A Real Time Facial Expression Recognition System Using Deep Learning

Yu Miao

Thesis submitted to the University of Ottawa
in partial Fulfillment of the requirements for the
Master of Applied Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

This thesis presents an image-based real-time facial expression recognition system that is capable of recognizing basic facial expressions of several subjects simultaneously from a webcam. Our proposed methodology combines a supervised transfer learning strategy and a joint supervision method with a new supervision signal that is crucial for facial tasks. A convolutional neural network (CNN) model, MobileNet, that contains both accuracy and speed is deployed in both offline and real-time frameworks to enable fast and accurate real-time output.

Evaluations for both offline and real-time experiments are provided in our work. The offline evaluation is carried out by first evaluating two publicly available datasets, JAFFE and CK+, and then presenting the results of the cross-dataset evaluation between these two datasets to verify the generalization ability of the proposed method. A comprehensive evaluation configuration for the CK+ dataset is given in this work, providing a baseline for a fair comparison. It reaches an accuracy of 95.24% on JAFFE dataset, and an accuracy of 96.92% on 6-class CK+ dataset which only contains the last frames of image sequences. The resulting average run-time cost for recognition in the real-time implementation is reported, which is approximately 3.57 ms/frame on an NVIDIA Quadro K4200 GPU. The results demonstrate that our proposed CNN-based framework for facial expression recognition, which does not require a massive preprocessing module, can not only achieve state-of-art accuracy on these two datasets but also perform the classification task much faster than a conventional machine learning methodology as a result of the lightweight structure of MobileNet.

Acknowledgements

First of all, I would like to express my sincerest appreciation to my supervisor, Prof. Abdulmotaleb El Saddik, for his guidance, support, patience and encouragement for my entire path of graduate study. His enthusiasm for knowledge has always inspired me considerably and invigorated me to move forward. Without his understanding and precious help, I would not have been able to complete my research. I am so honored and proud of working with such a responsible and excellent supervisor. I am very grateful for all the vast amount of benefits I received from being one of his students—not only the knowledge itself but also the positive influence of his attitude towards life.

Furthermore, I thank my parents, who have always given their unconditional support and love to me. Throughout the entire course of my research, they have been my reliable harbor when I have encountered challenges. I feel so lucky to be their daughter and receive their love.

In addition, I appreciate the experience of working in the Multimedia Communications Research Laboratory (MCRLab) and the valuable advice from my friend Yang.

Last, but not least, I would like to thank Yang Liu and all the members of MCR Lab for their help, cooperation, and understanding and for being great friends.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivations of the Problem

The vision of a digital twin as stated in [1] by Prof. El Saddik is a digital replication of a living or non-living physical entity. By bridging the physical and the virtual worlds, data are transmitted seamlessly, allowing the virtual entity to exist simultaneously with the physical entity. A digital twin facilitates the means to monitor, understand, and optimize the functions of the physical entity and provides continuous feedback to improve quality of life and wellbeing. A digital twin is hence the convergence of several technologies such

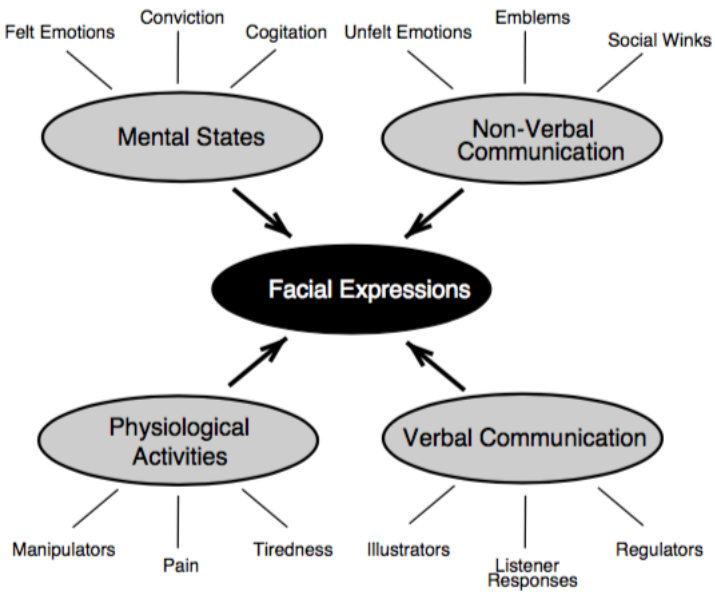


Figure 1.1: The sources of facial expressions.[4]

as AI, AR/VR and Haptics, IoT, Cybersecurity and Communication Networks.

One component of the digital twin vision is so-called affective computing, or the task of recognizing the emotion of a given subject in real time. Affect computing [2] has been an emerging trend in the last decade, as it meets the goal of intelligent human-computer interaction (HCI), which is to seek efficient communication between human and machine. According to [3], examples of channels regarded as communication intermediaries between human and computer are the auditory channel (carrying speech and vocal intonation) and visual channel (carrying facial expressions and body movements). Fasel et al. [4] noted that emotional voice, gesture, facial expressions, etc., constitute the factors of human emotions, see Figure 1.1. Facial expressions, among these factors mentioned above, play the most critical role in affect analysis.

Facial changes of facial expressions are responses to a person's internal emotional states, intentions, or social communications [5]. Darwin et al. [6] established facial expression analysis as a research field in 1872. Since then, facial expression recognition (FER) has received a great deal of attention and been an active research topic across a variety of disciplines, such as biology [6], neuroscience [7], psychology [8], and com-

puter vision. Especially in computer vision, for its impact and prominent potentiality, automatic FER has been growing in an extensive range of applications, e.g., HCI, biometric identification, surveillance and security [9], intensive care monitoring [10], aerial image analysis [11], driver state surveillance [12], and human entertainment industry and virtual reality.

The origins of the growing interest in FER in the past decade can be summarized in two main points. The first gives credit to the developing progress accomplished in related research fields such as machine learning, image processing, and human cognition [12] and tasks such as face detection, face tracking and face recognition. The second is due to the recent availability of relatively cheap computational power [4].

Recently, the approaches of deep learning have flourished due to the inexpensive computational power, and one example, called the convolutional neural network (CNN), has obtained excellent state-of-the-art results in the field of computer vision (e.g., image classification, face recognition, object detection). CNNs have been successfully applied to FER [13, 14, 15] and have shown better results than many conventional methods for its efficiency in feature learning and representations.

1.2 Challenges

1.2.1 Challenges for FER

Although FER under controlled conditions is already mature and no longer a substantial problem, it is still a challenge for computers to make accurate inferences in real-life scenarios. The challenges of achieving computational facial expression analysis can be classified into four main aspects. The first one is illustrated in Figure 1.2, in which the images differ in terms of head shapes and ethnic groups across the dataset. Different subjects in the same dataset express the same emotion to various extent. Figure 1.3 presents the second challenge: for the same dataset, images can vary substantially in illumina-

tion, head-pose, or background. The third problem is that most existing FER datasets (e.g., CK+ [16], JAFFE [17], MMI [18], RaFD [19], Oulu-CASIA [20], FERG-DB [21] etc.) contain posed facial expressions that are presented by professional actors/actresses or stylized characters, which is different from real-life scenarios, in which people generally do not express their emotions with exaggerated facial expressions (although the occurrence of spontaneous expressions has prompted another, related research field called facial micro-expression recognition [22, 23, 24]). Occlusions such as eyeglasses, beards, hats, and scarves also increase the probability of classification failures, which draws attention to this specific issue [25, 26]. High similarity between two specific classes of facial expressions, e.g., *disgust* with *angry* and *sadness* with *fear* (shown in Figure 1.4), which sometimes leads to misclassification, represents the fourth challenge.

(a) (b) (c) (d)

Figure 1.2: Example of subjects in the same class *disgust* in the CK+ dataset having different head shapes and ethnicities.

(a) (b) (c) (d)

Figure 1.3: Example of various illuminations ((a) and (b) in the CK+) and head orientations ((c) and (d) in the JAFFE) for subjects in the same dataset.

(a) Disgust (b) Angry (c) Sadness (d) Fear

Figure 1.4: Example of high similarity between facial expressions in two different classes [16, 17].

1.2.2 Challenges for CNN-based FER

In spite of CNN's strong performance, robust FER based on CNN remains a challenging unsolved problem. Deep convolutional neural networks (DCNNs) achieve high accuracy in face recognition tasks with high efficiency, requiring the size of labeled training data in the millions as emphasized in [27]. However, the total number of samples in most datasets for facial expression analysis is quite small (e.g., hundreds for [16, 17] and thousands for [18, 19, 20]), which is far from sufficient for training a CNN. Another problem, also mentioned in [28], is that many evaluations of the state-of-the-art work on the specific datasets in the literature are not applied consistently (e.g., in terms of the number of classes of emotions to be evaluated), which sometimes leads to falsely high accuracies.

1.3 Objectives

The objectives of our thesis are as follows:

- Propose a new training method for CNN that combines a supervised transfer learning and joint supervision of loss function. The employment of this method to obtain discriminative deep features achieves accuracy superior to that of current state-of-the-art methods on two publicly available datasets.

- Design and implement a system that can realize facial expression recognition from a webcam, achieving real-time operation on images while combining efficiency and

accuracy.

- Provide a comprehensive evaluation configuration for a specific dataset for better comparison with the state-of-the-art work that does not perform consistent evaluation methodology.

1.4 Thesis Statement

The convolutional neural network concept has already shown its prominence in feature learning and classification power relative to conventional methods. In our work, a new CNN model [29] is applied to accomplish this FER task, which has a much smaller size and lower computation complexity than CNN benchmarks (e.g., AlexNet [30], VGG [31]), showing excellent performance in both speed and accuracy. To address the problem of the insufficient size of those small facial expression datasets, a two-stage fine-tuning strategy is used in the CNN training process. In addition, a new supervision signal, center loss, is leveraged jointly with the softmax loss function in optimization for inter-class dispersion and intra-class compactness [32], which are essential to facial tasks. The proposed system was evaluated on the CK+ and JAFFE datasets and achieved competitive results, see Chapter 4. The average run-time cost for real-time classification through a webcam is 3.57 ms per frame, which can be readily applied to real-time applications or incorporated into mobile devices.

The main contributions of this work are as follows:

- A two-stage fine-tuning strategy is used in this work to solve the problem of the insufficient data size of facial expression datasets.

- Another center loss is added to the loss function under joint supervision with the softmax loss for enhancing the discriminatory power of the proposed system.

- Implementation of a real-time FER system that leverages a new CNN model is

designed and achieves a smaller run-time cost than previous implementations.

- Comprehensive evaluation configurations for a specific dataset are carried out to provide a baseline for valid comparisons with other related work.

1.5 Thesis Outline

The remainder of the thesis is organized as follows:

- Chapter 2 elaborates the background and related work of facial expression analysis from conventional methods to novel state-of-the-art deep learning related methods.
- Chapter 3 presents the methodology used in the proposed system and the detailed training process of CNN.
- Chapter 4 introduces the details of the datasets used in this work.
- Chapter 5 provides different experiments on two datasets, showing the effectiveness of the proposed methodology and results. Evaluations are performed using a variety of configurations adopted from other work for better comparison.
- Chapter 6 discusses the issues appearing during the process and the attained results.
- Chapter 7 summarizes the merits of this work, notes the limitations, and discusses planned future work.

Chapter 2

Background and Related Work

Regarded as one of the significant applications of facial expression recognition, HCI has often emphasized the necessity for algorithms capable of endowing computers with the ability to interact with humans in a more natural way, much closer to the way that humans interact with each other [33, 34]. Facial expression, as one of the critical communication routes, can be used as an effective tool for emotion detection, thus facilitating HCI. As mentioned by Mehrabian [35], considering the effect of the spoken message as a whole; the facial expressions of the speaker contributes 55 percent, while the vocal part (e.g., vocal intonation) and the verbal part (i.e., spoken words) contribute only 38 percent

and 7 percent, respectively. This condition also emphasizes that the facial expression, as the most significant part of nonverbal communication, is the primary modality used to convey emotions.

After giving a brief introduction of the facial expression recognition task in Section 2.1, the strategies for facial expression analysis in the literature are elaborated in Section 2.2. After presenting the characteristics of convolutional neural networks in Section 2.3, Section 2.4 reviews selected CNN-based related work involving facial expression recognition.

2.1 Facial Expression Recognition Task

According to Lopes et al. [28], the two main branches of facial expression recognition systems are as follows: those addressing static images [36, 37] and those addressing dynamic image sequences [38, 39]. Systems that work with static images consider only one still image at a time (frame-by-frame) and do not use temporal information. In contrast, those involving dynamic image sequences encoding a range of frames within a temporal window as an individual concentrate more on analyzing temporal variation [40]. This work adopts the frame-based scheme.

Automatic FER systems generally receive the two kinds of expected input (still images or a sequence of frames) and output one of the seven basic universal emotions (i.e., angry, disgust, fear, happiness, sadness, surprise and neutral) that were classified by Ekman [41] in 1975 in a cross-cultural study on the existence of *"universal categories of emotional expressions"*. In 1978, Ekman and Friesen developed the Facial Action Coding System (FACS) to describe observable facial muscle movements as action units (AU) [42]. This system taxonomizes these basic emotions by decomposing each facial expression into core AU and has been widely applied to FER tasks, providing a powerful tool for feature extraction.

2.2 Facial Expression Analysis

As illustrated by Jain and Li [5], the automatic facial expression analysis typically involves three steps: face acquisition, facial feature extraction and representation, and facial expression recognition (classification).

2.2.1 Face acquisition

Face acquisition refers to detecting the face region in a frame (face detection). One of the most widely used face detectors was proposed by Viola and Jones [43] and is termed the VJ detector.

Instead of working only with image intensities (which consumes substantial computational power), Papageorgiou et al. [44] developed a framework based on Haar wavelet representation in 1998. Later, in 2001, Viola and Jones further developed this idea by proposing the Haar-like features that represent the changes of texture or edges of special facial regions and can be operated much faster than pixels in systems. They used three types of Haar-like features, as shown in Figure 2.1: the two-rectangle feature (Figure 2.1.(A),(B)) and the three-rectangle feature (Figure 2.1.(C)). The features compute the differences between the sums of pixels within those rectangular areas. This kind of feature is then organized using summed-area tables (called integral images) and an algorithm (called a cascade of classifiers) [43] to speed up computation.

Based on these three key points, the VJ detector can perform robust and efficient face detection in real time. However, it concentrates only on frontal faces. Considering more realistic situations, some of the work also employed head pose estimation [45, 46, 47] to facilitate the field of face detection further. In our work, the VJ detector is used, and we ignore the non-frontal situation since the primary concentration is on emotion recognition.

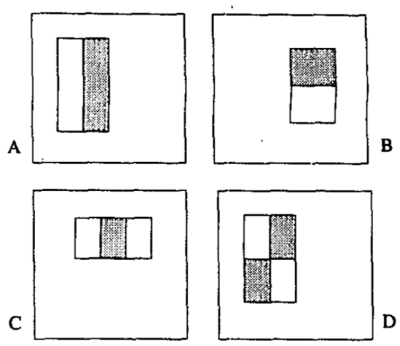


Figure 2.1: Examples of rectangular Haar-like features [43]

2.2.2 Facial Feature Extraction

Facial features in facial expression analysis, as mentioned in [4], can be divided into two types: intransient and transient. Intransient facial features refer to those that are always present in the face (e.g., eyes, mouth, nose, and eyebrows) but may be deformed when people display facial expressions. Transient facial features, on the other hand, represent wrinkles or other texture changes, especially in regions surrounding the eyes and the mouth, that appear with facial expressions.

How to extract or represent the facial features is one of the key aspects of facial expression analysis and is the focus of substantial previous work. One way to classify the approaches applied to facial feature extraction and representation is into geometric-based (shape-based) [48, 49] or appearance-based [50, 51] categories.

Geometric-based (shape-based) representation considers the shape information (intransient features, e.g., facial points or locations of eyebrows, eyes, the mouth, the nose) explicitly and ignores the texture. The face geometry is represented in the feature vectors extracted from that shape information. Appearance-based representation uses the intensity values or the pixels that refer to the textural changes [40], such as facial wrinkles as mentioned above. The facial features can be extracted from the entire face or particular facial regions. Most of the appearance-based work has adopted techniques such as the local binary pattern (LBP) [52, 2, 53], Gabor wavelet representation [54, 50, 51], scale-

invariant feature transform (SIFT) [14], or histograms of oriented gradients (HOG) [36, 37]. Notably, shape-based representation is typically vulnerable to illumination variation. It is challenging to maintain high accuracy and reliable facial component detection in real-time systems due to the varied environments [55]. In addition, according to Lopes et al. [56], the appearance-based approaches show better performance, particularly for low-resolution images, than shape-based approaches. Consequently, many shape-based approaches implement appearance-based approaches as a complementary process.

Zhang et al. [54] were the first to investigate and compare two kinds of feature extractions into the FER system. One type included 34 fiducial points selected manually, and another type was extracted with 2D Gabor transforms with three spatial frequencies and six distinct orientations; ultimately, 612 Gabor wavelet coefficients were extracted. These two sets of extracted features were then fed into a two-layer perceptron for training. The authors reported that the features extracted with the 2D Gabor wavelet were much more powerful than geometric facial points.

Following the work of Zhang et al., Zheng et al. [57] later proposed a method using KCCA which involved learning the correlation between a labeled graph (LG) vector converted from the 34 fiducial points and a six-dimensional semantic expression vector.

A real-time FER framework based on HOG features was proposed by Kumar [36]. The face was detected with 68 facial landmark points reported by Vahid Kazemi and Josephine Sullivan [58] and tracked using the Lucas-Kanade optical flow method [59]. Instead of extracting features from the whole face, seven critical empirical patches were selected for feature extraction. The feature vectors were obtained by applying nine bin histograms for each 4×4 cells in a single patch. These HOG features were then trained with linear support vector machine (SVM) and radial basis function (RBF) kernels.

Shan et al. [56] conducted a comprehensive study on LBP-based low-resolution FER and used a 59-bin LBP operator to divide the face images into 42 subregions. The resulting LBP features with a length of 2478 (59×42) were then used to recognize expressions

by different machine learning strategies including template matching, linear discriminant analysis (LDA), SVM and linear programming. The best result was achieved by further adopting the Adaboost technique with SVM to learn the most discriminative LBP features.

Facial features can also be roughly classified into the categories of hand-crafted and learned according to [34]. Most of the state-of-art work, including that mentioned above, uses hand-crafted features (e.g., Gabor wavelet coefficients, histograms of LBP, and HOG) so that both the computational power and programming efforts are needed for consideration.

2.2.3 Empirical Classifiers for FER

After the facial features are available, the last step for facial expression analysis is the classification of those basic emotions, which is referred to facial expression recognition. As indicated in [60], three stages of the training process in FER consists of feature learning, feature selection and classifier construction. Feature learning extracts features associated with facial expression. For feature selection, the ideal deeply learned features to be selected maximize interclass differences (to be separable) and minimize intraclass variations (to be discriminative) [32]. Ultimately, a classifier is constructed to infer the facial expression using these features selected in the previous step for each class.

Most of the previous work performed the FER with various types of classifiers, for example, neural networks (NN) [49, 54], LDA [51, 50], SVM [61, 15], Bayesian networks (BN) [39, 62], and rule-based classifiers [63, 64]. These kinds of classifiers leverage the extracted and selected hand-crafted or learned features related to facial expressions to execute the classification.

2.3 Convolutional Neural Network

The convolutional neural network (CNN) that was first introduced by LeCun et al. [65] in 1998 is characterized by its parameter-sharing architecture and the concept of receptive field. Recently, CNN has gained revived interest and achieved excellent state-of-the-art results relative to conventional methods in the field of computer vision. For example, especially for the task of face detection, approaches using CNN [66, 67] outperformed those using conventional approaches such as LBP [68] and deformable part models (DPMs) [69]. The same outcome was achieved in the related area of object detection [70, 71, 72].

The reason for the increasing popularity of CNNs may arise from its ability to learn and extract features directly from raw input data (even distorted images) that conventional machine learning and computer vision techniques require for manually extracted features. CNNs combine the three steps of FER mentioned in Section 2.2 (feature learning, feature selection and classifier construction) into one step and require minimal preprocessing. In addition, with the advantage of the graphical processing unit (GPU) technology, tasks that require intensive computation can achieve promising results at low power consumption.

2.3.1 Basic Architecture of CNN

- Neurons from the Artificial Neural Network

  The model of biological neurons initially inspired the use of artificial neurons as the elementary components in the artificial neural network (ANN). As a mathematical model shown in Figure 2.2, the neuron takes k inputs X_1 through X_k and weights w_1 through w_k in addition to a bias b. The weighted sum of the input volume is calculated, and a bias is added as the output of the kth neuron, as expressed in

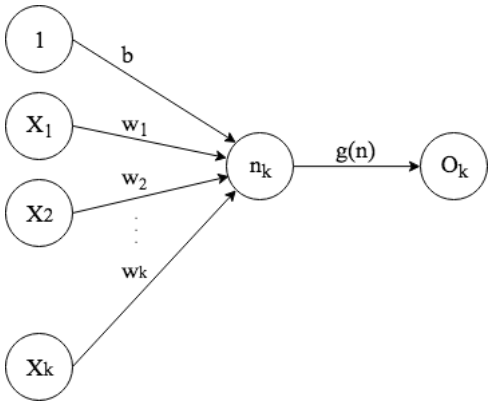


Figure 2.2: The sources of facial expressions.

Equation 2.1:

$$O_k = g(\sum_{i=1}^{k} w_i X_i + b) \tag{2.1}$$

where w_i is a fixed set of weights that are shared during convolution and $g(.)$ represents the nonlinear activation function, as defined below.

- Convolution Layer

  The principal goal of the CNN is to extract features from input images in the 'convolution' step, which convolves the pixels of an input image with the locally connected small area called the receptive field of the neuron. In CNN terminology, this receptive field is also called 'kernel' or 'filter' working as a feature detector, and the resulting dot product is the so-called 'feature map' obtained by sliding these filters over images. Since the feature perception moves from local to global, the convolution layer is also known as the locally connected layer, where the input images are connected with this local area of the receptive field. Every neuron shares a fixed set of weights with the receptive fields in a locally connected layer, which is called the weight-sharing scheme.

- Activation Function

The activation function that brings 'nonlinearity' to the CNN can be seen as a decision of whether to activate or 'fire' a neuron. The derivative of the linear function is a constant unrelated to the input, which leads to the problem in optimization that the gradient (which is constant) does not depend on the changes in the input so that the parameters are not updated. In addition, if all the activation functions are linear, the connection of layers in the CNN is a kind of linearity in which the whole network can be represented in a combination of linear manner, which is insufficient for the CNN to represent features. For this reason, the nonlinear activation function is introduced to the CNN, which provides an element-wise operation to compress pixels into a specific range. The rectified linear unit (ReLU) is one of the most famous examples of activation functions used in the CNN that replaces the negative values by zero. The function of the ReLU is shown in the Formula 2.2:

$$f(x) = max(x, 0) \tag{2.2}$$

- Subsampling Layer

  After obtaining the features from the convolution layer, the next step is to use these features to perform the classification. However, it is not advantageous to learn these millions of features with a classifier; furthermore, this scheme is prone to overfitting. To solve this problem, the subsampling (also called the pooling or downsampling) layer is used to decrease the spatial dimensionality to reduce the computation of training. The commonly used subsampling layer is the max-pooling layer, which takes the maximum value in the defined region with a pooling window (typically a 2×2 filter). In addition to the max-pooling layer, another widely used pooling method is to take the average value of the region, termed average-pooling. In practice, max-pooling shows better performance than average-pooling.

- Fully Connected Layer

The fully connected layer, which is similar to the traditional multi-layer perceptron neural network (MLP) [73], acts as a 'classifier' that connects every neuron of itself with every neuron of the previous layer and then outputs the scores of each class. Since the fully connected layer contains a vast number of parameters, to further reduce the amount of computation, some related work has replaced the fully connected layer with the global average pooling [74, 75, 76] instead.

2.3.2 Backward Propagation

Backpropagation was initially introduced by Rumelhart et al. [77] in 1986 and has been widely used for the training of deep models since it works much faster than the previously used approaches for learning. The first and most critical point of this algorithm is to compute the gradient of the cost (also called loss), function which is also called backward propagation of error. For its requirement of a known output, the backpropagation is taken as a supervised learning strategy using the gradient descent optimization algorithm proposed in [65]. Following the chain rule, this procedure computes the gradients of every layer iteratively from the end to the beginning of the network and then updates its weights and bias to determine the best map from the inputs to the correct outputs.

- Softmax Loss

  During the process of optimization, the key is to determine the distance between the true labels and the predicted labels, which is termed the loss (or cost) function. One of the most widely used choices of loss functions for CNN training is the softmax loss function [14, 78, 79]. This function actually consists of two parts: the softmax function and the cross-entropy loss. The softmax function is a generalization of logistic regression for a multiclass configuration that intuitively compresses the real-valued scores into probabilities that range from zero to one and sum to one. In the CNN, the input of the softmax classifier is typically the output from the

previous fully connected layer or global average pooling layer mentioned above. The softmax classifier is shown in Equation 2.3:

$$S_i = \frac{\exp V_i}{\sum_j \exp V_j} \tag{2.3}$$

where V_i is the ith output of the fully connected layer.

After obtaining the probabilities of each class, the loss is calculated using the cross-entropy function, as shown in Equation 2.4:

$$L_{crossentropy} = -\sum_j y_j \log S_J \tag{2.4}$$

where y_j stands for the jth true label. Since the probability distribution of the correct class has a total area of one, the equation above can be expressed as 2.5, which is also the softmax loss function:

$$L_s = -\sum_j \log S_J \tag{2.5}$$

2.3.3 Transfer Learning of the CNN

2.3.3.1 ImageNet: Large-scale Annotated Natural Image Datasets

To provide an unparalleled opportunity for researchers in the field of computer vision, Deng et al. organized a challenging large-scale image database, the ImageNet [80] in 2009, which follows the densely populated semantic hierarchical structure of WordNet [81] and is collected in the scheme with Amazon Mechanical Turk for large-scale labeling. This database contains more than 1.2 million 256×256 human-annotated images distributed over more than 1000 object class categories, where the objects in this dataset can have diverse positions, backgrounds and even occlusions. Some examples of Ima-

geNet are shown in Figure 2.3. The ImageNet is currently the largest high-quality visual recognition database; due to its public availability, it has become the standard benchmark and has been widely applied to large-scale visual tasks such as image classification and object recognition. Because of its intraclass variation, this database can help such data-driven systems improve their performances even with different domain problems, e.g., the medical image domain [82, 83].

(a) banana (b) cup (c) mouse (d) pizza (e) zebra

Figure 2.3: Some examples of different object categories in the ImageNet dataset.

2.3.3.2 Transfer Learning: Fine-tuning

CNN was heavily used in computer vision after being proposed in the 1990s, but with the rise of prevalence in SVM, the method fell out of fashion. With the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012, however, the CNN approach was revived by exhibiting its superior performance [30] on the large-scale ImageNet dataset resulting in a revolution with the aid of graphics processing units (GPU), dropout regularization and activation nonlinearity (ReLU).

Training deep models from scratch, which is also referred to full training as mentioned in [83], is a challenging task for several reasons in that it demands much experience, expertise and patience. First, it is difficult to find a large-scale image dataset in a particular domain that can be comparable to ImageNet in both data size and annotated quality. Training of the deep models is time-consuming and requires a large memory space and extensive computational power. In addition, overfitting or convergence-related problems sometimes occur that necessitate constant efforts to adjust the learning parameters and

architecture to ensure that the CNN learns better features and enables faster convergence.

An alternative option for the training of a CNN is first to initialize the network with a set of pretrained weights (and bias) based on a large-scale dataset from one task and retraining these parameters for a different target task. This kind of training process is termed fine-tuning. A common fine-tuning practice is to adapt the pretrained weights for initialization to all layers of the CNN except for the first and the last layer since the input resolution or the number of classes of the dataset of the new task may vary from the dataset used for pretraining. The set of weights of the first layer is often randomly initialized from a Gaussian distribution, and a new classification layer often replaces the last fully connected layer with the same number of neurons as that of the new task dataset. In general, the effectiveness of utilizing the fine-tuning training strategy has been addressed by various work, e.g., [83, 84] and applying the pretrained model based on the ImageNet to the training of CNN also has become a backbone for image classification, achieving promising results on object detection [70, 71, 72], in the medical image domain [82, 83, 85] and on action recognition [84, 86].

2.4 CNN-based Work of FER

In general, CNN is an end-to-end trainable [14] method of supervised learning that combines a feed-forward neural network with backpropagation using iterative algorithms such as the gradient descent method. The hierarchical structure of CNN—which is a design of local to global feature learning [87] comprising diverse convolution, subsampling (e.g., max-pooling, average-pooling) and fully connected layers—contains a strong feature representation capacity and can be a powerful tool in FER. CNN-based methods have already been successfully employed in FER [15, 88, 89] and have achieved outstanding performance. The frameworks of these work include image preprocessing, CNN

architecture, training schemes and evaluation configuration.

Burkert et al. [90] proposed a CNN architecture that does not depend on the hand-crafted features. Four parts compose this architecture, and the images are first preprocessed automatically through a convolutional layer. The images are then downsampled by the pooling layer in the second part. The next block, called the FeatEx, serves as the fundamental structure in this architecture, which was inspired by GoogleNet. Finally, the extracted features after two concatenated FeatEx blocks are fed into a fully connected layer to perform the classification. The deep features of different layers are visualized to show its validity, and evaluations are conducted with two standards datasets, namely, MMI and CK+. Their experiment on the CK+ dataset, which evaluated seven classes (*angry, disgust, fear, happiness, sadness, surprise, and contempt*), achieved a recognition rate of 99.6%.

Tang et al. [13] presented a framework that followed the convolution routines of Alex Krizhevsky [30] but replaced the last softmax layer with a linear SVM and optimized the loss from the L2-SVM (DLSVM) instead of the cross-entropy loss. The input images were first preprocessed by subtracting the mean value and setting the norm to 100 before being fed into the network for training. Superior results were obtained by using this DLSVM with softmax by evaluating two standard datasets, MNIST and CIFAR-10, as well as one of the largest recent FER datasets: FER-2013 [91]. The performance of the proposed framework won 1st place in the FER challenge of the ICML 2013 Workshop hosted on Kaggle, with an accuracy of 69.4% for the public validation set and 71.2% for the private test set; the human accuracy on FER-2013 was $65 \pm 5\%$.

To engage in the image-based static facial expression recognition sub-challenge of the EmotiW2015, Kim et al. [88] constructed a hierarchical committee of multiple CNNs with an ensemble method based on exponentially weighted decision fusion. First, face registration was achieved by four different pipelines, where the VJ detector and the Zhu-Ramanan (Z-R) model were used for face detection and the IntraFace was used

for the facial landmarks extraction; finally, the best method was selected by 2-D conventional alignment. Following Tang's CNN architecture [13], these approaches trained several CNN candidates varying in the size of kernels (receptive fields) and the number of neurons in the fully connected layer; finally, a decision was made by the ensemble method. In addition, to improve the performance on the SFEW2.0 dataset, a transfer learning method that used external datasets, namely, FER-2013, the Toronto Face Database (TFD) and the GENKI-4K database, was applied to the training process. This configuration defeated other candidates in this challenge by achieving a test accuracy of 61.6% on the SFEW2.0 dataset; the baseline for this dataset was 39.1%.

As an additional candidate in the contest of EmotiW2015, Ng et al. [89] emphasized the importance of using a transfer learning method, which was a supervised two-stage process. Two representative CNN architectures, (AlexNet and VGG-CNN-M-2048) were selected for their tradeoffs regarding accuracy and speed. By first fine-tuning the FER-2013 dataset based on the pretrained models from ImageNet and then fine-tuning the SFEW2.0 dataset (the target dataset), the authors reached an accuracy of 55.6% on the test set, which ranked 3rd place in the contest.

Inspired by the architecture of the AlexNet and GoogleNet [92], Mollahosseini et al. [93] proposed their own CNN architecture in 2016, which consisted of two conventional CNN modules (one of which contained a convolutional layer followed by a max-pooling layer), four *Inception* modules and two fully connected layers, having only 25M operations (compared to 100M in AlexNet). Face registration was performed to improve the performance of FER by using the bidirectional warping of the active appearance model (AAM) and a supervised method called IntraFace that adopted the SIFT features to extract 49 facial landmarks. Both subject-independent and cross-database experiments were carried out on seven public standards datasets (MultiPIE, MMI, CK+, DISFA, FERA, SFEW, and FER-2013), and six specific classes (angry, disgust, fear, happiness, sadness, surprise, excluding the neutral and contempt classes) were evaluated on the

CK+ dataset.

2.5 Summary

From all above, we can generally conclude from the literature review that conventional machine learning methods always require the manually extracted features, i.e., the hand-crafted features (LBP, HOG, Gabor wavelet coefficient, etc), so that both computational power and human efforts are needed for consideration. While the CNN-based methods can learn and extract features directly from raw input image and combines these steps of feature extraction and classification into one single step, and require minimal preprocessing. At the same time, it can also be able to obtain promising results or even outperform those using conventional methods. This is one promising advantage of the CNN-based method. One more thing is that, most of the approaches for FER are impractical for using in real-time scenario since they are constrained by the cost of the computational time.

Chapter 3

Methodology

In this chapter, we introduce the methodology of our facial expression recognition system. As reviewed in chapter 2, much related work has addressed this kind of FER task by either using empirical classifiers to accommodate hand-crafted features [54, 51, 50] or directly training the revived deep learning strategy, CNN, with the raw input of images [15, 78, 87]. However, most of these approaches are impractical for use in real-time scenarios because they are constrained by the cost of computational time.

As shown in Figure 3.1, the proposed methodology contains two parts. After the image data are prepared, the CNN training scheme—which consists of a transfer learning

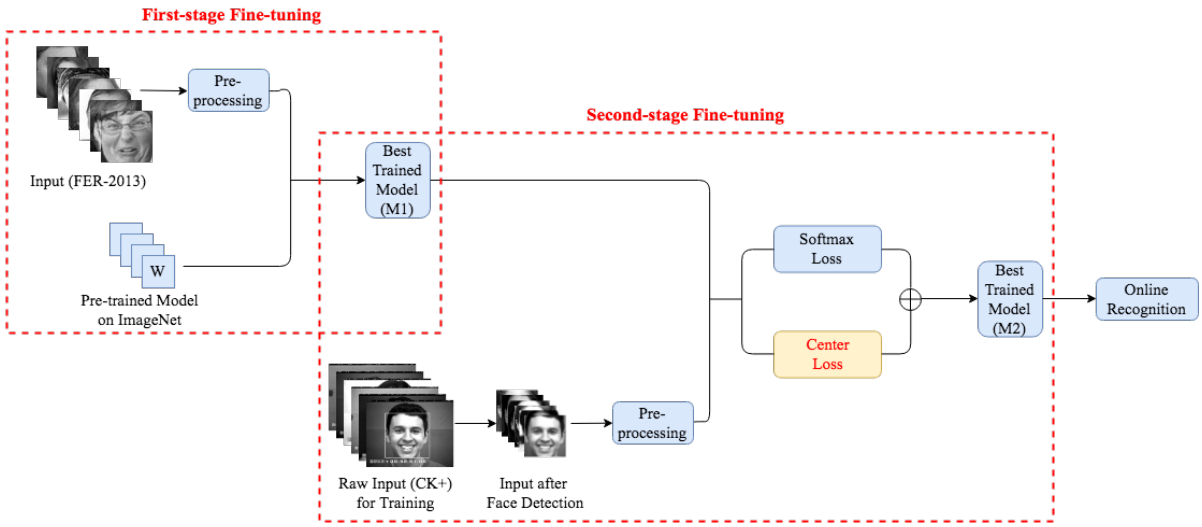


Figure 3.1: Overview of the proposed framework. A first stage of fine-tuning is applied to FER-2013 based on the pretrained model from ImageNet. After obtaining the best-trained model, a second stage of fine-tuning is then performed on a specific dataset. An additional center loss is used as a part of the supervision signal together with the softmax loss during optimization. Finally, the best-trained model is selected for online classification. This example involves the CK+ dataset.

strategy and an optimization method combining a newly proposed supervision signal with softmax loss—is applied to a recent proposed CNN model. Finally, the best-trained model is used to perform the online classification. This chapter describes all the details of the proposed methodology to accomplish this real-time FER task in classifying one of the basic facial expressions. The details for data preprocessing are described in Section 3.1. Finally, we present the adopted transfer learning method and the joint supervision strategy in Section 3.2 and Section 3.3, respectively.

3.1 Preprocessing

In the processing of FER, some interference may arise that influences the effectiveness of feature extraction due to the complexity of the environment and the various performance of the shooting equipment during data acquisition. Preprocessing of expression images can vary depending on factors such as the performance of the acquisition equipment or

changes in illustration conditions. This issue typically manifests in the images containing different levels of noise and the average pixel intensities of different images showing various brightness contrast. Thus, it is necessary to perform data preprocessing, the general purpose of which is to eliminate noise and to normalize and centralize the gray value of the image to provide a solid foundation for subsequent classification and identification. However, extensive image preprocessing may require large run-time cost, which threatens real-time capability. In our work, we perform a minimal amount of preprocessing while maintaining accuracy.

3.1.1 Face Detection

As stated by Viola and Jones, the three main stages of the original algorithm of the VJ detector are Haar feature selection with a representation of *IntegralImage*, Adaboost learning, and a construction of cascade-classifiers.

As shown in Figure 3.2 provided in the paper by Viola and Jones [43], the two satisfactory Haar features selected describe the differences in brightness intensities between certain regions of the human face; for example, the eyes are often darker than the upper cheeks and the nose bridge. To simplify the calculation of the sums of pixels under those black and white rectangles, an integral image representation was proposed. In this way, each calculation of the sum of pixels under a rectangle can be reduced to four array references so that any representation of two-rectangle and three-rectangle features can be computed in six and nine array references, respectively [43]. After features are selected, Adaboost learning is applied to find a final classifier that is a combination of a set of weak classifiers to choose the fewest optimal features that can distinguish positive and negative images. Finally, to enhance the efficiency, classifiers are constructed in a "cascade" fashion in which the latter classifier is triggered only when the former classifier obtains a positive result. Due to this principle, the VJ face detector is known for its fast speed of feature computation and efficient feature selection, providing the capability for

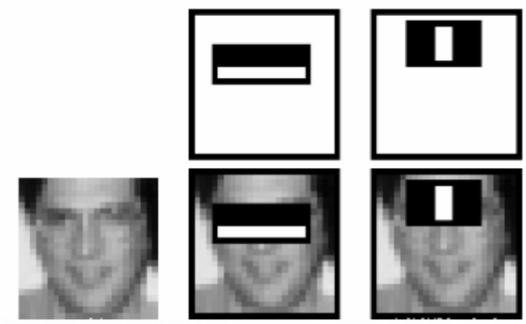


Figure 3.2: Two Haar features selected for face detection [43].

real-time analysis.

In our work, this Haar cascade classifier is adopted for face detection both in our offline and real-time systems, and we ignore the non-frontal situation since the primary concentration is on the FER part. The input images are loaded and converted into grayscale mode. If the classifier finds the faces, it returns the four coordinates of the rectangular region of interest (ROI) of the faces. Once the locations of this ROI are obtained, these four vertices are used to crop the faces, and the irrelevant backgrounds are deleted; image processing can be addressed later. Examples of detected ROI for faces of the two datasets (CK+ and JAFFE) are shown in Figure 3.3. Concerning the FER-2013 dataset, the low resolution of its images (48×48 pixels) and the various head orientations prevent reliable face detection, so no face detection was applied to this dataset. Even without face detection, the performance is remarkably enhanced when FER-2013 is applied to the first stage of fine-tuning, as shown in Chapter 5.

3.1.2 Data Augmentation

Data augmentation is often employed during the training of the CNN since the process itself incorporates a large quantity of data. In the training scheme of this work, the cropped faces are first distorted with a lightweight library in TensorFlow [94] before feeding them into the CNN. Each cropped face is randomly sampled by one of the distorted bounding boxes. The area of the sampled patch is [0.85, 1] of the original supplied

(a) CK+ (b) JAFFE

Figure 3.3: Examples of face detection.
Examples of detected ROIs for faces in the CK+ and JAFFE datasets. The white bounding boxes represent the ROI of the faces and the four vertices used to crop the faces.

image, and the number of generated images is as high as 100. After the sampling step, the sampled patches are rescaled to 48×48 pixels since the CNN training input must be square. The reason for the selection of this rescale parameter 48×48 is to remain consistent with the resolution of the FER-2013 dataset, as described later in this chapter. Furthermore, after rescaling, the images are also randomly flipped horizontally with a probability of 0.5 in order to have two times more data. Figure 3.4 shows examples of the randomly cropped images from two datasets. Finally, normalization is performed for faster convergence. The data are normalized into the range of $[-1, 1]$ instead of the typical $[0, 1]$ because the activation function ReLU, which is $\max(0, x)$, works better when negative values are also provided.

(a) CK+ (b) JAFFE

Figure 3.4: Example of data augmentation.
Examples of illustrating how random cropping works for data augmentation of the CK+ and JAFFE datasets. The yellow boxes are randomly generated in the range of 0.85 to 1 of the cropped faces to obtain additional cropped images during each iteration.

3.2 Framework and Transfer Learning

3.2.1 CNN structure

MobileNet [29], which was proposed in 2017, is a lightweight deep neural network produced by Google. Following a year's development, it has become an underlying network structure similar to GoogleNet and ResNet. Its starting point is to construct a lean, lightweight network based on a streamlined architecture, and it can be used efficiently on mobile and embedded devices that offer limited performance. It can be considered a method of network compression, but unlike other methods of compressing models, it does not perform operations such as pruning, quantification, and decomposition on an extensive network but rather outputs a new network structure using depthwise separable convolutions.

The first version of MobileNet (MobileNet V1) is employed as the CNN architecture in both offline and real-time systems of this work. The core of the MobileNet V1 is that it reduces the amount of computation by replacing conventional convolution with depthwise separable convolution. This convolution method decouples standard convolution into depthwise convolution, which plays a role in feature extraction, and a 1×1 pointwise convolution, which combines or fuses the extracted features into new features by additive means. The depthwise convolution acts as a filter in that it decomposes the channels and performs standard convolution for each channel individually, the kernel of which is $(D_K, D_K, 1, M)$, where D_K is the spatial size of input volume and M is the number of input channels. Then, pointwise convolution is used for the conversion of the channel, the kernel of which is $(1, 1, M, N)$, where N is the number of output channels.

The reduction of the computation and model size in the entire process is due to fewer filters extracting features. This kind of computation reduction between conventional convolution and two-step convolution is compared in the original paper [29] as shown in

Equation 3.1, where the denominator represents the former and the numerator represents the latter:

$$\frac{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F}{D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F} = \frac{1}{N} + \frac{1}{D_K^2} \tag{3.1}$$

where D_K and D_F are the spatial width/height of a kernel and an input feature map, respectively, and M and N represent the number of input and output channels, respectively.

In addition, the original paper [29] also defines two hyperparameters: the width multiplier α (the scaling factor), through which we can further reduce the amount of calculation, and the resolution multiplier ρ to control the size of the feature map. The body architecture of MobileNet V1 is shown in Table 3.1. Table 3.2 illustrates the performance on ImageNet when comparing MobileNet V1 with other popular CNN benchmarks. Relative to the other widely used models, MobileNet provides a better trade-off between speed and size.

As stated in the paper, all the merits mentioned above contribute to our decision to select the MobileNet structure as the framework of this work. The characteristics of small size, low complexity, and remarkable accuracy enable this FER task to maintain a favorable trade-off between speed and accuracy and to run in real time with small run-time cost, see Chapter 5.

3.2.2 Transfer Learning Scheme

One main problem for CNN-based FER is the insufficient size of the most of existing facial expression datasets. The required size of the labeled training data for CNN to learn and extract features and obtain high accuracies is asked to be in millions as mentioned in Chapter 2 while the size of most facial expression datasets is only hundreds or thousands. Training deep models with such limited amount of data is rather challenging since sometimes it may lead to the problem of overfitting (the model may have poor

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Table 3.1: The architecture of MobileNet [29]

Model	ImageNet Accuracy	Million Multi-Adds	Million Parameters
1.0 MobileNet V1 224	70.6%	569	4.2
AlexNet	57.2%	720	60
GoogleNet	69.8%	1550	6.8
VGG16	71.5%	15300	138
SqueezeNet	57.5%	1700	1.25

Table 3.2: Performance comparisons on ImageNet [29]

performance on the test data while attaining rather perfect performance on the training data). In addition, it is time-consuming for training from scratch without taking advantage of the pretrained model. One of the common ways to address this problem is to use inductive transfer learning [95]—the so-called fine-tuning strategy. A hallmark of this approach is the fine-tuning of a small target dataset based on the pretrained models on the ILSVRC-2012 (ImageNet), and much recent work [83, 84, 86, 82] has established the feasibility of this strategy. Although the target task (in this case, the FER task) is different from the source (which is object classification), the low-level features that are learned by the early layers of the CNN can still be applied to most of the other vision tasks, as noted previously [83]. Furthermore, as shown in Table 3.3, the JAFFE and CK+ datasets contain less than 1K of data and cannot be considered large scale, so this strategy is reasonable.

Dataset	Size of the dataset (K)
JAFFE	0.213
CK+	0.327 / 0.981
FER-2013	36.889

Table 3.3: The size of the datasets utilized in this work.

To further compensate for the small size of those two datasets using fine-tuning and overcome the difference between the target task and the source task, we follow the recent studies of [88, 89] to take advantage of the relatively large size of the FER-2013 dataset (which is more than 30K as shown in Table 3.3). Inspired by the associated transfer learning schemes, 'two-stage' fine-tuning is employed in the training scheme of CNN instead of only fine-tuning the already pretrained models from ImageNet. This technique is a kind of 'coarse-to-fine' training process as illustrated in Figure 3.1. To make use of the large FER-2013 dataset, we perform the first stage of fine-tuning following the approach elaborated in [95] that first fine-tunes the relatively small dataset FER-2013 (compared to ImageNet) in the target domain (in FER) by initializing the network with pretrained weights from ILSVRC-2012 in the source domain. Considering the distance between the

tasks of FER and object classification (the target and the source), we refer to this first-stage fine-tuning as 'coarse' fine-tuning. After obtaining the best-trained models from FER-2013, a second-stage fine-tuning step is applied to the JAFFE and CK+ datasets by transferring these set of pretrained weights to the network. Since the target and the source tasks are the same in this process, we term it 'refined' fine-tuning.

For the two fine-tuning schemes, the last fully connected layers are both replaced by a new classification layer classifying seven classes (the number of classes for CK+ may vary, as elaborated in Chapter 5). For the 'coarse' fine-tuning, the set of weights of the first convolutional layer are randomly initialized from a Gaussian distribution since the input of the images after preprocessing has a size of 48×48, while the original MobileNet V1 is 224×224. The initial learning rate of the first-stage fine-tuning is set to 0.001, which is relatively small, to 'tune' the pretrained weights of the early layers of the network from ImageNet slightly. For the 'refined' fine-tuning, the pretrained weights of the first layer from FER-2013 are directly set as the initialization at the beginning because of the same input size of 48×48. At this stage of fine-tuning, the initial learning rate is set to a relatively large value (e.g., 0.045, see the training details in Chapter 5) to 'lock' the weights of early layers (since FER-2013 and the targeted datasets CK+ and JAFFE are in the same domain) and to relearn the high-level features for the specific dataset (CK+ or JAFFE).

3.3 Joint Supervision

3.3.1 Center Loss

For recognition tasks in the computer vision field, particularly for FER in this case, the ideal learned features to be selected should contain not only interclass separation (separable) but also intraclass compactness (discriminatory). The softmax loss that had been widely used for optimization of CNN takes the exponential form and normalizes

the probability of a sample belonging to a specific class. Its key characteristic of nonnegativity eliminates the possibility of positive and negative value cancellation. However, the softmax loss tends to separate the deep features. Thus, Wen et al. [32] proposed a new supervision signal called center loss to enhance this discriminatory power. In this work, this center loss is used in the loss function together with the softmax loss for better optimization of the network training.

The principle of the center loss is to maintain a class center in the feature space for each class of the training set. In the training process, the center loss acts to reduce the intraclass differences by increasing the distance constraint between the features and its corresponding class center of the samples. The calculation of the center loss is given in Equation 3.2 below:

$$L_{CL} = \frac{1}{2} \sum_{i=1}^{N} \|x_i - c_{y_i}\|_2 \tag{3.2}$$

where x_i stands for the ith deep features extracted before the final classification layer (i.e., the features extracted after the 'Avg Pool' layer of this framework in Table 3.1) instead of those classified possibilities after the fully connected layer. c_{y_i} represents the learned center for the y_ith class. It is impractical to update the centers of each class concerning the entire training set due to the enormous computational load. The centers are updated within each iteration using a mini-batch strategy (the size of which is N in Equation 3.2) and computed by taking the average of the deep features of the corresponding class. Therefore, this formula ensures that the sum of the squares of the distances between each feature in a mini-batch and its corresponding class center is as small as possible, that is, the smaller the distance within the class (intraclass variations), the easier intraclass compactness can be achieved.

In addition, a hyperparameter α lying within $[0, 1]$ (the selection of α is described in Chapter 5) is also employed to control the learning rate of those centers during the

update within each iteration following the work of [32]:

$$c_{y_i}^{t+1} = c_{y_i}^{t} - \alpha \cdot \Delta c_{y_i}^{t} \tag{3.3}$$

And the update version of $c_{y_i}^{t}$ is computed as:

$$\Delta c_{y_i}^{t} = \frac{\sum_{i=1}^{N} \delta(j = y_i) \cdot (c_j - x_i)}{1 + \sum_{i=1}^{N} \delta(j = y_i)} \tag{3.4}$$

where N is the batch size and the $\delta(\cdot)$ function ensures the class center is updated only when the condition $(j = y_i)$ is satisfied, which means that the deep feature x_i belongs to its corresponding class ($\delta(j = y_i) = 1$; otherwise, it is 0).

3.3.2 Learning Algorithm with Joint Loss Function

To construct discriminative deep features of facial expressions that contain both intraclass compactness and interclass differences, the center loss is supervised under collaboration with the conventional softmax loss in our work, closely following the approach of [32]. The total loss used for network optimization is calculated in Equation 3.5:

$$L = L_S + \lambda \cdot L_{CL} = -\sum_{i=1}^{N} \log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{m} e^{W_j^T x_i + b_j}} + \frac{\lambda}{2} \sum_{i=1}^{N} \|x_i - c_{y_i}\|_2 \tag{3.5}$$

where the hyperparameter λ is selected to control the portion of the center loss added to the total loss and where m in the equation stands for the total number of classes. This collaboration of supervision is illustrated in Figure 3.5, where the deep features extracted after the 'Avg Pool' layer are used for calculation of the center loss and where those elements extracted after the final fully connected layer are collected to calculate the softmax loss.

With a proper value of λ balancing the two loss function, the network is optimized

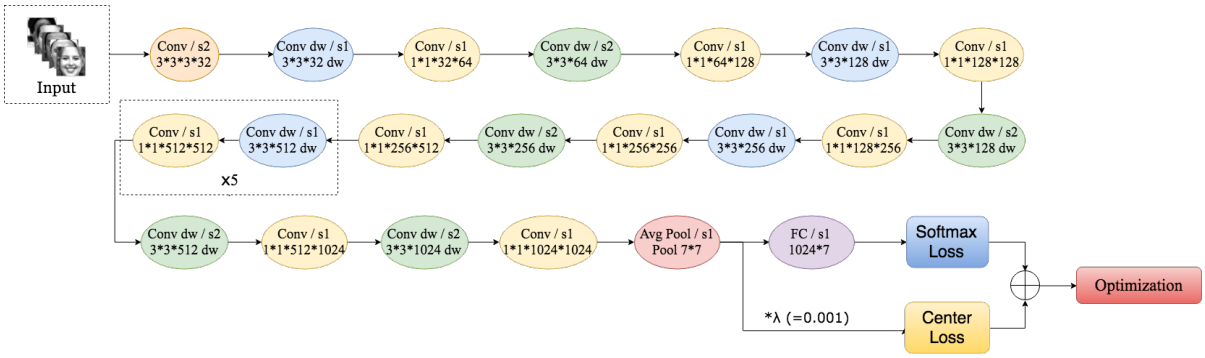


Figure 3.5: The proposed structure of this work, using collaborated supervision with center loss. Note that the deep features before the fully connected layer are used for calculating the center loss, while those after the fully connected layer are collected for the softmax loss.

using the stochastic gradient descent (SGD) [65] with momentum to stabilize the update and considerably speed up the convergence. The gradient of L_{CL} concerning the centers c_{y_i} and the deep features x_i are computed as follows:

$$\begin{aligned} \nabla x_i &= \frac{\partial L_{CL}}{\partial x_i} = x_i - c_{y_i} \\ \nabla c_j &= \frac{\partial L_{CL}}{\partial c_j} = \frac{\sum_{i=1}^{N} \delta(j = y_i) \cdot (c_j - x_i)}{N + 1} \end{aligned} \tag{3.6}$$

The details of the learning process are summarized in Algorithm 1; after initialization, the total loss is first computed during each iteration. Then, the weights and parameters of the network are updated through computing the gradient of backpropagation error.

Algorithm 1 The learning algorithm

Input : Training data x_i, initialized parameters θ_c and weights W of the network, hyperparameter λ, α, learning rate and initialized iteration: $t \leftarrow 0$.

Output: The updated parameters θ_c

while *not converge* **do**

$t \leftarrow t + 1$

Compute the total loss by $L^t = L_S^t + \lambda \cdot L_{CL}^t$

Compute the backpropagation error by $\frac{\partial L^t}{\partial x_i^t} = \frac{\partial L_S^t}{\partial x_i^t} + \lambda \cdot \frac{\partial L_{CL}^t}{\partial x_i^t}$

Update the weights W by $\frac{\partial L^t}{\partial W^t}$

Update the parameter c_j by $c_j^{t+1} = c_j^t - \alpha \cdot \Delta c_j^t$

Update the parameters θ_c by $\bigtriangledown \theta_c = \frac{\partial L_S}{\partial \theta_c} + \frac{\partial L_{CL}}{\partial \theta_c}$

end

Chapter 4

Datasets

In this chapter, the details of the datasets used in this work are presented. The primary basis for the computer performing the FER is the visual data from the FER image database. Due to the constraints of the research funding, time, energy and the requirements for evaluation of the algorithm performance, most researchers exploring the field of FER often rely on the existing facial expression datasets. The most commonly used existing facial expression datasets are CK+, JAFFE, MMI, SPEW, YaleFace, FER-2013, MultiPIE, TFD and so on. Among these, the two most widely used standard datasets for evaluation from the very beginning of expression studies to the present are the CK+

and JAFFE datasets, which are selected in this work for evaluation. Another dataset adopted in this work is the FER-2013 dataset, which is currently one of the largest facial expression datasets. The examples of basic facial expressions of the three datasets used in this work and the distributions of every class in each dataset are shown in Figure 4.1 and Table 4.1.

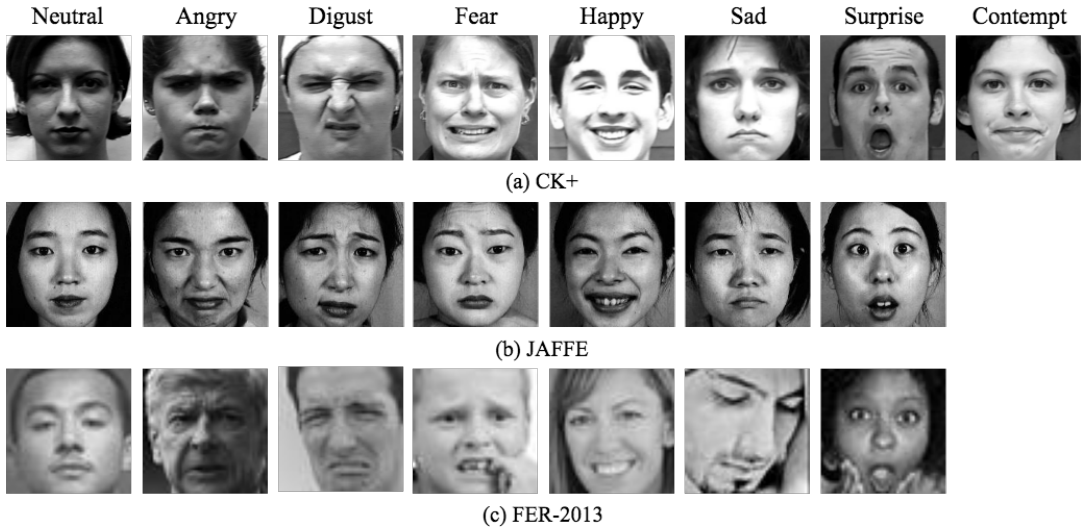


Figure 4.1: Examples of basic facial expressions of the three datasets.
Note that every dataset contains seven basic expressions (*neutral, angry, disgust, fear, happy, sad, surprise*), while CK+ also includes a *contempt* class. The facial expressions of FER-2013 are wild-type, while those in the CK+ and JAFFE datasets are posed and collected in a laboratory-controlled environment.

	FER-2013	CK+ (last frame)	JAFFE
Angry	4593	45	30
Disgust	547	59	30
Fear	5121	25	31
Happy	8989	69	31
Sad	6077	28	31
Surprise	4002	83	30
Neutral	6198	327	30
Contempt	0	18	0

Table 4.1: The distributions of every class in each dataset.

- The Japanese Female Facial Expression (JAFFE) Dataset

The ATR Human Information Processing Research Laboratory collected the Japanese Female Facial Expression (JAFFE) Dataset, which consists of seven basic facial expressions (six basic ones and a neutral one). The dataset contains 10 Japanese female expressors, each of whom posed $3 \sim 4$ examples of the six basic facial expressions and one for the neutral, resulting in a total of 213 static 256×256 images in the database. All images in the database are well-posed, with even illumination, a single imaging background and no occlusions such as eyeglasses. This database is characterized by relatively subtle emotion expression and presents a more challenging FER task.

- The Extended Cohn-Kanade Dataset (CK+)

  The CK+ dataset [16], which was released in 2010, is an extension of the Cohn-Kanade (CK) dataset that increased the number of sequences and the number of subjects by 22% and 27%, respectively. The dataset contains 327 image sequences digitized into 640×480 from neutral to peak emotions of both posed and non-posed (spontaneous) expressions with FACS-coded emotion labels for the peak frames. The 123 subjects in this database range from $18 \sim 50$ years of age (81% Euro-American, 13% Afro-American, 6% other ethnicities), 69% of whom are female. In addition to the seven basic facial expressions, another class *Contempt* is included in this dataset, resulting in a total of eight classes of facial expressions. Several baseline results and benchmarking protocols for shape and appearance feature tracking, as well as emotion and AU labels, are also provided in this dataset.

- FER-2013 Dataset

  The FER-2013 dataset was presented in the sub-challenge/competition Facial Expression Recognition Challenge of Challenges in Representation Learning in the ICML 2013 workshop [91], which was hosted by Kaggle. The dataset itself was retrieved from the Internet using the Google image search API consisting of 36,887

images, where 28,709 are used as training data, 3589 are used for public validation, and another 3589 are used for the private test. The dataset contains 48×48 pixel low-resolution grayscale images across seven basic facial expression classes. Because of the label noise and the variety of real-world conditions collected from the Internet, the FER-2013 has become by far one of the more widely used and also most challenging spontaneous dataset for FER, the human recognition rate of which is approximately 68%. For the contest itself, over 120 teams participated, and the first place was won by the team of Tang et al.[13], reaching an accuracy of 71.2% for the private test.

Chapter 5

Experiments and Evaluations

In this chapter, we present the details of the implementation of our work. Some of the details for data preparation are provided in Section 5.1. The training parameters for CNN training and the evaluation criterion are given in Section 5.2 and 5.3, respectively. In Section 5.4, we describe the offline experiments, including verifying the performance of the two-stage fine-tuning strategy and the joint supervision in addition to the self-evaluation on both JAFFE and CK+. In addition, Section 5.5 presents the implementation of our real-time facial expression recognition system and the evaluation of the run-time cost. Finally, the environment configuration is given in Section 5.6.

5.1 Data Preparation

For evaluation, the images in the JAFFE and CK+ datasets are randomly shuffled, respectively. Then each dataset is using the 5-fold cross-validation (i.e., the datasets is split into five groups during the CNN training, with four groups for training and the remaining group for validation each time and repeat it for five times). For the FER-2013 dataset, the total training set and the public test set are used for training and validation, respectively. For the CK+ dataset, the various situations of different sizes for training and validation are elaborated in Section 5.4.4. Note that the data for CNN training are preprocessed following the procedures elaborated in Section 3.1, while the data for evaluation does not conduct the data augmentation.

5.2 Training Parameters

The total loss function is optimized during backpropagation using the stochastic gradient descent (SGD) optimizer with a momentum of 0.9, weight decay of 0.00004 on the model weights, and a batch size of 64 for a mini-batch. The initial learning rate for the first-stage and the second-stage fine-tuning is 0.01 and 0.045, respectively. For the 'refined' fine-tuning, the initial learning rate is set to slightly larger than the 'coarse' fine-tuning since we do not intend to heavily alter these already pretrained weights and because we expect to fine-tune these high-level features related to facial expression. The learning rate exponentially decreases by 0.94 times every 15 epochs of training. To reduce the potential for overfitting, we apply a dropout strategy as proposed by Hinton [96] after the 'Avg Pool' layer and before the final fully connected layer with a probability of 0.5. The dropout mechanism randomly discards some units with 50% probability in training, which increases the sparsity of the network and avoids overfitting to some extent. For the selection of values of the parameters, we first select an approximate range for these

parameters based on the work where people using the CNN-based methods (e.g. [97, 28]). And then we determined a set of relatively good parameters through several sets of comparative experiments. All the training parameters are shown in Table 5.1

Values	1st-stage fine-tuning on FER-2013	2nd-stage fine-tuning on CK+	2nd-stage fine-tuning on JAFFE
Optimizer	SGD	SGD	SGD
Momentum	0.9	0.9	0.9
Weight decay	0.00004	0.00004	0.00004
Dropout probability	0.5	0.5	0.5
Initial learning rate	0.01	0.045	0.045
Learning rate decay factor	0.94	0.94	0.94
Number of epochs per decay	20	15	15

Table 5.1: The training parameters for three scenarios.

5.3 Evaluation Criterion

In this evaluation part, the accuracy and the confusion matrix for each experiment are presented. An example of the confusion matrix for multiclasses is illustrated in Table 5.2. Suppose we have an n-class problem with classes $C_1, C_2, \cdots, C_n$, which will result in an $n \times n$ confusion matrix, the rows and columns of which represent the actual classes and the predicted classes, respectively. The TP_n values at the diagonal are the true positive (TP) values for corresponding classes, which indicate the correct decisions, while the off-diagonal values are the misclassified errors (for example, E_{12} in this matrix indicates that the actual class C_1 is predicted as C_2).

	Predicted					
Actual		C_1	C_2	$\cdots$	$\cdots$	C_n
	C_1	TP_1	E_{12}	$\cdots$	$\cdots$	E_{1n}
	C_2	E_{21}	TP_2	$\cdots$	$\cdots$	E_{2n}
	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
	C_n	E_{n1}	E_{n2}	$\cdots$	$\cdots$	TP_n

Table 5.2: An illustration of the confusion matrix for multiclasses

In this case, the overall accuracy of a model can be calculated as the sum of correct decisions divided by the total number of classifications as shown in Equation 5.1:

$$Accuracy = \frac{\sum_{i=0}^{n} TP_i}{S_{total}} \tag{5.1}$$

where S_{total} stands for the total size of the validation set.

In addition, we computed the precision and recall values—two matrices widely used in the field of information retrieval and statistical classification to evaluate the quality of results—and the F1-Score. The precision shows the degree of relevance of the retrieved instances and is calculated as the ratio of the number of relevant instances retrieved (TP) to the total number of retrievals (which is the sum of the TP and false positives (FP)). The recall refers to the fraction of relevant instances that are retrieved (also called the sensitivity) and is calculated as the ratio of the number of relevant instances retrieved (TP) to the number of related instances that should be retrieved (which is the sum of the TP and false negatives (FN)). The calculation of the precision and recall of ith class C_i is shown in Equation 5.2 and Equation 5.3, respectively.

$$Precision_i = \frac{TP_i}{TP_i + FP_i} = \frac{TP_i}{TP_i + \sum_{j \neq i}^{n} E_{ji}} \tag{5.2}$$

$$Recall_i = \frac{TP_i}{TP_i + FN_i} = \frac{TP_i}{TP_i + \sum_{i \neq j}^{n} E_{ij}} \tag{5.3}$$

Finally, the average precision and recall for the multiclass problem can be obtained by taking the corresponding average value over all categories, as shown in Equation 5.4 and Equation 5.5, respectively:

$$Precision = \frac{\sum_{i=1}^{n} Precision_i}{n} \tag{5.4}$$

$$Recall = \frac{\sum_{i=1}^{n} Recall_i}{n} \tag{5.5}$$

The F1-Score, used to comprehensively reflect the overall performance of the model combining these two indicators (precision and recall), is calculated as shown in Equation 5.6.

$$F1\_Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{5.6}$$

5.4 Offline Experiments

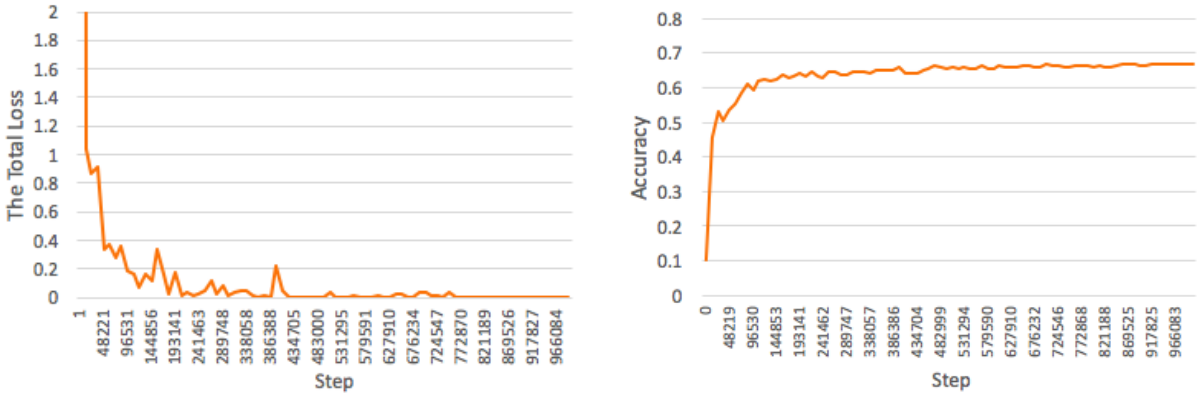
In this section, five kinds of offline experiments are presented to provide comprehensive insight into the performance of the overall proposed framework.

5.4.1 Effects of Two-stage Fine-tuning

First, the first experiment is conducted to show the effectiveness of adopting the two-stage fine-tuning training strategy. We compare the results of two scenarios that involve only fine-tuning from the pretrained weights based on ImageNet and these adopting two-stage fine-tuning using the FER-2013 dataset towards both the CK+ and JAFFE datasets.

As for the first-stage fine-tuning on FER-2013 as based on ImageNet, this approach obtains an accuracy of 67.03% on the public test set and 68.31% on the private test set, which is approximately 2.5% lower than that of the highest-ranked approach [13] in the 2013 challenge on Kaggle but reaches human recognition rate ($65 \pm 5\%$). The total loss (which converges quickly during training) and the accuracy for validation are shown in Figure 5.1.

Based on the best pretrained model from FER-2013, second-stage fine-tuning is then conducted on the CK+ and JAFFE datasets. To verify the effectiveness of this two-



(a) The total loss (b) Accuracy for validation

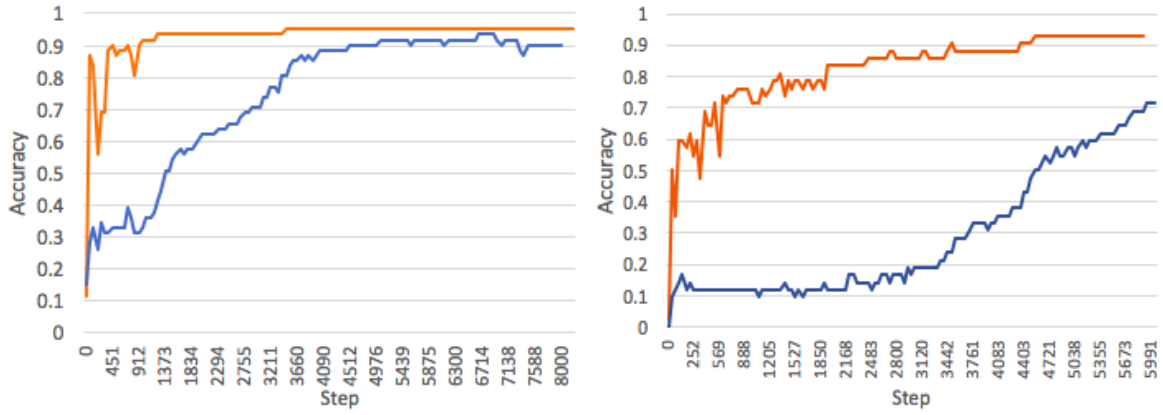
Figure 5.1: The total loss during training and accuracy for validation when fine-tuning the proposed model with FER-2013 based on the pretrained model from ImageNet

stage fine-tuning method, we directly fine-tune the CK+ and JAFFE datasets from the ImageNet for comparison. For the single-stage fine-tuning step, the initial learning rate is set to 0.01 for both datasets. The accuracy of the validation of two datasets for comparing the two situations is shown in Figure 5.2, and the exact accuracy of the two datasets achieved in both cases and the improvement in the accuracy rate are shown in Table 5.3.

Dataset	Single-stage Fine-tuning	Two-stage Fine-tuning	Accuracy Improvement
CK+	$91.80 \pm 1.64\%$	95.08%	**3.28±1.64%**
JAFFE	71.43%	92.86%	**21.43%**

Table 5.3: Illustration of the effectiveness of two-stage fine-tuning. The exact accuracy of the two datasets achieved in both cases and the improvement in the accuracy rate. Note that the accuracy for CK+ takes the six-class situation as the example.

From Figure 5.2, we can see that the adoption of the two-stage fine-tuning strategy can not only improve the accuracy ($3.28 \pm 1.64\%$ increase for CK+ and 21.43% increase for JAFFE) but also achieve much faster convergence than single-stage fine-tuning. The results presented above do not consider the center loss since its effectiveness is elaborated in Section 5.4.2. For CK+, this strategy improves the accuracy to the highest value at approximately 200 epochs, while the accuracy of the model that is directly fine-tuned



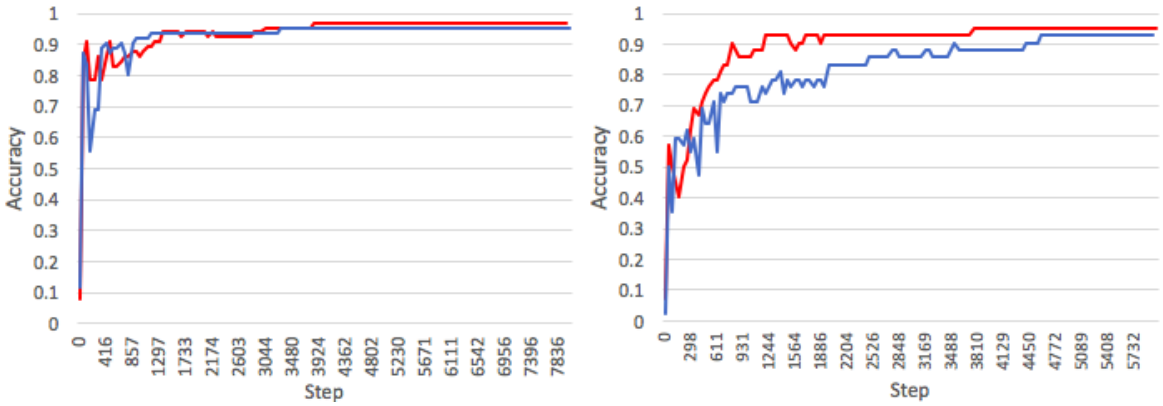
(a) The accuracy evaluated on CK+. (b) The accuracy evaluated on JAFFE.

Figure 5.2: The accuracy of validation on the two datasets for comparing single-stage fine-tuning with two-stage fine-tuning. Note that the orange line represents the accuracy after adopting two-stage fine-tuning and the blue line corresponds to conventional single-stage fine-tuning.

from the ImageNet increases much slower than the former. As a result, the result of 95.08% already reaches an accuracy superior to previous results [98, 37] (even without applying center loss). We take the result of evaluating six basic classes of facial expressions of this dataset (excluding *neutral* and *contempt*) as an example to make the comparison. For JAFFE dataset, the accuracy of 71.43% is improved to 92.86% after applying this strategy, which also surpasses the work of [98, 99]. These results suggest that it is advantageous to boost the performance when training relatively small datasets such as JAFFE and CK+ by leveraging a large dataset such as FER-2013 that lies in the same domain.

5.4.2 Effects of Center Loss

To further enhance the discriminatory power of the proposed framework, the center loss is employed as one part of the supervision signal. After verifying the effectiveness of the supervised fine-tuning strategy, in this section, we elaborate on the superiority of center loss for improving the results. Comparisons are carried out regarding both the JAFFE and CK+ datasets, and λ and α for the center loss are fixed to 0.001 for the



(a) The accuracy evaluated on CK+. (b) The accuracy evaluated on JAFFE.

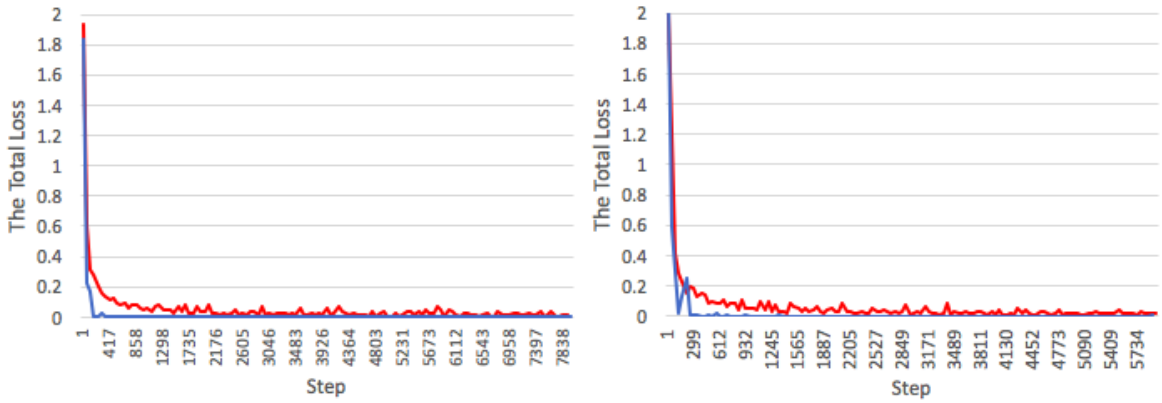
Figure 5.3: The accuracy of validation on the two datasets for comparing two scenarios of adopting the joint supervision and the one using only the softmax loss for supervision. Note that the red line represents the accuracy after adopting joint supervision, and the blue line corresponds to applying only the softmax loss.

CK+ and JAFFE datasets, respectively. The accuracy and the total loss of comparing two scenarios of adopting the joint supervision and the one only using the softmax loss for supervision are shown in Figure 5.3 and 5.4, respectively. The exact accuracy of the two datasets achieved in both cases and the accuracy improvement are reported in Table 5.4.

Dataset	Softmax Loss	Softmax Loss + Center Loss	Accuracy Improvement
CK+	95.08%	96.92%	**1.84%**
JAFFE	92.86%	95.24%	**2.38%**

Table 5.4: An illustration of the effectiveness of joint supervision. Exact accuracy of the two datasets achieved in both cases and the improvement in the accuracy rate. Note that the accuracy for CK+ takes the six-class situation as the example.

Figure 5.3 illustrates that using the center loss as an extra supervision signal can lead to an improvement in accuracy (1.85% for CK+ and 2.38% for JAFFE). All the results presented above involve the supervised fine-tuning method as stated in Section 5.4.1. For CK+, joint supervision begins to show its superiority at approximately 470 epochs, while for JAFFE the accuracy of using joint supervision starts to surpass that only using the softmax loss at approximately 50 epochs. Figure 5.4 shows that with the center loss, the



(a) The total loss evaluated on CK+. (b) The total loss evaluated on JAFFE.

Figure 5.4: The total loss of validation on the two datasets for comparing two scenarios of adopting the joint supervision and the one using only the softmax loss for supervision. Note that the red line represents the total loss after adopting the joint supervision, and the blue line stands for the result of applying only the softmax loss.

total loss converges slightly slower, fluctuates more, and cannot reach the minimum that corresponds to training with only softmax loss. However, this configuration may stimulate the model to continue updating the weights and parameters during backpropagation for better learning of discriminant deep features. As stated above, the performance can be effectively enhanced when combining the center loss with the conventional softmax loss during the training of the CNN for facial expression recognition tasks.

5.4.3 Evaluation on JAFFE Dataset

As mentioned in Section 5.4.1, most of the related work that evaluated on JAFFE dataset utilized conventional machine learning techniques for hand-crafted feature extraction and classification, such as [37, 100, 101]. After preparing the dataset as stated in Section 5.1, data preprocessing is applied to this dataset according to Section 3.1 before feeding the data into the network for training. As shown in Figure 5.3.(b) and Figure 5.4.(b), the training loss starts to converge at approximately 100 epochs and can ultimately reach an accuracy of 95.24% for evaluating seven basic classes in the dataset. The results of comparing our proposed model with the literature are shown in Table 5.5. This model

achieves an average precision of 96.19%, recall of 94.76% and F1-score of 95.47%. The confusion matrix for the validation set of the JAFFE dataset is also presented; as shown in Figure 5.5, the training loss converges quickly, although *disgust* is sometimes confused with *fear*.

Methodology	Validation Accuracy (%)
IVA + HOG + Adaboost & SVM [37]	88.20
LBP + SVM/Adaboost [99]	86.67
Boosted Deep Belief Networks (BDBN) [60]	93
2D-LDA + SVM [102]	94.13
Gaussian Process with Polynomial Kernels & Gaussian RBF [100]	93.43 $\sim$ 95.24
Local Fisher Discriminant Analysis (LFDA) [101]	94.37
Patch-based Gabor Feature + DL2 with SVM [12]	92.93
DCNN + SVM [15]	98.12
Advanced LBP + Tsallis Entropy + NLDA [103]	90.54 (48x48) 94.59 (64x64)
Feature-based Salient Facial Patches [98]	91.80
Proposed Method	**95.24**

Table 5.5: Performance comparison with the state-of-the-art methods on the JAFFE dataset.

	AN	DI	FE	HA	SA	SU	NE
AN	100	0	0	0	0	0	0
DI	0	80	20	0	0	0	0
FE	0	0	100	0	0	0	0
HA	0	0	0	100	0	0	0
SA	0	0	0	13.67	83.33	0	0
SU	0	0	0	0	0	100	0
NE	0	0	0	0	0	0	100

Figure 5.5: The confusion matrix for the validation set of the JAFFE dataset.

Relative to those conventional methods that employ geometric or appearance feature extraction techniques such as those proposed by [37, 99, 12, 103], our proposed framework not only does not require human effort in feature extraction but also can surpass the maximum accuracy levels of prior work. Examples include the 88.2% accuracy of

[37], which used both geometric-based (inter vector angle (IVA)) and appearance-based (HOG) feature extraction and the 92.93% accuracy of [12], which employed the patch-based Gabor feature extraction and dense L2 with SVM for classification. Table 5.5 reveals that although a previous study [15] (which also applied DCNNs) achieved an accuracy of 98.12%, surpassing our result slightly, the time cost for classification was much higher than ours (see Section 5.5).

5.4.4 Evaluation on the CK+ Dataset

The second dataset to be evaluated is the CK+ dataset, on which eight different evaluation configurations are applied. The size of data used for training and validation for eight evaluation configurations are presented in Table 5.6. The number of images in the neutral class is kept the same for both the small-size and large-size datasets containing the onset frame of each image sequence. The accuracy, precision, recall and F1-score values of eight situations evaluating the small-size and large-size datasets are reported in Table 5.7 and 5.8, respectively. The training loss and the confusion matrices for eight situations are shown in Figure 5.6, 5.7, respectively.

	6 Classes		7 Classes				8 Classes	
	Training	Validation	Neutral_excluded		Contempt_excluded		Training	Validation
			Training	Validation	Training	Validation		
Small	248	61	262	65	509	127	524	130
Large	742	185	785	196	1004	250	1047	261

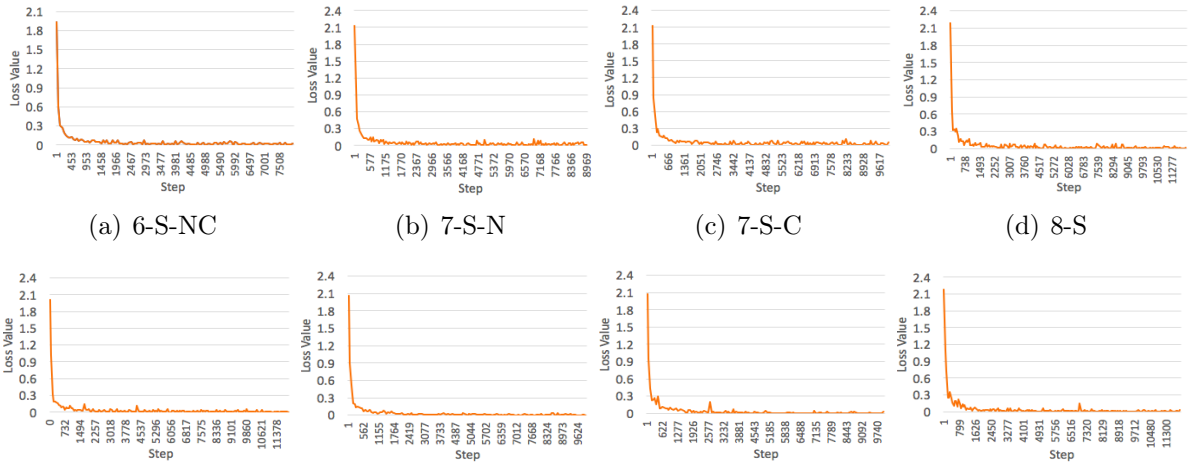
Table 5.6: The size of the data used for training and validation for eight evaluation configurations on the CK+ dataset.

Evaluation Configurations (Small Size)		Accuracy	Precision	Recall	F1-Score
6 Classes		96.92	92.78	95.95	94.34
7 Classes	Neutral_excluded	93.85	93.84	93.51	93.67
	Contempt_excluded	95.38	91.58	91.62	91.60
8 Classes		95.38	94.83	91.38	93.07

Table 5.7: The accuracy, precision, recall and F1-score values (%) of the small-size CK+ dataset with different evaluation configurations.

Evaluation Configurations (Large Size)		Accuracy	Precision	Recall	F1-Score
6 Classes		100	100	100	100
7 Classes	Neutral_excluded	100	100	100	100
	Contempt_excluded	98.80 ± 0.40	99.25	99.21	99.23
8 Classes		99.69 ± 0.31	99.17	99.81	99.49

Table 5.8: The accuracy, precision, recall and F1-score values (%) of the large-size CK+ dataset with different evaluation configurations.

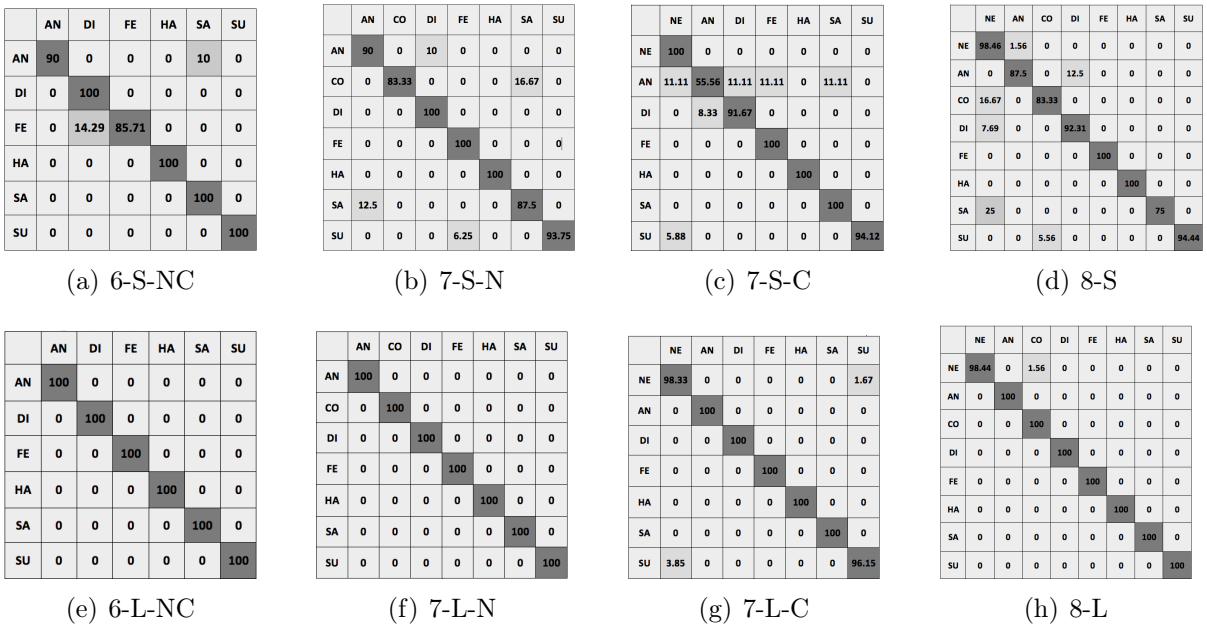


(a) 6-S-NC (b) 7-S-N (c) 7-S-C (d) 8-S

(e) 6-L-NC (f) 7-L-N (g) 7-L-C (h) 8-L

Figure 5.6: The training loss of 8 different configurations for the CK+ dataset. Note that 6, 7, 8: number of classes evaluated; S: small-size dataset (comprising the last frames of the image sequences); L: large-size dataset (comprising the last three frames of the image sequences); NC: *neutral* and *contempt* excluded; C: *contempt* excluded; N: *neutral* excluded.

As shown in Table 5.7 and 5.8, as the number of classes to be evaluated increases, the accuracy drops slightly since the process is more challenging with only a limited amount of data when the target task becomes more complex. For example, the accuracy of the small size 6-class dataset can reach 96.92%, while this value drops by 1.54% in the 8-class situation. Note also that the increase in the size of the dataset can significantly help to boost accuracy when comparing the performance on the small-size datasets (which contain only the apex frame of each image sequence) and the large-size datasets (which contain the last three frames of the image sequences for enlarging the data size). For example, using a large-size 8-class dataset provides an improvement of 4.31% over the accuracy of 95.38% for the corresponding small-size dataset.



(a) 6-S-NC (b) 7-S-N (c) 7-S-C (d) 8-S

(e) 6-L-NC (f) 7-L-N (g) 7-L-C (h) 8-L

Figure 5.7: The confusion matrices of 8 different configurations for evaluating the CK+ dataset. Note that 6, 7, 8: number of classes evaluated; S: small-size dataset (comprising the last frames of the image sequences); L: large-size dataset (comprising the last three frames of the image sequences); NC: *neutral* and *contempt* excluded; C: *contempt* excluded; N: *neutral* excluded.

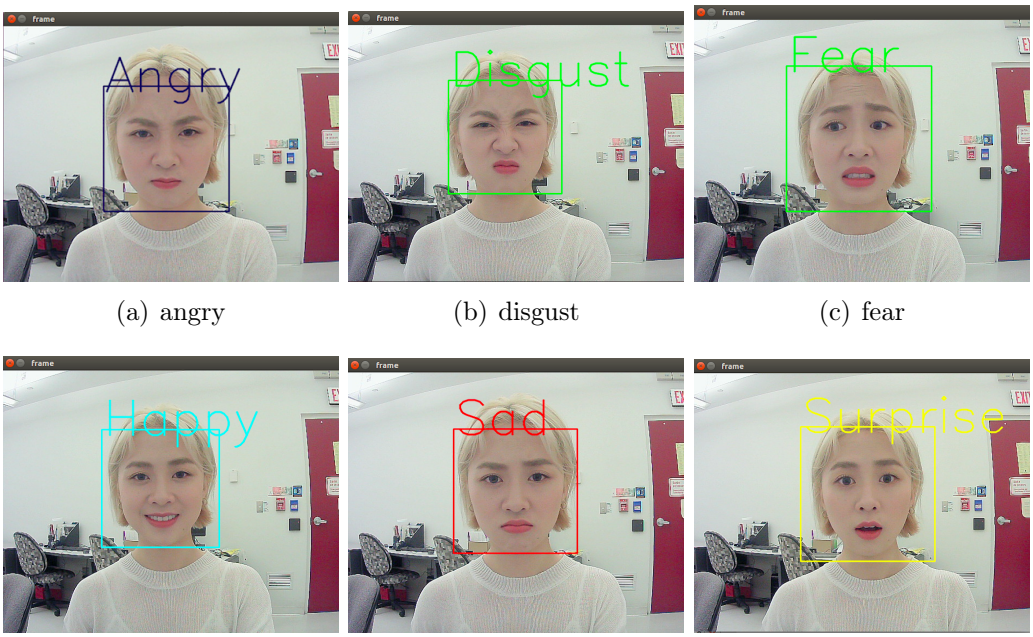
5.5 Real-time Experiment

To verify the ability to run in real time on the proposed system, we also design an implementation for real-time facial expression recognition from a standard webcam. After the webcam is connected to the network, the faces are preprocessed following the same procedures as described in Section 3.1 (without data augmentation). The data after preprocessing are then fed into the selected trained model, which implements the best evaluation result to perform the classification. The subject is asked to face the camera frontally and display one of the basic facial expressions. The computation time for classifying one single frame is evaluated, and the results of comparison with the literature are shown in Table 5.9, indicating that our proposed framework can perform classification (with a run-time of only approximately 3.57 ms/frame on average) much faster than the conventional classifiers such as [99, 60] and the one that also used CNN [15]. This

implementation can subsequently classify the facial expressions of an arbitrary number of faces simultaneously running in real time, even in non-laboratory-controlled conditions. Selected real-time results are presented in Figure 5.8.

Methodology	Classification Time (ms/frame)	System Arrangement
IVA + HOG + Adaboost & SVM [37]	66.7	2.4 GHz CPU with no GPU
LBP + SVM/Adaboost [99]	227	Intel i3 2.2 GHz CPU
Boosted Deep Belief Networks (BDBN) [60]	210	6-core 2.4GHz PC
2D-LDA + SVM [102]	35.7	Pentium IV with 2.80GHz
DCNN + SVM [15]	140 ∼ 145	Tesla K20Xm GPU with compute version 3.5 / CPU (700-900ms)
CNN + OVA Binary Classification [104]	230	Intel Core i7 3.4 GHz with a NVIDA GeForce GTX 660
68 Facial Landmarks + Optical Flow + SVM [36]	83.3	2.6 GHz Intel Core i5 CPU
Proposed Method	**3.57**	**NVIDIA Quadro K4200 GPU / Intel Xeon (R) E5-1603 v3 2.8 GHz * 4 CPU**

Table 5.9: The run-time cost comparison against the state-of-art methods for real-time facial expression recognition.

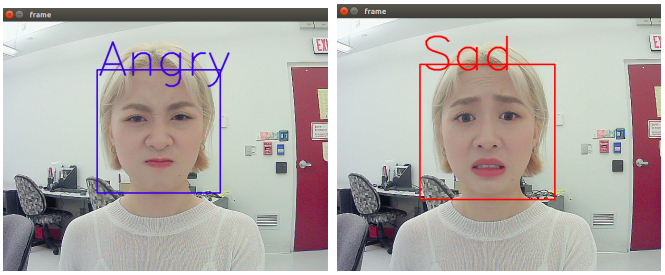


(a) angry (b) disgust (c) fear

(d) happy (e) sad (f) surprise

Figure 5.8: Examples of real-time classification for basic expressions.

Note that this computation time includes only the time cost required for the model to perform the classification (disregarding the preprocessing time for the faces). The OpenCV face detection and the data preprocessing module (e.g., resizing, conversion to grayscale) takes approximately 46.93 ms/frame and 7.49 ms/frame, respectively. Even



(a) disgust but angry (b) fear but sad

Figure 5.9: Real-time failures for classifying basic expressions.

including these preprocessing modules, the run-time of the complete pipeline for facial expression recognition is still suitable for running in real time. Note also that when taking an arbitrary number M of people in a single frame of the camera into account, the total time cost for this frame is $46.93 + M \times (7.49 + 3.57)$ ms. The proposed framework can successfully distinguish basic facial expressions except for occasional confusion between *angry* and *disgust* and between *fear* and *sadness*, as shown in Figure 5.9. It is possible that the subjects who presented these posed expressions are not professional actors; furthermore, these classes tend to be misclassified because of similar geometric and appearance features, which tend to lower even human accuracy when reviewing the results of evaluation on the JAFFE and CK+ datasets.

5.6 Environment Configuration

All the offline (training and testing) and real-time experiments in this work are implemented on an NVIDIA Quadro K4200 GPU based on the lightweight Slim [94] library with TensorFlow [105] backend. We use both OpenCV [106] and the Slim library for all image processing, e.g., random flips and rescaling. The bounding boxes of the ROI of each face for face detection are obtained using the Python wrapper in the OpenCV library.

Chapter 6

Discussion

In this chapter, issues concerning about the failure during face detection are given in Section 6.1, and an analysis of the attained results in Chapter 5 is presented in Section 6.2.

6.1 Face Detection Failure

Face detection is conducted on both the JAFFE and CK+ datasets using the Haar cascade classifier in OpenCV, as described in Section 3.1.1. Some face detection errors (false positive) arose within the CK+ dataset due to interface-related factors during

image recording (for example, the bold font on the image for recording the time when the image was taken). In the CK+ dataset, all of the images that failed in face detection (listed in Table 6.1) were cropped manually by using four vertices of the ROI of faces in an adjacent frame in the same video sequence that can be successfully extracted by the face classifier. Examples of face detection failure and manual cropping of faces are shown in Figure 6.1. For the FER-2013 dataset, no face detection was applied since faces in this dataset had already been automatically registered (see Figure 4.1).

Subject number	Frame of face detection failure
S032	S032_001_00000022.png S032_004_00000001.png S032_006_00000001.png
S034	S034_003_00000026.png
S052	S052_006_00000011.png
S068	S068_002_00000014.png S068_002_00000015.png S068_003_00000001.png S068_003_00000012.png S068_003_00000013.png S068_003_00000014.png
S077	S077_001_00000026.png
S099	S099_004_00000001.png

Table 6.1: The serial numbers of face detection failure in the CK+ dataset.

(a) S068_002_00000015 (b) S077_001_00000026

Figure 6.1: Two examples of face detection failure in CK+. The left image in each example is the one in which face detection failed, and the right one is the manually cropped face.

6.2 Problem Analysis

When evaluating the effectiveness of the two-stage fine-tuning strategy, we find that the JAFFE dataset achieves much lower accuracy (only 71.43%) than other reported methods [60, 102, 12] when fine-tuned only from ImageNet, and the accuracy does not rise until approximately 370 epochs. Moreover, there is hardly any related work that directly used a CNN to fine-tune this database. The reason may be that it is challenging to have the deep model to learn and extract features (in the domain of FER) directly based on the pretrained model obtained in the domain of the object classification (i.e., the target task is much different from the source task), particularly when the size of the target dataset (JAFFE) is very small and the interclass variation of which is nuanced (the subjects are all Asian and the facial expressions of which are very subtle). However, after using an additional auxiliary FER-2013 dataset in the same domain (facial expression) containing an enormous amount of data, the accuracy starts to rise much quicker and can achieve results superior to those of state-of-the-art methods [101, 103].

From the confusion matrix reported in the Figure 5.5, we notice that it is weird that *sadness* is even mixed up with the *happy*. When we look into the results, it is eventually because of the mislabeled one in the dataset as shown in Figure 6.2. After reviewing other related work, we do not find any of these work mentioned about removing this mislabeled data. Thus, we just keep it for fairness in the comparison with the literature. We also find that *disgust* is sometimes confused with *fear*, which may be due to some of the facial expressions in these two classes having similar features (for example, the rise of the eyebrows) as shown in Figure 6.3, especially when the difference between classes is less salient, as in the JAFFE dataset.

In addition, when reviewing the literature on CK+, we find that there is no consistent evaluation configuration with this dataset, as shown in Table 6.2. First, the number of classes evaluated in the related work varies from six to eight since there is an additional

Figure 6.2: Example of a mislabeled image in the JAFFE dataset. This image should have been labeled as *happy* but is labeled as *sadness* instead which eventually leads to the misclassification in the results as shown in the confusion matrix in the Figure 5.5.

(a) Disgust (b) Fear

Figure 6.3: Example of *disgust* and *fear* in the JAFFE dataset in which the difference is quite small.

contempt class in the CK+ dataset to supplement the seven basic facial expressions. Moreover, even with the seven classes, the evaluations differ in whether the neutral class or contempt class is excluded. Since the CK+ dataset contains 327 image sequences from neutral to peak emotion, another problematic aspect involves the size of the dataset in that some of the work selects only the last frame of each image sequence [107, 14, 37] comprising the dataset for training and testing, while other methods choose the last three frames [90, 60, 104] for data augmentation. Consequently, it is challenging to make a fair comparison on this dataset using the literature. To address this issue, eight different evaluation configurations for the CK+ dataset are carried out, providing a baseline for a much more comprehensive evaluation comparison with the related work.

In addition, as shown in Table 5.7, evaluating the small-size 7-class neutral-excluded dataset is more challenging than the contempt-excluded one (the accuracy of the former is only 93.85%, which is 1.53% lower than the latter, and the same situation arises with the average precision, recall and F1-score). This issue occurs because the number of contempt class is only 18 (only considering the last frame of the image sequences), which is much

Methodology	Evaluation Configuration	Validation Accuracy (%)
AAM + POSIT & LBP + NN Classifier [107]	7 classes (neutral excluded) Last frame	88.00
Hybrid CNN + SIFT Aggregator [14]	6 classes (neutral & contempt excluded) Last frame	99.10
Proposed CNN Model [90]	7 classes (neutral excluded) Last three frames	99.60
IVA + HOG + Adaboost & SVM [37]	6 classes (neutral & contempt excluded) Last frame	88.20
Zero-bias CNN + AD [108]	6 classes (neutral & contempt excluded) Last frame	98.30
	8 classes Last frame	96.40
Boosted Deep Belief Network (BDBN) [60]	7 classes (neutral excluded) Last three frames	96.70
CNN + One-verse-all (OVA) Binary Classification [104]	6 classes (neutral & contempt excluded) Last three frames	97.81
68 Facial Landmarks + Optical Flow + SVM [36]	7 classes (contempt excluded) Last frame	98.12
DCNN + SVM [15]	6 classes (neutral & contempt excluded) Last frame	97.08

Table 6.2: A review of the evaluation setups of the state-of-art methods on the CK+ dataset. These evaluation configurations vary in the number of classes (from 6 to 8), the specific classes and the size of the dataset (small or large) to be evaluated.

smaller than that of the neutral class (327), as illustrated in Table 4.1. Therefore, the data imbalance is the most significant factor affecting the performance on small-size datasets when considering the neutral and contempt classes. Moreover, confusion sometimes occurs among the angry, disgust and fear expressions in small datasets due to the similar geometric and appearance features and the limited number of training samples. The training difficulty for the CNN with small and unbalanced data, particularly in classes with nuanced differences, is illustrated by these examples.

Since most of these facial expression datasets are collected under a controlled laboratory environment, the comparison of classifiers across datasets becomes more reliable than traditional self-classification determining the capability of generalization of classifiers. In this work, we verify this ability of the proposed method by first training with the entire JAFFE dataset (6-class excluding the *neutral* class) and evaluating on the CK+ dataset (6-class excluding the *contempt* and *neutral* classes), and vice versa. Following the same data preprocessing procedure described in Section 3.1.2, evaluating the CK+ dataset when training with the JAFFE dataset achieves an accuracy of 72.49%,

Training	Testing	Proposed Method (%)
CK+	JAFFE	50.27
JAFFE	CK+	72.49

Table 6.3: The performance (accuracy) of the cross-dataset validation.

	AN	DI	FE	HA	SA	SU
AN	16.67	3.33	3.33	0	60.00	16.67
DI	0	30.00	16.67	0	36.67	16.67
FE	0	0	25.81	0	38.71	35.48
HA	0	0	9.68	64.52	3.22	22.58
SA	0	0	0	0	77.42	22.58
SU	0	0	0	3.33	10.00	86.67

(a) CK+_JAFFE

	AN	DI	FE	HA	SA	SU
AN	66.67	17.78	2.22	0	13.33	0
DI	32.20	66.10	1.70	0	0	0
FE	0	44.00	36.00	4.00	8.00	8.00
HA	0	10.14	0	88.41	1.45	0
SA	14.29	21.42	14.29	0	50.00	0
SU	0	3.57	8.33	2.38	0	84.52

(b) JAFFE_CK+

Figure 6.4: The two confusion matrices of the cross-dataset validation. Note that CK+_JAFFE: training with the CK+ and evaluating the JAFFE; JAFFE_CK+: training with the JAFFE and evaluating the CK+.

while evaluating the JAFFE dataset when training with the CK+ dataset achieves an accuracy of 50.27%. The performance of the cross-dataset validation towards these two datasets and the two confusion matrices are shown in Table 6.3 and Figure 6.4, respectively. Though Kumar et al. [36] also did the cross-dataset experiment on the CK+ and Shan et al. [56] did the cross-dataset experiment on the JAFFE, we still cannot do the comparison with their results since the deviation of the experimental method (where [36] trained with the Radboud faces Database (RafD) and tested on the CK+ and [56] trained with the Cohn-Kanade (CK) dataset and tested on the JAFFE). From Figure 6.4, we can also see that although two confusion matrices are both scattered, cross-dataset evaluation on the JAFFE dataset is more challenging than on the CK+ dataset due to the characteristics of the database itself. It may difficult to apply the CNN to classify a different dataset, especially when the dataset is race-related (the subjects of which are all Asians) and the interclass variation is not as dispersed as the source dataset used for training. This factor may be one of the limitations of the CNN, which can be described as 'dataset-sensitive' for classification.

Chapter 7

Conclusions

To meet the goal of intelligent human-computer interaction (HCI) and accomplish the subtask of the digital twin vision, a CNN-based system of estimating basic facial expressions is proposed in this work, achieving state-of-the-art results and running in real time with a webcam in an image-based manner. A newly proposed CNN model, MobileNet V1, is applied in our facial expression recognition task because of its ability to provide both speed and accuracy. CNN training always requires a significant amount of data, but most of the existing facial expression datasets cannot satisfy this criterion, which makes it challenging to train a deep CNN with insufficient data. The adoption of

supervised two-stage transfer learning with ImageNet and an auxiliary dataset (with a larger dataset such as FER-2013) solves the problem of small datasets (such as JAFFE and CK+) in CNN training to a large extent. These conditions also provide insight into related work when training a CNN with a limited amount of data. Moreover, to enhance the discriminability of the framework, an additional new supervision signal, center loss, together with the softmax loss is leveraged as a joint supervision signal for optimization during CNN training. The advantage of using the center loss over the triplet loss [27] and the contrastive loss [109] during the training scheme is that this configuration prevents the construction process of selecting the complex and indistinct sample pairs that can be easily implemented (and require only configuration at the feature output layer).

After verifying the effectiveness of applying the two-stage fine-tuning strategy and the center loss, self-evaluation was conducted on two publicly available datasets, JAFFE and CK+. In addition to the accuracy, the average precision, recall F1-score, and confusion matrix are determined in the evaluation. Since no consistent evaluation configuration is found for CK+ during the literature review, we provide an evaluation baseline for this dataset, including eight situations considering various classes and the size of the dataset to be evaluated. Overall, the performance of our proposed method on both JAFFE and CK+ obtains results superior to state-of-the-art method. The results illustrate that JAFFE is more challenging than CK+ since the facial expressions in this dataset are relatively subtle, with minimal interclass variation. Moreover, the results emphasize the importance of the dataset size in CNN training. To verify the ability of generalization of the proposed method, cross-dataset evaluation is also carried out between these two datasets after the self-evaluation.

Finally, a real-time system for recognizing facial expressions of several subjects simultaneously from the webcam is designed and implemented using the best-trained model to demonstrate its capability of running in real time. Relative to previous results, the proposed framework can perform classification much faster than conventional classifiers

or even similar CNN-based work as a result of the characteristics of MobileNet and the GPU system. Overall, it can be seen that this CNN-based framework for facial expression recognition is superior to conventional machine learning methods in that it eliminates much human effort for complex feature extraction and does not require extensive preprocessing procedures while obtaining state-of-the-art results. Although some reported studies outperform our accuracy, those methods either do not provide real-time implementation or incur a much higher run-time cost than our approach.

Limitations and Future Work

Putting aside what we have successfully achieved, several critical issues can be addressed for further improvements. First, without considering the influence of head pose variations, only frontal faces are taken for training and real-time implementation without face registration. It may be helpful in improving the accuracy to take registered faces from several views of the camera into account. In addition, the recognition capabilities of our system are still limited for exaggerated expressions rather than spontaneous ones due to the training datasets used. The field still lacks facial expression datasets containing high-quality spontaneous expressions, so this kind of customized dataset is needed. Further exploration of this problem may incorporate the micro-expression research field. Since this work is still carried out in an single-frame-based way, the spatial information of video sequences might also be considered to enhance the system further.

Acronyms

AAM Active Appearance Model. 22

AI Artificial Intelligence. 2

ANN Artificial Neural Network. 14

API Application Programming Interface. 40

AR Augmented Reality. 2

AU Action Units. 9, 40

BN Bayesian Networks. 13

CNN Convolutional Neural Network. ii, 3, 5, 6, 14–17, 19, 20, 24, 27, 29, 30, 32, 42, 43, 54, 62–65

DCNNs Deep Convolutional Neural Networks. 5, 52

DPMs Deformable Part Models. 14

FACS Facial Action Coding System. 9

FER Facial Expression Recognition. iv, v, 2–5, 9, 12–14, 20, 23–25, 27, 30, 33, 40, 41, 59

FN False Negative. 45

FP False Positive. 45

GPU Graphical Processing Unit. 14, 19, 56

HCI Human-Computer Interaction. 2, 3, 8, 63

HOG Histograms of Oriented Gradients. 12, 13, 23, 52

ILSVRC ImageNet Large Scale Visual Recognition Challenge. 19

IoT Internet of Thing. 2

IVA Inter Vector Angle. 52

KCCA Kernel Canonical Correlation Analysis. 12

LBP local binary pattern. 11–14, 23

LDA Linear Discriminant Analysis. 13

LG Labeled Graph. 12

MLP Multi-Layer Perceptron. 17

NN Neural Networks. 13

RBF Radial Basis Function. 12

ReLU Rectified Linear Unit. 16, 19

ROI Region of Interest. 27

SGD Stochastic Gradient Descent. 36, 43

SIFT Scale-Invariant Feature Transform. 12

SVM Support Vector Machine. 12, 13, 19, 21, 52

TP True Positive. 44, 45

VR Virtual Reality. 2

References

[1] Abdulmotaleb El Saddik. "Digital Twins: The Convergence of Multimedia Technologies". In: *IEEE MultiMedia* 25.2 (2018), pp. 87–92.

[2] SL Happy et al. "A real time facial expression classification system using local binary patterns". In: *Proceedings of the 2012 4th International Conference on Intelligent Human Computer Interaction (IHCI)*. IEEE. 2012, pp. 1–5.

[3] Maja Pantic et al. "Automatic analysis of facial expressions: The state of the art". In: *IEEE Transactions on pattern analysis and machine intelligence* 22.12 (2000), pp. 1424–1445.

[4] Beat Fasel et al. "Automatic facial expression analysis: a survey". In: *Pattern recognition* 36.1 (2003), pp. 259–275.

[5] Anil K Jain et al. *Handbook of face recognition*. Springer, 2011.

[6] Charles Darwin et al. *The expression of the emotions in man and animals*. Oxford University Press, USA, 1998.

[7] Winrich A Freiwald et al. "A face feature space in the macaque temporal lobe". In: *Nature neuroscience* 12.9 (2009), p. 1187.

[8] Paul Ekman et al. *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, USA, 1997.

[9] Di Huang et al. "Local binary patterns and its application to facial image analysis: a survey". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 41.6 (2011), pp. 765–781.

[10] Katharina Morik et al. *Combining statistical learning with a knowledge-based approach: a case study in intensive care monitoring*. Tech. rep. Technical Report, SFB 475: Komplexitätsreduktion in Multivariaten Datenstrukturen, Universität Dortmund, 1999.

[11] Stefan Kluckner et al. "A 3D teacher for car detection in aerial images". In: *Proceedings of the IEEE 11th International Conference on Computer Vision, 2007*. IEEE. 2007, pp. 1–8.

[12] Ligang Zhang et al. "Facial expression recognition using facial movement features". In: *IEEE Transactions on Affective Computing* 2.4 (2011), pp. 219–229.

[13] Yichuan Tang. "Deep learning using linear support vector machines". In: *arXiv preprint arXiv:1306.0239* (2013).

[14] Mundher Al-Shabi et al. "Facial Expression Recognition Using a Hybrid CNN-SIFT Aggregator". In: *arXiv preprint arXiv:1608.02833* (2016).

[15] Veena Mayya et al. "Automatic Facial Expression Recognition Using DCNN". In: *Procedia Computer Science* 93 (2016), pp. 453–461.

[16] Patrick Lucey et al. "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression". In: *Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. 2010, pp. 94–101.

[17] Michael Lyons et al. “Coding Facial Expressions with Gabor Wavelets”. In: *Proceedings of the 3rd. International Conference on Face & Gesture Recognition*. 1998, pp. 200–205.

[18] Maja Pantic et al. “Web-based database for facial expression analysis”. In: *Proceedings of the 2005 IEEE international conference on multimedia and Expo*. 2005, p. 5.

[19] Oliver Langner et al. “Presentation and validation of the Radboud Faces Database”. In: *Cognition and emotion* 24.8 (2010), pp. 1377–1388.

[20] Guoying Zhao et al. “Facial expression recognition from near-infrared videos”. In: *Image and Vision Computing* 29.9 (2011), pp. 607–619.

[21] Deepali Aneja et al. “Modeling Stylized Character Expressions via Deep Learning”. In: *Proceedings of the Asian Conference on Computer Vision*. 2016, pp. 136–153.

[22] Paul Ekman. “Darwin, deception, and facial expression”. In: *Annals of the New York Academy of Sciences* 1000.1 (2003), pp. 205–221.

[23] Tomas Pfister et al. “Recognising spontaneous facial micro-expressions”. In: *Proceedings of the 2011 IEEE International Conference on Computer Vision*. 2011, pp. 1449–1456.

[24] Wen-Jing Yan et al. “CASME database: a dataset of spontaneous micro-expressions collected from neutralized faces”. In: *Proceedings of the 2013 10th ieee international conference and workshops on Automatic face and gesture recognition (fg)*. 2013, pp. 1–7.

[25] Hasimah Ali et al. “Facial emotion recognition under partial occlusion using Empirical Mode Decomposition”. In: *Proceedings of the 2016 2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA)*. 2016, pp. 1–6.

[26] Irene Kotsia et al. “An analysis of facial expression recognition under partial facial image occlusion”. In: *Image and Vision Computing* 26.7 (2008), pp. 1052–1067.

[27] Florian Schroff et al. “Facenet: A unified embedding for face recognition and clustering”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 815–823.

[28] André Teixeira Lopes et al. “Facial expression recognition with convolutional neural networks: coping with few data and the training sample order”. In: *Pattern Recognition* 61 (2017), pp. 610–628.

[29] Andrew G Howard et al. “Mobilenets: Efficient convolutional neural networks for mobile vision applications”. In: *arXiv preprint arXiv:1704.04861* (2017).

[30] Alex Krizhevsky et al. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. 2012, pp. 1097–1105.

[31] Karen Simonyan et al. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[32] Yandong Wen et al. “A discriminative feature learning approach for deep face recognition”. In: *Proceedings of the European Conference on Computer Vision*. 2016, pp. 499–515.

[33] Zhihong Zeng et al. “A survey of affect recognition methods: Audio, visual, and spontaneous expressions”. In: *IEEE transactions on pattern analysis and machine intelligence* 31.1 (2009), pp. 39–58.

[34] Brais Martinez et al. “Advances, challenges, and opportunities in automatic facial expression recognition”. In: *Advances in Face Detection and Facial Image Analysis*. Springer, 2016, pp. 63–100.

[35] Albert Mehrabian. “Communication without words”. In: *Communication theory* (2008), pp. 193–200.

[36] Pranav Kumar et al. "A real-time robust facial expression recognition system using HOG features". In: *Proceedings of the International Conference on Computing, Analytics and Security Trends*. 2016, pp. 289–293.

[37] Rahul Islam et al. "SenTion: A framework for Sensing Facial Expressions". In: *arXiv preprint arXiv:1608.04489* (2016).

[38] Marian S Bartlett et al. *Automatic analysis of spontaneous facial behavior: A final project report*. Tech. rep. Technical Report UCSD MPLab TR 2001.08, University of California, San Diego, 2001.

[39] Irfan A. Essa et al. "Coding, analysis, interpretation, and recognition of facial expressions". In: *IEEE transactions on pattern analysis and machine intelligence* 19.7 (1997), pp. 757–763.

[40] Evangelos Sariyanidi et al. "Automatic analysis of facial affect: A survey of registration, representation, and recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 37.6 (2015), pp. 1113–1133.

[41] Paul Ekman et al. *Unmasking the face: A guide to recognizing emotions from facial cues*. 1975.

[42] E Friesen et al. "Facial action coding system: a technique for the measurement of facial movement". In: *Palo Alto* (1978).

[43] Paul Viola et al. "Robust real-time face detection". In: *International journal of computer vision* 57.2 (2004), pp. 137–154.

[44] Constantine P Papageorgiou et al. "A general framework for object detection". In: *Proceedings of the sixth international conference on Computer vision*. 1998, pp. 555–562.

[45] Hai Tao et al. "Explanation-based facial motion tracking using a piecewise bezier volume deformation model". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 1. 1999, pp. 611–617.

[46] Jing Xiao et al. "Robust full-motion recovery of head by dynamic templates and re-registration techniques". In: *International Journal of Imaging Systems and Technology* 13.1 (2003), pp. 85–94.

[47] Jeffrey F Cohn et al. "Detecting depression from facial actions and vocal prosody". In: *Proceedings of the 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*. 2009, pp. 1–7.

[48] Zhen Wen et al. "Capturing subtle facial motions in 3d face tracking". In: *Proceedings of the Ninth IEEE International Conference on Computer Vision*. 2003, pp. 1343–1350.

[49] Y-I Tian et al. "Recognizing action units for facial expression analysis". In: *IEEE Transactions on pattern analysis and machine intelligence* 23.2 (2001), pp. 97–115.

[50] Hong-Bo Deng et al. "A new facial expression recognition method based on local gabor filter bank and pca plus lda". In: *International Journal of Information Technology* 11.11 (2005), pp. 86–96.

[51] Michael Lyons et al. "Automatic classification of single facial images". In: *IEEE Transactions on pattern analysis and machine intelligence* 21.12 (1999), pp. 1357–1362.

[52] Shu Liao et al. "Facial expression recognition using advanced local binary patterns, tsallis entropies and global appearance features". In: *Proceedings of the 2006 IEEE International Conference on Image Processing*. 2006, pp. 665–668.

[53] Timo Ojala et al. "A comparative study of texture measures with classification based on featured distributions". In: *Pattern recognition* 29.1 (1996), pp. 51–59.

[54] Zhengyou Zhang et al. "Comparison Between Geometry-Based and Gabor-Wavelets-Based Facial Expression Recognition Using Multi-Layer Perceptron". In: *Proceed-*

ings of the 3rd. International Conference on Face & Gesture Recognition. 1998, pp. 454–459.

[55] Taskeed Jabid, Md Kabir, Oksam Chae, et al. "Robust facial expression recognition based on local directional pattern". In: *ETRI journal* 32.5 (2010), pp. 784–794.

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

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

[58] Vahid Kazemi et al. "One millisecond face alignment with an ensemble of regression trees". In: *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1867–1874.

[59] David Fleet et al. "Optical flow estimation". In: *Handbook of mathematical models in computer vision*. Springer, 2006, pp. 237–257.

[60] Ping Liu et al. "Facial Expression Recognition via a Boosted Deep Belief Network". In: *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1805–1812.

[61] Marian Stewart Bartlett et al. "Recognizing facial expression: machine learning and application to spontaneous behavior". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 2. 2005, pp. 568–573.

[62] Ira Cohen et al. "Facial expression recognition from video sequences: temporal and static modeling". In: *Computer Vision and image understanding* 91.1-2 (2003), pp. 160–187.

[63] Maja Pantic et al. “Expert system for automatic analysis of facial expressions”. In: *Image and Vision Computing* 18.11 (2000), pp. 881–905.

[64] Maja Pantic et al. “Facial action recognition for facial expression analysis from static face images”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 34.3 (2004), pp. 1449–1461.

[65] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[66] Haoxiang Li et al. “A convolutional neural network cascade for face detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 5325–5334.

[67] Shuo Yang et al. “From facial parts responses to face detection: A deep learning approach”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 3676–3684.

[68] Olexa Bilaniuk et al. “Fast LBP face detection on low-power SIMD architectures”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2014, pp. 616–622.

[69] Rajeev Ranjan et al. “A deep pyramid deformable part model for face detection”. In: *Proceedings of the 2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. 2015, pp. 1–8.

[70] Ross Girshick et al. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[71] Ken Chatfield et al. “Return of the devil in the details: Delving deep into convolutional nets”. In: *arXiv preprint arXiv:1405.3531* (2014).

[72] Maxime Oquab et al. “Learning and transferring mid-level image representations using convolutional neural networks”. In: *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1717–1724.

[73] Matt W Gardner et al. “Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences”. In: *Atmospheric environment* 32.14-15 (1998), pp. 2627–2636.

[74] Min Lin et al. “Network in network”. In: *arXiv preprint arXiv:1312.4400* (2013).

[75] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[76] Forrest N Iandola et al. “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size”. In: *arXiv preprint arXiv:1602.07360* (2016).

[77] David E Rumelhart et al. “Learning representations by back-propagating errors”. In: *nature* 323.6088 (1986), p. 533.

[78] Bo-Kyeong Kim et al. “Hierarchical committee of deep convolutional neural networks for robust facial expression recognition”. In: *Journal on Multimodal User Interfaces* 10.2 (2016), pp. 173–189.

[79] Jieru Wang et al. “Facial Expression Recognition with Multi-scale Convolution Neural Network”. In: *Proceedings of the Pacific Rim Conference on Multimedia*. 2016, pp. 376–385.

[80] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255.

[81] Christiane Fellbaum. *WordNet*. Wiley Online Library, 1998.

[82] Hoo-Chang Shin et al. “Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning”. In: *IEEE transactions on medical imaging* 35.5 (2016), pp. 1285–1298.

[83] Nima Tajbakhsh et al. “Convolutional neural networks for medical image analysis: Full training or fine tuning?” In: *IEEE transactions on medical imaging* 35.5 (2016), pp. 1299–1312.

[84] Minyoung Huh et al. “What makes ImageNet good for transfer learning?” In: *arXiv preprint arXiv:1608.08614* (2016).

[85] Yaniv Bar et al. “Chest pathology detection using deep learning with non-medical training”. In: *Proceedings of the 2015 IEEE 12th International Symposium on Biomedical Imaging*. 2015, pp. 294–297.

[86] Karen Simonyan et al. “Two-stream convolutional networks for action recognition in videos”. In: *Advances in neural information processing systems*. 2014, pp. 568–576.

[87] Zhiding Yu et al. “Image based static facial expression recognition with multiple deep network learning”. In: *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. 2015, pp. 435–442.

[88] Bo-Kyeong Kim et al. “Hierarchical committee of deep convolutional neural networks for robust facial expression recognition”. In: *Journal on Multimodal User Interfaces* 10.2 (2016), pp. 173–189.

[89] Hong-Wei Ng et al. “Deep Learning for Emotion Recognition on Small Datasets Using Transfer Learning”. In: *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. 2015, pp. 443–449.

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

[91] Ian J Goodfellow et al. "Challenges in representation learning: A report on three machine learning contests". In: *Proceedings of International Conference on Neural Information Processing*. 2013, pp. 117–124.

[92] Christian Szegedy et al. "Going deeper with convolutions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 1–9.

[93] Ali Mollahosseini et al. "Going deeper in facial expression recognition using deep neural networks". In: *Proceedings of the 2016 IEEE Winter Conference on Applications of Computer Vision*. 2016, pp. 1–10.

[94] N. Silberman S. Guadarrama. *TensorFlow-Slim: a lightweight library for defining, training and evaluating complex models in TensorFlow*. 2016. URL: `https://github.com/tensorflow/tensorflow/tree/master/tensorflow/contrib/slim`.

[95] Sinno Jialin Pan et al. "A survey on transfer learning". In: *IEEE Transactions on knowledge and data engineering* 22.10 (2010), pp. 1345–1359.

[96] George E Dahl et al. "Improving deep neural networks for LVCSR using rectified linear units and dropout". In: *Proceedings of 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2013, pp. 8609–8613.

[97] Christopher Pramerdorfer and Martin Kampel. "Facial Expression Recognition using Convolutional Neural Networks: State of the Art". In: *arXiv preprint arXiv:1612.02903* (2016).

[98] SL Happy et al. "Automatic facial expression recognition using features of salient facial patches". In: *IEEE transactions on Affective Computing* 6.1 (2015), pp. 1–12.

[99] Rohit Verma et al. "Fast facial expression recognition based on local binary patterns". In: *Proceedings of the 2013 26th Annual IEEE Canadian Conference on Electrical and Computer Engineering*. 2013, pp. 1–4.

[100] Fei Cheng et al. “Facial expression recognition in JAFFE dataset based on Gaussian process classification”. In: *IEEE Transactions on Neural Networks* 21.10 (2010), pp. 1685–1690.

[101] Yogachandran Rahulamathavan et al. “Facial expression recognition in the encrypted domain based on local fisher discriminant analysis”. In: *IEEE Transactions on Affective Computing* 4.1 (2013), pp. 83–92.

[102] Frank Y Shih et al. “Performance comparisons of facial expression recognition in JAFFE database”. In: *International Journal of Pattern Recognition and Artificial Intelligence* 22.03 (2008), pp. 445–459.

[103] Shu Liao et al. “Facial expression recognition using advanced local binary patterns, tsallis entropies and global appearance features”. In: *Proceedings of 2006 IEEE International Conference on Image Processing*. 2006, pp. 665–668.

[104] André Teixeira Lopes et al. “A Facial Expression Recognition System Using Convolutional Networks”. In: *Proceedings of the 2015 28th SIBGRAPI Conference on Graphics, Patterns and Images*. 2015, pp. 273–280.

[105] Martın Abadi et al. “TensorFlow: A System for Large-scale Machine Learning”. In: *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*. 2016, pp. 265–283.

[106] Itseez. *Open Source Computer Vision Library*. `https://github.com/itseez/opencv`. 2015.

[107] Kamlesh Mistry et al. “Intelligent Appearance and shape based facial emotion recognition for a humanoid robot”. In: *Proceedings of the 8th International Conference on Software, Knowledge, Information Management and Applications*. 2014, pp. 1–8.

[108] Pooya Khorrami et al. "Do Deep Neural Networks Learn Facial Action Units When Doing Expression Recognition?" In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2015, pp. 19–27.

[109] Yi Sun et al. "Deep learning face representation by joint identification-verification". In: *Advances in neural information processing systems*. 2014, pp. 1988–1996.