with the feature vector, the probability distribution has a smaller length, and the GRU utilized probability distribution as input also has fewer parameters.

In this chapter, we propose a real-time dynamic FER system based on MobileNetV3 and GRU. For the real-time capability of the system, we chose the MobileNetV3 architecture to minimize the system parameters. The proposed system is extended to a frame-to-sequence approach by exploiting temporal information with GRU. Finally, we demonstrate the performance improvement by using the proposed system on some public datasets.

In the whole system, we intercepted a certain length of frames as the basic unit of expression recognition. In this frame sequence, we first calculate the probability distribution of each frame through CNN in real-time. When the last frame of the sequence is reached, the probability distribution of each frame is combined into a sequence, and the expressions of this sequence are calculated through GRU. In the expression database, we also cut the video based on this standard and obtained 6 basic expressions and data under natural conditions. The experimental results prove the effectiveness of the proposed system.

The remainder of this chapter is organized as follows. Section 4.2 detailed the proposed facial expression recognition system from processing to all networks utilized in our work. Section 4.3 presented the whole framework of our system. Section 4.5 presented experimental results to verify the effectiveness of the proposed method. Section 6 discussed the details of differences with related research. Finally, the summary was drawn in section 4.6.

4.2 FER System with Lightweight Model

4.2.1 Motivation

The motivation of our work is very simple: build a model constructed by CNN-RNN with as little computation as possible to process facial expression sequences. To achieve this goal, we have investigated various existing CNN and RNN structures. One way to get an idea of the speed of models is to simply count how many computations it does, we can calculate its multiply accumulate operations (MACC) to approximate the calculation of the model. In most image processing research, the static image is treated as a three-dimensional matrix of size H × W × C, where H is the height of the image, W the width, and C the number of channels usually 3 in RGB images. The input and output to convolutional layers are also three-dimensional feature maps of size H × W × C. For a 2D convolutional layer with kernel size K, the number of MACCs is:

$$K \times K \times Cin \times Hout \times Wout \times Cout \qquad \text{(Equation 4.1)}$$

The *Hout* & *Wout* are the height and width of the output feature map, *Cin* & *Cout* are the input and output channels of a convolutional layer. We are ignoring the bias and the activation function here. Through Equation 4.1, we can see that the calculation amount of the CNN model depends on the input image size and the setting of the convolutional layer. We reviewed the existing FER methods and proposed a variety of ways to reduce the calculation of the proposed FER system.

4.2.2 Input Image Size

In papers [42] & [72], the input image size was set as 64 × 64 and 96 × 96, where the input size was normally 224 × 224 in most existing image recognition & FER research. Both papers proposed a compact CNN to recognize the small size of an input image and

got a good recognition accuracy in experiments. In this work, we choose to reduce the input image size to reduce the calculation of the proposed method.

4.2.3 The CNN Model

Many CNN structures have optimized the structure of the convolutional layer, such as the Inception series network. Depthwise-separable convolution was utilized to build lightweight models as MobileNets. A depthwise-separable convolution is a factorization of a regular convolution into two smaller operations: a depthwise convolution and a pointwise convolution. Together they take up a lot less memory (fewer weights) and are much faster, with the approximate performance of the regular convolutional layer doing. Compared to a regular convolution, the depthwise convolution does not combine the input channels. There is always the same number of output channels as input channels. The total number of MACCs for a depthwise convolution is:

$$K \times K \times Cin \times Hout \times Wout \qquad \text{(Equation 4.2)}$$

Compared to Equation 4.1, it does a factor of *Cout* less work, making this a lot more efficient than a regular convolutional layer.

The depthwise convolution alone will not change the channels of input & output, the following part is a pointwise convolution. The pointwise convolution is the same as a regular convolution but with a 1×1 kernel, the number of MACCs is:

$$1 \times 1 \times Cin \times Hout \times Wout \times Cout \qquad \text{(Equation 4.3)}$$

The total mass for a depthwise-separable layer by Equation 4.2 & 4.3 is:

$$Cin \times Hout \times Wout \times (K \times K + Cout) \qquad \text{(Equation 4.4)}$$

By comparing this to Equation 4.1 for a regular convolution layer, MACCs are reduced by a factor as $K \times K \times Cout / (K \times K + Cout)$, If K=3, MACCs reduced almost 9 times less costly.

Depthwise-separable layers are the main building block in MobileNetV1. MobileNetV2 utilized an “expansion block” to add more channels to the feature map, for an expansion factor, the input channels were expanded into:

$$Cexp = Cin \times expansion\_factor \quad \text{(Equation 4.5)}$$

In the entire block, first, utilize a 1×1 convolutional layer to expand the number of channels, and then utilize depthwise-separable convolution, the total MACCs as:

$$(K \times K + Cout + Cin) \times Cexp \times Hout \times Wout \quad \text{(Equation 4.6)}$$

By comparing this to Equation 4.1, MACCs reduced by a factor as:

$$K \times K \times Cout / ((K \times K + Cout + Cin) \times Expansion\_factor) \quad \text{(Equation 4.7)}$$

In normal MobileNetV2, set the K=3, Cin=64, Cout=128, Expansion_factor=6, the MACCs of an expansion block have roughly the same computational cost but worked with 6 times channels.

While optimized the parameters of each block, MobileNetV3 also added an SE block, which allows the network to automatically learn the importance of each feature channel. There is no major change in the convolutional layer. The SE block is equivalent to adding 2 FC and some elementwise multiplies calculations. The MACCs were Cout × (Cout/r) × Cout + Cout, where r is a reduced ratio of SE block. The SE block increases accuracy while appropriately increasing the number of parameters.

In this work, we choose the MobileNetV3 as the CNN architecture to extract the static facial expression feature.

4.2.4 The Frame-to-Sequence Model

An image sequence in standard facial expression databases usually begins with a neutral expression and gradually proceeds to a peak expression. There are two more common methods for processing image sequences, which are 3D-CNN [42] and CNN-RNN [90, 91].

4.2.4.1 3D CNN

The first method is to utilize 3D CNN. By extending the 2D convolution kernel to 3D, to deal with the input sequence as a Four-dimensional matrix of H × W × T × C, where adding T as the frame number of sequences. For a 3D convolutional layer with kernel

size K, the number of MACCs is:

$$K \times K \times K \times Cin \times Hout \times Wout \times Tout \times Cout \quad \text{(Equation 4.8)}$$

As a MobileNetV3 structure with 3D convolution layers, compared with the 2D version Equation 4.6, the MACCs and parameters of each expansion block will have exponential growth.

4.2.4.2 CNN-RNN

The second method is to utilize the CNN-RNN cascade model. RNN involves doing two big matrix multiplies and some elementwise multiplies. Essentially, the MACCs the same as 2 FC layers, and so the number of MACCs primarily depends on the size of the input, hidden state, and output vectors. LSTM has increased 3 gates and the MACCs are four times of simple RNN, while GRU has increased 2 gates and the MACCs are three times of simple RNN. The cascade model MACCs will be increased into a little more than T times compared with CNN.

4.2.4.3 Proposed Frame-to-Sequence Model

In this work, by considering the amount of calculation, we choose the CNN-RNN cascade model as our Frame-to-Sequence Model. Here, we use a sequence of probability distributions computed from the CNNs of frames in sequence instead of a feature vector. The probability distributions with a length of 7 (neutral + six basic expressions) while the feature vector computed from the CNNs with a length of hundreds will produce a huge difference in calculation. The architecture of our CNN-RNN model is composed of a single GRU layer with 128 hidden states and a softmax layer.

4.2.5 The Framework of The Proposed System

For this work, we proposed a FER system shown in Figure 4.1. The proposed system utilized two stages fine-tuning learning approach to improve the recognition performance. The proposed system aims to recognize the dynamic facial expression sequence, we have established an end-to-end framework for recognition from videos.

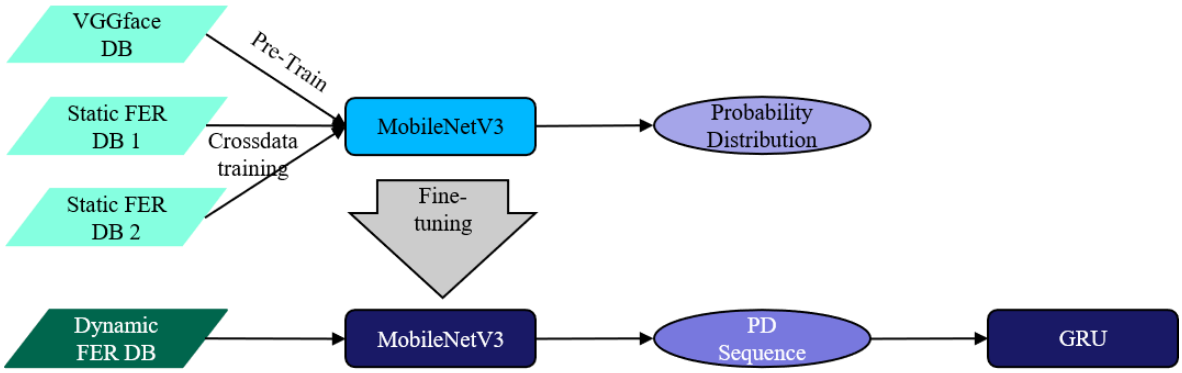


Figure 4.1 Proposed FER system, From CNN part to CNN-GRU model utilized two stages of fine-tuning.

4.2.6 Two Stages Fine-tuning

On the CNN part, compared with training the model by utilizing the facial expression database directly, we first trained the model on a large-scale face recognition database. Existing FER databases are usually on a small scale, and training on these databases directly is prone to overfitting. To avoid this situation, pre-training on a larger database is a common method. In this work, we utilized the VGGFace database [46] in the field of face recognition for pre-training. Compared to utilize the database of object

recognition, face recognition can better extract the texture features based on the human face, and it is easier to transfer to the research of FER. VGGFace contains 2.6 million images of 2622 subjects, images have large variations in pose, age, illumination, ethnicity, and profession, which helps our FER system recognize the facial expression of the wild condition.

After pro-trained the CNN model on the face recognition database, we go further training the CNN part on static facial expression databases. Training the CNN-GRU model on the dynamic database directly has a higher cost, and it is also easier to fall into overfitting.

The overall two stages fine-tuning process is shown in Figure 4.1. The CNN part was first trained on the VGGFace database and then transfer into static FER databases. The CNN-RNN model was fine-tuned from the CNN part.

4.2.7 Data Pre-processing

This work aims to recognize the wild condition facial expression, both the wild database and real situation need to pretreatment for our model. For the CK+ database, we split the whole sequence into 3 parts: neutral to peak, neutral to neutral, and peak to peak. The neutral to neutral intercepts the previous part of the sequence, the peak to peak intercepts the last part. The neutral to the peak is downsampling the entire sequence. For the AFEW database, we split videos into individual frames. Those frames are processed further by filtering the facial images. In our proposed system, we employ OpenCV DNN face detection to extract the faces and cropped face images (96 × 96 pixels). Through the experiment, we found that locating facial landmarks will take a long time for the CPU process and affect real-time performance. So, we also did experiments without face standardization. In that part, we changed the training method as Figure 4.2. For the training part, face images were aligned by locate the 68 facial landmarks and then affine transformation. Both aligned face images and unaligned images were utilized to training the model, but only unaligned face images were utilized for the test.

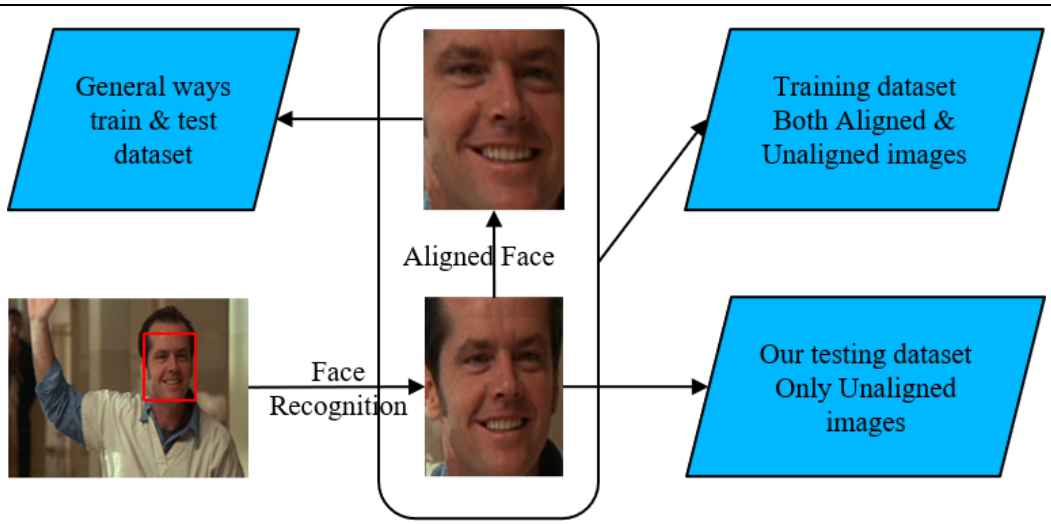


Figure 4.2 Training model utilized unaligned images.

In the real-time FER system, frames captured by the camera will also extract facial images through OpenCV DNN face detection. In this part, the CNN-GRU model will be separated into CNN and RNN parts. For each frame, the CNN part will extract the probability distribution immediately. After reaching a certain frame interval, the

probability distributions will be spliced into a sequence for recognition by the GRU part. The real-time FER system is shown in Figure 4.3.

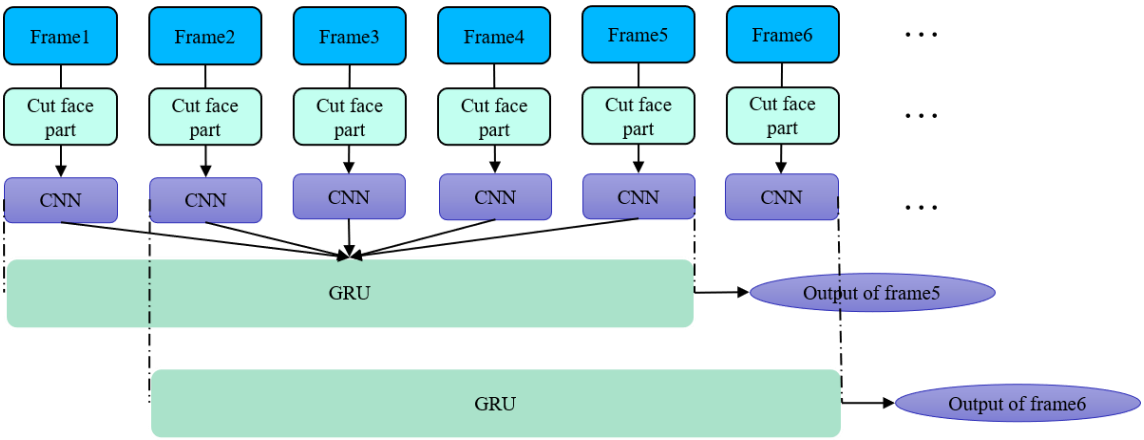


Figure 4.3 Proposed FER system on real-time recognition. For each 5 frames output the recognition result.

4.3 Experiments

4.3.1 Experimental Setup

This proposed system was verified on both static and dynamic databases to evaluate the CNN model only and whole model performance. For the static database, we utilized CK+, FER2013, and SFEW, for the dynamic database we utilized CK+ and AFEW.

The input of each database was pretreatment with resized the face frames to 96×96 pixels and then utilized data augmentation. The first and last 3 frames of each sequence in CK+ were extracted as neutral and peak expressions to train the static CNN network. And we spilled the CK+ database into 3 parts for each sequence and utilized downsampling or upsampling into 5 frames to do the experiments in the dynamic model.

4.3.2 Experimental Result of Static Database

In CNN part experiments, we add the results to choose the MobileNet V2 & V3 as the CNN part. We also compared the results between whether to pre-train with VGGFace. Besides, we add other databases as the training samples in a single database. For example, when training on the CK+ database, we added FER2013 and SFEW. But when applying CNN to the CNN-RNN model, the corresponding static and dynamic database is independent.

Table 4.1 showing the experimental results of the CK+, FER2013, and SFEW database in the CNN part. In the CK+ database, the CNN part got an accuracy of 98.3%. Compared with the direct initialization model, the pre-training on the VGG-face database makes the recognition results have higher accuracy. At the same time, pre-training on other expression databases will further improve the model effect. The

result on the FER2013 database got 73.6% and on the SFEW database got 52.2% has proven the effectiveness of pre-training.

Table 4.1 The experimental results of CK+, FER2013, and SFEW databases on static CNN.

Accuracy (%)	CK+	FER2013	SFEW
MobileNetV2	94.4	70.6	48.7
MobileNetV3	96.8	71.1	48.9
MobileNetV2 with pre-train	96.1	72.5	50.8
MobileNetV3 with pre-train	97.6	73.1	51.9
MobileNetV2 with cross databases	96.8	72.7	51.0
MobileNetV3 with cross databases	98.3	73.6	52.2

Figure 4.4 shows the confusion matrices of the CNN part network in each database. In both FER2013 & SFEW, Happy emotion has the highest recognition accuracy. Followed by Neutral and Surprised. Neutral and Sadness are easier to be confused. Anger and Fear are more likely to be mistaken as Sadness and Surprised. Those results are also more consistent with our observations on the databases.

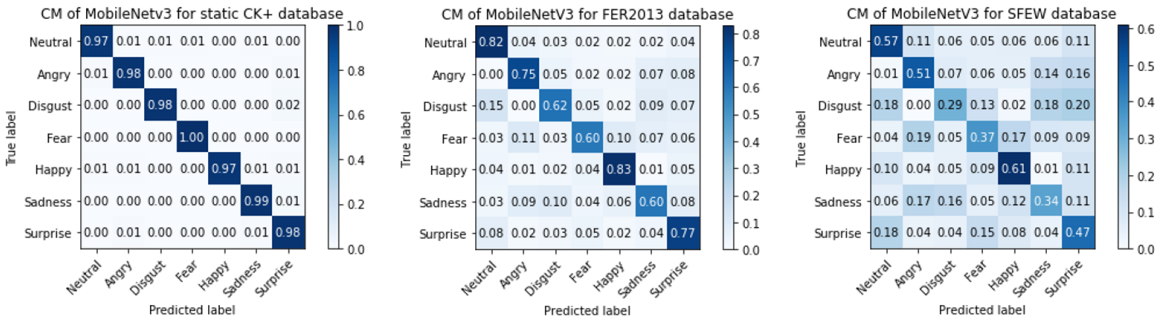


Figure 4.4 Confusion matrices of the CNN part network. From left to right are CK+, FER2013, and SFEW.

4.3.3 Experimental Result of Dynamic Database

Table 4.2 showing the experimental results of the CK+ and AFEW database in the whole model. We compared the model performance after adding a layer of LSTM and GRU after the CNN part, and GRU has better recognition accuracy. In the CK+ database, the dynamic model got an accuracy of 98.7%, and in the AFEW database, the accuracy got 48.7%.

Table 4.2 The experimental results of CK+ and AFEW databases on proposed dynamic model.

Accuracy (%)	CK+	AFEW
MobileNetV3 + LSTM	98.1	48.5
MobileNetV3 + GRU	98.7	48.8

Figure 4.5 shows the confusion matrices of the proposed dynamic network in each database. The static and dynamic results on the CK+ databases are very close, and both have high recognition accuracy. Compared with the results of the SFEW database, the recognition results of AFEW are roughly the same, but the overall results have declined.

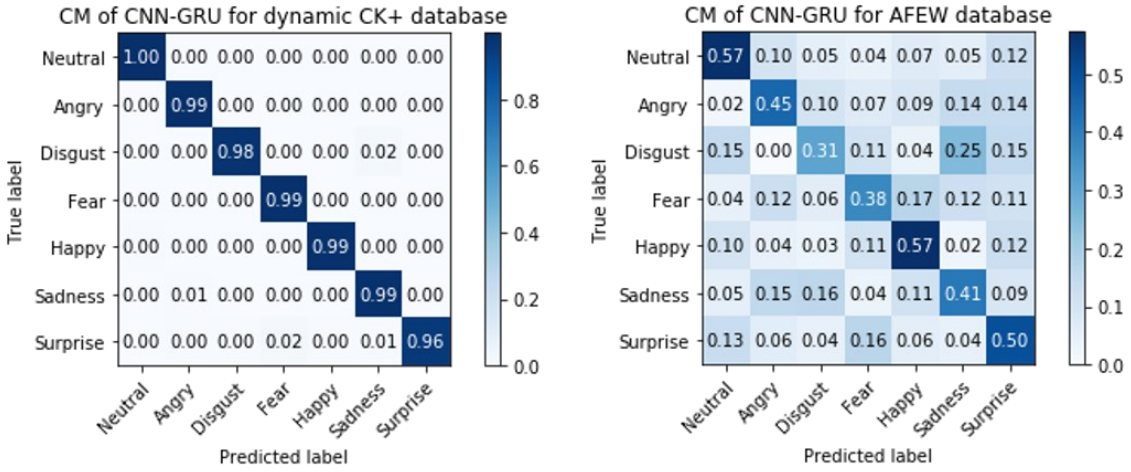


Figure 4.5 Confusion matrices of the CNN-GRU network. From left to right are CK+, and AFEW.

4.3.4 Experimental Result with Unaligned Face Images

We also conducted experiments on unaligned face images. The training set of each data set will be doubled because both aligned and unaligned face images were put in. The test set only utilized unaligned face images. Table 4.3 showing the experimental results of the CK+, and AFEW database in the proposed whole model by testing the unaligned image sequence. In the CK+ database, the accuracy of 98.0%, and in the AFEW database, the accuracy got 45.5%. Compared with aligning face data, the results on the CK+ database have not changed much because the CK+ database is lab-controlled, and the face in the database is facing the camera. But the recognition accuracy on the AFEW database dropped by 3.3%. It may be because the face part in the AFEW database has large changes in orientation and face angle and whether the faces are aligned or not has a relatively greater impact.

Table 4.3 The experimental results of CK+ and AFEW databases on the proposed dynamic model with an unaligned face image sequence.

Accuracy (%)	**CK+**	**AFEW**
Aligned face data	98.7	48.8
Unaligned face data	98.0	45.5

4.3.5 Comparison with State-of-the-art

Table 4.4 showing the comparison of our results with state-of-the-art in the CK+ database. Table 4.5 showing the comparison of our results with state-of-the-art in the SFEW & AFEW database. We also compared the model parameters. Compared with other models, our model has smaller parameters except for DTGAN. The recognition accuracy of the DTGAN is lower than our method on the CK+ database, and the results

of our experiments on the wild database are even more unacceptable. On the other hand, on the AFEW database, we only list these results obtained by the best single models in previous works. While our model has acceptable recognition accuracy, it has the smallest model parameters. Considering the size of the input image sequence and the optimization of the MobileNet series, our model has the least calculation. In the case of ensuring recognition accuracy, our model also has the real-time capabilities of mobile devices.

Table 4.4 Comparison with state-of-the-art in the CK+ database.

Method	Parameter Number	Data type	Accuracy (%)
AUDN [78]	-	Static	92.1
Inception-CNN [37]	23.5M	Static	93.2
3DCNN [41]	-	Dynamic	92.4
Two-Stream-CNN [40]	55.6M	Dynamic	93.7
DTAGN [42]	1.85M	Dynamic	97.3
FN2EN [92]	4.31M	Static	96.8
Frame attention networks (FAN) [93]	11.2M	Dynamic	99.7
Our MobilieNetV2	2.40M	Static	96.8
Our MobilieNetV3	3.12M	Static	98.3
Our MobilieNetV3+GRU	3.17M	Dynamic	98.7
Our MobilieNetV3+GRU with unaligned faces	3.17M	Dynamic	98.0

Table 4.5 Comparison with state-of-the art in the SFEW & AFEW database.

Method	Parameter Number	Data set	Accuracy (%)
FN2EN [92]	4.31M	SFEW	52.5
VGGFace-LSTM [94]	-	AFEW	45.4
Frame attention networks (FAN) [93]	11.2M	AFEW(Static)	51.2
Our MobilieNetV2	2.40M	SFEW	51.0
Our MobilieNetV3	3.12M	SFEW	52.2
Our MobilieNetV3+GRU	3.17M	AFEW	48.8
Our MobilieNetV3+GRU with unaligned faces	3.17M	AFEW	45.5

4.4 Summary

In this chapter, we proposed a real-time dynamic FER System based on CNN-GRU. We utilized a variety of methods to make our model have fewer calculations. In our CNN part, because of the optimization of the MobileNet series models, we have reduced the number of calculations for the CNN part. And we utilized the probability distribution of the CNN model's output as the input of the GRU part, further reducing the calculation amount of the entire model. When training the model, we utilized two fine-tuning methods. Through pre-training on the VGGFace database, the performance of the model was improved. The experiments have been conducted on multiple FER databases, and the experimental results have proved the effectiveness of our proposed model. We also conducted experiments on unaligned face data. Although the experimental results are slightly lower than that of aligned faces, the process of landmarking faces is reduced in the overall process, which improves the real-time performance of the overall FER system.

In future work, we will further examine the less computationally intensive and faster network architecture and try to integrate the process of face detection and landmarking into the network to make a faster and more accurate FER system.

Chapter 5

Pruning LBC Network for FER

5.1 Introduction

In the previous research, we focused on the dynamic FER of the deep model of the cascade architecture of CNN-RNN. In this chapter, we return to static images and keep our attention to the CNN part. In the previous research, we used plain CNN and MobileNet to extract features of facial expression images. Although its calculation speed is faster than other CNN models, its accuracy is not dominant. In this chapter, we try to analyze the specific structure of the convolutional layer and use other techniques to improve the CNN part to improve the recognition accuracy while maintaining a faster inference speed.

In recent years, with the development and wide application of deep learning, due to higher requirements for deep model results, deeper and more complex deep learning network structures have been proposed. But with it comes the exponential growth of model parameters and memory requirements, which makes it difficult to implement on various hardware platforms, such as mobile devices. To improve the calculation speed of the model, in addition to further improving the calculation speed of the hardware, many researchers also try to reduce the number of parameters required by changing the model structure. In addition, some researchers pay attention to the method of model compression to modify the trained model and compress it to minimize the computational space and time consumption of the model. Model pruning is a type of model compression. It is based on an assumption, or the current consensus, which is the

over-parameterization of deep neural networks. Over-parameterization means that we need a lot of parameters in the training phase to capture the tiny information in the data, and once the training is completed to the inference phase, we do not need so many parameters. This assumption supports that we can simplify the model before deployment.

By pruning model parameters with different granularities, model pruning methods can be divided into unstructured pruning and structured pruning. In unstructured pruning, the weight of the network is pruned at the neuron level, while the structured pruning method is pruning at the channel or even level. Compared with unstructured pruning, the structured pruning algorithm does not require additional computing library support and is more user-friendly in implementation and deployment. By scoring the filter or channel of the model, the parts with lower scores are removed, thereby reducing the model parameters. The structured pruning method is more of an optimization of the model structure. In paper [70], it is pointed out that after pruning, the more important thing is the preservation of the model structure rather than its parameters. From this point of view, the structured pruning method is also a neural architecture search (NAS) [95], but because it only involves layer dimensions, the search space is smaller.

In this context, we thought of the LBC network [54]. LBC is an application based on Local Binary Pattern (LBP) [96] features in traditional machine learning and uses a non-trainable convolution kernel with only three values of -1, 0, and 1. The convolution kernel of LBC is binarized and has advantages over traditional convolution kernels in terms of calculation speed and storage space. In the case of the same convolution kernel size, the non-trainable convolution kernel parameters reduce the overall training difficulty of the model. However, because the LBC kernel is not trainable, the model result has a great correlation with the number of kernels. Since the effect of LBC has a strong positive correlation with the number of LBC kernels, many randomly generated LBC kernels must be redundant. This assumption is in line with the theory of model pruning. Model pruning based on LBC not only needs to find the model structure after pruning, but it is more important to find the non-trainable LBC kernel parameters. In our work, we did not use the existing structured model pruning scoring method but used the weight of the data-driven SE block as the evaluation of the channel. At the same time, a depthwise convolution layer structure is introduced, so that the LBC kernel

corresponds to the SE optimization weight. Compared with the evaluation function of the existing model pruning method, the Squeeze-and-Excitation (SE) optimization [89] weight is obtained through training, and the relationship with the convolution channel is more intuitive.

In this chapter, we propose a basic network structure based on mobile inverted bottleneck [84, 88, 97] convolutional layer and squeeze-and-excitation optimization. By changing its inverted bottleneck expand ratio, adjusting the number of LBC kernels in each LBC layer. We conducted experiments on the image classification database CIFAR10 and FER database FER2013. The result proves the effectiveness of our proposed method: in the models with smaller expand ratios, its recognition accuracy still maintains the same level.

The remainder of this chapter is organized as follows. Section 5.2 detailed the proposed pruning method based on depthwise LBC network and SE optimization architecture. Section 5.3 presented experimental results to verify the effectiveness of the proposed method. Finally, Section 5.5 summarizes our works and outlines the direction of future work.

5.2 Model Pruning Based on SE Optimization Weight

In this work, we propose a mobile inverted bottleneck convolutional block based on a depthwise LBC layer and SE optimization, and score the LBC kernel according to the SE optimization weight, and get the LBC kernels that perform better on the model. By expanding the ratio in the Mobile Inverted Bottleneck, we get a large model with more LBC kernels and more parameters. According to the statistics of the output of the SE weight of each block of the model on the database, we get the score corresponding to each LBC kernel. And based on this score, the large model can be pruned.

5.2.1 Depthwise LBC Layer and LB Mobile Inverted Bottleneck Block

In this work, we use the LBC kernel to replace the general convolution kernel and perform structured pruning to avoid the above problems. In the structure of the LBCNN proposed in [54], each LBC block is composed of two parts. The first part is a sparse LBC layer with non-trainable parameters. The structure of this layer is the same as the standard convolutional layer, the convolution kernel size is set to 3×3 without backpropagation. The parameters of the LBC kernel are to first generate a set of all-zero matrices and then replace a part of 0 with a random 1 or -1 according to the Bernoulli distribution to generate a sparse LBC kernel. The second part is a standard convolution layer, the size of the convolution kernel is 1×1. In such an LBC block, it consists of a non-trainable convolutional layer and a trainable convolutional layer, so that the model can be trained normally. The 1×1 convolutional layer in the second part only provides a parameter that can be trained for the previous LBC layer.

Figure 5.1 showing the difference between the standard LBC and the LBC blocks we proposed. In this work, we used depthwise convolution to construct the LBC layer. Compared with the standard LBC layer, the depth-separable LBC not only reduces the

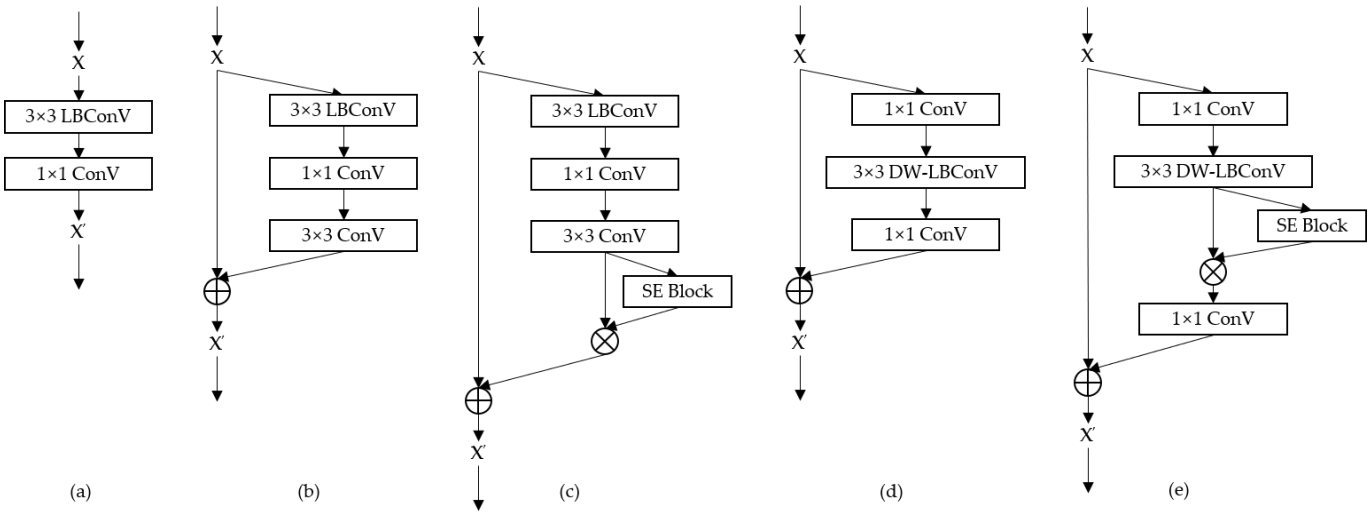


Figure 5.1 The difference between the standard LBC and the LBC blocks we proposed.

(a) Standard LBC layer; (b) Standard LBC with Residual module; (c) Standard LBC with SE-Residual module; (d) Depthwise LBC with Mobile Inverse Bottleneck module; (d) Depthwise LBC with SE-Mobile Inverse Bottleneck module.

parameters but also corresponds to each input feature map, there is only a 3×3 sparse LBC kernel for feature extraction, making the LBC kernel more Intuitively reflect the result of its feature extraction. At the same time, we introduced SE optimization to add attention weights to the feature maps extracted from the depthwise LBC layer, also to use SE optimization weights to replace the 1×1 convolution of the traditional LBC layer. In the entire mobile inverted bottleneck, the first and last two 1×1 convolutional layers are mainly used to adjust the number of input and output model channels. Even if we change the number of LBC kernels through model pruning, the input kernel output of the whole block will not change, which is convenient for the model to calculate the residual.

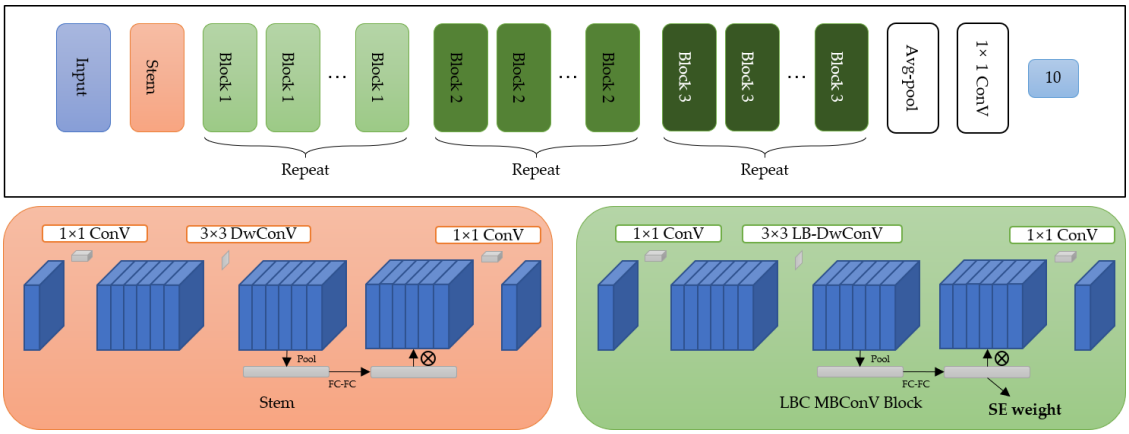


Figure 5.2 The proposed baseline LB-MBNet model framework.

5.2.2 Baseline LB-MBNet Model

By combining the LBC mobile inverted bottleneck block, we propose our baseline model. The mobile inverted bottleneck has applications in many model structures and has also been proven to be an efficient model structure. In the EfficientNet [97], the input resolution, depth, and width of the model are all quantified, and an optimization search is performed to find the best performance model structure under different parameters. In our proposed model, the network width is also quantified, but we only

change the number of LBC kernels in the depthwise LBC layer of each block. In the input of the model, we added a stem block to perform the first step of processing the input image. In this step, we used the standard convolutional layer instead of the LBC layer. In the subsequent model pruning, we do not prune the stem part, only pruning the block that uses the LBC. The quantization parameter r in the model, which is the expansion ratio of the LBC layer in each block, can be individually adjusted to adapt to different pruning scales. Figure 5.2 showing the proposed baseline model structure.

5.2.3 Model Pruning Based on SE Optimization Weight

As mentioned earlier in this chapter, we directly connect the SE block and the depth-wise LBC layer and use the SE optimization weight to express the importance of the LBC kernels. By setting the hyperparameter r to a larger value, we can get a large model with more parameters. Through experiments on the database, we compared the accuracy performance of the large model and the small model and concluded that the results of the large model have better performance. This is also the basis for our model pruning.

SE optimization is to optimize the attention weight of features in different feature channels for each sample. Corresponding to different samples, SE optimization weights are also different. But we can calculate the mean value and distribution of SE optimization weights in largescale samples, and score the characteristic channels in disguised form. Figure 5.3 showing the relationship between each LBC kernel and SE optimization weight in the proposed model. For the input *Xc*, *c* is the number of input feature channels. First, channel expansion is performed through 1×1 convolution to obtain *Xrc*. For *Xi* ($i \in 0 \ldots rc$) is the *i-th* feature map in *Xrc*, *X'i* is the output of the depthwise LBC layer. After the processing of the SE block, *Wi* ($i \in 0 \ldots rc$) is obtained.

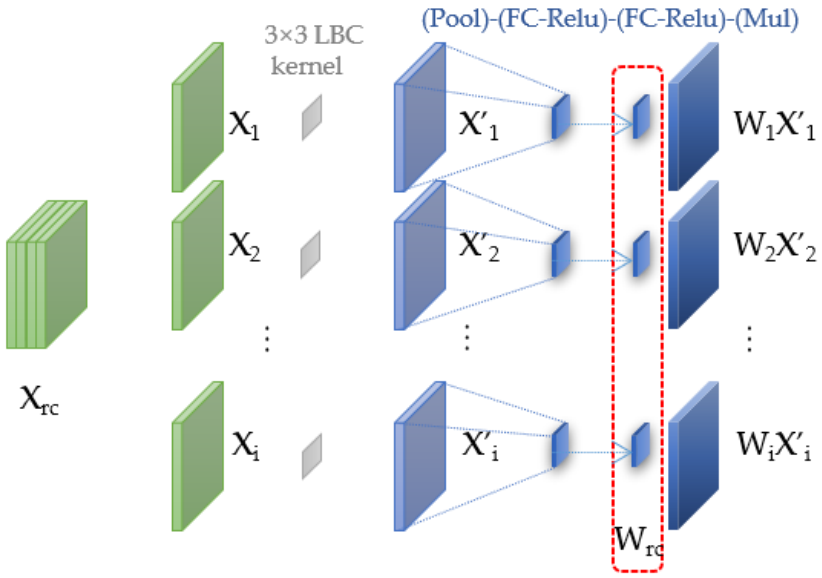


Figure 5.3 Relationship between LBC kernel and SE weight, each weight corresponds to a 3×3 LBC kernel.

For each sample, *Wrc* is the weight of this sample for each feature channel, and it is also the weight of the corresponding depthwise LBC kernel. By extracting the SE optimization weights of the overall sample of the database, we can count the mean and distribution of the SE optimization weights. Figure 5.4 showing the statistics of SE optimization weights of the training set on the CIFAR10 database. In most of the blocks of (b), The number of SE optimization weights is large, the SE optimization with higher weights only accounts for a small part of the overall channel number, and many weights are close to 0. This also means that the LBC kernel corresponding to the channels with lower weight cannot extract significant features well in the entire data set. By filtering the SE optimization weights, we can get better LBC kernel parameters, each of which is a 3×3 sparse matrix. After model pruning, the SE weight distribution of the reconstructed model (c) is more concentrated than that of (a), and the LBC kernel has similar importance.

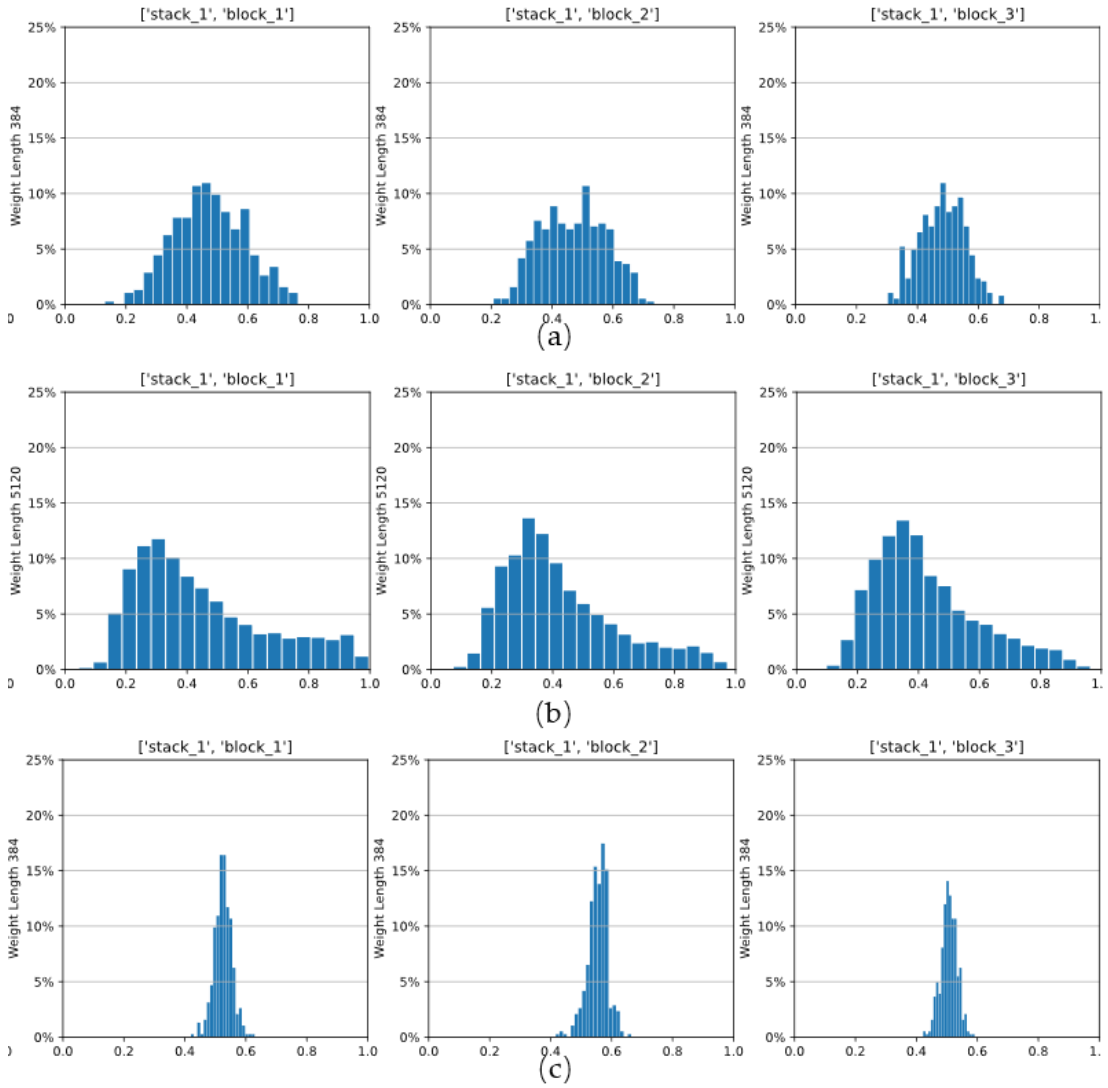


Figure 5.4 The statistics of SE optimization weights of the training set on the CIFAR10 database.

The statistics of SE optimization weights of the training set on the CIFAR10 database. (a) is the model with the expansion ratio r=6, the result obtained by random initialization training; (b) is the model with r=80, the result obtained by random initialization training; (c) a is the model with the expansion ratio r=6, but the result obtained by pruning and retraining from the model with r=80. Each subfigure represents the 3 blocks in stack 1 of each model, the x-axis is the mean value of each SE weight, and the y-axis is the proportion of the corresponding weights.

5.3 Experiments

5.3.1 Experiment Settings

For this work, we use the CIFAR10 database for image classification experiments. And FER2013 database for FER experiments.

In our test models, the model depth is fixed. In addition to the stem part, there are 4 MB blocks with LBC. Except for Block 0, each block is repeated 6 times, and the scale of the feature map is halved in the initial layer of each block. Out of the SE block, each model has 59 convolutional layers, and there are 18 sets of SE optimization weights involving LBC in the last 3 blocks to participate in model pruning. Specific parameters are shown in Table 1. Besides, we also set the number of repetitions of each block to 3 and established a model of 32 convolutional layers.

Table 5.1 Proposed Baseline LB-MBNet-59 Model structure.

The repeat numbers in () are the parameters of the LB-MBNet-32 Model.

Stack	Operator	Resolution	Channel	Repeat Num	Expansion ratio
Stem	Conv3×3	32×32	32	1	-
0	LB-MBConv, k3×3	32×32	16	1	1
1	LB-MBConv, k3×3	32×32	64	6 (3)	6
2	LB-MBConv, k3×3	16×16	128	6 (3)	6
3	LB-MBConv, k3×3	8×8	256	6 (3)	6
Top	Conv1×1 & Pooling & FC	4×4	512	1	-

For the input data, we use data augmentation to expand the training data. The operations include width shift, height shift, and horizontal flip, the shift range is 4 pixels. All models are compiled using the SGD method, the initial learning rate is set to 0.1, and decays by 0.1 at 80, 140, and 180 epochs. Each model is first trained for 200 epochs and then pruned, and then trained again for 200 epochs using the same hyperparameters.

5.3.2 Experimental Result in CIFAR10

In this chapter, we will evaluate the effect of the LBC layer in the proposed baseline model and compare its performance with that of the standard convolutional layer in the CIFAR10 database. At the same time, by changing the value of the expansion ratio r, the impact of different model scales on the recognition results is evaluated. Table 5.2 showing the model recognition result obtained while adjusting the expansion ratio r. We calculated the parameters and FLOPS of the model.

Table 5.2 Model recognition result with different expansion ratio r in CIFAR10.

Model name	Accuracy rate (%)	Trainable Params	FLOPs (10^8)	Ratio to r6 model
LB-MBNet-59-r6	92.47	7.11M	4.57	1×
LB-MBNet-59-r10	93.65	11.75M	7.57	1.7×
LB-MBNet-59-r18	93.81	21.03M	13.6	3.0×
LB-MBNet-59-r30	94.40	34.96M	22.5	4.9×
LB-MBNet-59-r60	95.05	69.78M	45.0	9.8×
LB-MBNet-59-r80	96.06	92.99M	59.9	13.1×
LB-MBNet-32-r6	92.88	3.19M	2.16	1×
LB-MBNet-32-r80	95.63	40.84M	27.7	12.8×

From Table 5.2, we can conclude that with the increase of the r, the parameter amount and number of calculation operations (FLOPs) of the model also increase almost linearly. The accuracy of the model has also increased. When r=6, the recognition accuracy of the LB-MBNet-59-r6 Model reached 92.47%, LB-MBNet-32-r6 Model reached 92.88%. When r=80, the recognition accuracy of the LB-MBNet-59-r80 Model reached 96.06%, LB-MBNet-32-r80 Model reached 95.63%, increased 3.59% and 2.75% respectively.

For the experiment of model pruning, we use the model with the best results and the maximum number of parameters as the basic model and obtain the mean value of SE optimization weight through 50K samples of the training data set. For different optimization scales, we only keep the part with the larger SE optimization weight value and pruning the corresponding layers in each block of the model. In our experiment, we separately prune the model in segmented and one-shot ways. In the segmented experiment, the r-value will gradually drop to the optimal scale we expect. In the

one-shot experiment, we use the basic model to directly modify the r-value to the optimal scale we expect. Table 5.3 showing the result of model pruning. We use the r80 model as the basic model for one-shot pruning. Here we transfer the weight of the corresponding part to the rebuilt model according to the fixed r.

Table 5.3 Model recognition result with rebuild model from r80 model in CIFAR10.

Model name	Accuracy rate (%)	Trainable Params	FLOPs (10^8)	Ratio to r6 model
LB-MBNet-59-r80	96.06	92.99M	59.9	13.1×
Re-LB-MBNet-59-r40	96.15	46.57M	30.0	6.6×
Re-LB-MBNet-59-r20	95.76	23.36M	15.0	3.3×
Re-LB-MBNet-59-r18	95.58	21.03M	13.6	3.0×
Re-LB-MBNet-59-r10	95.65	11.75M	7.57	1.7×
Re-LB-MBNet-59-r6	95.24	7.11M	4.57	1×
Re-LB-MBNet-59-r3	93.97	3.63M	2.33	0.5×
LB-MBNet-32-r80	95.63	40.84M	27.7	12.8×
Re-LB-MBNet-32-r6	94.90	3.19M	2.16	1×

From Table 5.3, we can conclude that after the model is pruned, the accuracy rate remains at a relatively high level.

In LB-MBNet-59 Model, compared with the basic r80 model, the accuracy of the r40 model reaches 96.15%, which is an improvement of 0.09% on the basic model. We believe it is because the parameters of the r80 model are relatively high, and the model has overfitting. In the case of reducing the half of parameters while maintaining a more effective LBC kernel, the results of the r40 model have risen slightly. As we continue to reduce the expansion ratio, although the accuracy of the model has dropped slightly, compared with the randomly initialized model, the results of the pruning-rebuild model are significantly better. When the expansion ratio r is reduced to 6, the model parameters and FLOPs return to the level of the basic r6 model we proposed, but the rebuilt model result has been improved by 2.77% to 95.24%. At the same time, compared with the r80 model at the beginning of pruning, the accuracy rate is only reduced by 0.82%, and the parameter amount and FLOPs are reduced by 92% of the r80 model. This result proves that our proposed method is effective.

We have further reduced the expansion ratio to r3. At this time, the Re-LB-MBNet-59-r3 Model accuracy rate has dropped significantly, reaching 93.97%,

which is a 1.27% drop compared to the r6 model. However, compared with the randomly initialized r6 model, the parameters and FLOPs of this result decreased by 50%, but the result increased by 1.50%. We think that because the expansion ratio is too small, the generalization ability of the model is not enough, which leads to overfitting at 93.97%. We believe that because the expansion rate is too small, the generalization ability of the model is insufficient, which leads to overfitting, but because the LBC kernel is non-trainable, and after the selection by model pruning, the result is better than the random initialization.

In LB-MBNet-32 Model, the accuracy of the rebuilt Re-LB-MBNet-32-r6 model reaches 94.90%. Compared with the LB-MBNet-32-r80 model, it has only decreased by 0.73%. But compared with the basic LB-MBNet-32-r6 model, it has increased by 2.02%. Compared with the LB-MBNet-59-r3 model, LB-MBNet-32-r6 has a similar level of the result but fewer parameters and FLOPs.

5.3.3 Compare with State-of-the-art in CIFAR10

We also compared the results of our proposed method with the state-of-the-art methods. At the same time, because many model pruning methods use ResNetV2-56 [98] as the basic model on the CIFAR10 database for pruning. The baseline model we proposed has a similar number of convolutional layers (59 layers). We also compared the model pruning method based on ResNetV2-56. But it should be noted that since we have applied the SE optimization part in the proposed model, the amount of model parameters has also increased.

Table 5.4 showing the comparison results between our proposed method and state-of-the-art methods.

Table 5.4 Comparison results with state-of-the-art methods on the CIFAR10 database.

1 Baseline model of model pruning.

Methods	Accuracy rate (%)	Params	Params pruned (%)	FLOPs (10^8)	FLOPs pruned (%)
ResNetV2-56[1][98]	93.01	0.597M	-	1.71	-
Li et al. [99]	93.03	0.516M	13.7	1.24	27.6
NISP [100]	92.98	0.343M	42.6	0.96	43.6
DCP-A [101]	93.02	0.177M	70.3	0.90	47.1
CP [68]	92.01	0.597M	0	0.86	50.0
AMC [102]	92.11	0.597M	0	0.86	50.0
GBN-40 [103]	93.34	0.278M	53.5	0.68	60.1
ResNetV2-164[1][98]	94.58	1.73M	-	4.97	-
L_1-Sparse[69]	94.92	1.44M	16.8	3.81	23.3
ResNetV2-1001[98]	95.08	10.48M	-	30.3	-
MobileNetV2[1][88]	91.93	2.2M	-	0.88	-
AutoSlim-MobileNetV2[104]	93.20	1.5M	31.8	0.88	0
MobileNetV3[84]	92.97	1.52M	-	0.35	-
LB-MBNet-59-r6[1]	92.35	7.11M	-	4.57	-
Re-LB-MBNet-59-r6	95.24	7.11M	0	4.57	0
Re-LB-MBNet-59-r3	93.97	3.63M	48.9	2.33	49
LB-MBNet-32-r6[1]	92.88	3.19M	-	2.16	-
Re-LB-MBNet-32-r6	94.90	3.19M	0	2.16	0

Compared with the state-of-the-art method, although the overall model we proposed is not superior in terms of parameters and FLOPs. But first, the result of our method is the best. Secondly, the LBC kernel is used in our method, which is a 50% sparse convolution kernel. These zero calculations are still counted in FLOPs before using further suitable tools. Among the Re-LB-MBNet-59-r6 models, the most similar result is the Res-NetV2-1001[98] model, but the model parameters are much less, and the FLOPs are only about 15%. Besides, in the Re-LB-MBNet-32-r6 model, the closest results are the Res-NetV2-164[98] and L1-Sparse[69] models. Although our models have more model parameters and fewer FLOPs.

Besides, the proposed method uses data-driven SE optimization weights as the evaluation of pruning, and the results are obtained based on the results of model training, which are more accurate than some manually designed evaluation indicators.

The basic model we proposed can evolve towards faster and more miniaturization. It is necessary to further optimize the depth and width of the model. However, this has already involved the research field of NAS and has not been carried out in this chapter. Experimental results have been able to prove the effectiveness of our current proposed method.

5.3.4 Experimental Result in FER2013

In this chapter, we will evaluate the effect of the LBC layer in the proposed baseline model in the FER2013 database and change the value of the expansion ratio r, the impact of different model scales on the recognition results is evaluated. The input image of the CIFAR10 database is 32×32, while the image in FER2013 is 48×48. We also change the input of the model to 48×48. Compared with the image classification database, the FER database has a smaller number of samples, and because the images are concentrated in the face part, the training of the model is easier to overfit. This is especially obvious when the expansion ratio r is increased. We set the number of layers of the baseline model to 59 and 32, which correspond to the Repeat Num in Table 5.1 as 6 and 3. Table 5.5 showing the model recognition result obtained while adjusting the expansion ratio r. We calculated the parameters and FLOPS of the model.

Table 5.5 Model recognition result with different expansion ratio r in FER2013.

Model name	Accuracy rate (%)	Trainable Params	FLOPs (10^8)	Ratio to r3 model
LB-MBNet-59-r3	69.56	3.63M	2.33	1×
LB-MBNet-59-r20	71.79	23.35M	15.0	6.4×
LB-MBNet-59-r50	70.74	58.17M	37.5	16.1×
LB-MBNet-59-r100	69.74	116.20M	74.9	32.1×
LB-MBNet-32-r3	68.78	1.67M	1.12	1×
LB-MBNet-32-r20	70.17	10.31M	6.99	6.2×
LB-MBNet-32-r50	70.72	25.58M	17.4	15.5×
LB-MBNet-32-r100	71.19	51.01M	34.6	30.6×

It can be seen from Table 5.5 that when the model parameters increase to a certain

level, the accuracy rate reaches the best, and then as the model parameters increase, the accuracy rate will decrease instead. On this basis, we further reduce the expansion ratio r for retraining, and the retrained model results are shown in Table 5.6.

Table 5.6 Model recognition result with rebuild model from r100 model in FER2013.

Model name	Accuracy rate (%)	Trainable Params	FLOPs (10^8)	Ratio to r3 model
LB-MBNet-59-r3	69.56	3.63M	2.33	1×
LB-MBNet-59-r20	71.79	23.35M	15.0	6.4×
Re-LB-MBNet-59-r3	70.46	3.63M	22.5	1×
Re-LB-MBNet-59-r20	72.15	23.35M	15.0	6.4×
LB-MBNet-32-r3	68.78	1.67M	1.12	1×
LB-MBNet-32-r100	71.19	51.01M	34.6	30.6×
Re-LB-MBNet-32-r3	70.05	1.67M	1.12	1×
Re-LB-MBNet-32-r20	72.07	10.31M	6.99	6.2×

It can be seen from Table 5.6 that after retraining, the overall accuracy is not only better than the model with r=3, but also does not cause overfitting due to too many model parameters, and the result is better than the model with r=100. In the Re-LB-MBNet-59-r20 model, the recognition accuracy reaches 72.15%, best in our overall experiments in FER2013.

5.4 Summary

Inspired by LBC and SE optimization, we propose a depthwise LBC and SE optimization model structure in this chapter. The importance of the non-trainable LBC kernel is obtained through statistical data-driven SE optimization weights. By increasing the expansion ratio of the inverse bottleneck structure in the model, a large model with higher accuracy but a huge amount of parameters can be obtained through training. According to the SE optimization weight, we perform channel-based model pruning of the basic model and only retain the depthwise LBC convolution channel that contributes more to the result. Our experimental results of image classification on the CIFAR-10 database and FER task on FER2013 prove the effectiveness of our proposed

method.

As we envisioned, the method of superimposing the non-trainable layer and the trainable layer can reduce the training parameters while allowing the model to be normally constructed into an end-to-end structure. But the initialization of the non-trainable layer is very important. All parameters in the standard convolutional network can be trained, and reasonable parameters can be found through the optimization method of the model. However, in a convolution model with non-trainable parameters, it is necessary to filter out reasonable parameters through the methods like model pruning. In this work, not only the advantages of LBC are maintained, which has a sparse and non-trainable binary convolution kernel, and at the same time, more reasonable LBC parameters are found through the method of model pruning, the model with high recognition accuracy and less of the model parameters is obtained. Compared with the traditional structured model pruning method, this work not only optimizes the model structure but also optimizes the non-trainable parameters.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This thesis focuses on facial expression recognition research and has proposed some related methods for perusing more accurate expression prediction. The concrete summaries and contributions are displayed as follows:

1) A multi-stream CNN-RNN cascade network based on the fusion of RGB expression sequences and LBP-TOP spatiotemporal features is proposed to identify dynamic FER tasks. The experimental results prove that the LBP-TOP feature can supplement the features extracted by deep learning, realize information enhancement, and make up for the neglected or missing information in the basic network.
2) A real-time FER system is proposed to recognize dynamic FER tasks. Through the study of the internal calculation of the depth model, the real-time of the overall system is realized. At the same time, other processing parts of the FER system were discussed.
3) A static image recognition model based on depthwise LBC and SE blocks is proposed, and the model pruning method is used to optimize the parameters and calculations of the model in the part of training and inference. While maintaining good recognition accuracy, the speed of reasoning is greatly improved. The experiment has obtained good results in both the image classification task and the FER task, which proves the effectiveness of the

proposed method.

6.2 Future Work

There is still much room for improvement in our work. In the dynamic FER task, we only studied the CNN-RNN cascade model. In the process of facial expression changes, appropriate handcrafted features can supplement the deep network. This thesis has conducted some works, but not completely. And 3D CNN needs to be improved in the dynamic FER task to meet the requirements of the real-time system. Future work will try to explore more effective methods to integrate handcrafted features, cascaded networks, and 3D CNN, and study better fusion and extraction methods for spatial and temporal features in dynamic FER. In our research on model architecture and lightweight models, it is also limited to the recognition of static images. For example, the LBC module also has a lot of room for expansion towards 3D convolution. Untrainable convolution kernels and normal convolutional networks can also be merged by more free methods.

Furthermore, FER research will also develop into multi-label, more complex recognition targets. Emotion recognition tasks will also further integrate multi-modal research such as facial expressions, speech, and text, to sort and recognize human emotions from a more macro perspective, can form new research directions

Bibliography

[1] Ren F, Matsumoto K. Semi-automatic creation of youth slang corpus and its application to affective computing[J]. IEEE Transactions on Affective Computing, 2015, 7(2): 176-189.

[2] Ren F, Huang Z. Automatic facial expression learning method based on humanoid robot XIN-REN[J]. IEEE Transactions on Human-Machine Systems, 2016, 46(6): 810-821.

[3] Darwin C. The expression of the emotions in man and animals[M]. University of Chicago press, 2015.

[4] Tian Y I, Kanade T, Cohn J F. Recognizing action units for facial expression analysis[J]. IEEE Transactions on pattern analysis and machine intelligence, 2001, 23(2): 97-115.

[5] Ekman P, Friesen W V. Constants across cultures in the face and emotion[J]. Journal of personality and social psychology, 1971, 17(2): 124.

[6] Ekman P. Strong evidence for universals in facial expressions: a reply to Russell's mistaken critique[J]. 1994.

[7] Matsumoto D. More evidence for the universality of a contempt expression[J]. Motivation and Emotion, 1992, 16(4): 363-368.

[8] Jack R E, Garrod O G B, Yu H, et al. Facial expressions of emotion are not culturally universal[J]. Proceedings of the National Academy of Sciences, 2012, 109(19): 7241-7244.

[9] Ekman P, Friesen W V. Facial action coding system: Investigator's guide[M]. Consulting Psychologists Press, 1978.

[10] Gunes H, Schuller B. Categorical and dimensional affect analysis in continuous input: Current trends and future directions[J]. Image and Vision Computing, 2013, 31(2): 120-136.

[11] Valstar M F, Mehu M, Jiang B, et al. Meta-analysis of the first facial expression recognition challenge[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2012, 42(4): 966-979.

[12] Lucey P, Cohn J F, Kanade T, et al. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression[C]//2010 ieee computer society conference on computer vision and pattern recognition-workshops. IEEE, 2010: 94-101.

[13] Zhao G, Huang X, Taini M, et al. Facial expression recognition from near-infrared videos[J].

Image and Vision Computing, 2011, 29(9): 607-619.

[14] Kuroda T, Jia M, Ren F, et al. The Construction of The Facial Expression Video Database[C]//2006 International Conference on Communication Technology. IEEE, 2006: 1-4.

[15] Goodfellow I J, Erhan D, Carrier P L, et al. Challenges in representation learning: A report on three machine learning contests[C]//International conference on neural information processing. Springer, Berlin, Heidelberg, 2013: 117-124.

[16] Dhall A, Goecke R, Lucey S, et al. Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark[C]//2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops). IEEE, 2011: 2106-2112.

[17] Dhall A, Ramana Murthy O V, Goecke R, et al. Video and image based emotion recognition challenges in the wild: Emotiw 2015[C]//Proceedings of the 2015 ACM on international conference on multimodal interaction. 2015: 423-426.

[18] Dhall A, Goecke R, Lucey S, et al. Collecting large, richly annotated facial-expression databases from movies[J]. IEEE multimedia, 2012, 19(03): 34-41.

[19] Dhall A, Goecke R, Lucey S, et al. Acted facial expressions in the wild database[J]. Australian National University, Canberra, Australia, Technical Report TR-CS-11, 2011, 2: 1.

[20] Dhall A, Goecke R, Ghosh S, et al. From individual to group-level emotion recognition: Emotiw 5.0[C]//Proceedings of the 19th ACM international conference on multimodal interaction. 2017: 524-528.

[21] Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images[J]. 2009.

[22] Ahonen T, Hadid A, Pietikainen M. Face description with local binary patterns: Application to face recognition[J]. IEEE transactions on pattern analysis and machine intelligence, 2006, 28(12): 2037-2041.

[23] Chan C H, Kittler J, Poh N, et al. (Multiscale) local phase quantisation histogram discriminant analysis with score normalisation for robust face recognition[C]//2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops. IEEE, 2009: 633-640.

[24] Donato G, Bartlett M S, Hager J C, et al. Classifying facial actions[J]. IEEE Transactions on pattern analysis and machine intelligence, 1999, 21(10): 974-989.

[25] Kotsia I, Pitas I. Facial expression recognition in image sequences using geometric deformation features and support vector machines[J]. IEEE transactions on image processing, 2006, 16(1): 172-187.

[26] Chen J, Chen D, Gong Y, et al. Facial expression recognition using geometric and appearance features[C]//Proceedings of the 4th international conference on internet multimedia computing and service. 2012: 29-33.

[27] Lee S H, Plataniotis K N, Ro Y M. Intra-class variation reduction using training expression images for sparse representation based facial expression recognition[J]. IEEE Transactions on Affective Computing, 2014, 5(3): 340-351.

[28] Yeasin M, Bullot B, Sharma R. Recognition of facial expressions and measurement of levels of interest from video[J]. IEEE Transactions on Multimedia, 2006, 8(3): 500-508.

[29] Jain S, Hu C, Aggarwal J K. Facial expression recognition with temporal modeling of shapes[C]//2011 IEEE international conference on computer vision workshops (ICCV workshops). IEEE, 2011: 1642-1649.

[30] Hasani B, Mahoor M H. Spatio-temporal facial expression recognition using convolutional neural networks and conditional random fields[C]//2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017). IEEE, 2017: 790-795.

[31] Zhao G, Pietikainen M. Dynamic texture recognition using local binary patterns with an application to facial expressions[J]. IEEE transactions on pattern analysis and machine intelligence, 2007, 29(6): 915-928.

[32] Jiang B, Valstar M, Martinez B, et al. A dynamic appearance descriptor approach to facial actions temporal modeling[J]. IEEE transactions on cybernetics, 2013, 44(2): 161-174.

[33] Hu M, Zheng Y, Yang C, et al. Facial expression recognition using fusion features based on center-symmetric local octonary pattern[J]. IEEE Access, 2019, 7: 29882-29890.

[34] Fasel B. Robust face analysis using convolutional neural networks[C]//Object recognition supported by user interaction for service robots. IEEE, 2002, 2: 40-43.

[35] Fasel B. Head-pose invariant facial expression recognition using convolutional neural networks[C]//Proceedings. Fourth IEEE International Conference on Multimodal Interfaces. IEEE, 2002: 529-534.

[36] Matsugu M, Mori K, Mitari Y, et al. Subject independent facial expression recognition with robust face detection using a convolutional neural network[J]. Neural Networks, 2003, 16(5-6): 555-559.

[37] Mollahosseini A, Chan D, Mahoor M H. Going deeper in facial expression recognition using deep neural networks[C]//2016 IEEE Winter conference on applications of computer vision (WACV). IEEE, 2016: 1-10.

[38] Chao L, Tao J, Yang M, et al. Long short term memory recurrent neural network based

multimodal dimensional emotion recognition[C]//Proceedings of the 5th international workshop on audio/visual emotion challenge. 2015: 65-72.

[39] Khorrami P, Le Paine T, Brady K, et al. How deep neural networks can improve emotion recognition on video data[C]//2016 IEEE international conference on image processing (ICIP). IEEE, 2016: 619-623.

[40] Feng D, Ren F. Dynamic facial expression recognition based on two-stream-cnn with lbp-top[C]//2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS). IEEE, 2018: 355-359.

[41] Liu M, Li S, Shan S, et al. Deeply learning deformable facial action parts model for dynamic expression analysis[C]//Asian conference on computer vision. Springer, Cham, 2014: 143-157.

[42] Jung H, Lee S, Yim J, et al. Joint fine-tuning in deep neural networks for facial expression recognition[C]//Proceedings of the IEEE international conference on computer vision. 2015: 2983-2991.

[43] Hasani B, Mahoor M H. Facial expression recognition using enhanced deep 3D convolutional neural networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2017: 30-40.

[44] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Advances in neural information processing systems, 2012, 25: 1097-1105.

[45] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.

[46] Parkhi O M, Vedaldi A, Zisserman A. Deep face recognition[J]. 2015.

[47] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 1-9.

[48] Ng H W, Nguyen V D, Vonikakis V, et al. Deep learning for emotion recognition on small datasets using transfer learning[C]//Proceedings of the 2015 ACM on international conference on multimodal interaction. 2015: 443-449.

[49] Levi G, Hassner T. Emotion recognition in the wild via convolutional neural networks and mapped binary patterns[C]//Proceedings of the 2015 ACM on international conference on multimodal interaction. 2015: 503-510.

[50] Lowe D G. Object recognition from local scale-invariant features[C]//Proceedings of the seventh IEEE international conference on computer vision. Ieee, 1999, 2: 1150-1157.

[51] Zhang T, Zheng W, Cui Z, et al. A deep neural network-driven feature learning method for

multi-view facial expression recognition[J]. IEEE Transactions on Multimedia, 2016, 18(12): 2528-2536.

[52] Asim M, Ming Z, Javed M Y. CNN based spatio-temporal feature extraction for face anti-spoofing[C]//2017 2nd International Conference on Image, Vision and Computing (ICIVC). IEEE, 2017: 234-238.

[53] Cai J, Meng Z, Khan A S, et al. Feature-level and model-level audiovisual fusion for emotion recognition in the wild[C]//2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR). IEEE, 2019: 443-448.

[54] Juefei-Xu F, Naresh Boddeti V, Savvides M. Local binary convolutional neural networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 19-28.

[55] Kumawat S, Verma M, Raman S. LBVCNN: Local binary volume convolutional neural network for facial expression recognition from image sequences[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2019: 0-0.

[56] Denton E L, Zaremba W, Bruna J, et al. Exploiting linear structure within convolutional networks for efficient evaluation[C]//Advances in neural information processing systems. 2014: 1269-1277.

[57] Lebedev V, Ganin Y, Rakhuba M, et al. Speeding-up convolutional neural networks using fine-tuned cp-decomposition[J]. arXiv preprint arXiv:1412.6553, 2014.

[58] Courbariaux M, Hubara I, Soudry D, et al. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1[J]. arXiv preprint arXiv:1602.02830, 2016.

[59] Rastegari M, Ordonez V, Redmon J, et al. Xnor-net: Imagenet classification using binary convolutional neural networks[C]//European conference on computer vision. Springer, Cham, 2016: 525-542.

[60] Romero A, Ballas N, Kahou S E, et al. Fitnets: Hints for thin deep nets[J]. arXiv preprint arXiv:1412.6550, 2014.

[61] LeCun Y, Denker J S, Solla S A. Optimal brain damage[C]//Advances in neural information processing systems. 1990: 598-605.

[62] Hassibi B, Stork D G. Second order derivatives for network pruning: Optimal brain surgeon[M]. Morgan Kaufmann, 1993.

[63] Han S, Pool J, Tran J, et al. Learning both weights and connections for efficient neural networks[J]. arXiv preprint arXiv:1506.02626, 2015.

[64] Han S, Liu X, Mao H, et al. EIE: Efficient inference engine on compressed deep neural network[J]. ACM SIGARCH Computer Architecture News, 2016, 44(3): 243-254.

[65] Molchanov D, Ashukha A, Vetrov D. Variational dropout sparsifies deep neural networks[C]//International Conference on Machine Learning. PMLR, 2017: 2498-2507.

[66] Louizos C, Welling M, Kingma D P. Learning sparse neural networks through L0 regularization[J]. arXiv preprint arXiv:1712.01312, 2017.

[67] Luo J H, Wu J, Lin W. Thinet: A filter level pruning method for deep neural network compression[C]//Proceedings of the IEEE international conference on computer vision. 2017: 5058-5066.

[68] He Y, Zhang X, Sun J. Channel pruning for accelerating very deep neural networks[C]//Proceedings of the IEEE international conference on computer vision. 2017: 1389-1397.

[69] Liu Z, Li J, Shen Z, et al. Learning efficient convolutional networks through network slimming[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2736-2744.

[70] Liu Z, Sun M, Zhou T, et al. Rethinking the value of network pruning[J]. arXiv preprint arXiv:1810.05270, 2018.

[71] Simonyan K, Zisserman A. Two-stream convolutional networks for action recognition in videos[J]. arXiv preprint arXiv:1406.2199, 2014.

[72] Kuo C M, Lai S H, Sarkis M. A compact deep learning model for robust facial expression recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2018: 2121-2129.

[73] Cootes T F, Taylor C J, Cooper D H, et al. Active shape models-their training and application[J]. Computer vision and image understanding, 1995, 61(1): 38-59.

[74] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[75] Szegedy C, Ioffe S, Vanhoucke V, et al. Inception-v4, inception-resnet and the impact of residual connections on learning[C]//Thirty-first AAAI conference on artificial intelligence. 2017.

[76] Yue-Hei Ng J, Hausknecht M, Vijayanarasimhan S, et al. Beyond short snippets: Deep networks for video classification[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 4694-4702.

[77] Taheri S, Qiu Q, Chellappa R. Structure-preserving sparse decomposition for facial expression

analysis[J]. IEEE Transactions on Image Processing, 2014, 23(8): 3590-3603.

[78] Liu M, Li S, Shan S, et al. Au-inspired deep networks for facial expression feature learning[J]. Neurocomputing, 2015, 159: 126-136.

[79] Zhang X, Mahoor M H, Mavadati S M. Facial expression recognition using lp-norm MKL multiclass-SVM[J]. Machine Vision and Applications, 2015, 26(4): 467-483.

[80] Klaser A, Marszałek M, Schmid C. A spatio-temporal descriptor based on 3d-gradients[C]//BMVC 2008-19th British Machine Vision Conference. British Machine Vision Association, 2008: 275: 1-10.

[81] Guo Y, Zhao G, Pietikäinen M. Dynamic facial expression recognition using longitudinal facial expression atlases[C]//European Conference on Computer Vision. Springer, Berlin, Heidelberg, 2012: 631-644.

[82] Liu M, Shan S, Wang R, et al. Learning expressionlets on spatio-temporal manifold for dynamic facial expression recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 1749-1756.

[83] Zhao X, Liang X, Liu L, et al. Peak-piloted deep network for facial expression recognition[C]//European conference on computer vision. Springer, Cham, 2016: 425-442.

[84] Howard A, Sandler M, Chu G, et al. Searching for mobilenetv3[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 1314-1324.

[85] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 2818-2826.

[86] Howard A G, Zhu M, Chen B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications[J]. arXiv preprint arXiv:1704.04861, 2017.

[87] Zhang X, Zhou X, Lin M, et al. Shufflenet: An extremely efficient convolutional neural network for mobile devices[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 6848-6856.

[88] Sandler M, Howard A, Zhu M, et al. Mobilenetv2: Inverted residuals and linear bottlenecks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 4510-4520.

[89] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7132-7141.

[90] Yan J, Zheng W, Cui Z, et al. Multi-cue fusion for emotion recognition in the wild[J]. Neurocomputing, 2018, 309: 27-35.

[91] Ouyang X, Kawaai S, Goh E G H, et al. Audio-visual emotion recognition using deep transfer learning and multiple temporal models[C]//Proceedings of the 19th ACM International Conference on Multimodal Interaction. 2017: 577-582.

[92] Ding H, Zhou S K, Chellappa R. Facenet2expnet: Regularizing a deep face recognition net for expression recognition[C]//2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017). IEEE, 2017: 118-126.

[93] Meng D, Peng X, Wang K, et al. Frame attention networks for facial expression recognition in videos[C]//2019 IEEE International Conference on Image Processing (ICIP). IEEE, 2019: 3866-3870.

[94] Vielzeuf V, Pateux S, Jurie F. Temporal multimodal fusion for video emotion classification in the wild[C]//Proceedings of the 19th ACM International Conference on Multimodal Interaction. 2017: 569-576.

[95] Elsken T, Metzen J H, Hutter F. Neural architecture search: A survey[J]. The Journal of Machine Learning Research, 2019, 20(1): 1997-2017.

[96] Ojala T, Pietikainen M, Harwood D. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions[C]//Proceedings of 12th international conference on pattern recognition. IEEE, 1994, 1: 582-585.

[97] Tan M, Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks[C]//International Conference on Machine Learning. PMLR, 2019: 6105-6114.

[98] He K, Zhang X, Ren S, et al. Identity mappings in deep residual networks[C]//European conference on computer vision. Springer, Cham, 2016: 630-645.

[99] Li H, Kadav A, Durdanovic I, et al. Pruning filters for efficient convnets[J]. arXiv preprint arXiv:1608.08710, 2016.

[100] Yu R, Li A, Chen C F, et al. Nisp: Pruning networks using neuron importance score propagation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 9194-9203.

[101] Zhuang Z, Tan M, Zhuang B, et al. Discrimination-aware channel pruning for deep neural networks[J]. arXiv preprint arXiv:1810.11809, 2018.

[102] He Y, Lin J, Liu Z, et al. Amc: Automl for model compression and acceleration on mobile devices[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 784-800.

[103] You Z, Yan K, Ye J, et al. Gate decorator: Global filter pruning method for accelerating deep convolutional neural networks[J]. arXiv preprint arXiv:1909.08174, 2019.

[104] Yu J, Huang T. Autoslim: Towards one-shot architecture search for channel numbers[J]. arXiv preprint arXiv:1903.11728, 2019.