AUTOMATED FACIAL EMOTION RECOGNITION: DEVELOPMENT AND APPLICATION TO HUMAN-ROBOT INTERACTION

by

XIAO LIU

Submitted in partial fulfillment of the requirements

for the degree of Master of Science

Department of Mechanical and Aerospace Engineering

CASE WESTERN RESERVE UNIVERSITY

August, 2019

CASE WESTERN RESERVE UNIVERSITY

SCHOOL OF GRADUATE STUDIES

We hereby approve the thesis of

Xiao Liu

candidate for the degree of **Master of Science***.

Committee Chair

Dr. Kiju Lee

Committee Member

Dr. Kathryn Daltorio

Committee Member

Dr. Frank Merat

Date of Defense

June 3, 2019

*We also certify that written approval has been obtained

for any proprietary material contained therein.

Contents

List of Tables **iv**

List of Figures **vii**

Acknowledgments **xi**

Abstract **xiii**

1 Introduction **1**

1.1 Related Work 3

1.1.1 FER with Neural networks 4

1.1.2 FER with Geometric feature 5

1.2 Research Objectives 6

2 FER using CNN **8**

2.1 Image Pre-processing Filters 8

2.1.1 Training & Test Datasets 9

2.1.2 Adjusting Brightness & Contrast 10

2.1.3 Edge Extraction Filter 10

2.2 Integration with Learning Algorithms 12

2.2.1 Learning Evaluation 14

2.2.2 Parameter Optimization for Pre-Filters 14

2.3 Algorithm Evaluation 17
2.3.1 Learning Results Visualization 17
2.3.2 Accuracy & Efficiency Evaluation 18
2.4 Conclusion 20

3 AU-based FER **21**
3.1 Algorithm 22
3.1.1 Facial Landmark Detection 22
3.1.2 Facial Action Units and Facial Segments 22
3.1.3 Feature Extraction 24
3.1.4 SVM for FER 27
3.2 Parameter Optimization 29
3.2.1 Dataset Preparation 30
3.2.2 Feature Type & Interpolating Order Determination 30
3.2.3 FS Selection 32
3.2.4 SVM Parameter Tuning 34
3.3 Experimental Results 36
3.3.1 Cross-Validation 37
3.3.2 Algorithm Efficiency Improvement 40
3.3.3 Real-time FER Performance 42
3.4 Discussion 46

4 FER for HRI application **48**
4.1 Development of the Robot 48
4.1.1 Mechanical Design 49
4.1.2 Electrical Design 50
4.1.3 Kinematic Analysis of the Arm 51
4.2 Interactive Features 56

4.2.1 Facial Features 56
4.2.2 Gestures 56
4.3 Graphical User Interface 59

5 Conclusion and Future Work **61**

List of Tables

2.1 Efficiency and accuracy comparison 19

2.2 Classification results with the proposed techniques for seven emotions denoted as A (Angry), D (Disgust), F (Fear), H (Happy), S (Sad), Su (Surprise), and N (Neutral) in % 20

2.3 Classification results without pre-filter optimization in % 20

3.1 AUs, FSs with description, and associated landmarks for each FS grouped into five categories. 24

3.2 Analysis of LC and VL features and polynomial interpolation order . 32

3.3 LC-VL Feature Improvements based on AU Combinations for CK+ . 33

3.4 LC-VL Feature Improvements based on AU Combinations for MUG . 33

3.5 CK+ dataset Classification results for seven and six emotions denoted as A (Anger), D (Disgust), F (Fear), H (Happiness), Su (Surprise), S (Sadness) and N (Neutral) with accuracy in % 36

3.6 MUG dataset Classification results for seven and six emotions denoted as A (Anger), D (Disgust), F (Fear), H (Happiness), Su (Surprise), S (Sadness) and N (Neutral) with accuracy in % 36

3.7 Cross validation (CK+ for training and MUG for testing) results for seven and six emotions denoted as A (Anger), D (Disgust), F (Fear), H (Happiness), Su (Surprise), S (Sadness) and N (Neutral) with accuracy in % 38
3.8 Cross validation (MUG for training and CK+ for testing)results for seven and six emotions denoted as A (Anger), D (Disgust), F (Fear), H (Happiness), Su (Surprise), S (Sadness) and N (Neutral) with accuracy in % 39
3.9 Cross validation (CK+/MUG for training and CK+ for testing) results for seven and six emotions denoted as A (Anger), D (Disgust), F (Fear), H (Happiness), Su (Surprise), S (Sadness) and N (Neutral) with accuracy in % 39
3.10 Cross validation (CK+/MUG for training and MUG for testing) results for seven and six emotions denoted as A (Anger), D (Disgust), F (Fear), H (Happiness), Su (Surprise), S (Sadness) and N (Neutral) with accuracy in % 40
3.11 Cross validation (CK+/MUG for training and CK+/MUG for testing) results for seven and six emotions denoted as A (Anger), D (Disgust), F (Fear), H (Happiness), Su (Surprise), S (Sadness) and N (Neutral) with accuracy in % 40
3.12 Selected FSs with description, and associated landmarks for algorithm efficiency improvement. 41
3.13 CK+ dataset Classification results with only keeping left facial features for seven and six emotions denoted as A (Anger), D (Disgust), F (Fear), H (Happiness), Su (Surprise), S (Sadness) and N (Neutral) with accuracy in % 41

3.14 MUG dataset Classification results with only keeping left facial features for seven and six emotions denoted as A (Anger), D (Disgust), F (Fear), H (Happiness), Su (Surprise), S (Sadness) and N (Neutral) with accuracy in % 42
3.15 Comparison with different Number of Selected FSs 43
3.16 Recent FER Approaches Comparison. 46

4.1 D-H parameters for 4-link Woody left arm 51

List of Figures

2.1 Overview of the presented optimization process: 1) use the processed images generated by two pre-filters with varying sets of parameters; 2) analyze the learning outcomes from the CNN; 3) select the best parameter set; and 4) apply the SVM classifier. 9

2.2 Sample images from FER2013 datasets exhibiting the following emotions: angry, disgust, fear, happy, sad, surprise, and neutral. 10

2.3 Implementing BCF to a face image: Gradually increasing brightness and contrast (top) vs. gradually decreasing brightness and contrast (bottom). 11

2.4 Applying BCF with two opposite values of Ω ((a) positive and (b) negative) and then applying EEF with gradually increasing *minVal* and *maxVal* from 20, 50, and 100. 12

2.5 CNN-based learning results shown in the training accuracy ([0,1]) over iterations ([1, 3,500]) using the pre-filters with three different sets: (a) the first parameter set, showing $\mathcal{I} \approx 2,288$; (b) the second data set with $\mathcal{I} \approx 2,101$; (c) and the third parameter set, resulting in $\mathcal{I} \approx 1,168$. . 15

2.6 3D Visualizations of pre-filter parameters after being conducted by CNN and learning evaluation. X, Y and Z axes represent BCF, EEF (2 variable embedded) parameters separately. With [χ_1, χ_2, χ_3] fixed but various Ψ iteration, iterate for 300 times (a), 500 times (b) and supporting vectors in 3D space (c); stochastic iteration of [χ_1, χ_2, χ_3] and Ψ, 3D scatter graph for 5∗100 times iteration with color sequence (d) and supporting vectors in 3D space (e) 18

3.1 Proposed method with 4 steps: (a) landmark detection; (b) Action Unit extraction with landmark; (c) LC and VL feature extraction and improvements; and (d) SVM training with improved parameters. . . . 22

3.2 (a) Landmark detection and Action Units group described with image from CK+ dataset; (b) Landmark detection and Action Units group described with image from MUG dataset; 23

3.3 Feature extraction from given landmarks associated with FS_1 and FS_2: LC feature (κ_i) at landmark (x_i, y_i), and VL feature $\vec{g}_j = (d_j, \theta_j)^T$ at (x_j, y_j). 25

3.4 Overview of algorithm improvements by tuning parameters related to feature extraction and SVM. 29

3.5 (a) Landmark detection with image cropping and re-sizing implemented on CK+ dataset: examples of 7 labeled emotions (i.e., “anger”, “disgust”, “fear”, “happy”, “neutral”, “sadness” and “surprise”) (row 1); after performing landmark detection (row 2); and after cropping and re-sizing (row 3); (b) MUG dataset examples organized the same way as (a). 31

3.6 Optimum weights and penalty parameter for CK+ datasets (a)∼(c) and MUG datasets (d)∼(e). 35

3.7 Cross validation results among CK+, MUG and merged (CK+ & MUG) dataset. 38
3.8 Video clip sample from MUG dataset. 44
3.9 Real-time result on video clips from MUG dataset; (a) before applying FIR filter; (b) after applied FIR filter. 45

4.1 CAD model of Woody (left) and fully assembled hardware prototype (right). 49
4.2 Circuit diagram (block diagram) of the embedded electronics in Woody. 51
4.3 D-H coordinate frame assignment for the manipulators; Note the red frames are auxiliary frame which illustrate the particular Woody's arm installation. (a) woody left arm frame assignment; (b) frame assignment from top view. 52
4.4 Workspace Shown by Forward Kinematics Trajectory of End-effector; (a) side view; (b) front view; (c) top view; (d) axonometric drawing. . 54
4.5 Geometric model of Woody's left arm (point downwards) with base frame {0} and the end-effector frame {5}; (a) geometric model; (b) geometric model seen from the view ① (side view); (c) seen from the view ②. 55
4.6 Woody's head features with varied eyebrow movements. (a)∼(c) show "cat" head feature with eyebrow movements from sadness to anger; (d)∼(f) show "bear" head feature with corresponding eyebrow movements; (g)∼(i) show "rabbit" head feature with corresponding eyebrow movements. 57
4.7 Woody's recognizable gestural cues. (a) shows "wave" gesture; (b) shows "nod" gesture; (c) shows "weep" gesture; (d) shows "excited" gesture. 58
4.8 Woody GUI main menu. 59

4.9 Real-time interaction function interface. Left: window shows real-time FER results; Right: window with generated gesture buttons for interaction. 60

Acknowledgments

I would like to show my gratitude to my advisor Prof. Kiju Lee for giving me the opportunity to work on human-robot interaction (HRI) project at Distributed Intelligence and Robotics Laboratory (dirLAB). She has not only been a professional teacher, but also a humble, reliable, and optimistic mentor to me and many other students. I have learned fundamental concepts of robot manipulation to modern robotic tools implementation from Prof. Lee throughout my graduate study. She has continuously brought state-of-the-art concepts into my research. She did not engage in fancy boasting of the artificial intelligence of robotics realm, but rather focused on fundamental theories. Working with Prof. Lee has been fun, rewarding, and free of pressure. I remember a meeting with Prof. Lee asking about a research position in her lab after my first year at Case Western Reserve University (CWRU). At that time, I was lost and had no confidence about the choice of my major. She said "I was once an international student, too, and I know things can be very hard sometimes." She encouraged me to explore further. With her guidance, I opened my mind to learn new things, embrace the challenges, and make connections with colleges and other faculty members. I appreciated all professional, personal, and emotional supports and guidance from Prof. Lee.

I would like to thank my committee members, Prof. Kathryn Daltorio and Prof. Frank Merat, for agreeing to serve on my committee under such tight timeline. Your time and insightful feedback and advice on my thesis project are greatly appreciated.

I am also thankful of all the help I received from my colleagues, Xiangyi Cheng, who collaborated on the facial emotion recognition project with me; Chuanqi Zheng, a great friend who has supported and provided instructions to my course work and career choices; Yanzhou Wang, who brought interesting discussions on robotics; Daniel Hayosh and Alexander Brandt, experienced CAD designers who built the social robot "Woody" and collaborated with me on algorithm implementations; Tao Liu, a doctoral student with many advanced ideas which helped my research; and many others, including Alan Waterhouse, John Wylie, Matthew Trowbridge, and Yang Liu. These friends and colleagues have made my time in dirLAB and Case Western Reserve University more fun and meaningful.It has been a great two years in dirLAB at CWRU. My thesis work has been shaped by many constructive conversations with Prof. Lee and my co-workers.

I would like to express my love to my dearest community, my friends and family here in Cleveland and also back in China. They have provided unconditional love and support I needed to thrive these three years. They have also made me to realize "Better is the end of a thing than its beginning, and the patient in spirit is better than the proud in spirit".

Automated Facial Emotion Recognition: Development and Application to Human-Robot Interaction

Abstract

by

XIAO LIU

This thesis presents two image processing algorithms for facial emotion recognition (FER). The first method uses two pre-processing filters (pre-filters), i.e., brightness and contrast filter and edge extraction filter, combined with Convolutional Neural Network (CNN) and Support Vector Machine (SVM). By using optimal pre-filter parameters in the pre-processing of the training images, the classification of FER could reach 98.19% accuracy using CNN with 3,500 epochs for 3,589 face images from the FER2013 datasets. The second approach introduces two geometrical facial features based on action units – landmark curvatures and vectorized landmarks. This method first detects facial landmarks and extracts action unit (AU) features. The extracted facial segments based on the action units are classified into five groups and input to a SVM. The presented method show how individual parameters, including detected landmarks, AU group selection, and parameters used in the SVM, can be examined and systematically selected for the optimal performance in FER. The results after parameter optimization showed 98.38% test accuracy with training using 1,479 labeled frames of Cohn-Kanade (CK+) database, and 98.11% test accuracy with training using 1,710 labeled frames of Multimedia Understanding Group (MUG) database for 6-emotion classification. This technique also shows the real-time processing speed of 6.67 frames per second (fps) for images with a 640×480 resolution.

The novelty of the first approach is combining image processing filters with CNN to enhance CNN performance. As for the second approach, it systematically analyzed the effectiveness of proposed geometric features and implemented FER in real-time. The demonstrated algorithms have been applied on human-robot interaction (HRI)

application platform - social robot “Woody” for testing. The presented algorithms have been made publicly available.

Chapter 1

Introduction

Modern technologies have offered exciting new ways to augment user experiences in human-computer interaction (HCI) and human-robot interaction (HRI). However, user engagement and preference towards such interactive technologies is difficult to measure. It is not only related to the interaction time, frequency, and performance, but also highly linked to how much they enjoy the interactions. It is well known that emotions arise from cognitive appraisals and organize adaptive behavioral responses. A recent work pointed out that physiological results of participants are mirrored in the subjective reports provided by the participants [1]; in another word, psycho-physiological techniques for measuring users experience can provide valuable information related to user engagement, satisfaction, and enjoyableness.

Game theory is a set of mathematical tools for modeling interactive decision-making of users [2]. Social emotions involved in game theory can provide useful parameters for setting up constructive game models [3] for human computer/robot interaction. For instance, guilt and anger can be modeled with utility functions that depend on both material and psychological payoffs, and their effect on behavior can be mathematically described by game theory. Emotion in games is not only related to cooperation of players, but also associated with neural activation consistent with

positive or negative affective states. Therefore, objective and accurate assessment of user emotions is important in game design and evaluation.

The capability of understanding a user's emotion through facial expressions in interactive technologies can play a significant role in understanding user experience and establishing long-term engagement between the technology and the user. Facial expression is one of the fundamental social cues, thus often used to understand social deficits or differences in individuals with certain health conditions, such as autism spectrum disorder [4] and Alzheimer's disease [5]. Facial expressions were also reported as one unidimensional tool commonly used to determine the level of pain intensity by visual analogue scales [6].

Automated facial emotion recognition (FER) aims for a computer with a vision system to detect and classify the facial images into a finite number of emotion classes. While common social understanding allows most of us to properly recognize others' emotions through facial expressions, significant individual differences in perceptual and expressive capabilities exist. In addition to individual differences, such vision-based methods are often sensitive to external conditions, such as lighting, distortion, and occlusion, also add complexity to FER [7]. When FER is intended for interactive applications, such as human computer/robot social interaction designed to use FER as user inputs, minimal latency is a key to realize natural, social engagement between the two. Ideally, the processing speed for existing FER systems should at least range from 0.4-8 frames per second (fps) [8]. However, research on interactive machine learning raises one important technical challenge. The requirement for rapid model updates often necessitates trading off accuracy with speed. The resulting models are therefore suboptimal [9].

Some existing software for FER includes AFFDEX software development kit (SDK) [10], EmotioNet [11], CERT [12], and OpenFace 2.0 [13]. AFFDEX SDK provides an easy interface for processing multiple faces within a video or live stream

in real-time with 7 emotions in the speed of 10 fps [10]. EmotioNet was proposed to annotate a "face in the wild" by calculating facial action units' intensity at the speed of 30 fps [11]. Although these methods presented a good real-time application, the testing accuracy of those proposed methods was not publicly available. CERT showed 90.10% accuracy for classifying 7 emotions [12] but lacking in the real-time processing capability. OpenFace 2.0 [13] demonstrated state-of-the-art results in FER task with open source but was still missing testing accuracy and detailed analysis.

In this thesis, two algorithms for automated FER are presented. The first is based on two pre-processing filters (prefilters) (i.e., brightness and contrast filter and edge extraction filter) combined with Convolutional Neural Network (CNN) based learning and classification by a Support Vector Machine (SVM). This method achieved 98.19% accuracy using CNN with 3,500 epochs for given 3,589 face images FER2013 datasets. The second method based on geometric feature extraction and parameter optimization achieved a testing accuracy of 98.38% for static frames of CK+ and 98.11 % for MUG database. It runs at a speed of 6.67 fps with resolution 640×480 pixels for online processing.

1.1 Related Work

More than ten thousand different expressions can be shown on a face and each person has a unique way of expressing their emotions through facial expression [14]. Even people from different backgrounds and cultures share many common expressive features, which can be divided into six emotions: happiness, anger, sadness, disgust, surprise, and fear [14]. Most prior research focuses on accurate classification of these six emotions or seven emotions including "neutral".

1.1.1 FER with Neural networks

Skin color detection, Haar features extraction, Gabor wavelet algorithm, and Local Binary Pattern detection are commonly used techniques for FER [15]. Local Binary Patterns (LBP) not only represent the shape and texture information of the faces but also form an LBP feature vector in an efficient way. The LBP-based algorithm can be combined with other approaches, such as principle component analysis (PCA), to achieve improved performance [16]. Admittedly, advanced algorithms can achieve high recognition rates in some ways, but with the advent of Neural Networks and Deep Learning, facial expression recognition training accuracy can reach up to 98% [17]. Complicated algorithms such as image pre-processing steps are no longer considered efficient for solving facial expression recognition problems.

With recent advancement in machine learning, CNN and Deep Belief Networks (DBN) have been used for feature extraction, classification and recognition tasks. CNN has achieved state-of-the-art results in various applications, including object recognition, face recognition [18], and scene understanding [19]. A multi-path CNN approach integrated with the complementary information from multi-scale perspectives [20] showed a competitive performance when compared with the most modern CNNs on specific datasets, such as the Chalearn Challenge Dataset [21]. Another method based on a novel Multi-Angle Optimal Pattern-based Deep Learning (MAOPDL) method could rectify the problem of sudden illumination changes and find a proper alignment of a feature set by using Multi-Angle-Based optimal configurations [17]. A modern structure of Deep Neural Network includes five major processes: Extended Boundary Background Subtraction (EBBS) [22], Multi-Angle Texture Patter+ Scanning Tunneling Microscopy (STM), Densely Extracted SURF+Local Occupancy Pattern (LOP), Priority Particle Cuckoo Search Optimization (PPCSO) [17] and Long Short-Term Memory Convolutional Neural Network (LSTM-CNN) [23]. The complexity of neural networks results in a high accuracy of training and valida-

tion, requiring repeated tuning of coefficients and modification in parameters within the network as well as outside the network in pre-processing filters. Achieving high accuracy and efficiency at the same time is challenging.

1.1.2 FER with Geometric feature

Performance of automatic FER largely depends on accurate representation of facial features. Facial Action Coding System (FACS) systematically associated facial expressions with observable facial movements in terms of Action Units (AUs), which provided the descriptive requirement for FER [24]. Among 44 AUs specified by FACS, 30 of them are considered anatomically related to specific facial muscles with more than 7,000 different AU combinations [25]. Key geometric points associated with these AUs can be defined and localized for automatic detection of AUs [26].

Geometry-based features focus on describing the shape of the face and its components especially its movements and shape transformation. Classification are typically based on the locations of the facial landmarks and distances among them. Traditional Histogram of Oriented Gradient (HOG) and Scale-Invariant Feature Transform (SIFT) can also extract similar information from a face image, but it is time consuming and not comprehensive [27]. In [28], Salient Facial Patches were extracted from key areas of human faces by marking coarse region of interest (ROI) as geometric features. Another recent algorithm combined the Delaunay triangulation with the Gabor filter for classifying AUs and capturing intensity of each AU [29]. The method used points, lines, and triangle shapes extracted from the facial key points. In this research, selective multi-class AdaBoost combined with the extreme learning machine (ELM) based classification was applied, achieving 95.05% accuracy in 6-emotion classification. Another geometric feature-based technique used 18 critical candidates/landmarks with 16 significant distance data selected by applying correlation-based feature subset selection (CFS) method [30].

Geometry-based features can also be integrated for FER in 3D space. The active appearance model (AAM) is widely used as the geometric feature-based approach. AAM variations are also considered for tracking a dense set of facial key points. In [31], various AAM-based algorithms were compared and evaluated for FER applications. Facial geometric features can also be extracted from 3D faces. Exploiting 3D texture and geometric scattering features between frames of 3D facial geometry sequences is also a dynamic method for FER [32]. In [33], onset and offset segments of 3D sequence of facial expressions were all included, and the geometric facial motion projected in vector field were captured and then trained by GentleBoost classifiers.

Once facial features are extracted, the data are provided to a classification module. Widely used methods include hidden Markov model (HMM), Gaussian mixture models (GMM), dynamic Bayesian networks (DBN), and support vector machine (SVM). HMM is common in handling sequential data and thus also not applicable [29]. GMM is sensitive to noise and cannot be used to model fast variation in consecutive frames [34]. Therefore, it is not suitable for real-time applications. A drawback of DBN is that network structure depends on variable order. If the order is chosen carelessly, the resulting network structure may fail to reveal conditional independencies [35]. SVM is a discriminating classifier which maps a feature vector to a higher dimensional plane. Since SVM can classify static geometrical features and obtain an accurate recognition rate, it is selected for this system.

1.2 Research Objectives

Despite the recent advancement and promising results of FER, there is a lack of open-source, real-time FER system readily available for a broad range of HRI and HCI applications. Three research objectives of this thesis project are:

1. Develop reliable FER algorithms for broad range of HCI and HRI applications.

2. Achieve improved FER performance compared to existing methods in terms of accuracy and speed.

3. Validate the algorithms using two publicly available databases which are commonly used for FER research.

Chapter 2

FER using CNN

The first method for FER uses an image pre-processing technique, systematically integrating two machine learning algorithms, CNN and SVM. As illustrated in Fig. 2.1. The presented method optimizes parameters used in image pre-processing filters based on the learning outcomes from the CNN and thus achieves improved performance in facial image classification.

2.1 Image Pre-processing Filters

The presented algorithm is based on a CNN-SVM classifier, which has been already used for various pattern recognition applications, recently [36, 37, 38]. The difference in our method is in the image pre-processing techniques that tune the parameters used in the pre-filters in order to achieve optimal performance in the CNN-SVM classifier. Face images often involve different lighting distortions and illumination, which can significantly alter the appearance of the faces [39]. To minimize the effect of such external conditions and effectively extract features for emotion recognition, brightness and contrasts are adjusted using the brightness and contrast filter (BCF) and then edges are extracted by using the edge extraction filter (EEF) prior to the CNN-based learning for pattern recognition.

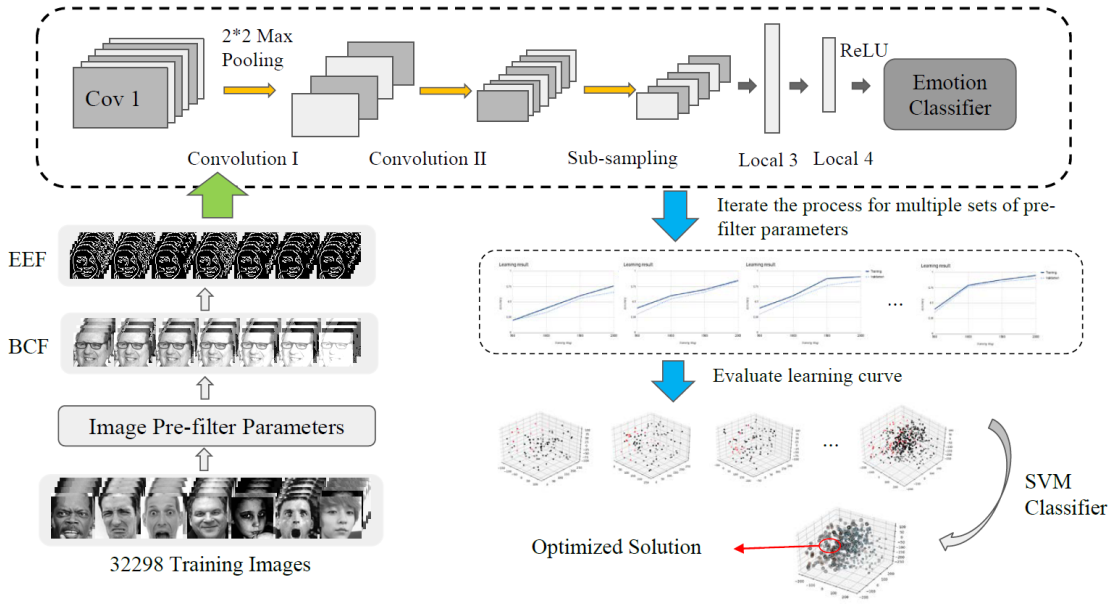


Figure 2.1: Overview of the presented optimization process: 1) use the processed images generated by two pre-filters with varying sets of parameters; 2) analyze the learning outcomes from the CNN; 3) select the best parameter set; and 4) apply the SVM classifier.

2.1.1 Training & Test Datasets

For algorithm development and evaluation, the Facial Expression Recognition (FER 2013) dataset was used [40]. This dataset was created using Google image search Application Programming Interface (API) with 184 emotion related keywords, such as blissful, enraged, and heartening. These images were grouped into the corresponding fine-grained emotion classes by rejecting incorrectly labeled frames and adjusting cropped regions. The resulting data contains nearly 36,000 gray-scale images with 48×48 pixels, and are divided into 7 effective expressions, i.e., 0-angry, 1-disgust, 2-fear, 3-happy, 4-sad, 5-surprise and 6-neutral (See Fig. 2.2). From this dataset, 28,709 images were used for training, 3,589 were used for validation and 3,589 were used for testing.

Figure 2.2: Sample images from FER2013 datasets exhibiting the following emotions: angry, disgust, fear, happy, sad, surprise, and neutral.

2.1.2 Adjusting Brightness & Contrast

BCF aims to manipulate the three channel values of Hue, Saturation and Value (HSV) in each pixel within a color image to eliminate the effect of different lighting conditions. For an input pixel, with its HSV values defined as $\vec{g} = [h, s, v]^T$, BCF results in the output ($\vec{g\prime}$) given by

$$\vec{g\prime} = \Omega\vec{g} + \Delta_{HSV} \tag{2.1}$$

where

$$\Omega = \begin{pmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & \gamma \end{pmatrix}; \quad \Delta_{HSV} = \begin{pmatrix} \Delta h \\ \Delta s \\ \Delta v \end{pmatrix}$$

Ω is the coefficient matrix for scaling the original HSV values of the pixel, and Δh, Δs and Δv are additional small increments added to the scaled values. For efficiency, images are often converted into a gray scale and in this case Ω is a scalar value. Fig. 2.3 shows how the image changes with gradually increasing (top) and decreasing (bottom) brightness and contrast by changing Ω. However, which Ω would result in better recognition performance cannot be determined.

2.1.3 Edge Extraction Filter

Edge Extraction Filter (EEF) is based on Canny edge detection [41]. It is a multi-stage algorithm, which first smoothens the image using a Gaussian filter to eliminate

Figure 2.3: Implementing BCF to a face image: Gradually increasing brightness and contrast (top) vs. gradually decreasing brightness and contrast (bottom).

the noise and then finds the image gradient using the edge detector, which highlights the edge regions, followed by suppression of the pixels which are not at the maximum. The edge detector calculates the magnitude of gradient changes along the horizontal and vertical directions from the smoothened image, derivatives as Gx and Gy are given by

$$\nabla I = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right]^T = [G_x, G_y]^T \tag{2.2}$$

where I represents the image, and x and y represent the horizontal and vertical directions. A full scanner selector checks whether the magnitude of the gradient (∇I) is within a specific range. The selector is integrated with two threshold values $minVal$ and $maxVal$

$$I' = \begin{cases} I(i,j) = 255 & minVal \leq \mid \nabla I \mid \leq maxVal \\ I(i,j) = 0 & else \end{cases} \tag{2.3}$$

where I' is the output of EEF and the magnitude of ∇I is given by $\mid \nabla I \mid = \sqrt{G_x^2 + G_y^2}$.

The boundary conditions for $\mid \nabla I \mid$, i.e., $minVal$ and $maxVal$, determine the range of the gradient intensity. Only part of pixels of I can be stored with specific range of magnitude and then highlighted ($I(i,j) = 255$). Fig. 2.4 shows the results from applying EEF for two images produced by applying the BCF with different Ω values. For both images, varying threshold values were used to demonstrate how they affect the edge detection results. For facial expressions, EEF is a useful tool to extract

desired features efficiently. However, as shown in Fig. 2.4, it may be hard to tell what parameters are better because it can only be concluded after these filtered images are fed into the CNN-based learning and by analyzing the results.

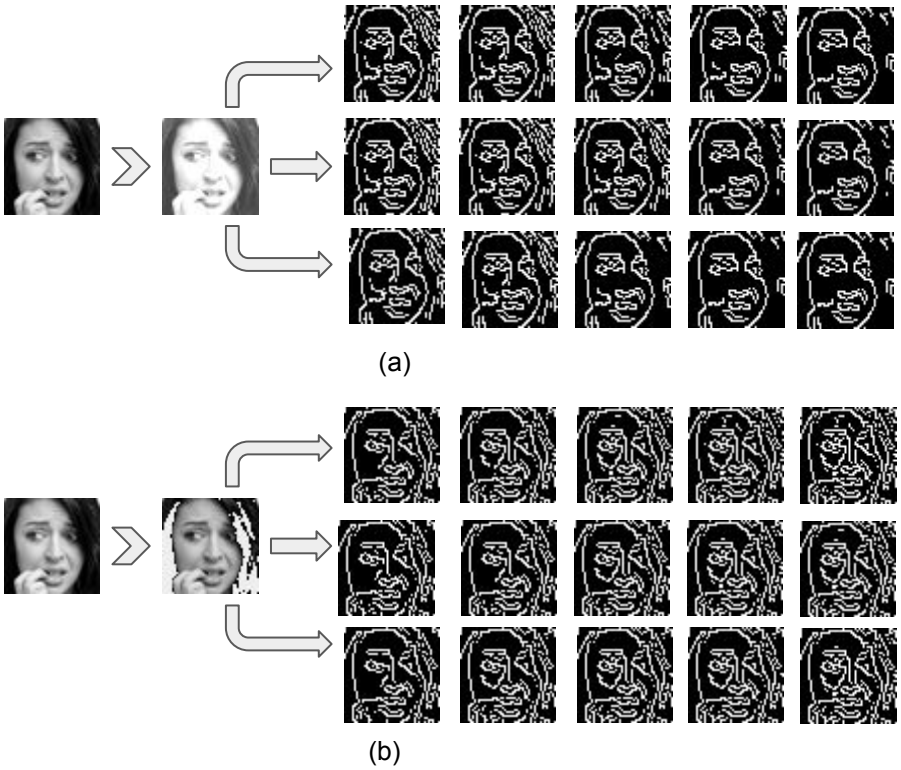


Figure 2.4: Applying BCF with two opposite values of Ω ((a) positive and (b) negative) and then applying EEF with gradually increasing $minVal$ and $maxVal$ from 20, 50, and 100.

2.2 Integration with Learning Algorithms

This section focuses how to obtain optimized solution for parameter selection in the pre-filters. Fig. 2.1 shows the overall process. The original data set is first pre-processed with pre-filters, and put into CNN for training, followed by the learning evaluation. After evaluation, the best performance with higher learning accuracy is determined and SVM classifies all collected pre-filter data. During this process a large random number of pre-filter parameter sets are fed to SVM for obtaining the

optimized solution of pre-filter parameters.

A classic feed-forward CNN has been implemented in our work[42] [43]. The architecture of CNN consists of 4 hidden layers. For convolution and pooling operations, Vanilla Backpropagation [44] is selected. An input image is a 48×48 matrix of pixel values. Convolutions use a stride of one and are zero-padded to ensure that the output maintains the same size as the input. The first convolutional layer is set for computing 64 features in each 5×5 (5 pixels width and height) patch. Its weight tensor has the volume size of $5 \times 5 \times 1$ and 64 output channels to the subsequent convolutional layer. Then, output feature of first convolution follows the process of convolving training pattern with weight tensor, adding bias, applying a rectified linear unit (ReLU) function [45], and max pooling, resulting in a 24×24 matrix. The second layer of convolution has 128 features for each 5×5 patch. Therefore, the weight tensor has the volume size of $5 \times 5 \times 64$ with 128 output channels. After following the same convolution and max pooling process, the image size is then reduced to a 12×12 matrix.

After two layers of convolution, two fully-connected layers ('local fully-connected layer 3 (local 3)' and 'local fully-connected layer 4 (local 4)' shown in Fig. 2.1) are followed. Local 3 includes 3,072 inter neurons with a weight tensor size of 3072×1536 and local 4 has 1,536 inter neurons with a weight tensor size 1536×7. Each of these layers are also followed by a pooling layer. Intermediate input is multiplied by a weight tensor, added by a bias, and then applied to a ReLU function. To avoid overfitting, a dropout is implemented prior to the readout layer. The logistic loss function is selected as a probabilistic and linear classifier. A gradient optimizer was initially considered, but did not perform well for the training images. Instead, the Adam optimizer was implemented.

2.2.1 Learning Evaluation

Evaluation focuses on testing the effect of image pre-processing filters, i.e., BCF and EEF, on CNN-based learning. Fig. 2.5 shows the learning curve using the images produced by the two pre-processing filters with three different sets of parameters. The parameters used for Fig. 2.5(a) shows the best learning performance, with the largest value of $\mathcal{I}$ among the three, while Fig. 2.5(c) resulted in the smallest value of $\mathcal{I}$. To further analyze the results, a linear fitting was applied and shown on the graphs. Linear fitting, however, cannot differentiate between (a) and (b) clearly. In each case, the scale of accuracy during learning shows an ascending trend. Therefore, $\mathcal{I}$ in Eq 2.4 is used as a suitable measurement criterion. Thus, when labeling the learning curve into good or bad, the shaded area (under curve) will generate a score for labeling so that the datasets of learning results will be reachable for SVM. Trapezoidal rule approximation used here to evaluate the areas ($\mathcal{I}$), given by

$$\mathcal{I} = \int_a^b f(x)dx \approx \sum_{k=1}^{N} \frac{f(x_{k-1}) + f(x_k)}{2} \Delta x_k \tag{2.4}$$

Where $f(x)$ describes the learning curve, x_k is the trapezoidal approximation step size.

2.2.2 Parameter Optimization for Pre-Filters

The primary novel contribution of this paper is in the optimization technique for the parameters used in the two image pre-processing filters, i.e., BCF and EEF. The pseudocode for this process is shown in Algorithm 1. In this algorithm, the criterion of labeling good or bad learning results depends on how precise the CNN is required for each application. Obtaining high standard training accuracy requires strict learning evaluation, and therefore a larger area (*desired_area*) is considered a better learning outcome. Moreover, another useful property of Algorithm 1 is that the algorithm can

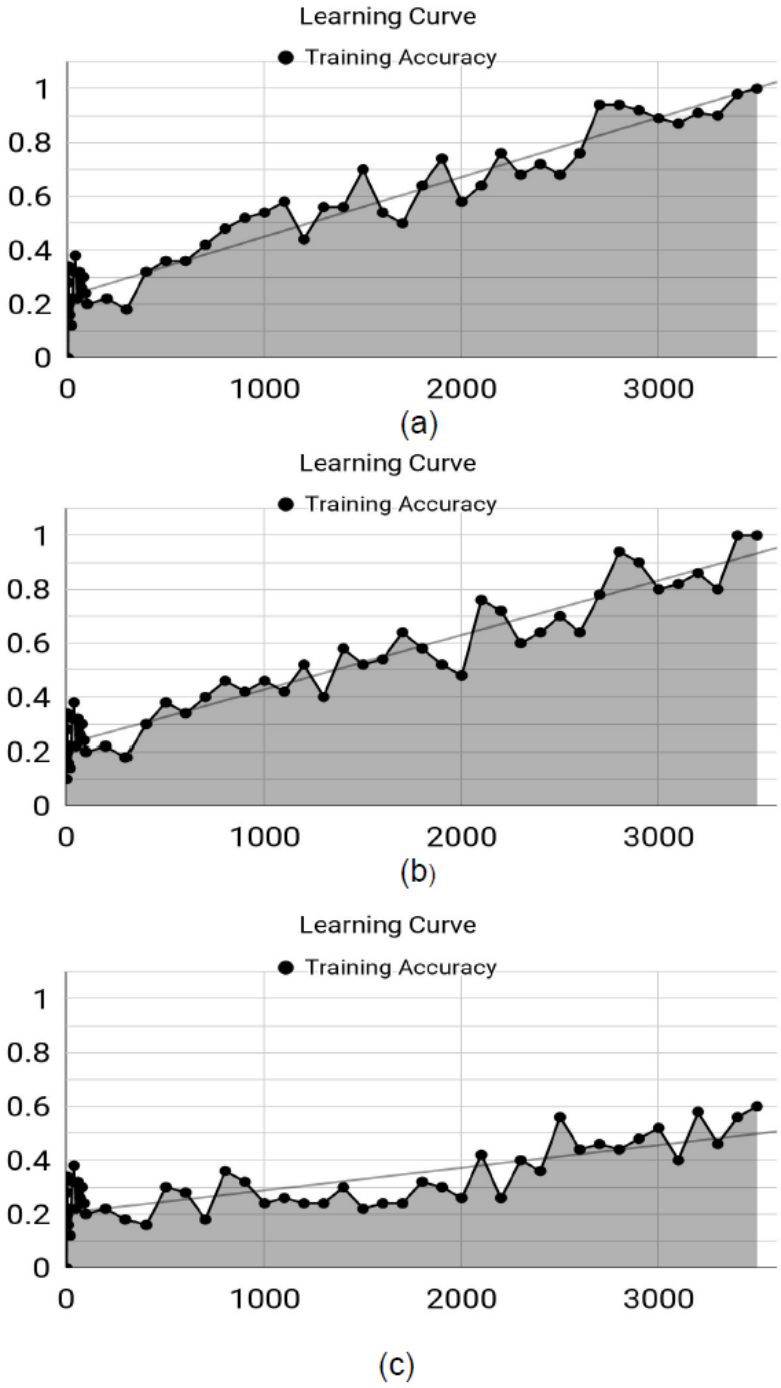


Figure 2.5: CNN-based learning results shown in the training accuracy ([0,1]) over iterations ([1, 3,500]) using the pre-filters with three different sets: (a) the first parameter set, showing $\mathcal{I} \approx 2,288$; (b) the second data set with $\mathcal{I} \approx 2,101$; (c) and the third parameter set, resulting in $\mathcal{I} \approx 1,168$.

Algorithm 1: Get optimized pre-filter

```
input  : datasets
output: optimized_PF
for  n ∈ {1, ...desired_value} do
    PF = random [χ1, χ2, χ3, ··· χn];
    while PF iteration time ≤ desired_value do
        Ψ = random [ψ1, ψ2, ψ3, ··· ψn] ∗ Δh;
        PF = PF + Ψ;
        perform BCF according to 2.1;
        implement EEF according to 2.2 2.3;
        proceed to CNN with 3500 epoch;
        Compute learning curve areas according to 2.4;
        if area ≥ desired_area then
            Update: append [label = 1] to labels;
            Update: append area to areas;
        else
            Update: append [label = 0] to labels;
            Update: append area to areas;
        end
        Update: append PF to PF_n;
    end
    Update: horizontal stack PF_n to PF_final ;
end
for  n ∈ {1, ...desired_value} do
    Plot 3D. scatter with PF_n;
end
normalize PF_final;
fit SVM classifier with PF_final & labels using kernel = linear;
Plot 3D. scatter with PF_final ;
for  i ∈ {1, ...predictiontime} do
    PF = random [χ1, χ2, χ3,··· χn];
    prediction = SVM classifier.predict(PF);
    Update: append prediction to predictions;
    if prediction == 1 then
        Update: append PF to optimized_PF;
    end
end
print (optimized_PF);
```

be used for datasets other than faces, and produces an optimal set of parameters for the pre-filters.

In Algorithm 1, two stochastic iterations are included to generate random values of $[\chi_1, \chi_2, \chi_3, \cdots \chi_n]$ (i.e., pre-filter parameters) and $[\psi_1, \psi_2, \psi_3, \cdots \psi_n] * \Delta h$ (i.e., increments with step size $\Delta h = 1$). Firstly, iterating increments with $[\chi_1, \chi_2, \chi_3, \cdots \chi_n]$ fixed. By implementing BCF, EEF, and CNN, each parameter set for the pre-filters is labeled as either "good (label =1)" or "bad (label =0)." When setting strict criteria for labeling, SVM can be very accurate as it only selects the best pre-filter sets as its output. Using Algorithm 1, it can produce the best optimized set of parameters for the pre-filters given the datasets.

2.3 Algorithm Evaluation

2.3.1 Learning Results Visualization

Learning results evaluation has been conducted in the following procedures shown in Fig. 2.1. Ranges of BCF and EEF were being set to 0∼255 and -100∼100 based on minimum and maximum of image features, in order to cover all situations with various pre-filters. In this paper, we embedded pre-filter with three parameters (Ω value for BCF, *minVal* and *maxVal* for EEF), thus for each iteration, pre-filter can be visualized as a single dot in 3D space. For the first usage of Algorithm 1, with $[\chi_1, \chi_2, \chi_3]$ fixed, iterate $[\psi_1, \psi_2, \psi_3] * \Delta h$ (step size $\Delta h = 1$) for 300 and 500 times, $\mathcal{I} \approx 2,101$, shown in Fig. 2.5(b) is chosen as the criteria for labeling, results shown as Fig. 2.6(a-c).

Another trial conducted with both $[\chi_1, \chi_2, \chi_3]$ and $[\psi_1, \psi_2, \psi_3] * \Delta h$ iterating. We randomized pre-filter in the outer loop for five times and randomize Ψ in the inner loop for 100 times, iteration adds up to 5×100 times. The plotted 3D scatter graph as Fig. 2.6(d-e) with color sequences (light tone = 'good', cold tone = 'bad')

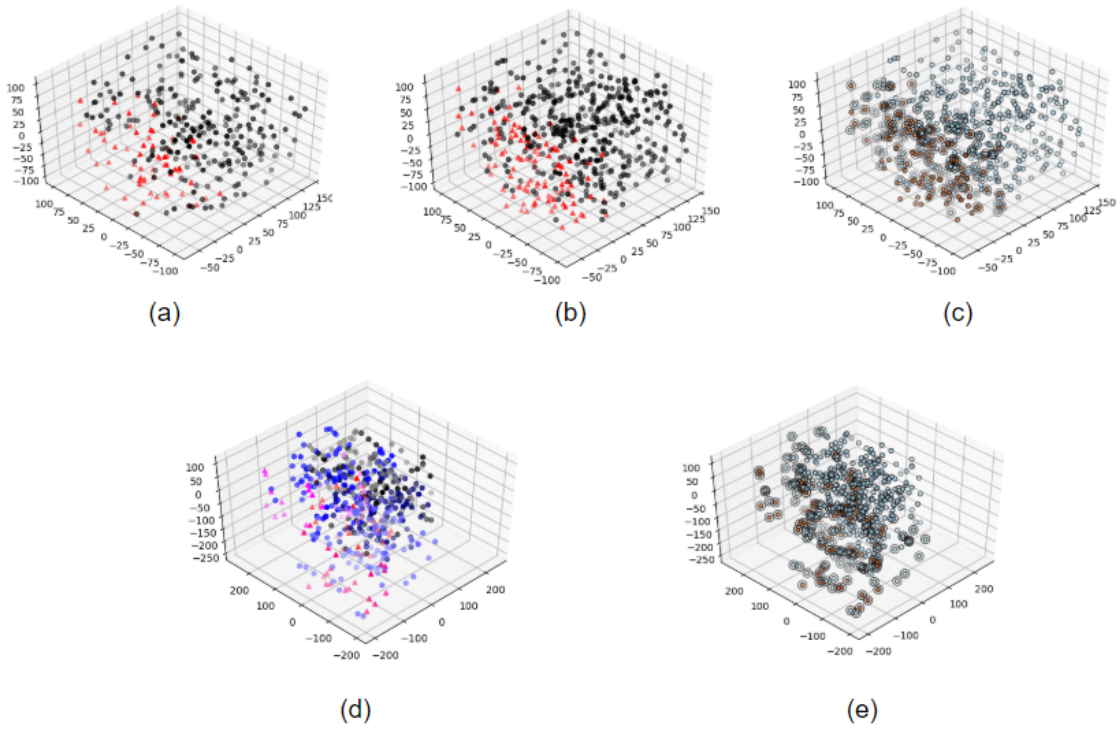


Figure 2.6: 3D Visualizations of pre-filter parameters after being conducted by CNN and learning evaluation. X, Y and Z axes represent BCF, EEF (2 variable embedded) parameters separately. With [χ_1, χ_2, χ_3] fixed but various Ψ iteration, iterate for 300 times (a), 500 times (b) and supporting vectors in 3D space (c); stochastic iteration of [χ_1, χ_2, χ_3] and Ψ, 3D scatter graph for 5∗100 times iteration with color sequence (d) and supporting vectors in 3D space (e)

is representing each pre-filter iteration. The result shows that a general optimized solution within large range for pre-filter can still be found from the output although the initial pre-filter is random.

2.3.2 Accuracy & Efficiency Evaluation

For testing accuracy and efficiency, five different cases were considered: (a) the presented optimized pre-filters + CNN-SVM, CNN with 3,500 epochs; (b) a fixed set of parameters in the pre-filters + CNN-SVM, CNN with 3,500 epochs; (c) CNN-SVM with no pre-filter, CNN with 3,500 epochs; (d) a fixed set of parameters in the pre-filters + CNN-SVM, CNN with 7,800 epochs; and (e) CNN-SVM with no pre-filter,

CNN with 9,600 epochs (See Table 2.1).

Table 2.1: Efficiency and accuracy comparison

Pre-filter	CNN epoch	Comp. Time	FER Accuracy
(a) Optimized	**3,500**	**70.32 sec**	**98.19%**
(b) Fixed	3,500	70.37 sec	71.99%
(c) None	3,500	70.47 sec	50.68%
(d) Fixed	7,800	142.25 sec	96.43%
(e) None	9,600	172.23 sec	95.99%

GTX1080Ti, cudnn 7.0 and Cuda 9.1 toolkit was used for CNN training on GPU. For (1) 5,000 randomized sets of pre-filter parameters were provided to SVM for image classification and good pre-filter parameters were stored as output. A pre-filter parameter set was selected arbitrarily from an optimized pool, and then BCF, EEF, and CNN were performed. As shown in Table 2.1 (a-c), if the number of epochs for CNN training is fixed at 3,500, the computational time remains the same, regardless of the use of pre-filters, while the accuracy differs significantly. The optimized pre-filters achieved 97.85% accuracy, while fixed pre-filters and no pre-filters resulted in 71.99% and 50.68%, respectively. For the system to achieve $\geq$ 95% accuracy with a fixed parameter set in the pre-filters or without using pre-filters, it requires 7,800 and 9,700 epochs, respectively, in the CNN resulting in significantly higher computational times (Table 2.1 (d-e)).

To visualize the performance of the pre-filter optimization, Table 2.2 & 2.3 show the emotion classification results from (a) and (c).

The number of testing images was 3,589. The total number of correctly classified images with the optimized pre-filters was 3,524, with accuracy of 98.19%; however, the number of correctly classified images without pre-filters was only 1,819 with accuracy of 50.68%.

Table 2.2: Classification results with the proposed techniques for seven emotions denoted as A (Angry), D (Disgust), F (Fear), H (Happy), S (Sad), Su (Surprise), and N (Neutral) in %

Emotion	A	D	F	H	S	Su	N
A	**98.36**	0.20	0.20	0.00	0.60	0.00	0.60
D	0.00	**96.36**	0.00	0.00	0.18	0.00	0.18
F	1.15	0.00	**96.74**	0.00	1.53	0.00	0.58
H	0.69	0.00	0.00	**98.97**	0.00	0.11	0.22
S	1.80	0.00	0.34	0.00	**98.15**	0.00	0.34
Su	0.72	0.00	0.24	0.72	0.00	**98.31**	0.00
N	1.13	0.00	0.00	0.00	0.97	0.00	**98.23**

Table 2.3: Classification results without pre-filter optimization in %

Emotion	A	D	F	H	S	Su	N
A	**39.22**	0.82	10.27	10.27	14.58	3.49	17.25
D	20.00	**43.64**	10.91	9.09	7.27	1.82	9.09
F	12.67	0.19	**31.48**	9.40	17.66	8.64	15.16
H	6.18	0.00	3.32	**67.28**	7.44	3.32	14.87
S	16.16	0.67	10.44	16.33	**41.92**	3.37	21.04
Su	5.80	0.48	4.83	6.04	3.14	**73.19**	6.76
N	8.04	0.48	7.07	13.99	15.76	4.02	**48.23**

2.4 Conclusion

The presented algorithm, combining optimized pre-processing filters with a CNN-SVM classifier, improved time and accuracy in facial emotion recognition. This method serves as not only an optimization technique for achieving improved performance of CNN for detecting facial emotions, but also a useful technique for comprehensive tuning of pre-filters for various images beyond human faces. Evaluation results revealed that the presented algorithm is efficient and accurate, and therefore suitable for embedded applications, such as human-computer, or even human-robot, interaction.

Chapter 3

AU-based FER

The second approach for FER using two types of geometrical facial features, i.e., Landmark Curvature (LC) and Vectorized Landmark (VL), extracted from facial Action Units (AUs). This method follows five steps (Fig. 3.1):

1) 68 facial landmarks are detected using Ensemble of Regression Trees [46] through a machine learning toolkit called Dlib [47].

2) Selected landmarks closely related to facial expressions are chosen for fitting facial AUs.

3) Two geometrical features, LC and VL are extracted based on selected landmarks.

4) LC and VL features are provided to SVM for classification.

5) Parameters associated with landmark selection, LC and VL feature extraction, and SVM are tuned for improved FER performance.

In this section, we first present the FER algorithm for LC/VL-based emotion classification, (Step 1 - Step 4), and detailed methods for parameter optimization (Step 5) are described in Section 4.

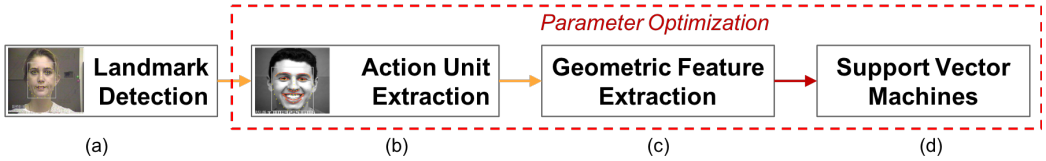


Figure 3.1: Proposed method with 4 steps: (a) landmark detection; (b) Action Unit extraction with landmark; (c) LC and VL feature extraction and improvements; and (d) SVM training with improved parameters.

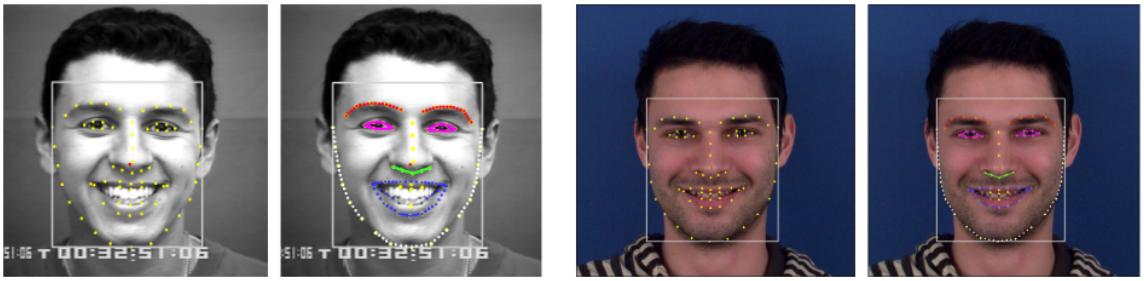
3.1 Algorithm

3.1.1 Facial Landmark Detection

For the first step, facial landmarks are detected using the Dlib tool kit, which provides thoroughly documented implementation of face detection based on the HOG filter. This filter is trained by Max-Margin Object Detection (MMOD) [47] and followed by landmark detection using Ensemble of Regression Trees [46], which estimates facial landmark positions from the cascade regressor derived from a sparse subset of pixel values. This landmark detection produces 68 well-trained landmarks from Labeled Face Parts in the Wild (LFPW) dataset in few milliseconds [48]. The detected landmarks include corners of a mouth, eyebrows, eyes, and nose. The image is reorganized to have (a) two images from landmark detection only (left ones from the two sets) and (b) two right images. Fig. 3.2(a) shows one sample face image selected from the extended Cohn-Kanade (CK+) dataset [49] with the detected face shown in a grey rectangular box and 68 landmarks shown in yellow dots. Fig. 3.2(b) shows the results for an image selected from the Multimedia Understanding Group (MUG) Facial Expression database [50].

3.1.2 Facial Action Units and Facial Segments

Facial Action Coding System (FACS) is a system developed for encoding facial movements by distinctive momentary changes [24, 51]. FACS describes facial expressions



(a) (b)

Figure 3.2: (a) Landmark detection and Action Units group described with image from CK+ dataset; (b) Landmark detection and Action Units group described with image from MUG dataset;

based on AUs – distinctive units of facial actions driven by individual muscles or groups of muscles. According to the FACS, 30 AUs (12 for upper face, 18 for lower face) are considered anatomically related to contractions of a specific set of facial muscles generating facial expressions [25]. Among these 30 AUs, 12 AUs can be described using 59 out of 68 landmarks detected by the Dlib toolkit. To realize automatic FER, our method focuses on these 12 AUs.

The 12 AUs are re-classified into 16 Facial Segments (FSs) and divided into five groups as shown in Table 3.1. Group I with two AUs describes the left and right eyebrows, reclassified into four segments (FS_1-FS_4). Group II includes two AUs, reclassified as four segments (FS_5-FS_8), describing eyelid movements. Group III involves FS_9 associated with the nose wrinkler. Group IV contains 5 AUs associated with lip movements, reclassified into four segments (FS_{10}-FS_{13}). Group V consists of two AUs for the cheeks and chin (FS_{14}-FS_{16}). Fig. 3.2(a) and Fig. 3.2(b) illustrate these 12 AUs on a selected face image from each CK+ and MUG databases, respectively. Five different colors are used for five distinctive AUs groups: Group I (red), Group II (purple), Group III (green), Group IV (blue), and Group V (gray), according to Table 3.1.

Table 3.1: AUs, FSs with description, and associated landmarks for each FS grouped into five categories.

Group	AU	FS	Description	Index of Landmarks
Group I	AU_1	FS_1	Left inner brow raiser	20-22
		FS_2	Right inner brow raiser	23-25
	AU_2	FS_3	Left outer brow raiser	18-20
		FS_4	Right outer brow raiser	25-27
Group II	AU_5	FS_5	Left upper lid raiser	37-40
		FS_6	Right upper lid raiser	43-46
	AU_7	FS_7	Left lid tightener	37, 40-42
		FS_8	Right lid tightener	43, 46-48
Group III	AU_9	FS_9	Nose wrinkler	32-36
Group IV	AU_{10}	FS_{10}	Upper lip raiser	49-55
	AU_{12};AU_{15}	FS_{11}	Left lip corner	49, 61, 68
		FS_{12}	Right lip corner	55, 65, 66
	AU_{20};AU_{23}	FS_{13}	Lip stretched/tightener	49, 55-60
Group V	AU_{13}	FS_{14}	Left cheek puffer	1-6
		FS_{15}	Right cheek puffer	12-17
	AU_{17}	FS_{16}	Chin raiser	7-11

3.1.3 Feature Extraction

SVM takes one or more types of vectorized data as inputs and classifies them into distinctive classes. For SVM-based facial emotion classification, two types of geometric features, i.e., LC and VL, are extracted from the AU-related landmarks described above. First, the set of all landmarks is defined as S_L, such that

$$S_L = \{L_1, L_2 \dots L_N | L_i = (x_i, y_i),\ x_i, y_i \in \mathbb{R}, i = 1 \dots N\} \tag{3.1}$$

where N is the number of landmarks and each $L_i = (x_i, y_i)$ indicates the pixel location of the landmark with respect to $(0, 0)$ located at the upper left corner of the image frame. In our case, $N = 59$. As shown in Table 3.1, each FS is comprised of a unique subset of S_L. For example, FS_1 describes the left inner brow raiser using 3 landmarks, such that $FS_1 = \{L_{20}, L_{21}, L_{22}\}$.

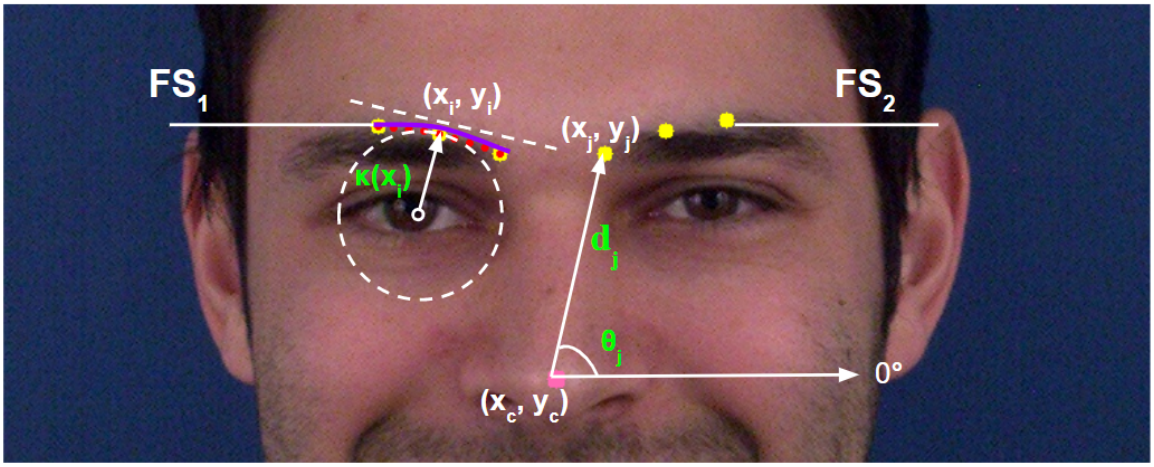


Figure 3.3: Feature extraction from given landmarks associated with FS_1 and FS_2: LC feature (κ_i) at landmark (x_i, y_i), and VL feature $\vec{g}_j = (d_j, \theta_j)^T$ at (x_j, y_j).

For each set of landmarks corresponding to individual FS_i, the least squares regression was applied to fit a curve. To avoid poor fitting conditions, linear interpolation–doubling the number of data points–were applied prior to fitting the points into a polynomial function. For example, if FS_i involves r landmarks, $r-1$ data points were added via linear interpolation. Then, for $2r-1$ data points, polynomial filling using least squares regression was applied, such that

$$f_{FS_i}(x) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M = \sum_{j=0}^{M} w_j x^j \tag{3.2}$$

where w_j for $j = 1, \cdots, M$ is calculated in order to minimize the squared error between the y value of data points and f_{FS_i}. It is noted that the highest order M determines the geometric shape of the curve, and therefore is expected to influence the training results. The effect of M in training outcomes can be experimentally evaluated and its value can be determined to optimize the performance. Such parameter tuning process is described in detail in Section 3.2.

After curve fitting, local curvatures at individual landmarks are calculated by

$$\kappa(x_i) = \frac{\mid f''_{\mathrm{FS}_i}(x_i) \mid}{\left\{1 + \left(f'_{\mathrm{FS}_i}(x_i)\right)^2\right\}^{\frac{3}{2}}} \tag{3.3}$$

where $f'_{\mathrm{FS}_i}(x_i)$ and $f''_{\mathrm{FS}_i}(x_i)$ are the first and second derivatives of the function with respect to x, respectively. If FS_i consists of r landmarks, S_{κ_i} is defined as the following set:

$$S_{\kappa_i} = \{\kappa(x_1), \kappa(x_2), \kappa(x_3), \ldots, \kappa(x_r)\} \tag{3.4}$$

For a given n FSs, the entire set of landmark curvatures can be written as

$$S_{LC} = \cup_{i=1}^{n} S_{\kappa_i}. \tag{3.5}$$

Fig. 3.3 demonstrated LC feature, $\kappa(x_i)$, at landmark (x_i, y_i) for FS_1, where the red dots and purple curve illustrated linear interpolation and least square regression curve fitting process. The dashed circle visualizes $\kappa(x_i)$. Algorithm 2 describes the LC feature extraction process.

Algorithm 2: Extract LC features

input : Input image
output: S_{LC}
if *landmark* ***not*** *detected* **then**
  return: False;
else
  AU ← Selected AU according to Table 3.1;
end
for i ***in range*** *(number of FSs)* **do**
  $\mathrm{f}_{\mathrm{FS}_i}$← apply curve fitting from eq. (3.2);
  $\kappa(x_i)$← curvature from eq. (3.3);
  S_{κ_i}← append curvature $\kappa(x_i)$;
  S_{LC}← append S_{κ_i};
end
S_{LC}← normalization S_{LC};
return S_{LC};

The set of the VL features is defined by the polar coordinates of the landmarks described with respect to the geometric center of all landmarks calculated by

$$L_c = \begin{pmatrix} x_c \\ y_c \end{pmatrix} = \begin{pmatrix} \frac{1}{N}\sum_{i=1}^{N} x_i \\ \frac{1}{N}\sum_{i=1}^{N} y_i \end{pmatrix}.$$

where N is the total number of selected landmarks. First, the Euclidean distance of each landmark measured from L_c is calculated by

$$d_i = \|L_i - L_c\|_2. \tag{3.6}$$

Second, the angle relative to the horizontal axis (i.e., 0°) is then obtained for each landmark by

$$\theta_i = \tan^{-1}\frac{y_i - y_c}{x_i - x_c} \tag{3.7}$$

These transformed polar coordinates of the landmarks, $\vec{g}_i$, are considered as the VL features:

$$S_{VL} = \{\vec{g}_1, \vec{g}_2, \cdots, \vec{g}_N |\ \vec{g}_i = (d_i, \theta_i)^T,\ i = 1, \cdots, N\} \tag{3.8}$$

Fig. 3.3 illustrates $\vec{g}_j = (d_j, \theta_j)^T$ for a selected landmark (x_j, y_j). Algorithm 3 shows the VL extraction process. After executing Algorithm 1 and Algorithm 2, all LC and VL features extracted from 59 landmarks and merged into a single row vector, called the LC-VL vector.

3.1.4 SVM for FER

SVM is a powerful tool for both binary and multi-class classification and regression and thus is suitable for our FER application. SVM is also robust against outliers because it estimates optimal separating hyper-planes among different classes by maximizing the margin between the hyper-plane and closest points of the classes. This

Algorithm 3: Extract VL features

input : Input image
output: S_{VL}
if *landmark* ***not*** *detected* **then**
  return: False;
else
  $L_c \leftarrow$ geometric center of all landmarks;
end
for i ***in range*** *(number of landmarks)* **do**
  $d_i \leftarrow \|L_i - L_c\|_2$ from eq. (3.6);
  $\theta_i \leftarrow tan^{-1}\frac{y_i - y_c}{x_i - x_c}$ from eq. (3.7) ;
  $S_{VL} \leftarrow$ append $\vec{g}_i = (d_i, \theta_i)^T$;
end
$S_{VL} \leftarrow$ normalization S_{VL};
return S_{VL};

SVM optimization problem is formulated as [52]

$$\min(\vec{\omega}, b) = \frac{1}{2}\vec{\omega}^T\vec{\omega} + C\sum_{i=1}^{n}\max(1 - \tilde{y}_i(\vec{\omega}^T\vec{\phi}(\vec{\tilde{x}}_i) + b), 0)^2 \tag{3.9}$$

where $(\vec{\tilde{x}}_i, \tilde{y}_i)$ represents training pairs, normal vector $\vec{\omega}$ and scalar b determine the linear hyper-plane, $\vec{\phi}(\vec{\tilde{x}}_i)$ is the mapping function to map the training data into a higher dimensional space. L2-loss function, $\max(1 - \tilde{y}_i(\vec{\omega}^T\vec{\phi}(\vec{\tilde{x}}_i) + b), 0)^2$, is chosen for better contributing to multi-classification problems [52]. C is the essential regularization parameter, which controls the trade-off between achieving low error on the training data and minimizing the norm of the weights. Obtaining a high-level training performance is determined by tuning C properly. To determine the attributes of VL and LC features respectively in the training session, weight factor W_1 and W_2 are introduced prior to constructing training pairs $(\vec{\tilde{x}}_i, \tilde{y}_i)$. Weight factor W_1 and W_2, determine the portion of each feature in the training pairs, with the regularity term C in SVM altogether determined the training result.

3.2 Parameter Optimization

The presented FER algorithm in Section 3.1 involves the following four parameters:

- M: Order of curve fitting in Eq. (3.2)
- G: Different combinations of five groups
- W_1 & W_2: Weight factors for two sets of data in SVM
- C: Regularization parameter in Eq. (3.9)

This section presents the procedural method for selecting an optimal set of parameters for achieving the highest accuracy given a dataset. Although cross-validation of parameter selection for different datasets would be ideal, such a process can be highly time consuming with an excessive amount of data [53]. Instead, we selected two commonly used facial expression datasets, i.e., CK+ and MUG, and experimentally optimized the parameters. Note that both datasets are randomly shuffled and divided into 80% for training and 20% for testing. The algorithm was performed on Ubuntu 17.10 system, with intel i7-8700 CPU (3.20 GHz) and 16G RAM.

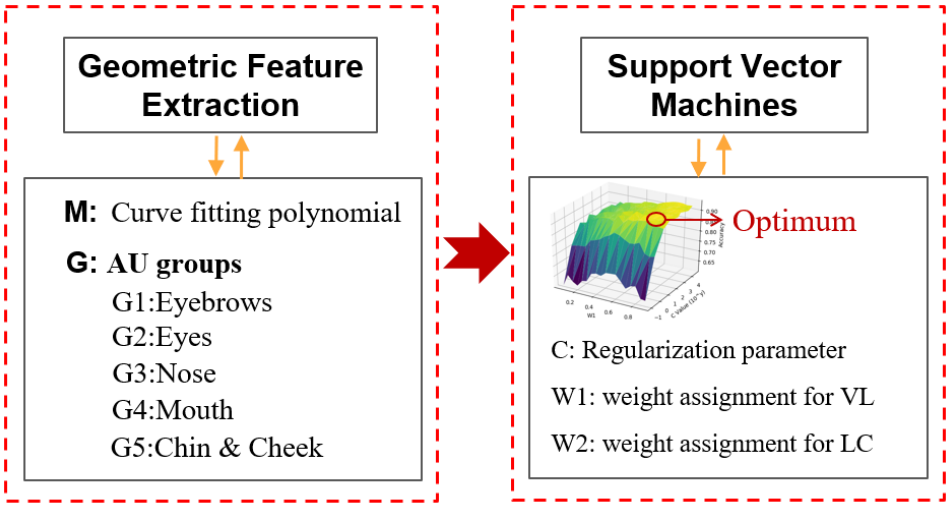


Figure 3.4: Overview of algorithm improvements by tuning parameters related to feature extraction and SVM.

3.2.1 Dataset Preparation

CK+ and MUG Facial Expression database mentioned in section 2.1 were used for training and testing of the presented algorithm shown in Fig. 3.5. The CK+ database consists of 593 sequences of face images taken from 123 subjects. Each sequence starts with onset (neutral expression) and ends with a peak expression (last frame) which shows the change of one's facial expression from neutral to a certain emotion. From 327 selected labeled sequences, we extracted 6 frames from each sequence into the organized sets. These 6 frames include the start frame (neutral) and the last five frames (the labeled emotion). Based on this approach, 1,872 frames displaying anger (225), disgust (295), fear (125), happiness (345), neutral (327), sadness (140), and surprise (415) separately, were extracted.

MUG Facial Expression database contains 1,462 color image sequences from 86 subjects with different facial expressions. Similar to the CK+ dataset, frames representing the peak expression were extracted into organized sets from each sequence. Processed MUG dataset contains 2,658 images in total, including anger (318), disgust (366), fear (207), happiness (520), neutral (521), sadness (339), and surprise (380).

Fig. 3.5(a) shows seven face images arbitrarily selected from the CK+ dataset (top) and the detected face shown in a rectangular box and 68 landmarks shown in yellow dots for each image (middle). The bottom row displays cropped and enlarged face images for better visualization of the detected landmarks. Fig. 3.5(b) shows the same for the seven face images arbitrarily selected from the MUG Facial Expression database.

3.2.2 Feature Type & Interpolating Order Determination

The first step in the parameter optimization is to determine whether using both VL and LC results in a higher FER accuracy than using only VL or LC. Since the LC features are determined by the order of curve fitting (M), each case was tested for

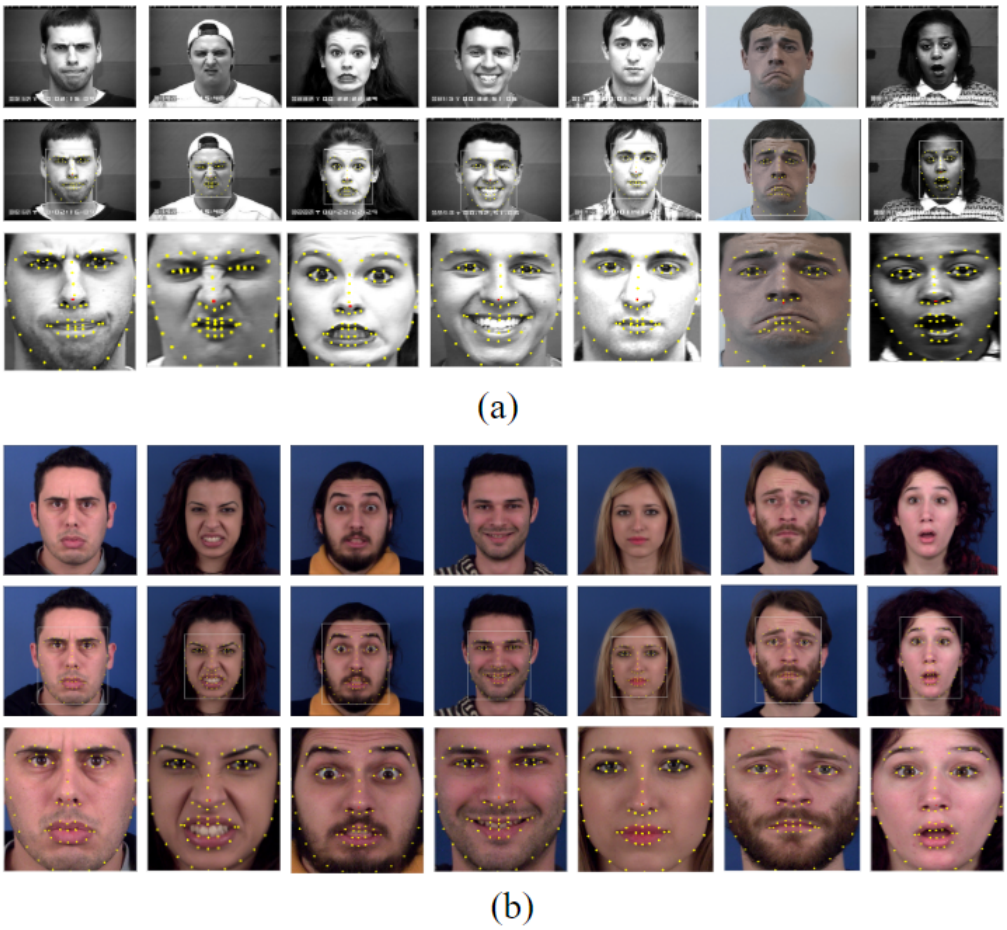


Figure 3.5: (a) Landmark detection with image cropping and re-sizing implemented on CK+ dataset: examples of 7 labeled emotions (i.e., "anger", "disgust", "fear", "happy", "neutral", "sadness" and "surprise") (row 1); after performing landmark detection (row 2); and after cropping and re-sizing (row 3); (b) MUG dataset examples organized the same way as (a).

different value of $M = 2, 3, 4$. At this stage, the SVM-related parameters were set to $C = W_1 = W_2 = 1$.

Table 3.2 shows the results in terms of training time and recognition accuracy using VL, LC, or both VL and LC for each database. In both datasets, using both VL and LC with $M = 3$ resulted in the highest accuracy in FER (i.e., 88.01% for CK+ and 88.83% for MUG). When only one of the features was used, the training accuracy remained similar while VL required nearly twice the training time of LC. Using both VL and LC has slightly increased the training time compared to the case

Table 3.2: Analysis of LC and VL features and polynomial interpolation order

VL	LC	M (curve fitting)	CK+ Dataset		MUG Dataset	
			time (sec)	Accuracy	time (sec)	Accuracy
✓	✗	✗	402.33	76.24%	1387.54	82.20%
✗	✓	2	214.17	72.27%	719.11	78.60%
✗	✓	3	213.38	76.57%	719.36	84.09%
✗	✓	4	215.49	74.82%	708.44	77.84%
✓	✓	2	415.48	84.62%	1421.70	88.02%
✓	✓	**3**	**416.36**	**88.01%**	**1399.96**	**88.83%**
✓	✓	4	416.74	85.98%	1406.76	87.48%

using VL only. We also found that M showed no or little effect on the training time. A value of $M > 4$ lowered the accuracy due to over-fitting. The training time for MUG was relatively larger due to the size of MUG dataset, while the same trend appeared as in CK+.

3.2.3 FS Selection

The second step of parameter optimization is to determine what facial segments (FS) to be used. More number of segments – more data – would directly result in increased training time. In addition, more data does not guarantee better performance. To evaluate the effect of the facial segments on recognition accuracy and speed, 16 facial segments were divided into five groups, corresponding to facial elements, i.e., eyebrows (Group I), eyelids (Group II), nose (Group III), lips (Group IV), and cheek/chin (Group V), as listed in Table 3.1. For these five groups, all possible combinations were examined for FER performance in terms of accuracy.

Tables 3.3 and 3.4 show the results. An interesting finding was that for both datasets, using all five groups did not result in the highest accuracy. Instead, a combination of four groups (G_{1234}), excluding Group V showed the highest accuracy in both cases (i.e., 91.88% in CK+; 91.10% in MUG). In MUG, G_{124} also resulted in

the same highest accuracy of 91.10%. The results from both datasets were consistent – for any three groups G_{124} results in the highest accuracy and for any two groups G_{24} does. Increasing the number of landmarks did not have a significant effect on trainint time. For example, it took about 416 seconds when all five groups were used, while taking 405 seconds for G_{24}.

Table 3.3: LC-VL Feature Improvements based on AU Combinations for CK+

	Accuracy in Combined AU Group				
All 5 Groups	G_{12345} **88.01%**				
Any 4 Groups	G_{1234} **91.88%**	G_{1235} 73.88%	G_{1245} 87.84%	G_{1345} 81.40%	G_{2345} 83.54%
Any 3 Groups	G_{123} 76.39%	G_{124} **88.55%**	G_{125} 79.25%	G_{134} 85.69%	G_{135} 66.73%
	G_{145} 84.36%	G_{234} 87.30%	G_{235} 68.87%	G_{245} 81.22%	G_{345} 76.74%
Any 2 Groups	G_{12} 72.09%	G_{13} 61.18%	G_{14} 86.58%	G_{15} 65.65%	G_{23} 73.35%
	G_{24} **87.67%**	G_{25} 68.69%	G_{34} 80.08%	G_{35} 59.21%	G_{45} 74.42%

Table 3.4: LC-VL Feature Improvements based on AU Combinations for MUG

	Accuracy in Combined AU Group				
All 5 Groups	G_{12345} **88.83%**				
Any 4 Groups	G_{1234} **91.10%**	G_{1235} 77.84%	G_{1245} 87.31%	G_{1345} 81.09%	G_{2345} 85.42%
Any 3 Groups	G_{123} 78.79%	G_{124} **91.10%**	G_{125} 69.51%	G_{134} 89.02%	G_{135} 68.94%
	G_{145} 86.74%	G_{234} 87.69%	G_{235} 71.78%	G_{245} 86.74%	G_{345} 81.06%
Any 2 Groups	G_{12} 67.42%	G_{13} 60.98%	G_{14} 84.47%	G_{15} 66.10%	G_{23} 68.75%
	G_{24} **87.31%**	G_{25} 66.86%	G_{34} 82.39%	G_{35} 58.90%	G_{45} 77.27%

3.2.4 SVM Parameter Tuning

The third and final stage of parameter tuning targets the three SVM-related parameters, i.e., W_1, W_2, and C. The input to the SVM is the merged LC-VL vector, which is a non-standardized vector. Knowing that using both LC and VL significantly improves FER performance, this process aims to determine their individual attributes. To solve this multi-attribute problem, the optimization model in this case is presented by

$$A_{ij} = \max_{W_1,W_2,C} score(C^{(i)}, S_{VL}W_1^{(j)}, S_{LC}W_2^{(j)}) \tag{3.10}$$

where $i = 1, 2, \cdots, Q$ and $j = 1, 2, \cdots, P$. Function $score()$ returns the test accuracy of SVM classifier with the i^{th} regularity parameter C and different weight assignment as input. A_{ij} is the best alternative among all returned test accuracy, where W_1, W_2 are the weight of VL and LC feature, respectively. 5% is set as the step size of $W_1^{(j)}$ ranging from 5%∼95%, so $P = 19$ in this case, and $W_2^{(j)}$ is $1 - W_1^{(j)}$. Q is the number of regularization parameter C, where

$$C = 10^{\lambda}, \lambda \in \mathbb{R} \tag{3.11}$$

$C^{(i)}$ is obtained by changing λ in Eq.3.11. Therefore, once A_{ij} is found, W_1, W_2 under i^{th} penalty factor C can be found, the best tuned value of C and best weight assignment for each feature (i.e., VC and VL) can be obtained.

All alternative results are shown in Fig. 3.6. Three axes represent the weight of VL feature (W_1) distributed in the range 0%∼100%, the value of parameter C (10^{-1}∼10^5), and the training accuracy of the model, respectively. Fig. 3.6(a-c) are the results for CK+ and Fig. 3.6(d-f) are for MUG. For both datasets, we found that the accuracy increases as C increases; however, after C reaches a certain threshold value, the accuracy slightly dropped by 2%∼3% while elapsed training time continued

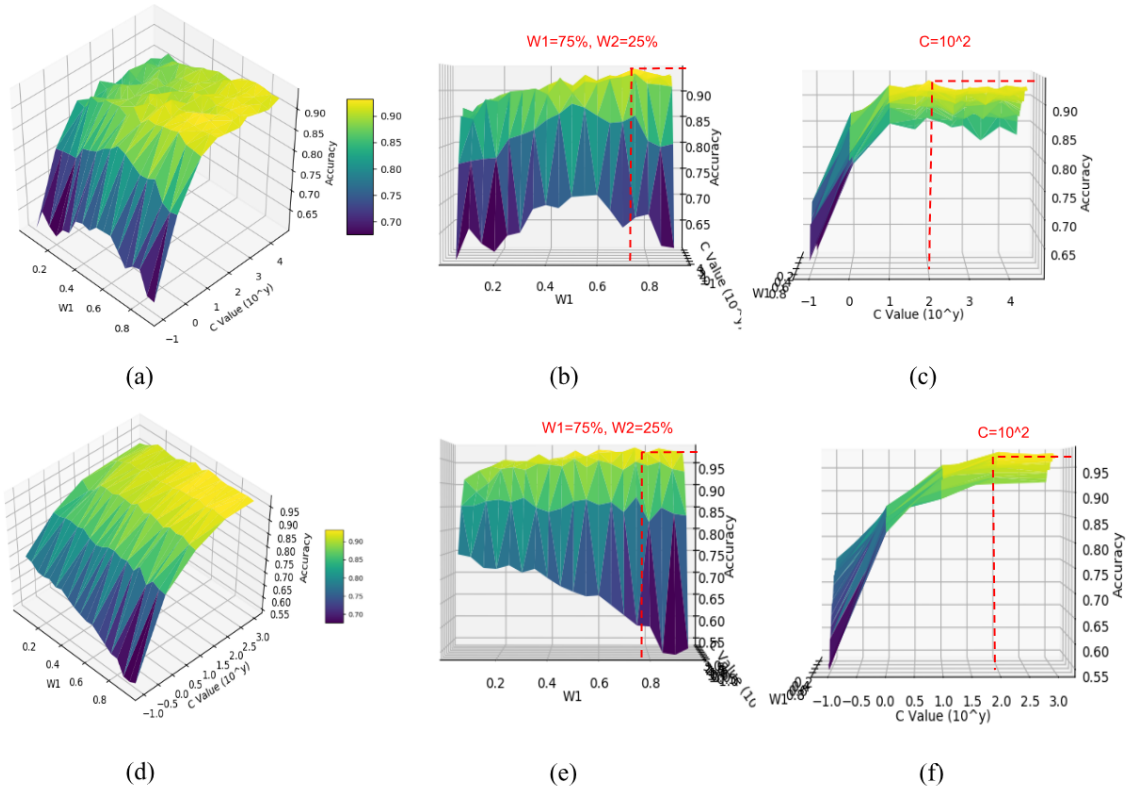


Figure 3.6: Optimum weights and penalty parameter for CK+ datasets (a)∼(c) and MUG datasets (d)∼(e).

to increase. For CK+, the optimal C was found at $C = 10^2$ and the optimal weight distribution was found at $W_1 = 75\%$ for VL features (i.e., $W_2 = 25\%$ for LC features). With these selected parameters in SVM, the classifier returned the testing accuracy of 96.06% on 7 classes and 98.38% on 6 classes (without neutral), which exceeds the performance of existing algorithms [28, 29, 54, 55]. Table 3.5 presented the results on CK+ database by implementing proposed algorithm with fine-tuned parameters. As shown in Fig. 3.6 (d-f), the highest accuracy for MUG was also found at $W_1 = 75\%$, $W_2 = 25\%$ and $C = 10^2$. The test accuracy resulted is 95.23% for 7 classes and with 6 classes (without the neutral emotion) reached 98.11%. The confusion matrix for the MUG database is in Table 3.6.

Table 3.5: CK+ dataset Classification results for seven and six emotions denoted as A (Anger), D (Disgust), F (Fear), H (Happiness), Su (Surprise), S (Sadness) and N (Neutral) with accuracy in %

Emo-	7 Class with Neutral (N)							6 Class without Neutral (N)					
tion	A	D	F	H	S	Su	N	A	D	F	H	S	Su
A	**92**	2	0	0	4	0	2	**100**	0	0	0	0	0
D	2	**98**	0	0	0	0	0	0	**100**	0	0	0	0
F	0	0	**96**	0	0	0	4	0	0	**92**	0	0	8
H	0	0	0	**100**	0	0	0	0	0	0	**100**	0	0
S	0	0	4	0	**94**	0	2	0	0	0	0	**100**	0
Su	0	0	0	0	0	**98**	2	1	1	0	0	1	**97**
N	2	2	0	2	2	0	**94**						

Table 3.6: MUG dataset Classification results for seven and six emotions denoted as A (Anger), D (Disgust), F (Fear), H (Happiness), Su (Surprise), S (Sadness) and N (Neutral) with accuracy in %

Emo-	7 Class with Neutral (N)							6 Class without Neutral (N)					
tion	A	D	F	H	S	Su	N	A	D	F	H	S	Su
A	**92**	0	2	0	4	0	6	**98**	0	0	0	2	0
D	0	**100**	0	0	0	0	0	0	**100**	0	0	0	0
F	0	0	**88**	0	0	12	0	0	0	**98**	0	0	2
H	0	0	0	**99**	0	0	1	1	0	0	**99**	0	0
S	0	0	0	0	**94**	0	6	1	1	0	0	**98**	0
Su	0	1	8	0	0	**91**	0	0	0	3	0	1	**96**
N	0	0	0	0	0	0	**100**						

3.3 Experimental Results

For a comprehensive experimental evaluation of the presented method and comparison with other existing FER algorithms, CK+ and MUG datasets as well as the combination of the two are considered. The cross-validation results on FER accuracy are reported. The previous work reporting the highest FER accuracy based on the CK+ and MUG datasets targets 6 facial emotions except for "neutral" among the 7 emotions we target in this paper. Therefore, our evaluation was conducted for both 6 and 7 emotion classes. In addition, short video clips provided by the MUG dataset

were also tested for real-time FER performance.

3.3.1 Cross-Validation

In Section 3.2, the FER results using the optimized parameters for each CK+ and MUG dataset were reported. Five additional experiments were conducted for cross-validation between the two datasets and the performance of the merged dataset (CK+/MUG), which means seven sets of experiments in total:

- CK+ for training and CK+ for testing (Section 3.2)
- MUG for training and MUG for testing (Section 3.2)
- CK+ for training and MUG for testing (Table 3.7)
- MUG for training and CK+ for testing (Table 3.8)
- CK+/MUG for training and CK+ for testing (Table 3.9)
- CK+/MUG for training and MUG for testing (Table 3.10)
- CK+/MUG for training and CK+/MUG for testing (Table 3.11)

The experiments were conducted for classifying 7 distinctive emotions (i.e., anger, disgust, fear, happiness, sadness, surprise, and neutral) and repeated for 6 emotions excluding the neutral emotion.

Each of the CK+, MUG and CK+/MUG datasets was divided into the training set and testing set by random shuffling (80% for training; 20% for testing). Fig. 3.7 shows the results from all seven experimental scenarios listed above. When CK+ was used for training, the accuracy reached up to 98.38% for the CK+ test set, 99.67% for the MUG test set for 6-emotion classification and 96.6% for the CK+ test set and 95.99% for the MUG test set for 7-emotion classification. Using MUG for training, the results show 98.11% for the MUG test set and 98.59% for the CK+ test set for

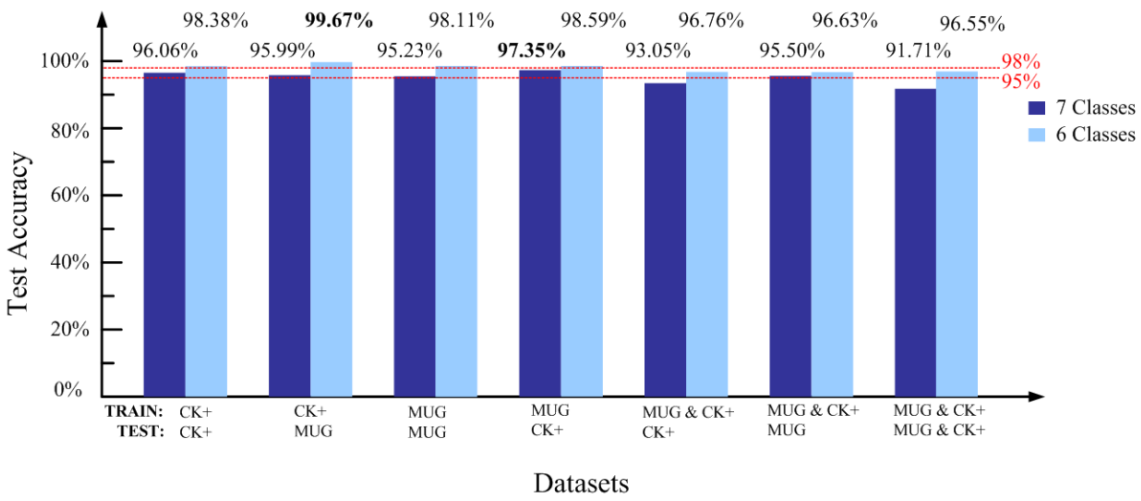


Figure 3.7: Cross validation results among CK+, MUG and merged (CK+ & MUG) dataset.

Table 3.7: Cross validation (CK+ for training and MUG for testing) results for seven and six emotions denoted as A (Anger), D (Disgust), F (Fear), H (Happiness), Su (Surprise), S (Sadness) and N (Neutral) with accuracy in %

Emo-	7 Class with Neutral (N)							6 Class without Neutral (N)					
tion	A	D	F	H	S	Su	N	A	D	F	H	S	Su
A	**93**	0	0	0	0	0	7	**98**	2	0	0	0	0
D	0	**97**	0	0	0	0	3	0	**100**	0	0	0	0
F	0	0	**92**	0	0	0	8	0	0	**100**	0	0	0
H	0	0	0	**100**	0	0	0	0	0	0	**100**	0	0
S	0	0	0	0	**100**	0	0	0	0	0	0	**100**	0
Su	0	0	0	0	0	**94**	6	0	0	0	0	0	**100**
N	3	0	0	0	2	0	**95**						

6-emotion classification. In 7-emotion classification, the accuracy was 95.23% for the MUG test set and 97.35% for the CK+ set. The combined CK+/MUG set has more amount of data but at the same time more diverse than individual datasets. While the FER accuracy still remained relatively high in all three cases (>96% for 6-emotion classification; >91% for 7-emotion classification), the results were not as accurate as the case using either only CK+ or MUG for training. The best performance was found when CK+ was used for training and MUG for testing in 6-emotion classification and

Table 3.8: Cross validation (MUG for training and CK+ for testing)results for seven and six emotions denoted as A (Anger), D (Disgust), F (Fear), H (Happiness), Su (Surprise), S (Sadness) and N (Neutral) with accuracy in %

Emo-	7 Class with Neutral (N)							6 Class without Neutral (N)					
tion	A	D	F	H	S	Su	N	A	D	F	H	S	Su
A	**97**	0	0	0	1	0	2	**97**	2	0	0	2	0
D	0	**100**	0	0	0	0	0	0	**100**	0	0	0	0
F	0	0	**93**	0	0	7	0	0	0	**95**	0	0	5
H	0	0	0	**96**	0	0	4	0	0	0	**100**	0	0
S	1	2	0	0	**94**	0	3	1	0	0	0	**99**	0
Su	0	0	0	0	0	**100**	0	0	0	1	0	0	**99**
N	0	0	0	0	1	0	**99**						

Table 3.9: Cross validation (CK+/MUG for training and CK+ for testing) results for seven and six emotions denoted as A (Anger), D (Disgust), F (Fear), H (Happiness), Su (Surprise), S (Sadness) and N (Neutral) with accuracy in %

Emo-	7 Class with Neutral (N)							6 Class without Neutral (N)					
tion	A	D	F	H	S	Su	N	A	D	F	H	S	Su
A	**98**	0	0	0	0	0	2	**97**	3	0	0	0	0
D	0	**82**	0	0	2	0	16	4	**91**	0	0	4	0
F	0	0	**100**	0	0	0	0	0	0	**100**	0	0	0
H	0	0	0	**100**	0	0	0	0	0	0	**100**	0	0
S	0	0	0	0	**89**	0	11	0	0	0	0	**100**	0
Su	0	0	0	0	0	**95**	5	0	1	1	0	2	**95**
N	0	9	0	0	6	0	**85**						

when MUG was used for training and CK+ used for testing in 7-emotion classification.

Table 3.7 to Table 3.11 show all confusion matrices based on cross-dataset validation with 7 and 6 emotion classifications. For each facial emotion specifically, "happiness" has the overall best average recognition rate at 98.80%, followed by "anger" and "surprise" both at 95.80%. "Disgust", "fear" and "normal" are slightly lower, with recognition rates of 92.24%, 92.20% and 93.40%, respectively. "Sadness" has the lowest recognition rate at 90.60%; However, though it is lower than any other emotion class, the accuracy still remains at a high level (>90%).

Table 3.10: Cross validation (CK+/MUG for training and MUG for testing) results for seven and six emotions denoted as A (Anger), D (Disgust), F (Fear), H (Happiness), Su (Surprise), S (Sadness) and N (Neutral) with accuracy in %

Emo-	7 Class with Neutral (N)							6 Class without Neutral (N)					
tion	A	D	F	H	S	Su	N	A	D	F	H	S	Su
A	**97**	1	1	0	0	0	0	**100**	0	0	0	0	0
D	5	**95**	0	0	0	0	0	5	**95**	0	0	0	0
F	0	0	**90**	0	0	10	0	2	0	**90**	0	3	5
H	0	0	0	**98**	0	0	2	0	1	0	**98**	0	1
S	1	4	0	0	**88**	1	5	4	1	0	0	**95**	0
Su	0	0	5	0	0	**94**	1	0	1	1	0	0	**98**
N	0	0	0	0	0	0	**100**						

Table 3.11: Cross validation (CK+/MUG for training and CK+/MUG for testing) results for seven and six emotions denoted as A (Anger), D (Disgust), F (Fear), H (Happiness), Su (Surprise), S (Sadness) and N (Neutral) with accuracy in %

Emo-	7 Class with Neutral (N)							6 Class without Neutral (N)					
tion	A	D	F	H	S	Su	N	A	D	F	H	S	Su
A	**94**	3	2	0	0	0	2	**97**	2	0	0	1	0
D	4	**88**	0	0	1	0	7	1	**96**	2	0	1	0
F	2	0	**86**	0	2	8	3	0	0	**89**	0	0	11
H	0	0	0	**100**	0	0	0	1	0	0	**99**	0	0
S	2	7	1	0	**82**	0	8	1	2	0	0	**96**	1
Su	0	0	3	0	0	**96**	1	0	0	2	0	1	**97**
N	2	6	1	0	3	1	**88**						

3.3.2 Algorithm Efficiency Improvement

The size of data is directly related to the training and detection time. In an attempt to further improve time efficiency, the number of selected FSs was reduced from 13 to 8. Considering symmetry in human faces, either the left or right side of FSs related to the eyes, eyebrows, and lip, was used. Table 3.12 shows the selected eight FSs, using the left side and common features.

Table 3.12 presents the selected FS with description by only keeping left facial features in the experiments. The results show that with the CK+ dataset, FER

Table 3.12: Selected FSs with description, and associated landmarks for algorithm efficiency improvement.

FS	Description	Index of Landmarks
FS_1	Left inner brow raiser	20-22
FS_3	Left outer brow raiser	18-20
FS_5	Left upper lid raiser	37-40
FS_7	Left lid tightener	37, 40-42
FS_9	Nose wrinkler	32-36
FS_{10}	Upper lip raiser	49-55
FS_{11}	Left lip corner	49, 61, 68
FS_{13}	Lip stretched/tightener	49, 55-60

accuracy was 83.96% with 213.03 (sec) as training time for 7-emotion classification, and 88.67% with 177.46 (sec) as training time for 6-emotion classification. For the experiment on MUG dataset, 7-emotion classification resulted in 89.02% FER accuracy with 698.85 (sec) training time, and 6-emotion classification resulted in 89.88% recognition accuracy with 659.95 (sec) training time.

Table 3.13: CK+ dataset Classification results with only keeping left facial features for seven and six emotions denoted as A (Anger), D (Disgust), F (Fear), H (Happiness), Su (Surprise), S (Sadness) and N (Neutral) with accuracy in %

Emo-	7 Class with Neutral (N)							6 Class without Neutral (N)					
tion	A	D	F	H	S	Su	N	A	D	F	H	S	Su
A	**82**	0	2	0	11	0	4	**82**	9	2	2	4	0
D	8	**83**	0	0	0	0	8	10	**85**	0	0	5	0
F	4	0	**88**	0	0	0	8	8	0	**88**	4	0	0
H	3	1	1	**93**	0	0	1	0	1	0	**99**	0	0
S	39	4	0	0	**36**	0	21	21	11	0	0	**68**	0
Su	0	1	0	0	0	**94**	5	1	1	1	0	2	**94**
N	12	2	3	0	0	0	**83**						

Table 3.13 and Table 3.14 show the confusion matrices of the classification results by 7 and 6 emotion classes based on CK+ and MUG datasets. Table 3.15 presents the results in recognition rate and training time based on different number of selected

Table 3.14: MUG dataset Classification results with only keeping left facial features for seven and six emotions denoted as A (Anger), D (Disgust), F (Fear), H (Happiness), Su (Surprise), S (Sadness) and N (Neutral) with accuracy in %

Emotion	7 Class with Neutral (N)							6 Class without Neutral (N)					
	A	D	F	H	S	Su	N	A	D	F	H	S	Su
A	**78**	3	0	0	8	0	11	**84**	2	2	5	8	0
D	3	**92**	0	3	1	0	1	5	**95**	0	0	0	0
F	0	0	**80**	2	0	12	5	2	2	**76**	5	2	12
H	0	1	0	**98**	0	0	1	1	3	1	**95**	0	0
S	4	0	4	1	**81**	0	9	3	0	4	1	**91**	0
Su	0	0	4	0	0	**95**	1	0	0	11	0	0	**89**
N	4	1	0	0	6	0	**89**						

FSs. CK+ dataset has 1,872 images and MUG dataset has 2,658 images, therefore the training process takes relatively longer for MUG dataset than CK+ dataset. Compared to the experimental results not removing all right side FSs, the accuracy was decreased from 95.23% to 89.02% for 7-emotion classification and from 98.11% to 89.88% for 6-emotion classification of the MUG dataset. As for the CK+ dataset, only keeping the left side FS resulted in a decrease of FER from 96.06% to 83.02% with 7-emotion classification and from 98.38% to 89.88% with 6-emotion classification. However, this can reduce the training time from 416 (sec) to 213 (sec) on the CK+ dataset, and 1400 (sec) to 714 (sec) on the MUG dataset. The detection speed for images with a resolution of 640×480 pixels can also be increased from 6.67 to 12.06 fps. Consequently, this efficiency improvement method can be applied on some scenarios which require a higher detection speed, but a lower recognition rate will be reported.

3.3.3 Real-time FER Performance

To test the technical integrity of the entire FER system, short video clips available from the MUG dataset were used for real-time performance. 18 videos clips in total

Table 3.15: Comparison with different Number of Selected FSs

Dataset	Number of FS	Class	Testing Accuracy	Training Time (sec)
CK+	13	6	98.38%	367.03
CK+	13	7	96.06%	408.58
MUG	13	6	98.11%	1203.90
MUG	13	7	95.23%	1399.96
CK+	8	6	88.67%	177.46
CK+	8	7	83.96%	213.03
MUG	8	6	89.88%	569.95
MUG	8	7	89.02%	698.85

recorded by 3 persons–not included in the training sessions were selected for testing. 6 videos from each person present their faces dynamically changing from neutral to 6 emotion peak and back to neutral.

Fig. 3.8 shows a time lapse sequence of one of the video clips used in this evaluation. For each person, 6 video clips were merged into a single video clip with exhibited facial emotions changing in sequence of Angry→ Disgust→ Fear→ Happy→ Sad→ Surprise. We note that each clip starts with a neutral face, changes to one of the six emotions, and returns to a neutral face. Each video clip lasted around 7 (sec) and thus the merged video for each person lasted around 49 (sec). For this real-time test, the SVM classifier for 7 emotion classes was trained using the MUG dataset.

Fig. 3.9(a) showed the real-time results when the presented FER system was directly used for the videos. The x-axis represents time in seconds and the y-axis shows the seven emotions. The results from the real-time data are noisy due to high sensitivity of the SVM classifier and dynamic facial emotion changes in the videos. Since facial images during the transition from one emotion to another are not included in the training, the classification can be somewhat unreliable. In addition, some emotions, such as "fear" and "surprise" can be naturally confusing. Furthermore, human facial expressions in real time dynamically change and subtle differences can be even difficult for humans to fully understand. The field of vision-based automatic

Figure 3.8: Video clip sample from MUG dataset.

FER cannot replicate a human's perception of facial emotions because of it is not a discrete classification problem and also involves significant individual heterogeneity. Automatic FER aims to capture at least "commonly" recognizable facial emotions as accurately as possible.

For real-time FER applications, such as social robots or other types of interactive systems, noisy FER results shown in Fig 3.9(a) may not be ideal. In such systems, a user's emotion status may be used as a part of control inputs which may lead to highly unreliable behavior in the system's response. Adding a finite impulse response

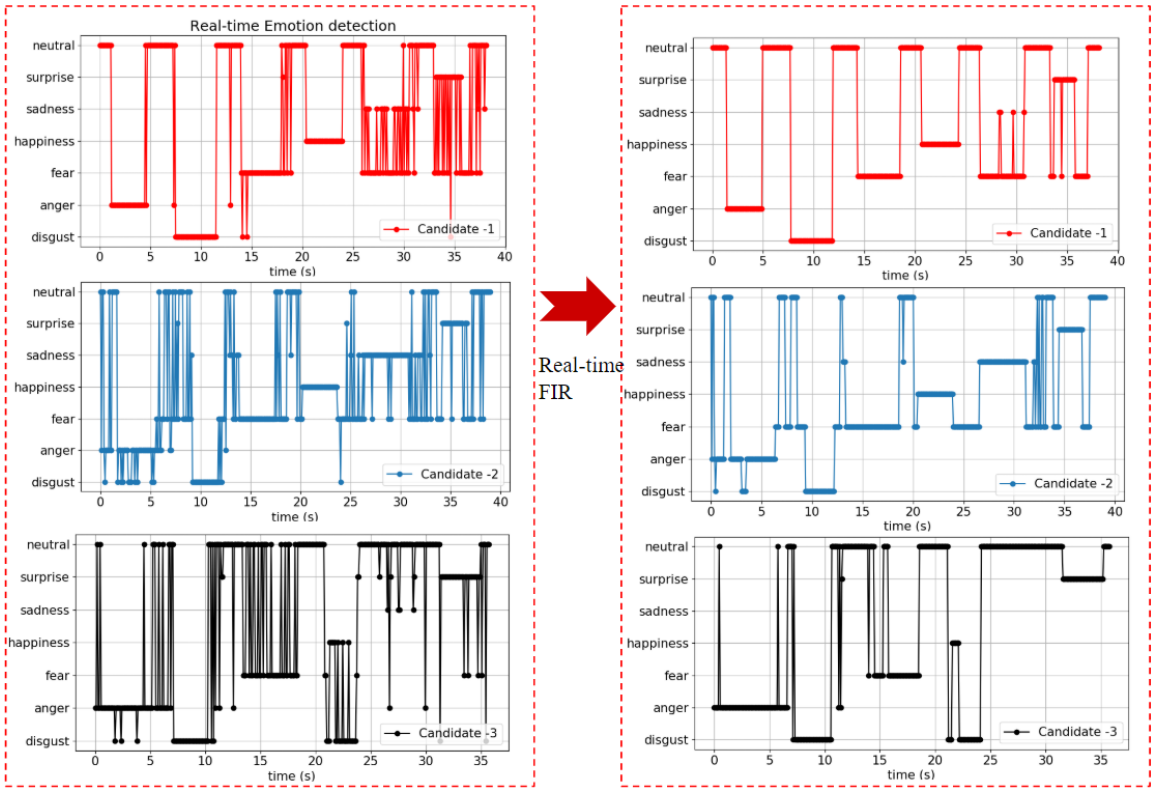


Figure 3.9: Real-time result on video clips from MUG dataset; (a) before applying FIR filter; (b) after applied FIR filter.

filter by introducing a small delay can reduce the noise during transition or between somewhat confusing emotions:

$$Y_i = \text{argmax } (count(Y_{i-3}, Y_{i-2}, \cdots, Y_{i+3})) \quad s.t\ i > 3 \tag{3.12}$$

where *count* is a function which returns the occurrence of each elements in one set of data, and Y is the SVM classifier resulting in predicted labels. For the i^{th} frame in real-time test, this filter captures three frames before and three frames after the i^{th} frame for stabilizing real-time response with small latency. Considering the system speed of 6.67 fps, the latency is kept at around 0.4 second. The results after applying the finite impulse response filter are shown in Fig 3.9(b).

3.4 Discussion

Table 3.16: Recent FER Approaches Comparison.

Dataset	Class	Feature	Real time	Open Source	Method	Accuracy (%)
EmotioNet	6	AU intensity [11]	✓	✓	KSDA	80.90
CK+	6	Discriminative response map fitting[28]	✗	✓	SVM	94.14
MUG	6	Triangle based geometric feature [29]	✗	✓	SVM	95.50
CK+	6					97.80
CK+	6	Geometric 8 facial points [54]	✗	✓	SVM	83.01
CK+	7					73.63
CK+	7	SFS Geometric distance variation [55]	✗	✓	SVM	88.70
MUG	6	Facial manifold structure [56]	✗	✓	SVM	92.76
CK+	6					94.31
CK	6	Graph-preserving sparse GSNMF [57]	✗	✓	Nearest-neighbor	93.50
CK	7					94.30
MUG	6	**Landmark Curvature (LC) and Vectorized Landmark (VL)**	✓	✓	SVM	**98.11**
MUG	7					**95.23**
CK+	6					**98.38**
CK+	7					**96.06**

The presented method combining two geometric features followed by the parameter tuning process has achieved a higher accuracy in FER than recently developed methods using the same datasets. Table 3.16 lists seven previously presented open-source based works using similar, but different, facial geometric features to ours with a selected classifier. Most of these methods, except for EmotioNet [11], are not ideal for real-time applications because the facial features were obtained from computationally complex procedures while also exhibiting relatively lower recognition rates. EmotioNet, on the other hand, can be used in real time; however, the FER accuracy was still low – 80.90%. Recent work using triangle-based geometric features presented in [29] showed improved accuracy for both CK+ and MUG datasets. This method

uses 370 triangle features in total, where each triangle feature has 4 computational components. While direct comparisons in terms of the processing speed have not been performed, the amount of data required for training and testing is significantly higher than our method. In our method, only 41 landmarks each with three calculated values (κ, d and θ) are used. More importantly, we have achieved the highest accuracy among all algorithms compared in the table.

The main technical contribution of this paper is the comprehensive procedural parameter optimization method. Experimental tuning process performed on the two distinctive datasets confirmed that the results are consistent. This implies that these selected parameters may be used for other datasets without repeating the parameter tuning process. While the presented work focused on achieving high FER accuracy while maintaining time efficiency for real-time applications, the method can be customized for higher efficiency with slightly lower accuracy if fast processing is critical. For example, considering largely symmetric facial geometric features, only left or right sides of facial segments for the eyebrows, eyelids, or cheeks can be used. This can reduce the training time from 416 to 213 (sec) with 89.30% recognition rate on CK+ dataset, 1400 to 714 (sec) with 86.74% recognition rate on MUG dataset. The detection speed for images with a resolution of 640×480 pixels can also be increased from 6.67 to 12.06 fps. Real-time evaluation results shown in Fig. 3.9 also reveal that the presented algorithm is efficient, accurate and relatively stable, and therefore suitable for many embedded applications, particularly in human-computer/robot interaction. Further evaluation of this method may involve additional datasets in order to confirm that the same set parameters also result in the highest performance and conduct additional cross-validation.

Chapter 4

FER for HRI application

Social robots feature unique technical functionalities that enable humans to communicate and interact with them using social cues, such as language, gestures, and facial displays. Several hardware platforms have been developed and some are commercially available; their application domains have been dramatically expanding from simple entertainment to health care, education, and specialized services. Unlike the robots developed for specific, often repetitive tasks, social robots use social cues provided by human users as control inputs and aim to generate socially acceptable responses. Many robots are designed not only to socially interact with users, but also to provide assistance, service, and/or care through interaction. In this chapter, a low-cost robot developed in the dirLAB is introduced and the technical feasibility on FER is demonstrated. This section of the thesis is resulted from close collaboration with Daniel Hayosh, a graduate student in the dirLAB.

4.1 Development of the Robot

"Woody" was developed as an open-source-based Do-It-Yourself (DIY) robotic hardware platform that can be constructed by a few college or high school school students with low-level engineering training within one or two days. The robot's mechanical

design is simple and modularized for easy construction, customization, and repair. Unlike most of existing robots made of plastic and/or metal, Woody uses laser-cut plywood for its mechanical structures which provides user-friendly appearance and an environmentally friendly fabrication process.

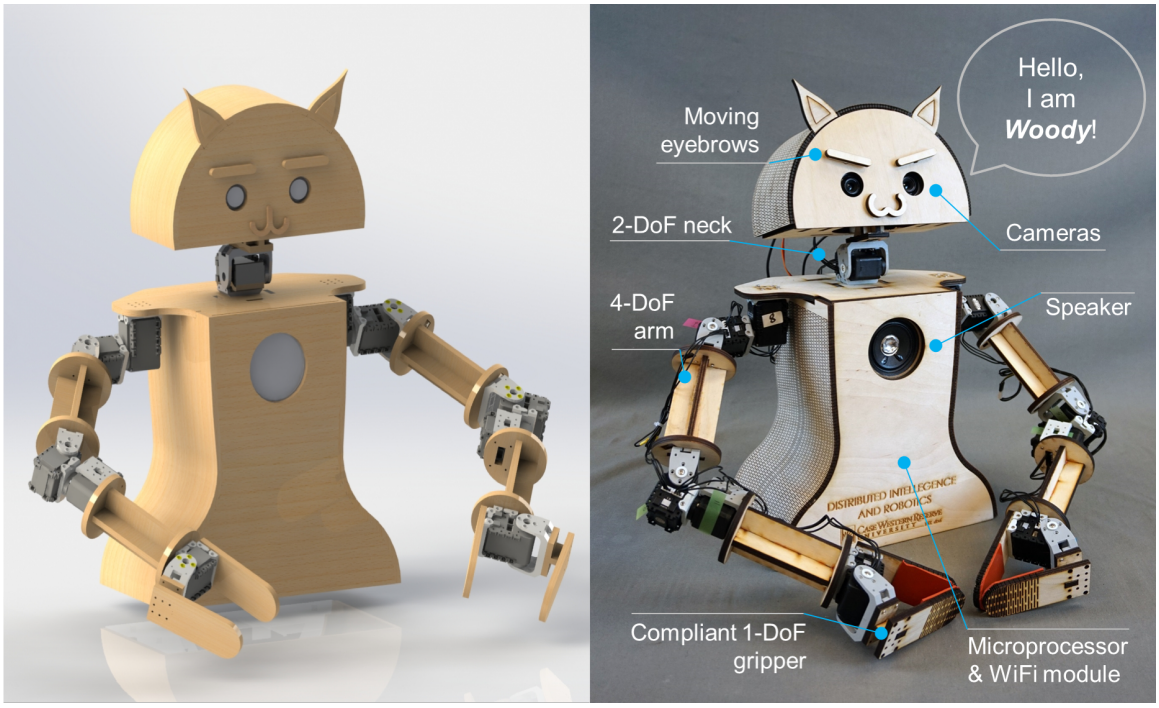


Figure 4.1: CAD model of Woody (left) and fully assembled hardware prototype (right).

4.1.1 Mechanical Design

Fig. 4.1 shows the fully assembled design of Woody in CAD and the physical prototype. The robot has 14 degrees of freedom (DoF): two for its neck, two for the eyebrows, four in each arm, and one in each gripper. The current prototype uses two mini servo motors for the eyebrows and 12 Dynamixel AX-12A motors for the rest. It is equipped with two cameras, a microphone, and a speaker. A Raspberry Pi or computer can be used as the main processing board. Portable devices, i.e. raspberry Pi, can be placed inside the torso. The speaker is mounted at the level of Woody's

chest.

The head of Woody is mounted on a 2-DoF neck for generating pan and tilt motions. Two cameras with built-in microphones are installed on the head in the locations of Woody's eyes, and thus can make Woody more intuitive when implementing vision algorithms. Positioned directly above the cameras are two small servos which control the eyebrows. Mechanically generating facial emotions involves high-level mechanical, electrical, and computational complexities. Moving eyebrows is a simple yet effective way of generating facial emotions. Alternatively, an LCD screen may be installed to display animated images. The head features slots for the ears and mouth where different ear and mouth designs can be inserted, making Woody's face customizable.

Woody contains two 4-DoF arms. Two motors are installed at the shoulder and another two at the elbow. An additional motor controls the gripper at the tip of each arm. Links in each arm is made of two wooden pieces assembled and glued together.

4.1.2 Electrical Design

The electrical design is simple to assemble, Fig. 4.2 shows the circuit diagram with all embedded electronic components in Woody. The robot itself is designed to be a semi-autonomous system with basic embedded functions, such as data transmission and motor control only. The portable control board (Raspberry Pi 3 model B) has a relatively high computational capability compared to other processing boards for embedded applications. It is sufficient to handle the basic image processing on board. It also has a Wi-Fi module to communicate within a local area network (LAN). Cameras, speaker, and microphone are also connected to this control board via serial ports.

The system controls two types of motors: 12 Dynamixel AX-12A and 2 Tower Pro SG90. The Dynamixel motors offer accurate motor position control and feed-

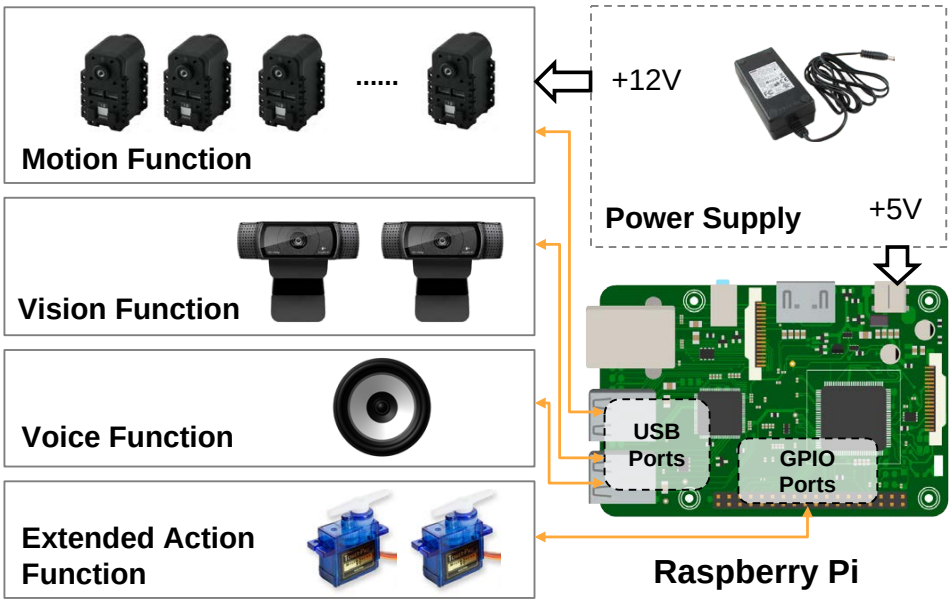


Figure 4.2: Circuit diagram (block diagram) of the embedded electronics in Woody.

back. They are used to generate the motion of the neck and two arms with grippers. The Tower Pro motors are used for moving the eyebrows to generate simple facial expressions. The robot can be battery powered or plugged into a continuous power source. The control board has extra serial ports for potential extension in the functionality.

4.1.3 Kinematic Analysis of the Arm

Forward Kinematics

Table 4.1: D-H parameters for 4-link Woody left arm

link	θ_i	d_i	a_i	α_i
1	0	0	0	-$\pi/2$
2	θ_1	l_1	0	$\pi/2$
3	θ_2	0	l_2	0
4	θ_3	0	0	$\pi/2$
5	θ_4	$l_3 + l_4$	0	0

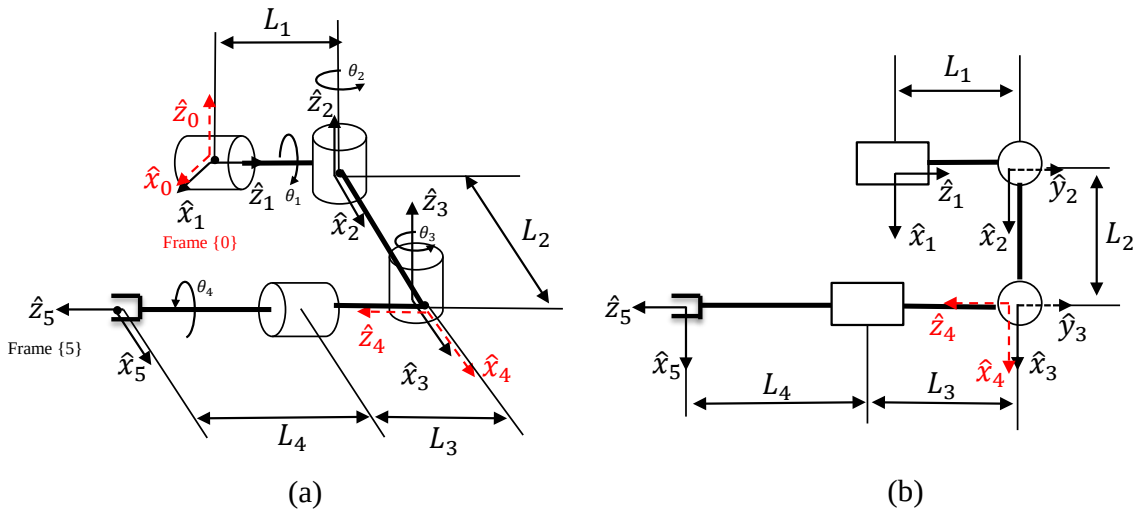


Figure 4.3: D-H coordinate frame assignment for the manipulators; Note the red frames are auxiliary frame which illustrate the particular Woody's arm installation. (a) woody left arm frame assignment; (b) frame assignment from top view.

Each arm of Woody is a 4-DoF RRRR manipulator. Forward kinematics equations of the arm, from the shoulder to the end-effector, are derived using the Denavit-Hartenberg (D-H) parameterization. For simplification, c_i represents $\cos\theta_i$ and $\cos(\theta_1+\theta_2)$ represents c_{12}. The two arms are mirror images, and therefore, the left arm of Woody is used for both forward kinematics and inverse kinematics derivation. Table 4.1 lists the D-H parameters for the left arm, note that D-H parameters of the link 1 in table 4.1 only represents the rotation of base frame because of the hardware installation. Reference frames attached to the arm and the parameters are also illustrated in Fig. 4.3.

Based on this parameterization, the rigid-body transformation from the frame {0}

to the frame {5} is calculated as:

$$T_5^0 = \begin{bmatrix} s_1s_4 + c_4c_1c_{23} & c_4s_1 - c_1s_4c_{23} & c_1s_{23} & d_x \\ c_4s_{23} & s_4s_{23} & -c_{23} & d_y \\ c_1s_4 - s_1c_4c_{23} & c_1c_4 + s_1s_4c_{23} & -s_1s_{23} & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.1}$$

$$\begin{aligned} d_x &= c_1s_{23}(l_3 + l_4) + l_2c_1c_2 \\ d_y &= l_1 - c_{23}(l_3 + l_4) + l_2s_2 \\ d_z &= -s_1s_{23}(l_3 + l_4) - l_2c_2s_1 \end{aligned} \tag{4.2}$$

After frame transformation, d_x, d_y and d_z showed the end-effector pose in the 3D Cartesian space referring to the base frame. Note that d_x, d_y and d_z are only determined by joint angle θ_1, θ_2 and θ_3, therefore, the workspace of robot arm can be illustrated by plotting the trajectory of the end-effector within a particular angle range of $joint_1$, $joint_2$ and $joint_3$. For Woody's left arm, the angle range for $joint_1$, $joint_2$ and $joint_3$ are -$\pi/2 \sim \pi/3$, $0\sim \pi/2$ and $0\sim \pi/2$ based on specific mechanical designing and assembling of the robot arm. Thus, the workspace for both Woody's left and right arm can be plotted in 3D simulation space. (see Fig. 4.4)

Inverse Kinematics

Equations for inverse kinematics of the arm are solved by geometric approaches. As shown in Fig. 4.5, given the two frames {0} and {5}, the values of θ_1, θ_2, and θ_3 can be solved geometrically as illustrated in Fig. 4.5 (b), (c). Given end-effector pose, $[d_x, d_y, d_z]^T$,

$$\theta_1 = \tan^{-1}(\frac{-d_z}{d_x}) \tag{4.3}$$

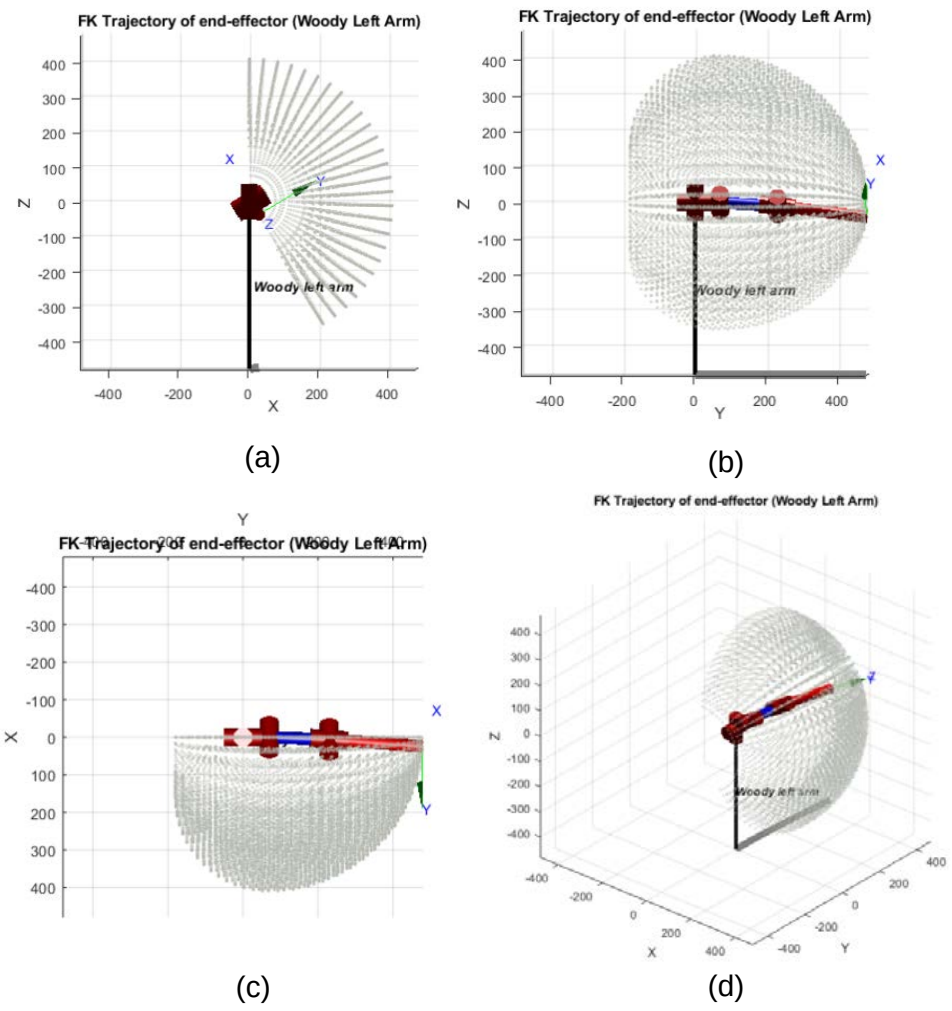


Figure 4.4: Workspace Shown by Forward Kinematics Trajectory of End-effector; (a) side view; (b) front view; (c) top view; (d) axonometric drawing.

Then, consider seeing the robot arm along the direction of the arrow in Fig. 4.5 (b). The left arm appears to be a typical two-link planar manipulator. Geometrically, define $d'_x = \frac{d_x}{cos\theta_1}$ and $d'_y = d_y - L_1$, θ_3 is given by

$$\theta_3 = cos^{-1}\left(\frac{-{d'_x}^2 - {d'_y}^2 + {L_2}^2 + (L_3 + L_4)^2}{2L_2(L_3 + L_4)}\right) - \frac{\pi}{2} \tag{4.4}$$

For deriving θ_2, β should be firstly calculated by θ_3

$$\beta = tan^{-1}\left(\frac{(L_3 + L_4)sin(\frac{\pi}{2} - \theta_3)}{(L_3 + L_4)sin(\frac{\pi}{2} - \theta_3) + L_2}\right) \tag{4.5}$$

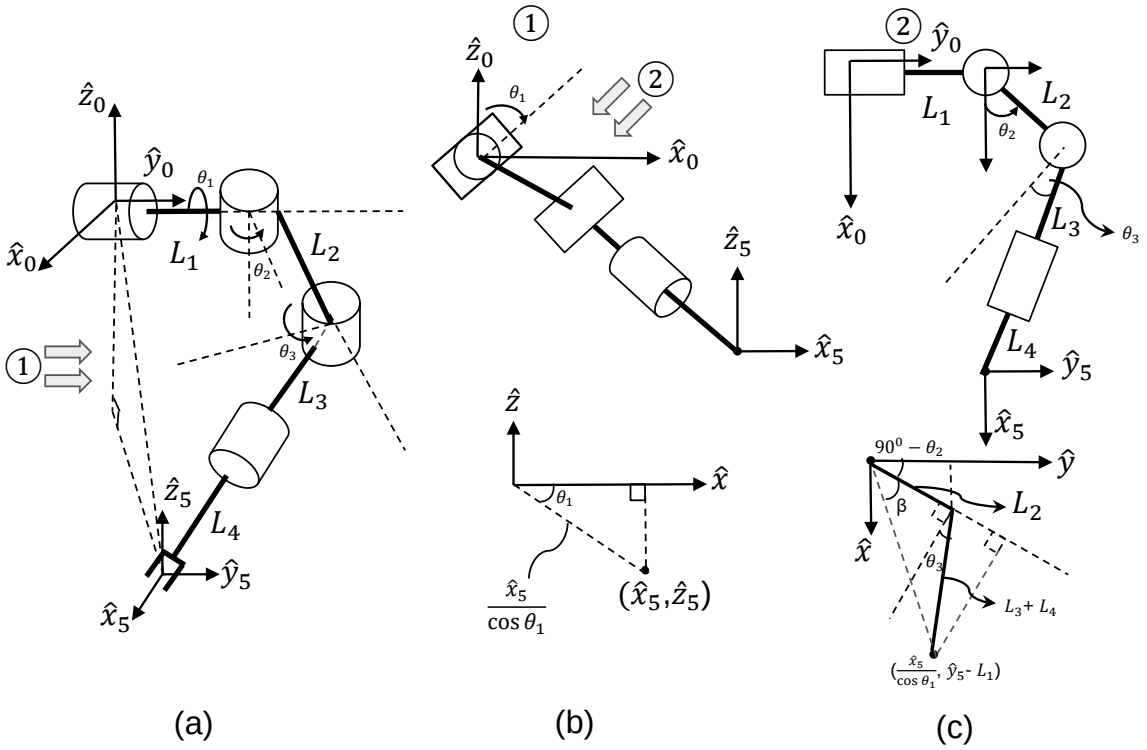


Figure 4.5: Geometric model of Woody's left arm (point downwards) with base frame {0} and the end-effector frame {5}; (a) geometric model; (b) geometric model seen from the view ① (side view); (c) seen from the view ②.

Therefore, θ_2 is given by

$$\theta_2 = \begin{cases} tan^{-1}(\frac{d'_y}{d'_x}) + \beta & d_y \geq 0 \\ \beta - tan^{-1}(\frac{d'_y}{d'_x}) & d_y < 0 \end{cases} \tag{4.6}$$

Note that Fig. 4.5 only includes the situation which d_y is positive, when d_y is a negative value, d'_y should be changed as $d'_y = -d_y + L_1$, the joint angle θ_2 should also be updated according to eq. (4.6).

In the previous section, the range for each joint angle is fixed, and all different poses of robot arm have been considered. The above solutions are the general form.

4.2 Interactive Features

In the field of HRI, a majority of emotion-related work focuses on robots expressing emotions across modalities like using non-anthropomorphic colored LEDs to communicate basic emotions [58] or producing multi-modal behaviors, like gestures that accompany facial expressions, to better convey a robots emotional state [59]. Therefore, Woody, as the low-cost HRI platform, must match the its emotions with humans emotions. Human emotions, especially facial emotions, can be categorized into 7 classes: anger, disgust, fear, happiness, neutral, sadness and surprise. Thus, Woody is designed to demonstrate these emotions through its features as well.

4.2.1 Facial Features

Two small servos mounted inside of Woody's head above two cameras serve as the actuators of the eyebrow movements. The eyebrows can both point upwards, downwards and stay in the center. Combined with varied head features- ears, nose and mouth, Woody can demonstrate different emotions along the spectrum from positive emotional state (i.e., happiness) to negative emotional state (i.e., anger or sadness).

Fig. 4.6 shows the eyebrow movements of Woody with various head features. Customizable head features, including eyebrows, nose/mouth, and ears, are designed for Woody to interact with users among different ages or gender. In this way, users' engagement/attention can be triggered in the manner of their own preference.

4.2.2 Gestures

In human-human collaboration and interaction, cooperative gestures play a key role in helping to communicate intent, instruct, lead, and build rapport. Humans communicate cooperatively, to inform others and to share interests from as early as 14 months of age [60]. In HRI, robots capable of recognizing human gestural cues (i.e.,

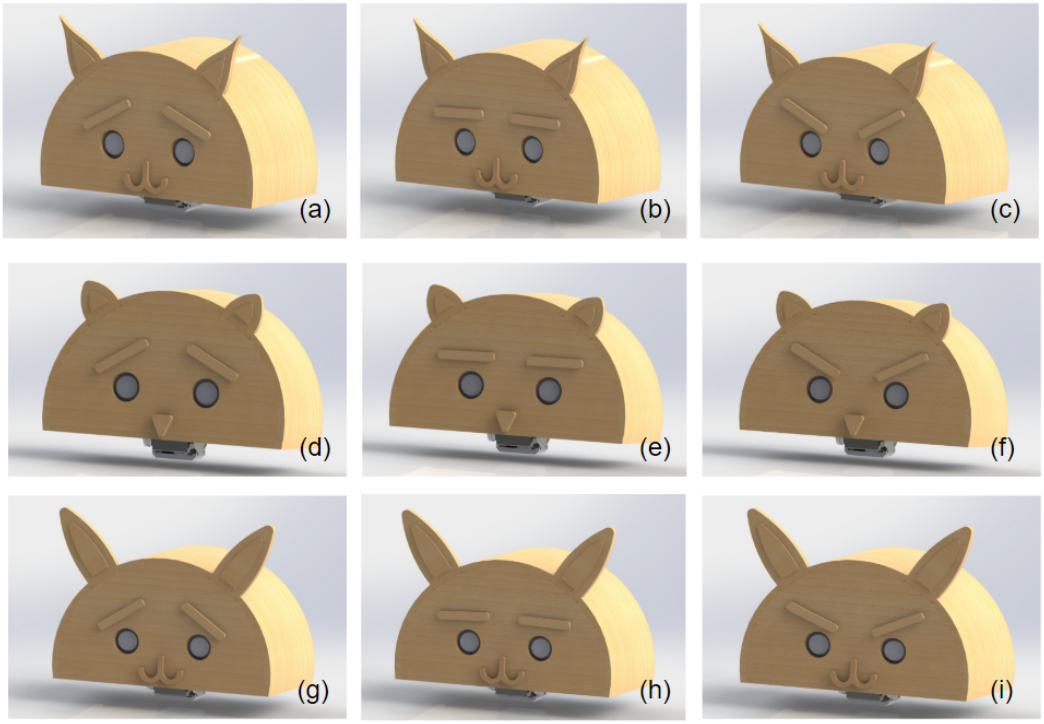


Figure 4.6: Woody's head features with varied eyebrow movements. (a)∼(c) show "cat" head feature with eyebrow movements from sadness to anger; (d)∼(f) show "bear" head feature with corresponding eyebrow movements; (g)∼(i) show "rabbit" head feature with corresponding eyebrow movements.

facial emotions) is important, however, an equally important area to successful HRI is making robots capable of generating meaningful, recognizable gestural cues to humans [61]. Thus, Woody is designed as a humanoid robot to be capable of generating recognizable gestural cues related to its emotion state.

Fig. 4.7 demonstrates some examples of recognizable gestures that Woody can perform in its interaction with users. Fig. 4.7 (a) is Woody waving to the user by performing a basic greeting gesture, Fig. 4.7 (b) is when Woody recognized the facial emotion of human as "anger" or "sadness", Woody is doing "nodding" as the gesture of expressing compassion and feeling sorry. Fig. 4.7 (c) illustrated the gesture of "weeping" once Woody detected the user's emotion state is "sadness", Fig. 4.7 (d) showed Woody is lifting both of its arms to express its emotion state is "high" and "excited" when the user has a happy facial expression. Truly, in real life scenarios,

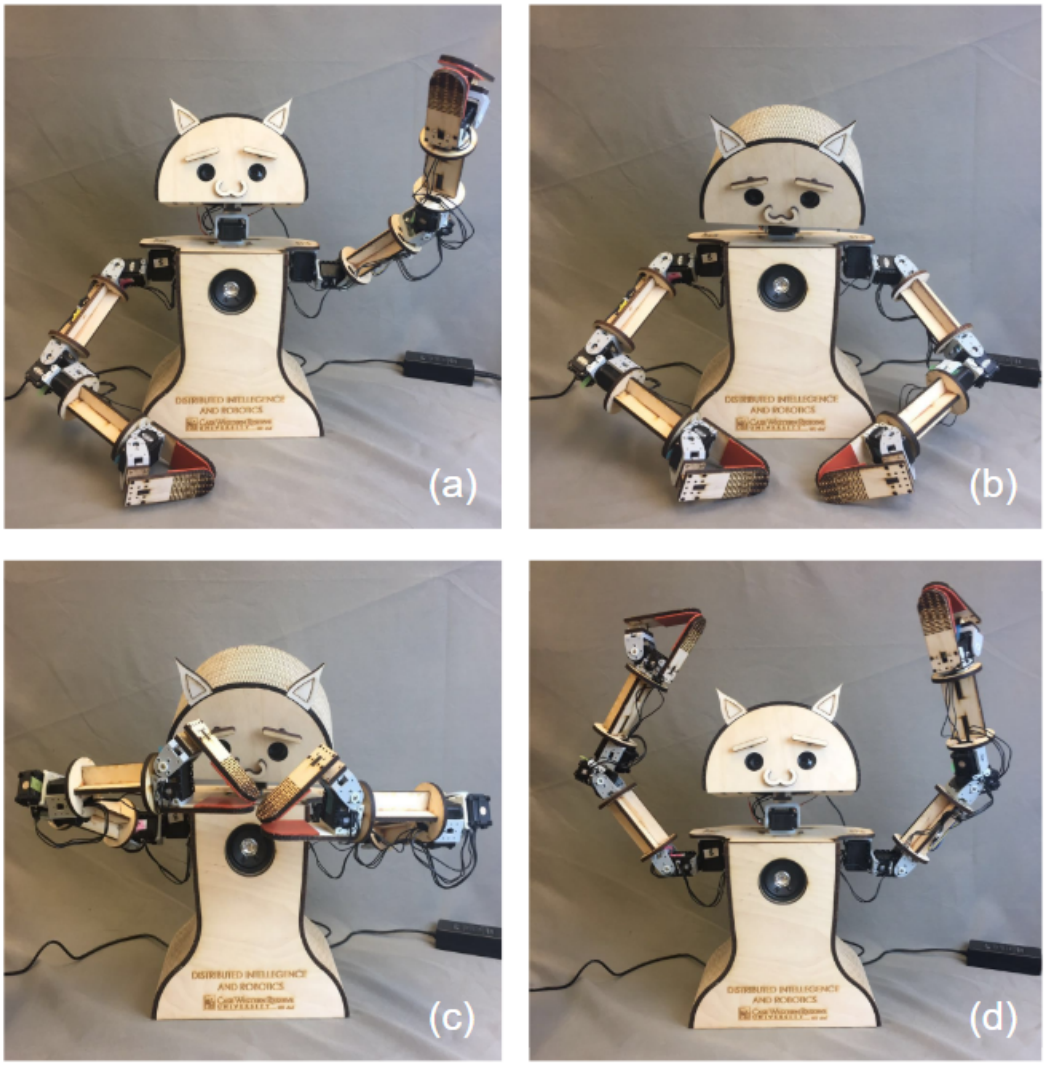


Figure 4.7: Woody's recognizable gestural cues. (a) shows "wave" gesture; (b) shows "nod" gesture; (c) shows "weep" gesture; (d) shows "excited" gesture.

these basic gestures are limited. Therefore, another functionality of this social robot platform is developed for users to record their preferred gestures based on their own perspective. For example, parents can record extra gestures on Woody and use those gestural cues to interact with their kids for building trust between their kids and the robot.

4.3 Graphical User Interface

In order to offer an easy-to-use way for users to build this DIY social robot platform and program it the way they want, a graphical user interface (GUI) was developed using python GTK+ 3.0 graphic user interface library under Ubuntu system.

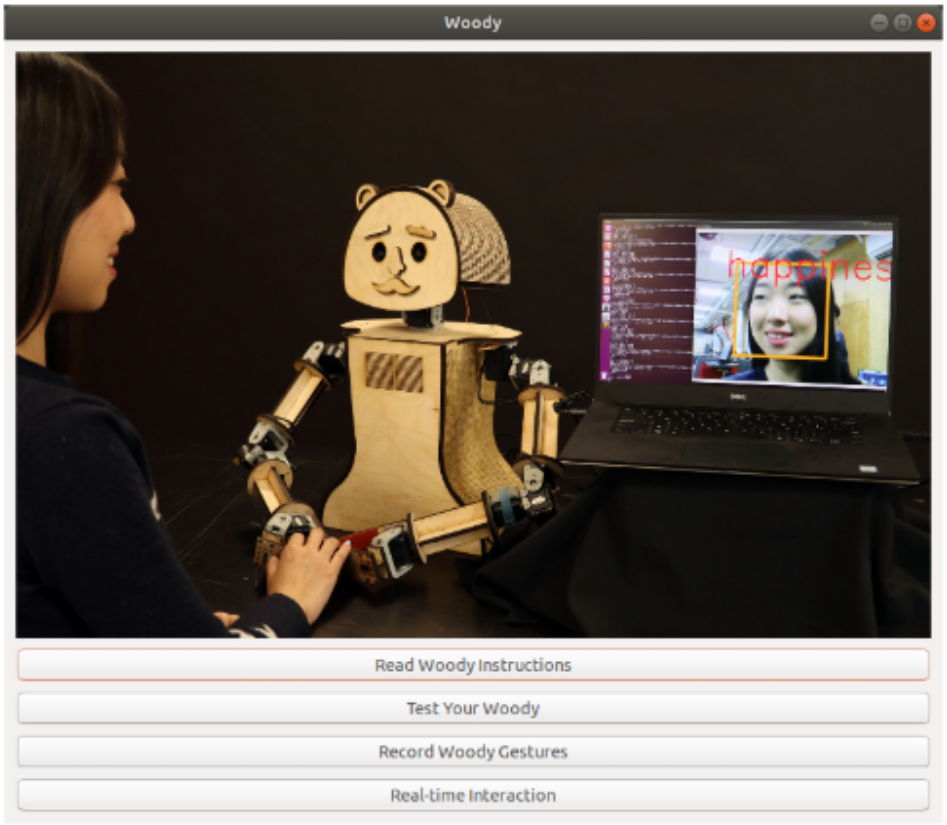


Figure 4.8: Woody GUI main menu.

The GUI's main menu (in Fig. 4.8) has four buttons which contain the basic and most important functions of this system. The buttons are designed using the "box container" feature from GTK+ library and are vertically packed together. The "Read Woody Instructions" button can guide users to a new page with text which contains mechanical assembly procedure and electronic set-up process. The second button "Test Your Woody" is designed for testing if the motor functions of the robot are correctly set-up and whether cameras connections are normal. A dialogue is supposed

to appear if something is not right for initialization. The third button "Record Woody Gestures" is the main feature of this system which provides customizable gestures recording function for users to make their own intuitive gestural cues of the robot, name them and add the gestures to the interaction stage.

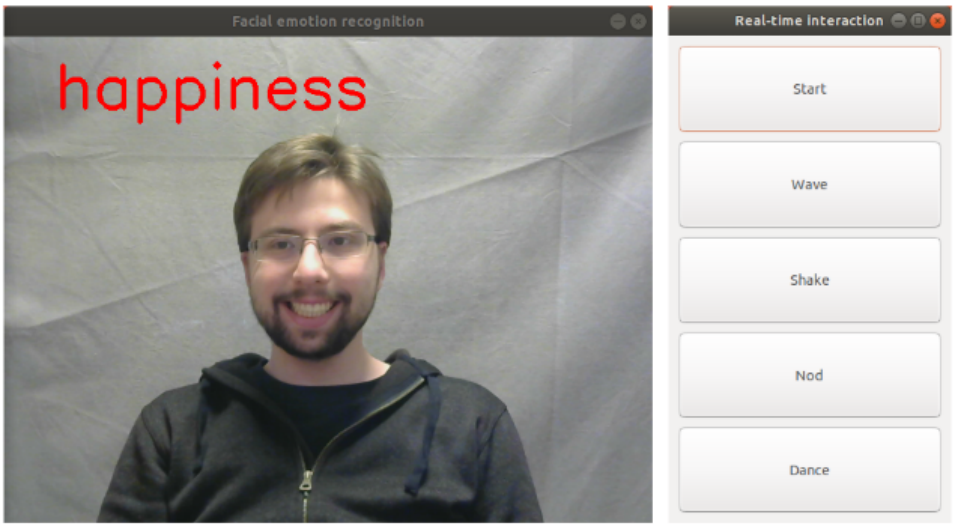


Figure 4.9: Real-time interaction function interface. Left: window shows real-time FER results; Right: window with generated gesture buttons for interaction.

Fig. 4.9 shows the window that appears when a user clicks the button "Real-time Interaction". Two separate windows pop up next to each other. Window Fig. 4.9 (left) presented the result of real-time FER, window Fig. 4.9 (right) presented a series of buttons named by user generated gestures. Two functions, the vision processing and motor controlling can run at the same time by using the "threading" package in python. Therefore, in the HRI scenarios, Woody can integrate FER with making corresponding social gestural cues so that the user can sense the intuition of the robot. Additionally, the GUI is easy to use. For example, this can benefit parents monitoring their kids emotional states, as they can make the robot perform gestures to improve human-robot social interactions. For another example, administrators of clinical psychological trials can use this GUI and system to make participants get more engaged during the tests.

Chapter 5

Conclusion and Future Work

This thesis describes three significant contributions. First, it provides an optimization method for CNN based FER by introducing pre-processing filters to adjust brightness and contrast, and conducts edge detection on input batch of images. In this way, it improves the CNN performance a great deal in terms of fixed epochs and training time efficiency. Second, this thesis proposes a new set of combined geometric facial features associated with AUs for further improving FER in terms of recognition rate and training time. It also performs the real-time applications on proposed algorithms and results in stable recognition performances using both the standard testing video clip and a live video stream from a camera. Meanwhile, it includes a comparison to the most recent FER technique based on geometric features, and proves to be effective and suitable for real-time implementation among the recent work in FER. Third, this thesis introduces a hardware and software FER application platform in HRI: the social robot "Woody". It describes clearly how FER can be integrated with other HRI functionality for social robots, as well an easy-to-use GUI for demonstrating robot embedded FER function. This thesis shows great potential in the FER and HRI fields.

Future work for the first CNN-based FER method includes 1) addressing the use of

CNN for FER, 2) finding other ways of optimizing parameters, and 3) comparing CNN based on CK+ and MUG datasets with pre-processing filters results in the selection of the same parameters. Additionally, the processing time for the pre-processing procedure, especially for the two different pre-filters can be reported. The CNN training process itself contains many parameters and different selections of training and testing images may be repeated for thorough evaluation and analysis of the results. In the second AU-based FER method, accurate detection of facial landmark is a critical precondition to successful FER. This thesis uses an existing toolkit (i.e., Dlib) for landmark detection. Further research on evaluating reliability and accuracy of landmark detection and investigating and comparing different techniques would be required. Lastly, Woody can be further developed with fully implemented FER capabilities.

Bibliography

[1] R. L. Mandryk, K. M. Inkpen, and T. W. Calvert, "Using psychophysiological techniques to measure user experience with entertainment technologies," *Behaviour & information technology*, vol. 25, no. 2, pp. 141–158, 2006.

[2] R. M. Heilman, L. G. Crişan, D. Houser, M. Miclea, and A. C. Miu, "Emotion regulation and decision making under risk and uncertainty.," *Emotion*, vol. 10, no. 2, p. 257, 2010.

[3] R. Gennari, A. Melonio, D. Raccanello, M. Brondino, G. Dodero, M. Pasini, and S. Torello, "Children's emotions and quality of products in participatory game design," *International Journal of Human-Computer Studies*, vol. 101, pp. 45–61, 2017.

[4] S. W. White, L. Abbott, A. T. Wieckowski, N. N. Capriola-Hall, S. Aly, and A. Youssef, "Feasibility of automated training for facial emotion expression and recognition in autism," *Behavior therapy*, vol. 49, no. 6, pp. 881–888, 2018.

[5] K. Asplund, A. Norberg, R. Adolfsson, and H. M. Waxman, "Facial expressions in severely demented patientsa stimulus–response study of four patients with dementia of the alzheimer type," *International Journal of Geriatric Psychiatry*, vol. 6, no. 8, pp. 599–606, 1991.

[6] H. Breivik, P. Borchgrevink, S. Allen, L. Rosseland, L. Romundstad, E. Breivik Hals, G. Kvarstein, and A. Stubhaug, "Assessment of pain," *BJA: British Journal of Anaesthesia*, vol. 101, no. 1, pp. 17–24, 2008.

[7] X. Guo, Y. Tie, L. Ye, and J. Yan, "Identifying facial expression using adaptive sub-layer compensation based feature extraction," *Journal of Visual Communication and Image Representation*, vol. 50, pp. 65–73, 2018.

[8] B. Fasel and J. Luettin, "Automatic facial expression analysis: a survey," *Pattern recognition*, vol. 36, no. 1, pp. 259–275, 2003.

[9] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza, "Power to the people: The role of humans in interactive machine learning," *AI Magazine*, vol. 35, no. 4, pp. 105–120, 2014.

[10] D. McDuff, A. Mahmoud, M. Mavadati, M. Amr, J. Turcot, R. el Kaliouby, and A. SDK, "A cross-platform real-time multi-face expression recognition toolkit," in *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems [Association for Computing Machinery (ACM), 2016]*, pp. 3723–3726.

[11] C. Fabian Benitez-Quiroz, R. Srinivasan, and A. M. Martinez, "Emotionet: An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5562–5570, 2016.

[12] G. Littlewort, J. Whitehill, T. Wu, I. Fasel, M. Frank, J. Movellan, and M. Bartlett, "The computer expression recognition toolbox (cert)," in *Face and gesture 2011*, pp. 298–305, IEEE, 2011.

[13] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L.-P. Morency, "Openface 2.0: Facial behavior analysis toolkit," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 59–66, IEEE, 2018.

[14] P. Ekman, W. V. Friesen, M. O'sullivan, A. Chan, I. Diacoyanni-Tarlatzis, K. Heider, R. Krause, W. A. LeCompte, T. Pitcairn, P. E. Ricci-Bitti, *et al.*, "Universals and cultural differences in the judgments of facial expressions of emotion.," *Journal of personality and social psychology*, vol. 53, no. 4, p. 712, 1987.

[15] M. A. Rahim, M. S. Azam, N. Hossain, and M. R. Islam, "Face recognition using local binary patterns (lbp)," *Global Journal of Computer Science and Technology*, 2013.

[16] L. Yuan, C.-m. Wu, and Y. Zhang, "Facial expression feature extraction using hybrid pca and lbp," *The Journal of China Universities of Posts and Telecommunications*, vol. 20, no. 2, pp. 120–124, 2013.

[17] D. K. Jain, Z. Zhang, and K. Huang, "Multi angle optimal pattern-based deep learning for automatic facial expression recognition," *Pattern Recognition Letters*, 2017.

[18] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1701–1708, 2014.

[19] H. F. Zaki, F. Shafait, and A. Mian, "Learning a deeply supervised multi-modal rgb-d embedding for semantic scene and object category recognition," *Robotics and Autonomous Systems*, vol. 92, pp. 41–52, 2017.

[20] H. Liu, J. Lu, J. Feng, and J. Zhou, "Group-aware deep feature learning for facial age estimation," *Pattern Recognition*, vol. 66, pp. 82–94, 2017.

[21] S. Escalera, J. Fabian, P. Pardo, X. Baro, J. Gonzalez, H. J. Escalante, D. Misevic, U. Steiner, and I. Guyon, "Chalearn looking at people 2015: Apparent age and cultural event recognition datasets and results," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 1–9, 2015.

[22] M. Braham and M. Van Droogenbroeck, "Deep background subtraction with scene-specific convolutional neural networks," in *2016 international conference on systems, signals and image processing (IWSSIP)*, pp. 1–4, IEEE, 2016.

[23] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4580–4584, IEEE, 2015.

[24] E. Friesen and P. Ekman, "Facial action coding system: a technique for the measurement of facial movement," *Palo Alto*, vol. 3, 1978.

[25] Y.-I. Tian, T. Kanade, and J. F. Cohn, "Recognizing action units for facial expression analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 2, pp. 97–115, 2001.

[26] Y. Wu and Q. Ji, "Constrained joint cascade regression framework for simultaneous facial action unit recognition and facial landmark detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3400–3408, 2016.

[27] H. Fleyeh and J. Roch, *Benchmark evaluation of HOG descriptors as features for classification of traffic signs*. Högskolan Dalarna, 2013.

[28] S. Happy and A. Routray, "Automatic facial expression recognition using features of salient facial patches," *IEEE transactions on Affective Computing*, vol. 6, no. 1, pp. 1–12, 2015.

[29] D. Ghimire, J. Lee, Z.-N. Li, and S. Jeong, "Recognition of facial expressions based on salient geometric features and support vector machines," *Multimedia Tools and Applications*, vol. 76, no. 6, pp. 7921–7946, 2017.

[30] A. Durmuşoğlu and Y. Kahraman, "Facial expression recognition using geometric features," in *2016 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pp. 1–5, IEEE, 2016.

[31] D. Ghimire and J. Lee, "Geometric feature-based facial expression recognition in image sequences using multi-class adaboost and support vector machines," *Sensors*, vol. 13, no. 6, pp. 7714–7734, 2013.

[32] Y. Yao, D. Huang, X. Yang, Y. Wang, and L. Chen, "Texture and geometry scattering representation-based facial expression recognition in 2d+ 3d videos," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 1s, p. 18, 2018.

[33] G. Sandbach, S. Zafeiriou, M. Pantic, and D. Rueckert, "Recognition of 3d facial expression dynamics," *Image and Vision Computing*, vol. 30, no. 10, pp. 762–773, 2012.

[34] Y. Xu, J. Dong, B. Zhang, and D. Xu, "Background modeling methods in video analysis: A review and comparative evaluation," *CAAI Transactions on Intelligence Technology*, vol. 1, no. 1, pp. 43–60, 2016.

[35] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning bayesian networks: The combination of knowledge and statistical data," *Machine learning*, vol. 20, no. 3, pp. 197–243, 1995.

[36] X.-X. Niu and C. Y. Suen, "A novel hybrid cnn–svm classifier for recognizing handwritten digits," *Pattern Recognition*, vol. 45, no. 4, pp. 1318–1325, 2012.

[37] M. Elleuch, R. Maalej, and M. Kherallah, "A new design based-svm of the cnn classifier architecture with dropout for offline arabic handwritten recognition," *Procedia Computer Science*, vol. 80, pp. 1712–1723, 2016.

[38] S. Guo, S. Chen, and Y. Li, "Face recognition based on convolutional neural network and support vector machine," in *2016 IEEE International Conference on Information and Automation (ICIA)*, pp. 1787–1792, IEEE, 2016.

[39] P. Viola, M. Jones, *et al.*, "Rapid object detection using a boosted cascade of simple features," *CVPR (1)*, vol. 1, pp. 511–518, 2001.

[40] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee, *et al.*, "Challenges in representation learning: A report on three machine learning contests," in *International Conference on Neural Information Processing*, pp. 117–124, Springer, 2013.

[41] J. Canny, "A computational approach to edge detection," in *Readings in computer vision*, pp. 184–203, Elsevier, 1987.

[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[43] B.-K. Kim, J. Roh, S.-Y. Dong, and S.-Y. Lee, "Hierarchical committee of deep convolutional neural networks for robust facial expression recognition," *Journal on Multimodal User Interfaces*, vol. 10, no. 2, pp. 173–189, 2016.

[44] J. L. McClelland, D. E. Rumelhart, P. R. Group, *et al.*, "Parallel distributed processing," *Explorations in the Microstructure of Cognition*, vol. 2, pp. 216–271, 1986.

[45] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for lvcsr using rectified linear units and dropout," in *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 8609–8613, IEEE, 2013.

[46] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1867–1874, 2014.

[47] D. E. King, "Max-margin object detection," *arXiv preprint arXiv:1502.00046*, 2015.

[48] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar, "Localizing parts of faces using a consensus of exemplars," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 12, pp. 2930–2940, 2013.

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

[50] N. Aifanti, C. Papachristou, and A. Delopoulos, "The mug facial expression database," in *11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10*, pp. 1–4, IEEE, 2010.

[51] J. Hamm, C. G. Kohler, R. C. Gur, and R. Verma, "Automated facial action coding system for dynamic analysis of facial expressions in neuropsychiatric disorders," *Journal of neuroscience methods*, vol. 200, no. 2, pp. 237–256, 2011.

[52] Y.-W. Chang and C.-J. Lin, "Feature ranking using linear svm," in *Causation and Prediction Challenge*, pp. 53–64, 2008.

[53] V. Cherkassky and Y. Ma, "Practical selection of svm parameters and noise estimation for svm regression," *Neural networks*, vol. 17, no. 1, pp. 113–126, 2004.

[54] A. Saeed, A. Al-Hamadi, R. Niese, and M. Elzobi, "Frame-based facial expression recognition using geometrical features," *Advances in Human-Computer Interaction*, vol. 2014, p. 4, 2014.

[55] C. Gacav, B. Benligiray, and C. Topal, "Greedy search for descriptive spatial face features," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1497–1501, IEEE, 2017.

[56] N. Aifanti and A. Delopoulos, "Linear subspaces for facial expression recognition," *Signal Processing: Image Communication*, vol. 29, no. 1, pp. 177–188, 2014.

[57] R. Zhi, M. Flierl, Q. Ruan, and W. B. Kleijn, "Graph-preserving sparse nonnegative matrix factorization with application to facial expression recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 1, pp. 38–52, 2011.

[58] D. O. Johnson, R. H. Cuijpers, and D. van der Pol, "Imitating human emotions with artificial facial expressions," *International Journal of Social Robotics*, vol. 5, no. 4, pp. 503–513, 2013.

[59] S. Costa, F. Soares, and C. Santos, "Facial expressions and gestures to convey emotions with a humanoid robot," in *International Conference on Social Robotics*, pp. 542–551, Springer, 2013.

[60] H. Moll and M. Tomasello, "Cooperation and human cognition: the vygotskian intelligence hypothesis," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 362, no. 1480, pp. 639–648, 2007.

[61] L. D. Riek, T.-C. Rabinowitch, P. Bremner, A. G. Pipe, M. Fraser, and P. Robinson, "Cooperative gestures: Effective signaling for humanoid robots," in *Proceedings of the 5th ACM/IEEE international conference on Human-robot interaction*, pp. 61–68, IEEE Press, 2010.