การรู้จำการแสดงสีหน้าโดยใช้ลักษณะเด่นบนพื้นฐานของกราฟและโครงข่ายประสาทเทียม

นาย ชัยสิทธิ์ ตันโชติศรีนนท์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์และสารสนเทศ ภาควิชาคณิตศาสตร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2553

FACIAL EXPRESSION RECOGNITION USING GRAPH-BASED FEATURES AND ARTIFICIAL NEURAL NETWORK

Mr. Chaiyasit Tanchotsrinon

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science Program in Computer Science and Information

Department of Mathematics

Faculty of Science

Chulalongkorn University

Academic Year 2010

Thesis Title	FACIAL EXPRESSION RECOGNITION USING GRAPH-BASED FEATURES AND ARTIFICIAL NEURAL NETWORK
By	Mr. Chaiyasit Tanchotsrinon
Field of Study	Computer Science and Information
Thesis Advisor	Suphakant Phimoltares, Ph.D.
Thesis Co-advisor	Assistant Professor Saranya Maneeroj, Ph.D.

Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment of the Requirements for the Master's Degree

.... Dean of the Faculty of Science

(Professor Supot Hannongbua, Dr.rer.nat.)

THESIS COMMITTEE

.... Chairman

(Professor Chidchanok Lursinsap, Ph.D.)

.... Thesis Advisor

(Suphakant Phimoltares, Ph.D.)

.... Thesis Co-advisor

(Assistant Professor Saranya Maneeroj, Ph.D.)

.... External Examiner

(Atchara Mahaweerawat, Ph.D.)

ชัยสิทธิ์ ตันโชติศรีนนท์ : การรู้จำการแสดงสีหน้าโดยใช้ลักษณะเด่นบนพื้นฐานของกราฟ และโครงข่ายประสาทเทียม. (FACIAL EXPRESSION RECOGNITION USING GRAPH-BASED FEATURES AND ARTIFICIAL NEURAL NETWORK) **อ. ที่ปรึกษาวิทยานิพนธ์หลัก: อ. ดร. ศุภกานต์ พิมลธเรศ, อ. ที่ปรึกษาวิทยานิพนธ์ร่วม: ผศ. ดร. ศรันญา มณีโรจน์** 53 หน้า.

การแสดงออกทางสีหน้าเป็นส่วนประกอบสำคัญต่อการสื่อสารของมนุษย์ เนื่องจากเป็นหนึ่งในภาษากายของเราที่จะเพิ่มความเข้าใจกันและกันระหว่างการสื่อสารข้อมูล จากงานวิจัยที่ได้ทำการศึกษาก็พบว่า วิธีการในการสกัดลักษณะเด่นมีผลกระทบต่อการรู้จำการแสดงสีหน้าโดยตรง ดังนั้นการเลือกคัดลักษณะเด่นที่ไม่เกี่ยวข้องกับการแสดงสีหน้าย่อมทำให้การรู้จำการแสดงสีหน้ามีความถูกต้องแม่นยำมากขึ้น ขั้นตอนในการรู้จำการแสดงออกทางสีหน้าในงานวิจัยนี้สามารถแบ่งออกได้เป็นสองขั้นตอน ขั้นตอนแรกคือการกำหนดจุด 14 จุดที่เกี่ยวข้องกับการแสดงสีหน้า จากนั้นหาระยะทางระหว่างแต่ละจุดเพื่อที่จะเก็บข้อมูลลักษณะของการแสดงสีหน้าเพื่อสร้างเป็นลักษณะเด่นบนพื้นฐานของกราฟ ส่วนขั้นตอนที่สองคือการรู้จำการแสดงสีหน้าด้วยโครงข่ายประสาทเทียมเพื่อที่จะจำแนกการแสดงสีหน้าพื้นฐานทั้งหกประเภทจากลักษณะเด่น ฐานข้อมูลการแสดงสีหน้าที่ถูกเข้ารหัสหน่วยการกระทำของคอห์นและคานาเดะได้ถูกเลือกสำหรับการประเมินค่าความถูกต้องและนำไปใช้ในระบบรู้จำภายใต้ลักษณะเด่นและเทคนิคการฝึกฝนหลายชนิด ผลการทดลองแสดงให้เห็นว่าการใช้ลักษณะเด่นบนพื้นฐานของกราฟโดยไม่ทำการตรวจสอบความสมเหตุสมผลและการใช้ลักษณะเด่นบนพื้นฐานของกราฟโดยทำการตรวจสอบความสมเหตุสมผลให้ค่าความถูกต้องที่ดีที่สุดคือ 95.24% ดังนั้นการใช้ลักษณะเด่นบนพื้นฐานของกราฟร่วมกับโครงข่ายประสาทเทียมจึงเป็นวิธีการที่มีประสิทธิภาพสำหรับการรู้จำการแสดงสีหน้า

ภาควิชา คณิตศาสตร์	ลายมือชื่อนิสิต ชัยสิทธิ์ ตันโชติศรีนนท์
สาขาวิชา วิทยาการคอมพิวเตอร์และสารสนเทศ	ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก Suphakant Phimoltares
ปีการศึกษา 2553	ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์ร่วม Saranya Maneeroj

5273603023: MAJOR COMPUTER SCIENCE AND INFORMATION

KEYWORDS : FACIAL FEATURES / MULTILAYER PERCEPTRONS / BACKPROPAGATION / FACIAL EXPRESSION RECOGNITION

CHAIYASIT TANCHOTSRINON: FACIAL EXPRESSION RECOGNITION USING GRAPH-BASED FEATURES AND ARTIFICIAL NEURAL NETWORK. ADVISOR: SUPHAKANT PHIMOLTARES, Ph.D., CO-ADVISOR: ASST.PROF. SARANYA MANEEROJ, Ph.D., 53 pp.

Facial expression is significant for face-to-face communication since it is one of our body language that increases data information during the communication. In recent surveys, it found that feature extraction methods have influenced on facial recognition directly and they can outperform if some irrelevant features are eliminated. Consequently, a procedure of facial expression recognition using graph-based features and artificial neural network is proposed in this thesis, and the procedure can be divided into 2 phases. For the first phase, fourteen points are manually located to create graph with edges connecting among such points, followed by computation of the Euclidean distances from those edges to define them as features for training in the next phase. For the next phase, Multilayer Perceptrons with back-propagation learning algorithm is implemented to recognize six basic emotions from the corresponding feature vectors. To evaluate the performance, Cohn-Kanade AU-Coded facial expression database is applied to the recognition system under various kinds of feature and training technique, and the experimental results have shown that MLP (Graph-based features) without validation and MLP (Graph-based feature) with cross validation can achieve the highest recognition rate (95.24%). Therefore, it has illustrated that the combination of graph-based features and artificial neural network is the efficient way for the facial expression recognition.

Department : Mathematics	Student's Signature
Field of Study : Computer Science and Information	Advisor's Signature
Academic Year : 2010	Co-advisor's Signature

Acknowledgements

I am deeply indebted to my Thesis adviser, Dr. Suphakant Phimoltares, and Assistant Professor Dr. Saranya Maneeroj, whose help, suggestions and encouragement helped me in all the time of research for and writing of this Thesis. Furthermore, I would like to express my gratitude to Professor Dr. Chidchanok Lursinsap and Dr. Atchara Mahaweerawat for devoting time and all those who gave me the possibility to support this Thesis.

Especially, I wish to express my deep sense of gratitude to my parent for their teaching, encouragement and love that helps me to pass the difficult times and complete this Thesis.

Contents

List of Tables

List of Figures

Figure Page

CHAPTER I

INTRODUCTION

1.1 Motivation and Problem Description

Recently, facial expression recognition becomes an interested topic for researchers since it is one of human body language that lead us to understand an interaction between human and human, or human and computer increasingly, as mentioned that "The facial expression sends the 55% information during the process of human communications while language sends only 7%" in [1]. Additionally, the facial expressions can be adapted to various kinds of application, for instance, animation, robotics, security, Human-Computer-Interaction (HCI), driver safety, and health care system. The following are the description of these applications deep in detail.

As mention in [2], Facial expressions can be applied to animation movies. To demonstrate this point, the actors or the actresses were masked a lot of points into their faces followed by using a video camera to record the changed positions of these points and also marking these points into a simulation of facial characters in an animation movie. Then, when the actor or the actress acts on script in the scene, the character in the animation movie will be moved as same as the actor or the actress did. This technique of facial expression recognition was applied to animation movies in order to increase animation characters realistically. In security, the facial expressions were used to improve investigation capabilities of detectives by providing a report of facial expression information of all suspects in an interrogation. Since the facial expressions represented emotions during communication and regulated interpersonal behaviors, the detectives can analyzed the facial expressions of all suspects during investigation. This process provided more critical resources to improve the investigation potentially.

Furthermore, the researchers in [3] have said that the facial expression recognition also plays an important role in Robotics and Human-Computer-Interaction (HCI). This is because the emotions make humans different from animals and things. Nowadays, computer technologies become an integral part of our daily lives. Most people can't live without computers anymore because they use them every day and they use them to deal with nearly everything. From that reasons, humans are social animals such that the researchers want to increase social interactions between humans and computers, thereby the computers can understand the human outlet emotions and can response back with rationality. For example, the system that was used to automatically select music or light in home will change these things according to the owner's moods.

In addition, the facial expression recognition can be used to increase driver safety. As mentioned in [4], a video camera collected information about head motions and eye tracking in real time, and the system then analyzed these collected information in order to detect and predict drowsiness of the driver. In part of health care system, [5] has applied the facial expressions to recognize a pain in some kinds of patients, such as, children, the patients with limited ability to communicate, the mentally impaired patients, and the patients who require for assisted breathing. This application helps the patient administrators to take care of their patients immediately when they need the assistances.

After survey in several researches in facial expression recognition, it has found that most research attempted to classify six basic emotion states, which are Surprise, Sad, Happy, Disgust, Fear and Anger, as mentioned in [6]. In Figure 1.1, it represents these basic emotion states.

(a)“Surprise” (b) “Sad” (c) “Happy”

(d) “Fear” (e) “Disgust” (f) “Anger”

Figure 1.1: Six basic emotions

In this thesis, the high accuracy recognition system based on machine learning with a reasonable number of samples and a small number of facial points is introduced. The number of points is fourteen, including two points of inner eyebrows, two points of middle eyebrows, two points of outer eyebrows, two points of inner eye, two points of outer eyes, and four points of mouths. These points are defined as the vertices of the graph. Then, connecting edge is formed from a pair of vertices along with its Euclidean distance. Such the distances are interpreted as features that send to the learning process to recognize one of six basic emotions: ‘Surprise’, ‘Sad’, ‘Happy’, ‘Disgust’, ‘Fear’, and ‘Anger’ using Artificial Neural Networks (ANN). In the architecture of recognition system, it contains five neural networks, whose relationship is like binary tree. To evaluate the method, the proposed systems are applied with Cohn-Kanade AU-Coded facial expression database and the experimental results have shown that the system can perform 95.24% accuracy which is higher than the existing method.

1.2 Problem Formulation

From above, there are three Problems interested in this thesis:

1. What do features affect facial expression recognition?
2. How to create graph-based model from derived features?
3. How to design the recognition system based on neural networks?

1.3 Objective of Research

The main objectives of this thesis are as follows:

1. To locate facial features using Graph-based method.
2. To develop facial expression recognition system with high accuracy.

1.4 Scope of Study

1. In this thesis, Cohn-Kanade AU-Coded Facial Expression Database is used because it is a reliable benchmark with a variety of samples.
2. The facial expression recognition system can be applied to only frontal-view face images without any occlusions (Sunglasses, scarf, mask).

1.5 Research methodology

The research methodology in this thesis is composed of 7 main tasks. The first task is literature review because it provides knowledge for understanding a development in an area of interest. After review, useful information in the literatures, such as their background knowledge, their research methodologies, their experimental results, and their future works are obtained, and it also found that their research methodologies have both of advantages and disadvantages, as described in Chapter 2. The problems are arisen from their techniques and the interested problems are stated in the second task. Then, a design and analysis of experiment will be required for the third task, as explained in the procedure of Chapter 4. Subsequently, implementation and testing tasks are needed for improving the recognition system in sixth task, and the last task is a conclusion of the experiment.

Table 1.1: Task Schedule.

No.	Task	Month Sequence											
		1	2	3	4	5	6	7	8	9	10	11	12
1	Literature review												
2	Problem statement												
3	Design and Analysis of experiment												
4	Implementation												
5	Testing												
6	Improvement recognition system												
7	Conclude experimental results												
8	Prepare for proposal test												
9	Prepare for proceeding												
10	Prepare dissertation												

1.6 Expected Beneficial Outcomes

The expected outcomes from this thesis are as follows:

1. Significant points directly corresponding to local feature locations.
2. Graph-based facial expression recognition system with high performance.

CHAPTER II

LITERATURE REVIEWS

2.1 Database & FACS Description

Widely used Databases in facial recognition are ORL, YALE, AR, FERET and JAFFE. Some of these databases were not updated for a long time. For example, ORL database was created in April 1992 and April 1994. In each sample of AR database, it consists of only three facial expressions but it has a lot of conditions, such as having only left light on face, having only right light on face, having both right and left light on face, wearing sun glasses, wearing sun glasses with having left or right or both left and right light on face, and wearing scarf with having left or right or both left and right light on face. The AR and YALE databases might be suited for facial recognition or facial identification more than facial expression recognition. JAFFE database contains only ten females with seven image per person and seven different emotionally facial expressions. Due to the small number of samples and the unbalance of sample in gender aspects in JAFFE, the researcher prefer to use CMU-Pitts burgh AU-Coded Face Expression Image Database or Cohn-Kanade dataset (CK+) in this experiment. The reason why CMU-Pitts burgh AU-Coded Face Expression Image Database was selected will be described later in Chapter four. Table 2.1 and Table 2.2 represents a detail of this database.

Table 2.1: Overview of CMU-Pitts burgh AU-Coded Face Expression Image Database

Subjects	
Number of subjects	210
Age	18 – 50 years
Women	69%
Men	31%
Euro-American	81%
Afro-American	13%
Other	6%

Table 2.2: Overview of CMU-Pitts burgh AU-Coded Face Expression Image Database

Digitized sequences	
Number of subjects	182
Resolution	640x490 for grayscale
	640x480 for 24-bit color
Frontal view	2105
30-degree view	Videotape only
Sequence duration	9 – 60 frames/sequence

Human face contains a lot of muscle so there are several ways to identify the facial expressions by using muscle movement. Therefore, many researches in facial expression recognition attempt to describe the expressions in term of facial movement in many ways. Consequently, a facial coding system has been constructed in order to reduce a variation and create correspondence in the identification of facial expressions.

The system is named as Facial Action Coding System or FACS and it becomes a standard for labeling face movements. Facial Action coding system consists of 44 action units proposed by T.Kanade [13]. These forty-four action units can be divided into two groups. The first group is a specific set of facial muscles which contains 30 action units, and the second one is an unspecific anatomic set of face which contains 14 action units, as shown in Table 2.3 and 2.4, respectively.

Table 2.3: List of Action Units (Group 1)

AU	Facial muscle	Description of muscle movement
1	Frontalis, pars medialis	Inner corner of eyebrow raised
2	Frontalis, pars lateralis	Outer corner of eyebrow raised
4	Corrugator supercilii, Depressor supercilii	Eyebrows drawn medially and down
5	Levator palpebrae superioris	Eyes widened
6	Orbicularis oculi, pars orbitalis	Cheeks raised; eyes narrowed
7	Orbicularis oculi, pars palpebralis	Lower eyelid raised and drawn medially
9	Levator labii superioris alaeque nasi	Upper lip raised and inverted; superior part of the nasolabial furrow deepened; nostril dilated by the medial slip of the muscle
10	Levator labii superioris	Upper lip raised; nasolabial furrow deepened producing square-like furrows around nostrils
11	Levator anguli oris (a.k.a. Caninus)	Lower to medial part of the nasolabial furrow deepened
12	Zygomaticus major	Lip corners pulled up and laterally
13	Zygomaticus minor	Angle of the mouth elevated; only muscle in the deep layer of muscles that opens the lips
14	Buccinator	Lip corners tightened. Cheeks compressed against teeth
15	Depressor anguli oris (a.k.a. Triangularis)	Corner of the mouth pulled downward and inward
16	Depressor labii inferioris	Lower lip pulled down and laterally
17	Mentalis	Skin of chin elevated
18	Incisivii labii superioris andIncisivii labii inferioris	Lips pursed
20	Risorius w/ platysma	Lip corners pulled laterally
22	Orbicularis oris	Lips everted (funneled)
23	Orbicularis oris	Lips tightened
24	Orbicularis oris	Lips pressed together
25	Depressor labii inferioris, or relaxation of mentalis, or orbicularis oris	Lips parted
26	Masseter; relaxed temporal and internal pterygoid	Jaw dropped
27	Pterygoids and digastric	Mouth stretched open
28	Orbicularis oris	Lips sucked
41	Relaxation of levator palpebrae superioris	Upper eyelid droop
42	Orbicularis oculi	Eyelid slit
43	Relaxation of levator palpebrae superioris; orbicularis oculi, pars palpebralis	Eyes closed
44	Orbicularis oculi, pars palpebralis	Eyes squinted
45	Relaxation of levator palpebrae superioris; orbicularis oculi, pars palpebralis	Blink
46	Relaxation of levator palpebrae superioris; orbicularis oculi, pars palpebralis	Wink

Table 2.4: List of Action Units (Group 2)

AU	Description of Movement	AU	Description of Movement
8	Lips toward	33	Blow
19	Tongue show	34	Puff
21	Neck tighten	35	Cheek suck
29	Jaw thrust	36	Tongue bulge
30	Jaw sideways	37	Lip wipe
31	Jaw clench	38	Nostril dilate
32	Bite lip	39	Nostril compress

2.2 Relationship between AU and Emotions

Lucey et al. [7] and Zhao et al. [8] have studied the relationship between action units and each basic emotion, as shown in Table 2.5 and 2.6, respectively. In Table 2.5 and 2.6, they explain criteria of lists of Action Units for identifying each basic emotion. Moreover, the Action Unit coding system that is used to represent the facial expressions has shown that the emotional expressions mostly came from four regions, including eyebrows, eyes, nose, and mouth regions, as hi-lighted with gray color in Table 2.7.

Table 2.5: Emotion description in term of facial Action Units proposed by Lucey

Emotion	Criteria
Surprise	Either AU 1+2 or 5 must be present and the intensity of AU 5 must no be stronger than B
Sadness	Either AU 1+4+15 or 11 must be present. An exception is AU 6+15
Happy	AU 12 must be present
Fear	Au combination of AU 1+2+4 must be present, unless AU 5 is of intensity E then AU 4 can be absent
Disgust	Either AU 9 or AU 10 must be present
Angry	AU 23 and AU 24 must be present in the AU combination

Table 2.6: Relation between Action Units and facial expressions by Zhao

Expression	Main related AU
Surprise	AU1 : 19.9%, AU2 : 20.5%, AU5 : 14.2%, AU25 : 21.6%, AU27 : 12.3%
Sadness	AU1 : 15.6%, AU2 : 8.1%, AU4 : 16.0% AU15 : 19.3%, AU17:27.0%
Happy	AU6 : 23.7%, AU12 : 36.5%, AU16 : 8.0%, AU25 : 23.0% AU26 : 5.1%
Fear	AU1 : 12.5%, AU4 : 25%, AU11 : 8.8% ,AU20 : 19.7% AU25 : 21.5%
Disgust	AU4 : 25.0%, AU 6 : 7.2%, AU7 : 22.0%, AU9 : 18.5% AU17 : 11.8%
Angry	AU7 : 15.0% AU17:16.3%, AU23 : 16.7% AU24 : 13.9%

Table 2.7: List of Action Units that influence on facial expressions

AU	Name	N	AU	Name	N	AU	Name	N
1	Inner Brow Raiser	173	13	Cheek Puller	2	25	Lips Part	287
2	Outer Brow Raiser	116	14	Dimpler	29	26	Jaw Drop	48
4	Brow Lowerer	191	15	Lip Corner Depressor	89	27	Mouth Stretch	81
5	Upper Lip Raiser	102	16	Lower Lip Depressor	24	28	Lip Suck	1
6	Cheek Raiser	122	17	Chin Raiser	196	29	Jaw Thrust	1
7	Lid Tightener	119	18	Lip Puckerer	9	31	Jaw Clencher	3
9	Nose Wrinkler	74	20	Lip Stretcher	77	34	Cheek Puff	1
10	Upper Lip Raiser	21	21	Neck Tightener	3	38	Nostril Dilator	29
11	Nasolabial Deepener	33	23	Lip Tightener	59	39	Nostril Compressor	16
12	Lip Corner Puller	111	24	Lip Pressor	57	43	Eyes Closed	9

2.3 Feature Extraction

Metallinou et al. [9] extracted features by using 46 facial markers, and then applied PCA, PFA, and Fisher methods to increase the distribution of data points simultaneously. The performance of this technique depends on the number of facial point markers. If the facial point markers are sufficient, the recognition accuracy will high. In contrast, if the number of the makers is too high, the overfitting and time consumption problems will occur. After applying this technique, the optimal number of markers is 30 that are reduced from 46 facial point markers with higher distribution. Similarly, L. Oliveira et al. [10] proposed a feature selection algorithm that was used to analyze coefficients and then reject some of them which have no effect to recognition system. Table 2.8 has shown the experimental results of feature selection algorithm proposed by L. Oliveira as well as the number of coefficients influenced on recognition rate of k-Nearest Neighbor (kNN) or Support Vector Machine (SVM). From this experiment, it can be noted that facial data points should be easily collected from any image and the number of points has resulted in the recognition rate.

Table 2.8: Experimental Results of Feature selection algorithm proposed by L. Oliveira

Strategy	Coefficients	Rec. Rate (%)	
		kNN	SVM
2DPCA	900	82.0	78.8
2DPCA + feature selection	58	91.0	94.0

2.4 Moment Invariants and Singular Value Decomposition

Guojiang et al. [11] proposed a method which combining moment invariants with a singular value. Moment invariants (MI) is a is a useful tool for extracting features from two-dimensional images which are invariant with respect to the images position, size, and orientation. Singular value decomposition (SVD) is a technique in many matrix computation and analyses for obtain singular value vector with the characteristics of image .This method extracted features, including eye, eye brow, and mouth regions, and then applied these features to Support Vector Machine (SVM) for classification process. Figure 2.1 has shown the flow chart of the method proposed by Guojiang, followed by its experimental results in Table 2.9. From this method, it has found that a major drawback is its complexity.

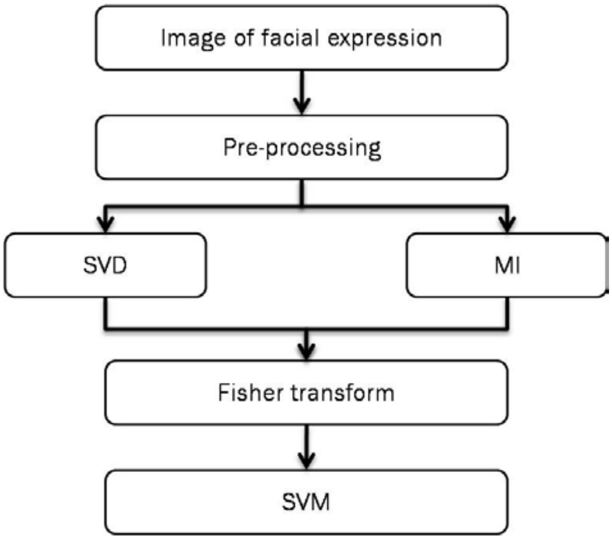


Figure 2.1: Flowchart of method proposed by Guojiang

Table 2.9: Experimental Results of MI, SVD, and MI+SVD proposed by Guojiang

Method	MI	SVD	MI + SVD
Rate (%)	73.81	76.19	90.48

2.5 Development of Research on Facial Expression Recognition

Chen et al. [12] describe an implementation of facial expression recognition for computers and divide it into 3 procedures: the scan of facial expression image, facial expression feature extraction, and facial expression recognition. A process of facial expression recognition proposed by Chen is illustrated in Figure 2.3. From this figure, the facial expression recognition process is composed of facial expression feature extraction and facial expression recognition. The facial expression feature extraction includes four methods: Method based of geometric feature, Method based on Gabor Wavelet, Method based on dynamic image, and Model-based method, while facial expression recognition includes 3 methods: Template-based method, Methods based on machine learning, and Methods based on probability models. The detail of the feature extraction and the recognition will be described as follows.

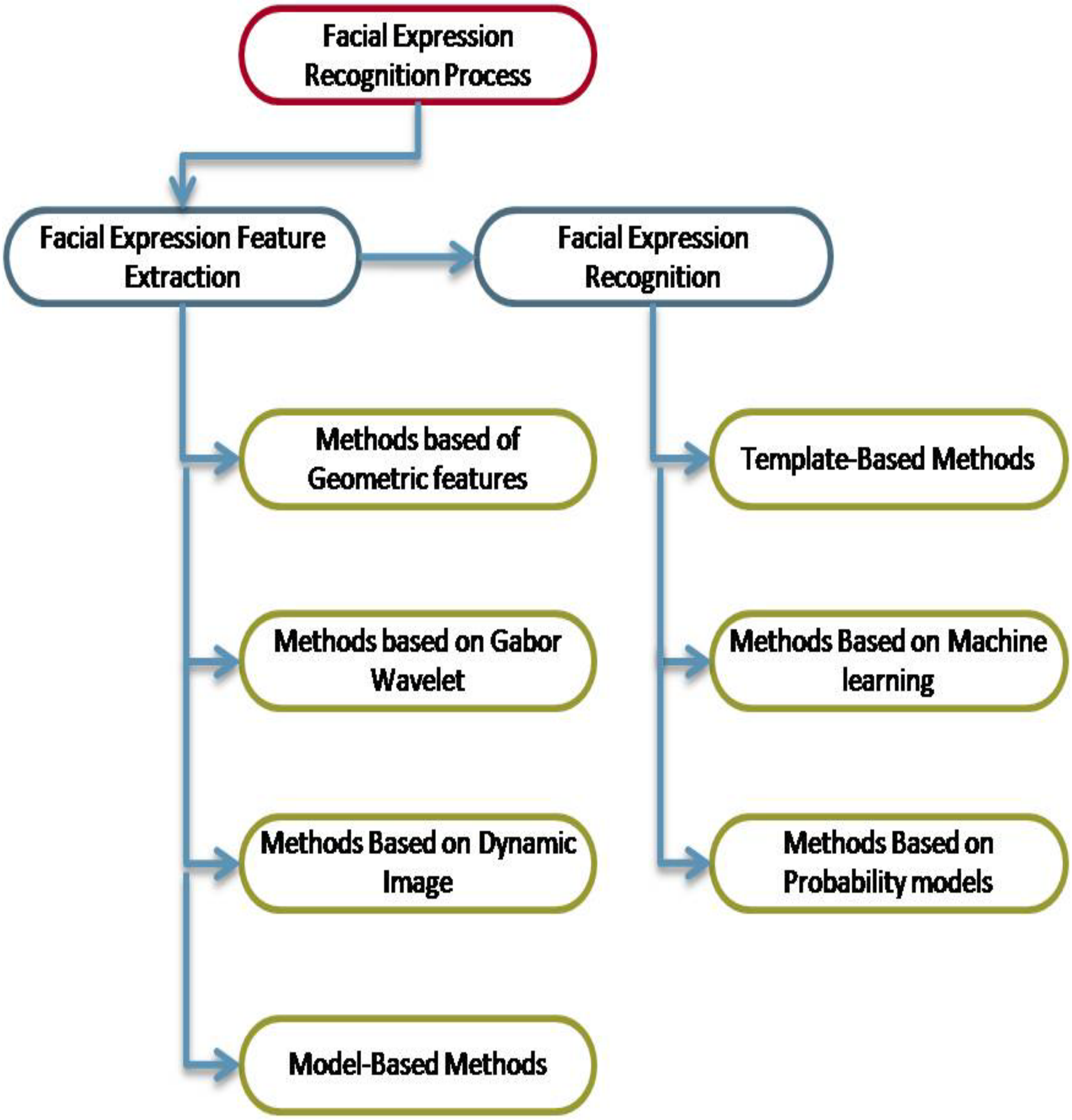


Figure 2.2: Facial Expression Recognition Process proposed by Chen

In part of facial expression feature extraction, it is a procedure to extract the information of characteristic expressions from face images or face image sequences. The information is defined as the facial expression features. The facial expression features are different in each facial image. If a difference of the information for expressions is high, an accuracy of the recognition process will be high as well. Since the feature extraction affected on the recognition process directly, it is significant for the recognition process.

In part of facial expression recognition, Chen et al. [12] described three categories of recognition methods, including Template-based method, Methods based on machine learning, and Methods based on probability models. Each method has its own advantages and disadvantages. This thesis has emphasized on the recognition method based on machine learning because it is simulated from human neural behavior with properties of self-learning, self-organization, association, and fault tolerance.

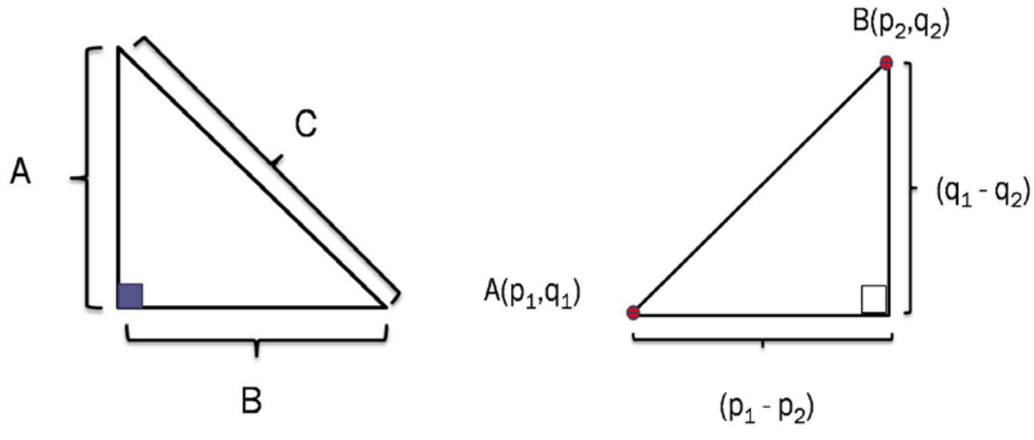
CHAPTER III

FUNDAMENTAL KNOWLEDGE

3.1 Euclidean Distance

Euclidean Distance is well-known method to find a distance in straight line between two points. The Euclidean distance is derived from Pythagorean Theorem. In right angled triangle, the square of the length of the hypotenuse equals the sum of the squares of the lengths of the two other sides. The Pythagorean Theorem is represented in the equation 3.1.1, where c is the length of hypotenuse, and a and b are the lengths of legs. Figure 3.1 (a) represents the right angled triangle and Figure 3.1 (b) displays the right angled triangle with coordinate.

$$C^2 = A^2 + B^2 \tag{3.1.1}$$



(a) (b)

Figure 3.1: (a) Right angled triangle, (b) Right angled triangle with coordinate

From the right angled triangle with coordinate in Figure 3.1 (b), the Euclidean distance between point A and B is calculated by the equation 3.1.2, and the Euclidean distance for pair of points used in this experiment is computed by the equation as well.

$$D(p,q) = \sqrt{(\Delta p)^2 + (\Delta q)^2} \qquad (3.1.2)$$

3.2 MLP Neural Networks

The network model that is adopted in this thesis for the recognition of facial expression is Multilayer Perceptrons [15]. For this network model, it typically consists of a set of sensory units (source nodes) that constitute the input layer, one or more hidden layers of computation nodes, and an output layer of computation nodes. The input signal propagates through the network in a forward direction, on a layer-by-layer basis. The architectural graph of a multilayer perceptron with two hidden layers and an output layer, are illustrated in Figure 3.2. In this architecture, the network is fully connected so a neuron in any layer of the network is connected to all the nodes/neurons in the previous layer. Signal flow through the network progresses in a forward direction, from left to right and on a layer-by-layer basis.

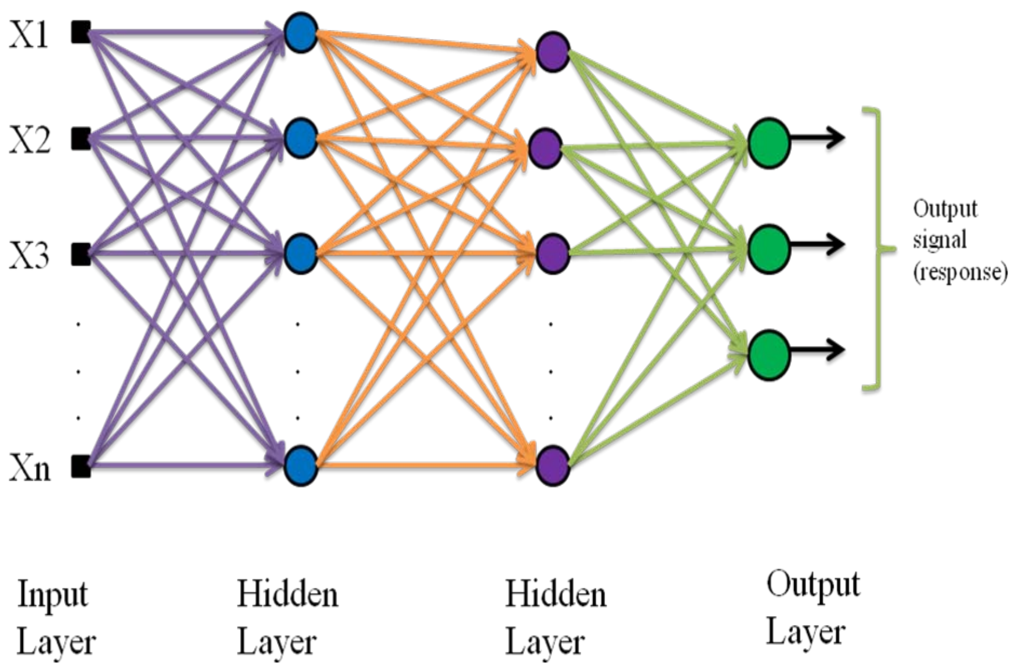


Figure 3.2: Architectural graph of a multilayer perceptron with two hidden layers

In order to describe the network model deep in detail, Figure 3.3 depicts a portion of the multilayer perceptron.

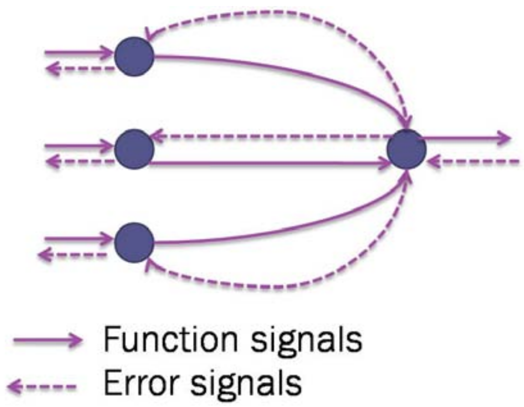


Figure 3.3: Illustration of the directions of two basic signal flows in a multilayer perceptron: forward propagation of function signals and back-propagation of error signals

From the above figure, two kinds of signals are identified in this network:

1) **Function Signals:** it is an input signal (stimulus) that comes in at the input end of the network, propagates forward (neuron by neuron) through the network, and emerges at the output end of the network as an output signal. There are two reasons why the signal is referred to as "function signal." First, it is presumed to perform a useful function at the output of the network. Second, at each neuron of the network through which a function signal passes, the signal is calculated as a function of the inputs and associated weights applied to that neuron. In addition, the function signal is also referred to as the input signal.

2) **Error Signals:** it originates at an output neuron of the network, and then propagates backward (layer by layer) through the network. It is referred to as an "error signal" because its computation by every neuron of the network involves an error-dependent function in one form or another.

The output neurons (the computational nodes) constitute the output layers of the network, and the remaining neurons (computational nodes) constitute hidden layers of the network. Therefore, the hidden units are not part of the output or input of the network. Besides, the first hidden layer is fed from the input layer made up of sensory units (source nodes); the resulting outputs of the first hidden layer are in turn applied to the next hidden layer; and so on for the rest of the network.

There are two computations that perform in each hidden or output neuron of a multilayer perceptron, as follows:

1) The computation of the function signal appearing at the output of a neuron, which is expressed as a continuous nonlinear function of the input signal and synaptic weights associated with that neuron.
2) The computation of an estimate of the gradient vector such as, the gradients of the error surface with respect to the weights connected to the inputs of a neuron, which is needed for the backward pass through the network.

3.3 Back-Propagation Algorithm

In this section, the learning algorithm used for adjusting the weight will be explained. In the thesis, Back-Propagation Algorithm is adopted as the learning algorithm for facial expression recognition. This algorithm is based on the error-correction learning rule, and the following detail is described for the theory of this learning algorithm [15].

The error signal at the output of neuron *j* at iteration *n* is defined by

$$e_j(n) = d_j(n) - y_j(n), \quad \text{neuron j is an output node} \qquad (3.3.1)$$

The instantaneous value of the error energy for neuron *j* is defined as $\frac{1}{2}e_j^2(n)$. Correspondingly, the instantaneous value $\varepsilon(n)$ of the total error energy is obtained by summing $\frac{1}{2}e_j^2(n)$ over all neurons in the output layer; these are the only "visible" neurons for which error signals can be calculated directly. Thus,

$$\varepsilon(n) = \frac{1}{2}\sum_{j \in C} e_j^2(n) \qquad (3.3.2)$$

The set C includes all the neurons in the output layer of the network. Let *N* denote the total number of patterns (examples) contained in the training set. The average squared error energy is obtained by summing $\varepsilon(n)$ over all *n* and then normalizing with respect to the set size *N*, as shown by

$$\varepsilon_{av} = \frac{1}{N}\sum_{n=1}^{N} \varepsilon(n) \qquad (3.3.3)$$

The instantaneous error energy $\varepsilon(n)$ and therefore the average error energy ε_{av}, is a function of all the free parameters (i.e., synaptic weights and bias levels) of the network. For a given training set, ε_{av} represents the cost function as a measure of learning performance. The objective of the learning process is to adjust the free parameters of the network to minimize ε_{av}. To do this minimization, an approximation similar in rationale to that used for the derivation of the LMS algorithm is adopted. Specifically, a simple method of training in which the weights are updated on a pattern-by-pattern basis until one epoch is considered, that is, one complete presentation of the entire training set has been dealt with. The adjustments to the weights are made in accordance with the respective errors computed for each pattern presented to the network.

The arithmetic average of these individual weight changes over the training set is therefore an estimate of the true change that would result from modifying the weights based on minimizing the cost function ε_{av} over the entire training set.

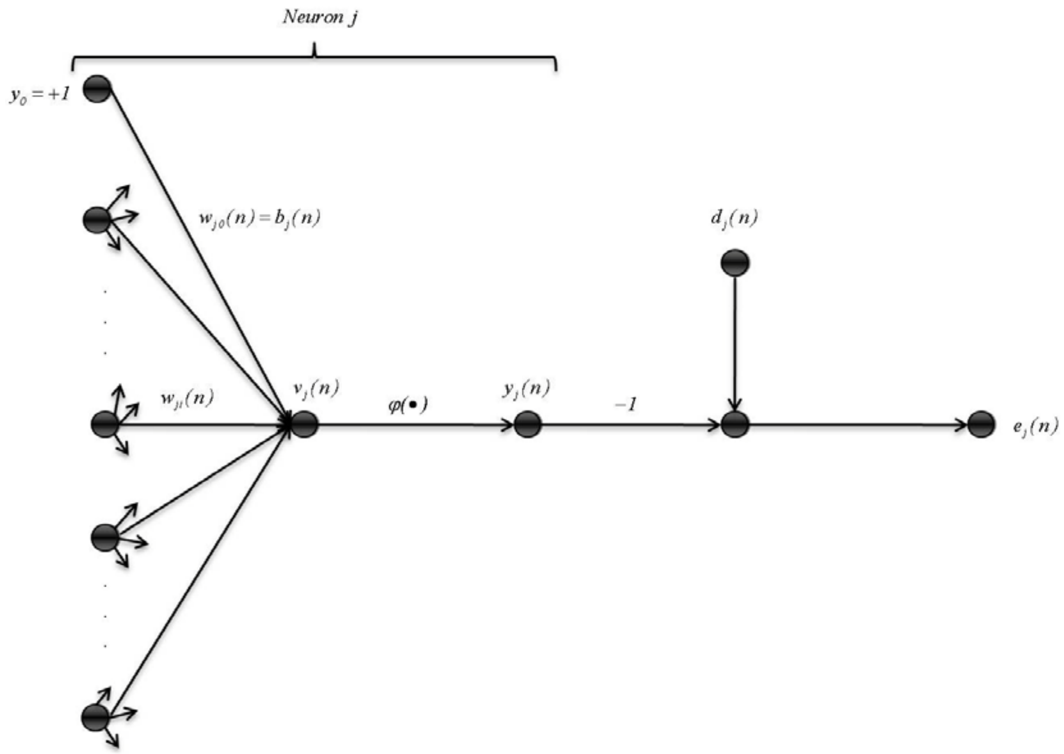


Figure 3.4: Signal-flow graph highlighting the details of output neuron *j*

Figure 3.4 depicts neuron *j* being fed by a set of function signals produced by a layer of neurons to its left. The induced local field (n) produced at the input of the activation function associated with neuron *j* is therefore

$$v_j(n) = \sum_{i=0}^{m} w_{ji}(n) y_i(n) \tag{3.3.4}$$

where m is the total number of inputs (excluding the bias) applied to neuron j. The synaptic weight w_{jo} (corresponding to the fixed input = +1) equals the bias applied to neuron j. Hence the function signal (n) appearing at the output of neuron j at iteration n is

$$y_j(n) = \varphi_j(v_j(n)) \tag{3.3.5}$$

Likewise the LMS algorithm, the back-propagation algorithm applies a correction $\Delta w_{ji}(n)$ to the synaptic weight $w_{ji}(n)$, which is proportional to the partial derivative $\partial\varepsilon(n)/\partial w_{ji}(n)$. According to the chain rule of calculus, this gradient is may express as:

$$\frac{\partial \varepsilon(n)}{\partial w_{ji}(n)} = \frac{\partial \varepsilon(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_i(n)} \frac{\partial y_i(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \tag{3.3.6}$$

The partial derivative $\partial\varepsilon(n)/\partial w_{ji}(n)$ represents a sensitivity factor, determining the direction of search in weight space for the synaptic weight w_{ji}.

Differentiating both sides of Eq. (3.3.2) with respect to $e_j(n)$, yields

$$\frac{\partial \varepsilon(n)}{\partial e_j(n)} = e_j(n) \tag{3.3.7}$$

Differentiating both sides of Eq. (3.3.1) with respect to $y_j(n)$, yields

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1 \tag{3.3.8}$$

Next, differentiating Eq. (3.3.5) with respect to $v_j(n)$, yields

$$\frac{\partial y_j(n)}{\partial v_j(n)} = \varphi_j'(v_j(n)) \tag{3.3.9}$$

where the use of prime (on the right-hand side) signifies differentiation with respect to the argument. Finally, differentiating Eq. (3.3.4) with respect to $w_{ji}(n)$, yields

$$\frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_j(n) \tag{3.3.10}$$

The use of Eqs. (3.3.7) to (3.3.10) in (3.3.6) yields

$$\frac{\partial \varepsilon(n)}{\partial w_{ji}(n)} = -e_j(n)\varphi_j'(v_j(n))y_i(n) \tag{3.3.11}$$

The correction $\Delta w_{ji}(n)$ applied to $w_{ji}(n)$ is defined by the delta rule:

$$\Delta w_{ji}(n) = -\eta \frac{\partial \varepsilon(n)}{\partial w_{ji}(n)} \tag{3.3.12}$$

where η is the learning-rate parameter of the back-propagation algorithm. The use of the minus sign in Eq. (3.3.12) accounts for gradient descent in weight space (i.e., seeking a direction for weight change that reduces the value of (n)). Accordingly, the use of Eq. (3.3.11) in (3.3.12) yields

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n) \tag{3.3.13}$$

where the local gradient $\delta_j(n)$ is defined by

$$\begin{aligned} \delta_j(n) &= -\frac{\partial \varepsilon(n)}{\partial v_j(n)} \\ &= -\frac{\partial \varepsilon(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \\ &= e_j(n)\varphi_j'(v_j(n)) \end{aligned} \tag{3.3.14}$$

The local gradient points to required changes in synaptic weights. According to Eq. (3.3.14), the local gradient $\delta_j(n)$ for output neuron *j* is equal to the product of the corresponding error signal $e_j(n)$ for that neuron and the derivative $\varphi_j'(v_j(n))$ of the associated activation function.

From the Equations (3.3.13) and (3.3.14), it can be seen that a key factor involved in the calculation of the weight adjustment $\Delta w_{ji}(n)$ is the error signal $e_j(n)$ at the output of neuron *j*. In this context, it can be identified two distinct cases, depending on where in the network neuron *j* is located. In case 1, neuron *j* is an output node. This case is simple to handle because each output node of the network is supplied with a desired response of its own, making it a straightforward matter to calculate the associated error signal. In case 2, neuron *j* is a hidden node. Even though hidden neurons are not directly accessible, they share responsibility for any error made at the output of the network.

Case 1 Neuron *j* is an Output Node

For this case, the neuron *j* is located in the output layer of the network, it is supplied with a desired response of its own. The Eq. (3.3.1) is used to compute the error signal $e_j(n)$ associated with this neuron, as shown in Figure 3.4. Having determined $e_j(n)$, it is a straightforward matter to compute the local gradient $\delta_j(n)$ using Eq. (3.3.14).

Case 2 Neuron *j* is a Hidden Node

For this case, the neuron *j* is located in a hidden layer of the network, there is no specified desired response for that neuron. Accordingly, the error signal for a hidden neuron would have to be determined recursively in terms of the error signals of all the neurons to which that hidden neuron is directly connected. Figure 3.5 depicts the situation that neuron *j* is a hidden node of the network.

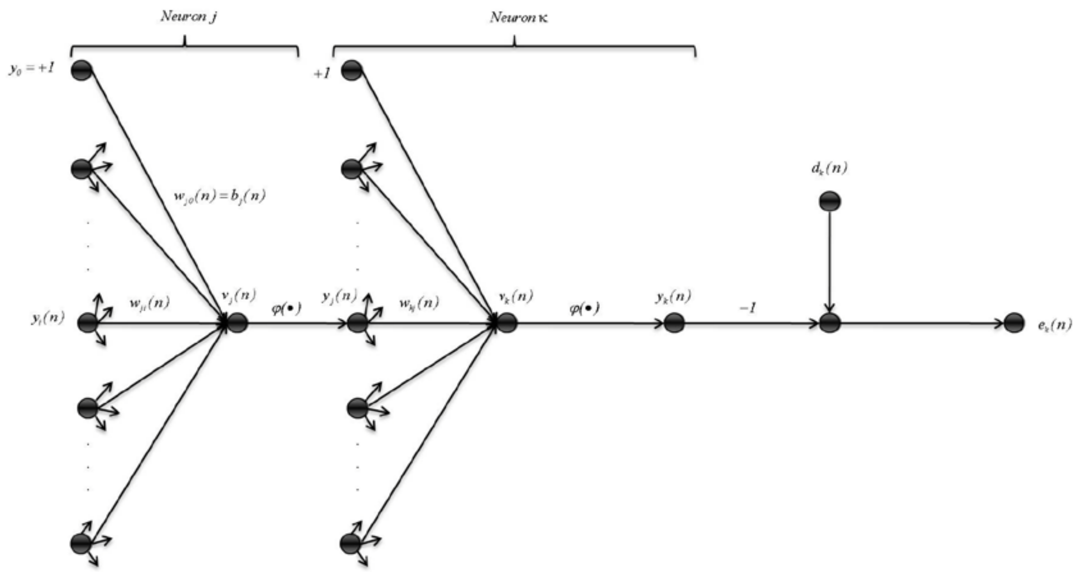


Figure 3.5: Signal-flow graph highlighting the details of output neuron *k* connected to hidden neuron *j*

According to Eq. (3.3.14), the local gradient $\delta_j(n)$ for hidden neuron *j* can be redefined as

$$\begin{aligned}\delta_j(n) &= -\frac{\partial \varepsilon(n)}{\partial v_j(n)}\frac{\partial y_j(n)}{\partial v_j(n)} \\ &= -\frac{\partial \varepsilon(n)}{\partial y_j(n)}\varphi_j'(v_j(n)),\end{aligned} \quad \text{neuron } j \text{ is hidden} \qquad (3.3.15)$$

where in the second line Eq. (3.3.9) is used. To calculate the partial derivative $\partial\varepsilon(n)/\partial y_j(n)$, it can be preceded as follows. From Figure 3.5, it can be seen that

$$\varepsilon(n) = \frac{1}{2}\sum_{k\in C} e_k^2(n), \text{neuron } k \text{ is an output node} \qquad (3.3.16)$$

which is Eq. (3.3.2) with index *k* used in place of index *j*. Differentiating Eq. (3.3.16) with respect to the function signal $y_j(n)$, yields

$$\frac{\partial \varepsilon(n)}{\partial y_j(n)} = \sum_k e_k \frac{\partial e_k(n)}{\partial y_j(n)} \qquad (3.3.17)$$

Next the chain rule is applied to the partial derivative $\partial e_k(n)/\partial y_j(n)$, and rewrite Eq. (3.3.17) in the equivalent form

$$\frac{\partial \varepsilon(n)}{\partial y_j(n)} = \sum_k e_k(n) \frac{\partial e_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial y_j(n)} \tag{3.3.18}$$

From the Figure 3.5, it is note that

$$\begin{aligned} e_k(n) &= d_k(n) - y_k(n) \\ &= d_k(n) - \varphi_k(v_k(n)), \end{aligned} \quad \text{neuron } k \text{ is an output node} \tag{3.3.19}$$

Hence

$$\frac{\partial e_k(n)}{\partial v_k(n)} = -\varphi_k'(v_k(n)) \tag{3.3.20}$$

It also note from the Figure 3.5 that for neuron *k* the induced local field is

$$v_k(n) = \sum_{j=0}^{m} w_{kj}(n) y_i(n) \tag{3.3.21}$$

where m is the total number of input (excluding the bias) applied to neuron *k*. Again, the synaptic weight $w_{k0}(n)$ is equal to the bias $b_k(n)$, applied to neuron *k*, and the corresponding input is fixed at the value +1. Differentiating Eq. (3.3.21) with respect to $y_j(n)$ yields

$$\frac{\partial v_k(n)}{\partial y_j(n)} = w_{kj}(n) \tag{3.3.22}$$

By using the equations. (3.3.20) and (3.3.22) in (3.3.18), the desired partial derivative is achieved:

$$\begin{aligned} \frac{\partial \varepsilon(n)}{\partial y_j(n)} &= -\sum_k e_k(n) \varphi_k'(v_k(n)) w_{kj}(n) \\ &= -\sum_k \delta_k(n) w_{kj}(n) \end{aligned} \tag{3.3.23}$$

where in the second line the definition of the local gradient $\delta_k(n)$ given in Eq. (3.3.14) with the index *k* substituted for *j* is used.

Finally, using Eq. (3.3.23) in (3.3.15), the back-propagation formula for the local gradient $\delta_j(n)$ is acquired as described:

$$\delta_j(n) = \varphi_j'(v_j(n)) \sum_k \delta_k(n) w_{kj}(n), \text{neuron } j \text{ is hidden} \qquad \textbf{(3.3.24)}$$

In Figure 3.6, demonstrates the signal-flow graph representation of Eq. (3.3.24), assuming that the output layer consists of neurons.

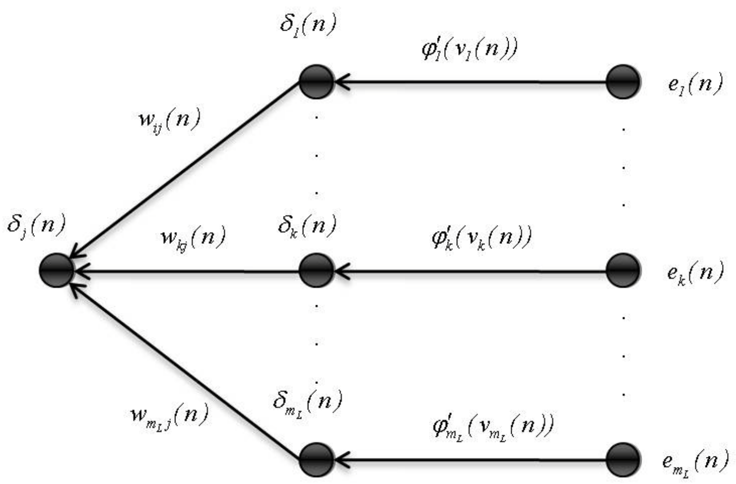


Figure 3.6: Signal-flow graph of a part of the adjoint system pertaining to back-propagation of error signals

The factor $\varphi_j'(v_j(n))$ involved in the computation of the local gradient $\delta_j(n)$ in the Eq. (3.3.24) depends solely on the activation function associated with hidden neuron *j*. The remaining factor involved in this computation, namely the summation over *k*, depends on two sets of terms. The first set of terms, the $\delta_k(n)$ requires knowledge of the error signals $e_k(n)$, for all neurons that lie in the layer to the immediate right of hidden neuron *j*, and that are directly connected to neuron *j*: see in the Figure 3.5. The second set of terms, the $w_{kj}(n)$, consists of the synaptic weights associated with these connections.

In summary, the relations derived for the back-propagation algorithm can be divided into two components. First, the correction $\Delta w_{ji}(n)$ applied to the synaptic weight connecting neuron *i* to neuron *j* is defined by the delta rule:

$$\begin{pmatrix} weight \\ correction \\ \Delta w_{ji}(n) \end{pmatrix} = \begin{pmatrix} learning - \\ rate\ parameter \\ \eta \end{pmatrix} \cdot \begin{pmatrix} local \\ gradient \\ \delta_j(n) \end{pmatrix} \cdot \begin{pmatrix} input\ signal \\ of\ neural\ j \\ y_i(n) \end{pmatrix} \tag{3.3.25}$$

Second, the local gradient $\delta_j(n)$ depends on whether neuron *j* is an output node or a hidden node:

1) If neuron *j* is an output node, $\delta_j(n)$ equals the product of the derivative $\varphi'_j\left(v_j(n)\right)$ and the error signal $e_j(n)$, both of which are associated with neuron *j*, as shown in the Eq. (3.3.14).
2) If neuron *j* is a hidden node, $\delta_j(n)$ equals the product of the associated derivative $\varphi'_j\left(v_j(n)\right)$ and the weighted sum of the δs computed for the neurons in the next hidden or output layer that are connected to neuron *j*, as shown in the Eq. (3.3.24).

CHAPTER IV

RESEARCH METHODOLOGY

4.1 Database

The CMU-Pitts burgh AU-Coded Face Expression Image Database is selected for this experiment because it is a reliable benchmark which has a variety of samples. For instance, it contains 210 adults whose ages range from 18 to 50 years old and 69% of them are females. For a race aspect, this database contains 81% of Euro-American, 13% of Afro-American, and 6% of other groups. This database was recorded by Panasonic AG-7500 cameras with two levels of image resolution: 640x490 or 640x480 pixel arrays, with 8 – bit gray-scale or 24 – color values. Participants were recorded in each sample in a series of 23 facial displays with frontal view and no occlusion. Each sample began and ended with neutral face display [7]. Table 2.1 in Chapter 2 has shown a description of this database.

4.2 Graph-based feature extraction

The graph-based feature extraction proposed in this thesis consists of two procedures which are locating points in face images and constructing feature vectors. Locating points in face images will be described about the selected regions that influence on the emotional expressions in order to construct feature vectors from distances between each point and then to find a relation of face geometry and six basic emotions.

4.2.1 Locating fourteen points

The relationships between emotions and facial Action Units (AU) are mentioned in [7] and [8]. From these two papers, the regions that influence on six basic expressions consist of eyes, eyebrows, nose and mouth whose numbers of Action Units are 1, 2, 4, 5, 6, 7, 10, 12, 13, 14, 15, 16, 17, 23, 24, 25, 26 and 27, as shown as the red stars in Table 2.5. In addition, the nose region can be ignored because of its minimal effect on the outlet of emotions. Twenty AU were used by P. Lucey and H. Zhao for description six basic emotions in term of facial Action Units. The nose region was represented by AU 9 and AU 11 so it is only two from twenty AU.

In order to achieve only the Action Units that are relevant to the emotional expressions, the graph construction based on fourteen points is introduced in this thesis. The fourteen points include two points of inner eyebrows, two points of middle eyebrows, two points of outer eyebrows, two points of inner eyes, two points of outer eyes and four points of mouths, and all these points are defined manually, as shown in Figure 4.1.

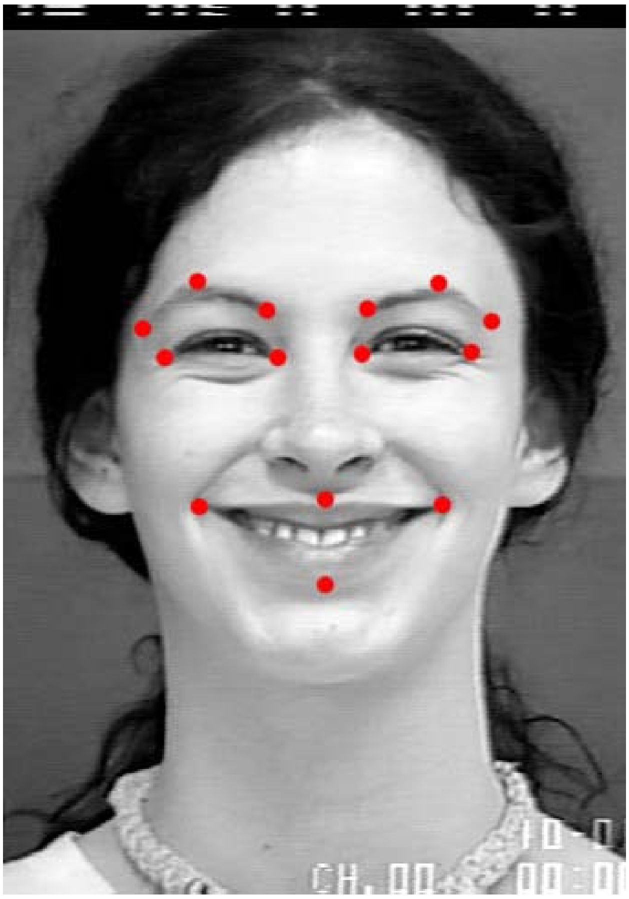


Figure 4.1: Locating fourteen points for constructing feature vectors

4.2.2 Constructing feature vectors

In this procedure, the formulation for computing the Euclidean distance is displayed in equation 4.2.1, and the Euclidean distances of all pairs of the fourteen points are computed for measuring the position change of face geometry and finding a relation of six basic emotions and the distances from 14 points in each image. Then, the 14x14 symmetric distance matrix is built, as shown in Table 4.1. Each matrix contains 196 feature distances of all pairs of the fourteen points, but there are 14 distances that can be discarded from the matrix due to the fact that the distance from each point to itself is equal to zero, as shown in a diagonal line in the matrix. So, the matrix contains 182 feature distances. Additionally, because of the symmetry in the matrix, it can be further reduced from 182 to 91 feature distances, as displayed in shading part of Table 4.1.

Table 4.1: 14x14 symmetric distance matrix computed from the located 14 points

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14
P1	0	1.632942	3.341676	4.971328	7.066944	8.049944	0.984073	3.219068	5.094948	7.374944	4.535813	7.552801	5.6364	6.922153
P2	1.632942	0	1.920026	3.523663	5.556564	6.610363	1.520691	2.154182	3.820366	6.020507	4.8956	6.857412	5.307928	6.85572
P3	3.341676	1.920026	0	1.631962	3.728874	4.720042	2.722499	0.891067	1.921067	4.101829	4.687729	5.304074	4.240212	5.953906
P4	4.971328	3.523663	1.631962	0	2.120849	2.120849	4.302615	2.013579	0.859884	2.527706	5.463854	4.601793	4.297732	6.018181
P5	7.066944	5.556564	3.728874	2.120849	0	1.229675	6.422904	4.109465	2.386001	1.364001	7.205699	4.946362	5.565618	7.123377
P6	8.049944	6.610363	4.720042	3.090582	1.229675	0	7.328547	4.942631	3.063609	0.947523	7.580237	4.555436	5.675509	7.053411
P7	0.984073	1.520691	2.722499	4.302615	6.422904	7.328547	0	2.402082	4.305125	6.598257	3.687221	6.576352	4.655857	5.977976
P8	3.219068	2.154182	0.891067	2.013579	4.109465	4.942631	2.402082	0	1.903182	4.196308	3.807571	4.723812	3.422426	5.110793
P9	5.094948	3.820366	1.921067	0.859884	2.386001	3.063609	4.305125	1.903182	0	2.293142	4.860422	3.760332	3.495211	5.196778
P10	7.374944	6.020507	4.101829	2.527706	1.364001	0.947523	6.598257	4.196308	2.293142	0	6.644833	3.718091	4.72889	6.129437
P11	4.535813	4.8956	4.687729	5.463854	7.205699	7.580237	3.687221	3.807571	4.860422	6.644833	0	4.545327	2.274071	2.655297
P12	7.552801	6.857412	5.304074	4.601793	4.946362	4.555436	6.576352	4.723812	3.760332	3.718091	4.545327	0	2.29264	2.810872
P13	5.6364	5.307928	4.240212	4.297732	5.565618	5.675509	4.655857	3.422426	3.495211	4.72889	2.274071	2.29264	0	1.74
P14	6.922153	6.85572	5.953906	6.018181	7.123377	7.053411	5.977976	5.110793	5.196778	6.129437	2.655297	2.810872	1.74	0

$$D(p,q) = \sqrt{(p_2 - p_1)^2 + (q_2 - q_1)^2} \qquad (4.2.1)$$

From the previous description, there are only 91 distances represented for all possible pairs in the located fourteen points. In addition, all feature distances are normalized by the distance of diagonal line across the face using the equation in 4.2.2 in order to reduce image scaling errors obtained from the distance between face and camera, as shown in Figure 4.2.

$$\hat{D}(p,q) = \frac{D(p,q)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \qquad (4.2.2)$$

Where $\hat{D}(p,q)$ is a normalized distance between p and q calculated from the ratio of $D(p,q)$, original distance, and the distance of diagonal line across the face with $(x_1, y_1)$is a left lower point of image and (x_2, y_2) is a upper right point of image

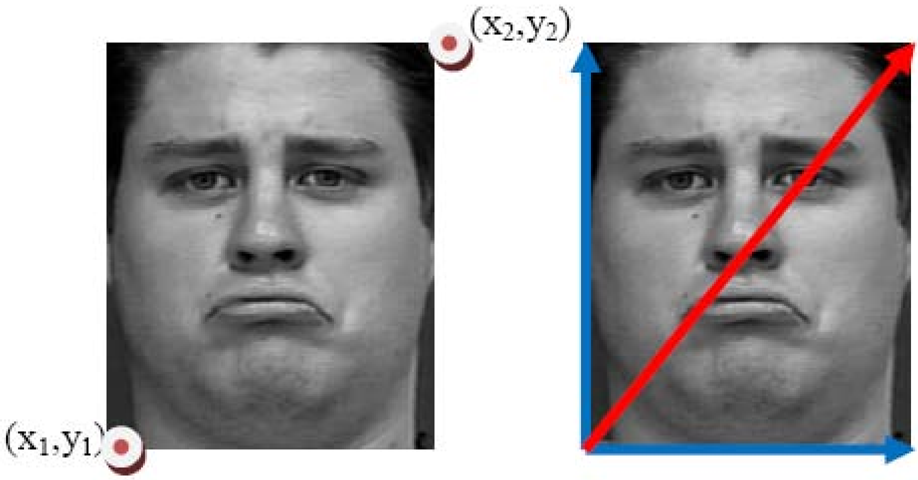
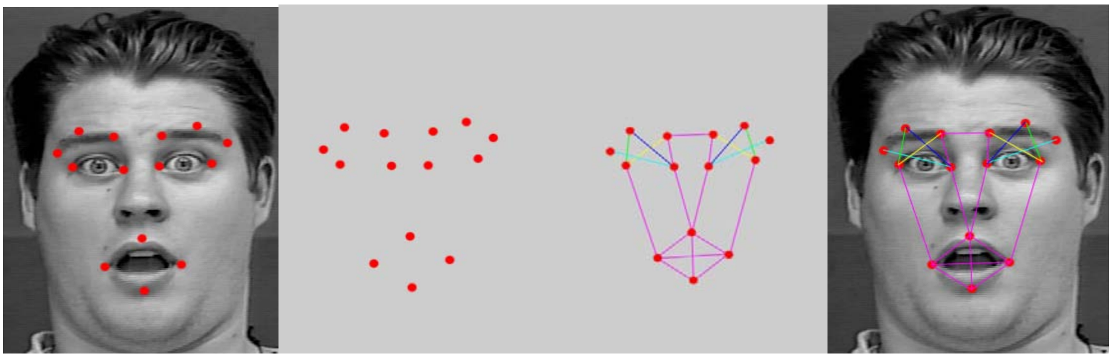


Figure 4.2: Normalized distance by dividing original distance with diagonal line across the face

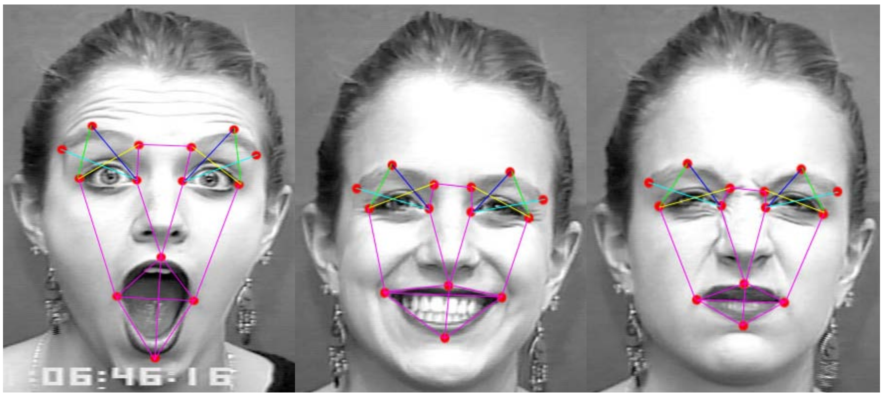
Even though the types of emotional expressions are changed, some distances, such as the distance between outer eyebrow and inner eyebrow, are almost unchanged. Therefore, this thesis focuses on the distances influenced on the relation of Action Units and emotional expressions.

Accordingly, 91 feature distances can be reduced to 21 feature distances. This yields an increase in data distribution. Figure 4.3 shows Graph-based feature construction in each stage and Figure 4.4 shows the comparison of graphs among six basic emotions.

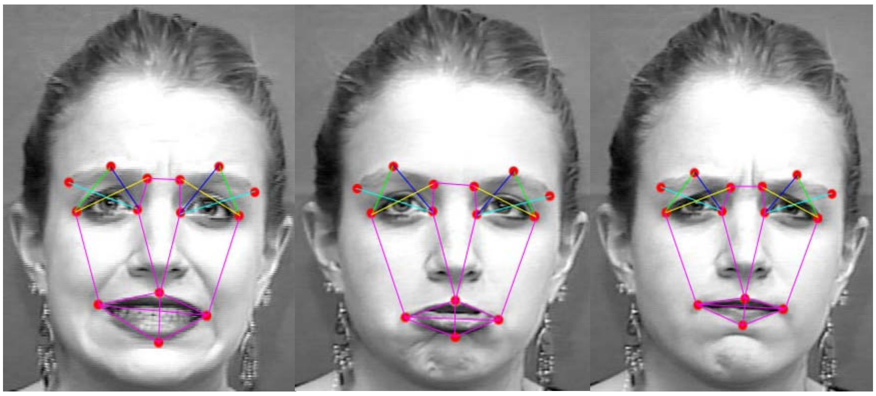


(a) (b) (c) (d)

Figure 4.3: Graph-based feature construction in each stages: (a) Locate 14 facial points, (b) Find distance between each points to form connecting edge



(a) 'Surprise' (b) 'Happy' (c) 'Disgust'



(d) 'Fear' (e) 'Sadness' (f) 'Anger'

Figure 4.4: Example of Graph-based features on six basic emotions

In addition, an angle between two lines is computed by an equation 4.2.3 in order to gain more extracted features for measuring the facial movements. A computation of angle used for this experiment consists of two steps: a construction of vector by subtracting coordinate values in each axis, and a computation if angle between two vectors by using the equation (4.2.3).

$$\hat{A} \cdot \hat{B} = |A||B| \cos\theta \qquad (4.2.3)$$

Where $\hat{A}$ is vector A and $|A|$ is length of A $\hat{B}$ is vector B and $|B|$ is length of B.

Since the Graph-based features in this experiment consist of 21 feature distances, as shown in the Figure 4.3 and 4.4, all possible angles from all pair of these 21 distances after discarding the permutation of the same distance pair is 26 angles, as shown in the Figure 4.5.

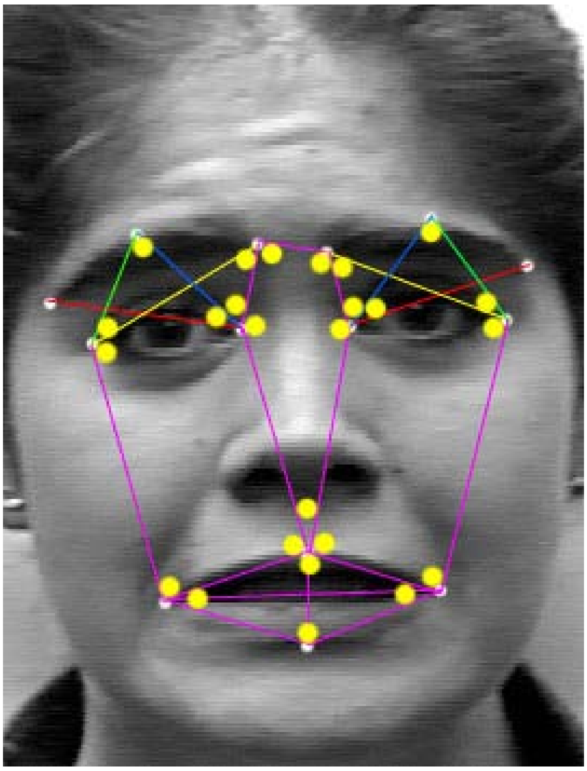


Figure 4.5: Example after applied angle features

4.3 Recognition system

Artificial Neural Networks (ANNs) are beneficial and are applied in various sciences, e.g. mathematics, statistics, physics, engineering, and computer science. Artificial Neural Networks can be used in several applications, such as time series analysis, signal processing, and pattern recognition, with learning ability from input vectors. There are various kinds of neural network model in Artificial Neural Networks, and Multilayer neural network is a widely used network model. In general, Multilayer neural network consists of three layer types: input layer, at least one hidden layer of computation neurons, and output layer of computation neuron. Multilayer-perceptron is widely used to solve some difficult and diverse problems by training them in supervised manner with the error calculated from back-propagation learning algorithm based on error-collection learning rules. Basically, the network consists of two parts in different directions: a forward pass and a backward pass. In forward pass, input vector is applied to the sensory neuron of the networks then output value was produced while, in the backward pass, all of synaptic weights were adjusted with respect to the difference between target and actual output.

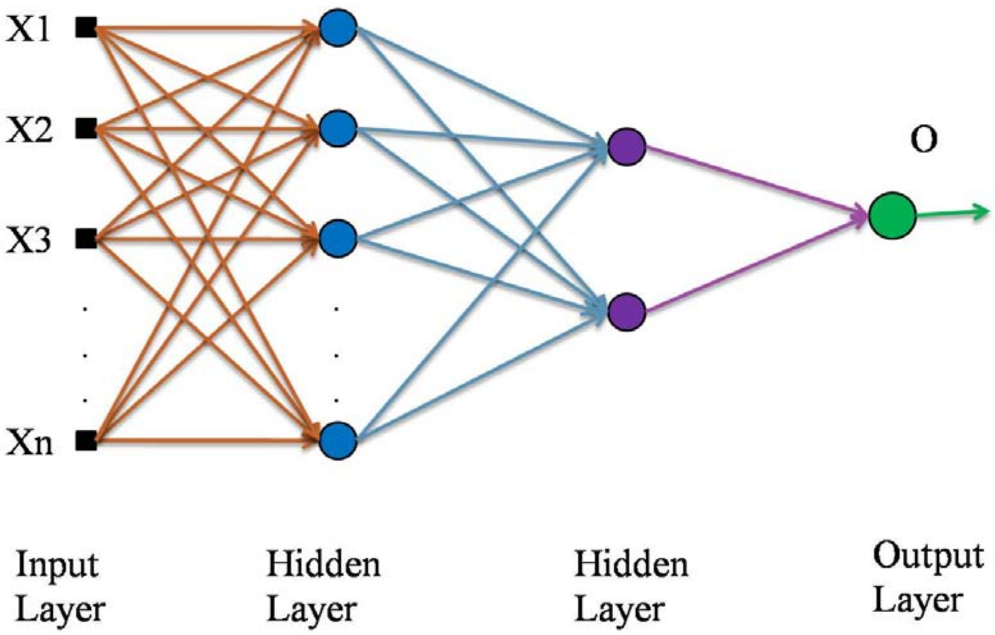


Figure 4.6: MLP with back-propagation learning algorithm with two hidden layers

In part of recognition system, the proposed method consists of five neural networks for the classification of emotional expressions. In this experiment, recognition architecture is Multilayer perceptron networks with back-propagation learning algorithm applied in term of binary tree, as displayed in Figure 4.7. According to the architecture, the decision vertices in bi-classification tree are sorted by easiness of classification. To illustrate this point, ‘Surprise’ is the simplest emotion to distinguish because it is obviously identified from mouth region covered by complete graph with four points as a shape of large parallelogram. The next three emotions are ‘Happy’, ‘Disgust’, and ‘Fear’, respectively. The last two emotions that are difficult to distinguish are ‘Sad’ and ‘Anger’ so they are in the last pair of bi-classification.

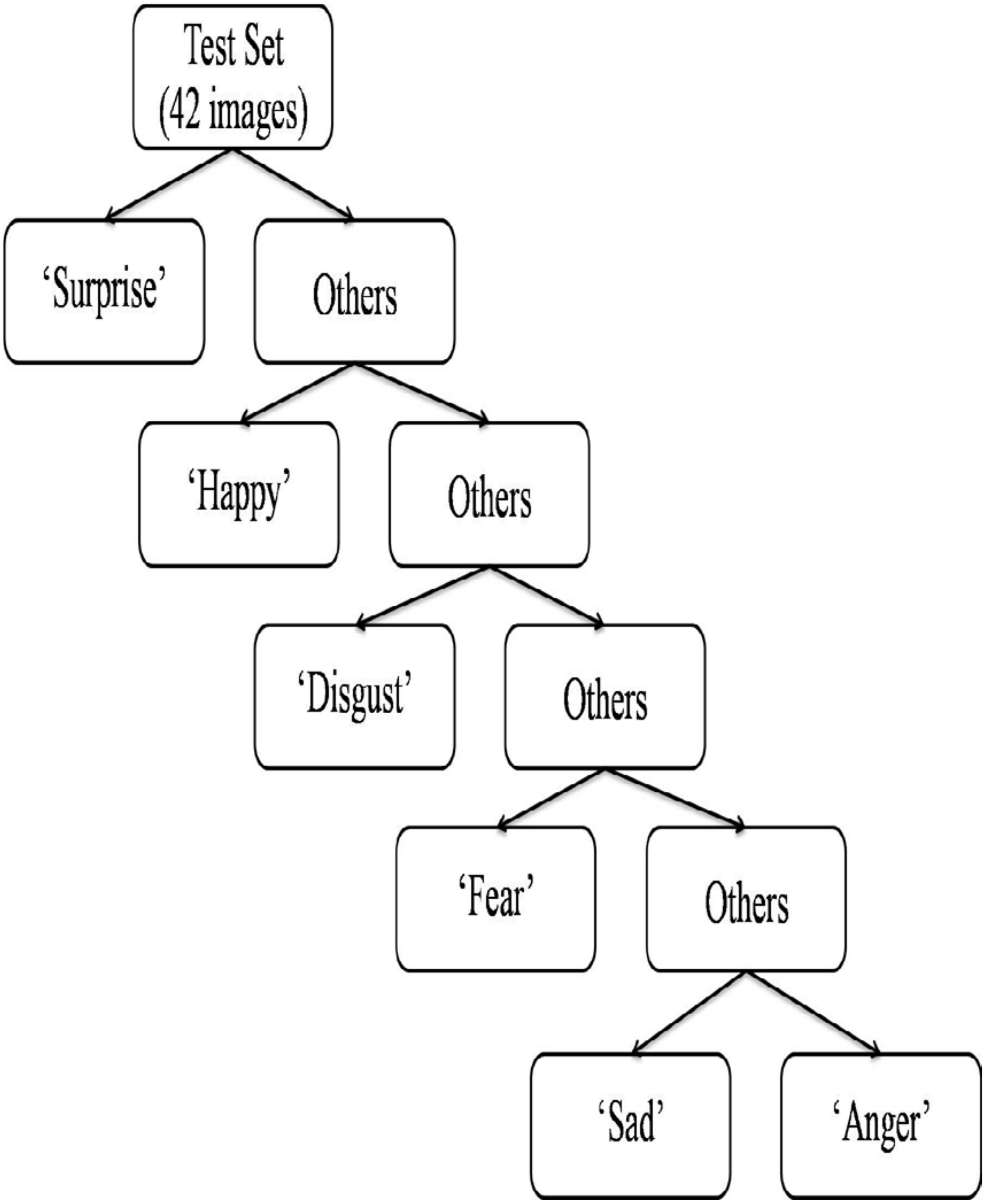


Figure 4.7: Binary tree of classification based on neural networks

Table 4.2: List of training algorithms

	Training Algorithm	Input Layer	Hidden Layer	Hidden Node	Output Layer
Surprise	Levenberg-Marquardt	21	1	12	1
Happy	Scaled Conjugate Gradient	21	2	32,16	1
Disgust	Fletcher-Powell Conjugate Gradient	21	2	16,2	1
Fear	Gradient Descent	21	1	8	1
Sad and Anger	BFGS Quasi-Newton	21	2	9	1

In addition, Table 4.2 is the description of specified condition used in the five neural networks. The learning rate and maximum epoch were 0.1 and 10000, respectively. The accepted threshold for activation function was 0 because training targets were set to be one and minus one.

Theoretically, single hidden layer is sufficient for classifying the emotional expressions. Occasionally, some types of data, such as low distribution data, are difficult to totally classify in single hidden layer. In these cases, multiple hidden layers can achieve the higher speed in the classification because they have capability of dividing the data more than single layer.

CHAPTER V

EXPERIMENTAL RESULTS

5.1 Experimental Results

To evaluate, 90 images from 15 persons are randomly chosen with the same criteria as recent research [11]. There are six basic emotions consisting of ‘Surprise’, ‘Sad’, ‘Happy’, ‘Disgust’, ‘Fear’ and ‘Anger’. In case of MLP without validation, those images can be divided into two groups: The first group is used for training set containing 48 images of eight persons, each of which consists of six images of emotions. The other group is used for test set that have 42 images of seven persons. In case of MLP with validation, 36 images are used for training, 12 images are used for the validation, and 42 images are used for testing so each emotion containing 15 persons will be composed of 6 persons for training, 2 persons for validation, and 7 persons for testing.

Five neural network models applied to this experiment are as follows:

1. MLP (Graph-based features) without validation
2. MLP (Graph-based features) with validation
3. MLP (Graph-based features + angle features) without validation
4. MLP (Angle features) without validation
5. MLP (Graph-based features) k-fold cross validation

5.1.1 MLP (Graph-based features) without validation

For the classes of ‘Surprise’, ‘Happy’, and ‘Disgust’, the images were correctly classified. Those emotions were the three first bi-classifications in binary tree. However, ‘Fear’, ‘Sad’, and ‘Anger’ are a bit overlapped so they are difficult to distinguish. As shown in figure 4.1, the test set contains 42 images classified into two classes: ‘Surprise’ and other emotions. All of them are correctly classified. Thus, the number of images classified into other emotions is 35. In the second level, 35 images from the previous level were classified into ‘Happy’ class of seven images and other emotions of 28 images. The third level is to classify the feature vectors into classes of ‘disgust’ and the rest. Again, all images in this level were correctly classified. In fourth step, 21 images were classified into eight ‘Fear’ images and 13 images of other emotions. However, only seven images were correctly classified so one image from ‘sad’ class was misclassified into this class. For the last level, 13 images were classified into seven ‘sad’ images and six ‘Anger’ images. Nevertheless, six images in ‘sad’ class were correctly classified while the misclassification came from one image belongs to ‘Anger’ class. To evaluate our system, confusion matrix is applied as shown in table 5.1. The row corresponds to six actual emotions while the column corresponds to six predicted emotions.

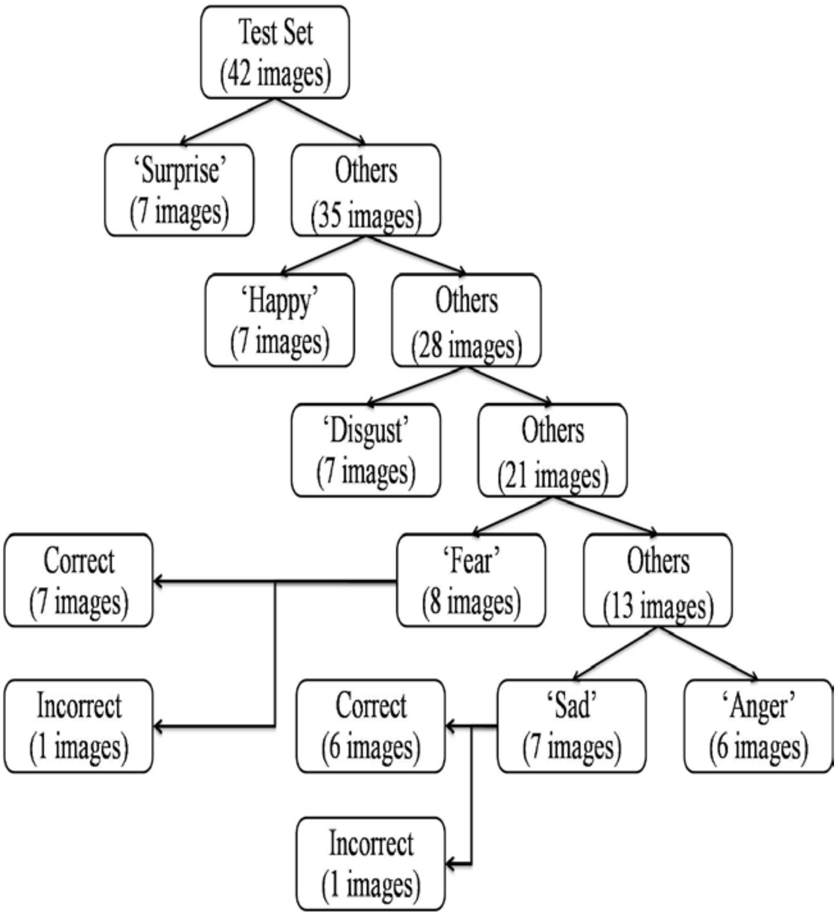


Figure 5.1: MLP (Graph-based features) without validation

5.1.2 MLP (Graph-based features) with validation

In case of MLP (Graph-based features) with validation, the experimental results in the first three levels are similar to the previous results, while a difference in the other level is slightly occurred as follows. For the 'Fear' class, 6 images are correctly classified in the fourth level while the other is misclassified to 'Anger' class. For the 'Sad' class, 6 images are also correctly classified in the last level while the other is misclassified to 'Anger' class. For the 'Anger' class, 6 images are also correctly classified in the last level while the other is misclassified to 'Sad' class.

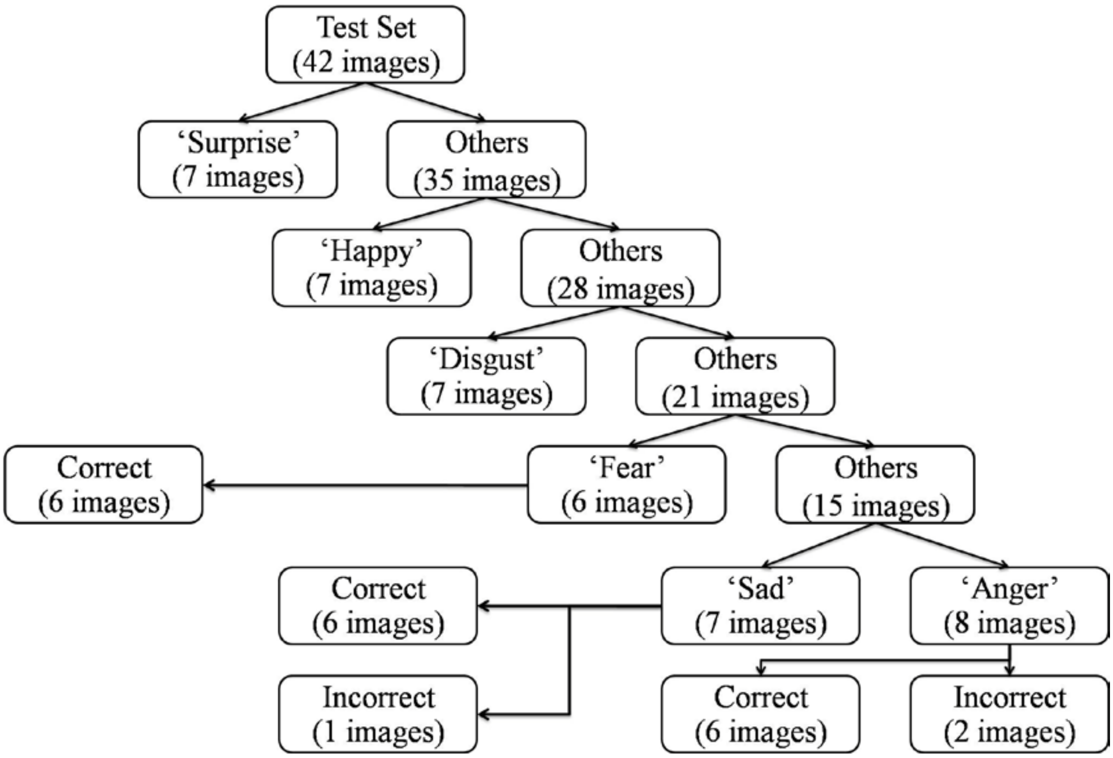


Figure 5.2: MLP (Graph-base features) with validation

5.1.3 MLP (Graph-based features + angle features) with no validation

In case of MLP (Graph-based features + angle features) with no validation, the experimental results in the first two levels are totally correct and the misclassifications are occurred as follows. For the third level, even though all images from the 'Disgust' class are totally correct, one image from 'Fear' class and two images from 'Anger' class are misclassified to the 'Disgust' class as well. For the fourth level, only five images of class 'Fear' are classified correctly. For the 'Sad' class in the last level, all images from the 'Sad' class are correctly classified while one image from 'Anger' class is misclassified to 'Sad' class. For the 'Anger' class in the last level, only four images of class 'Anger' are classified correctly due to the previous misclassification, while one image from 'Fear' class is misclassified to 'Anger' class.

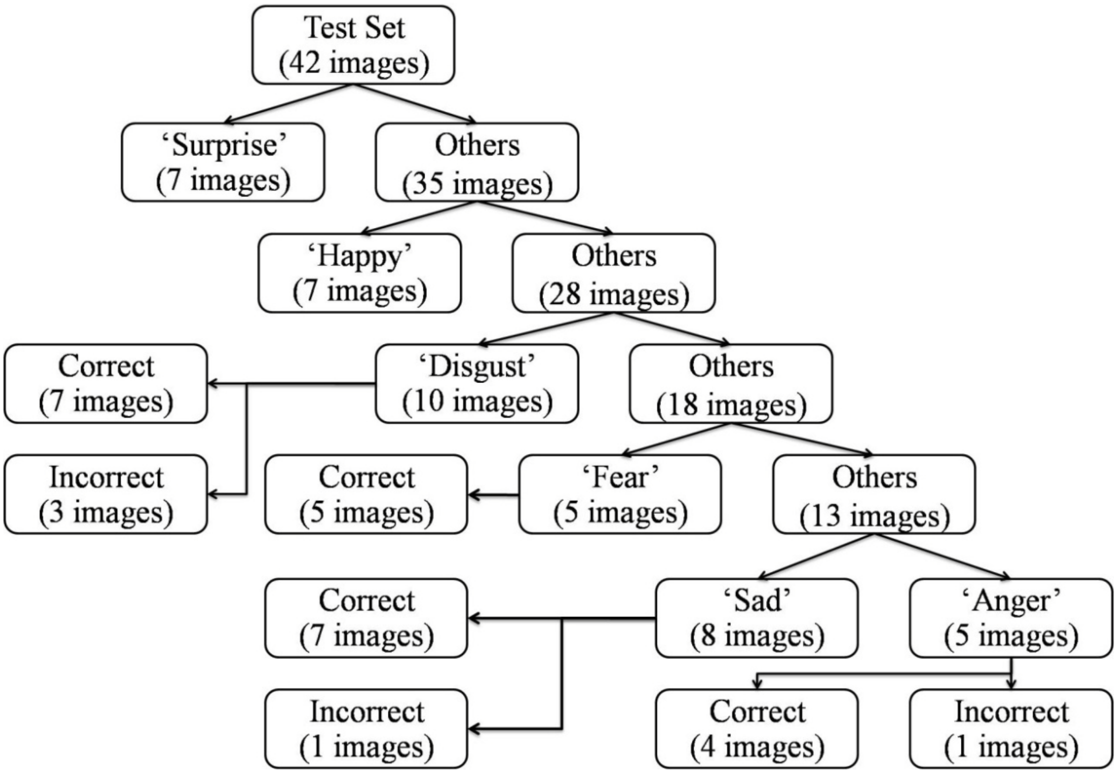


Figure 5.3: MLP (Graph-based features + angle features) without validation

5.1.4 MLP (Angle features) with no validation

In case of MLP (Angle features) with no validation, the experimental results in the first two levels are totally correct and the misclassifications are occurred in the other levels as follows. For the third level, even though all images from the 'Disgust' class are totally correct, one image from 'Anger' class are misclassified to the 'Disgust' class as well. For the fourth level, all images of class 'Fear' are correctly classified. For the 'Sad' class in the last level, six images are correctly classified while one image from 'Anger' class is misclassified this class. For the 'Anger' class in the last level, only five images are correctly classified while one image from 'Sad' class is misclassified to this class.

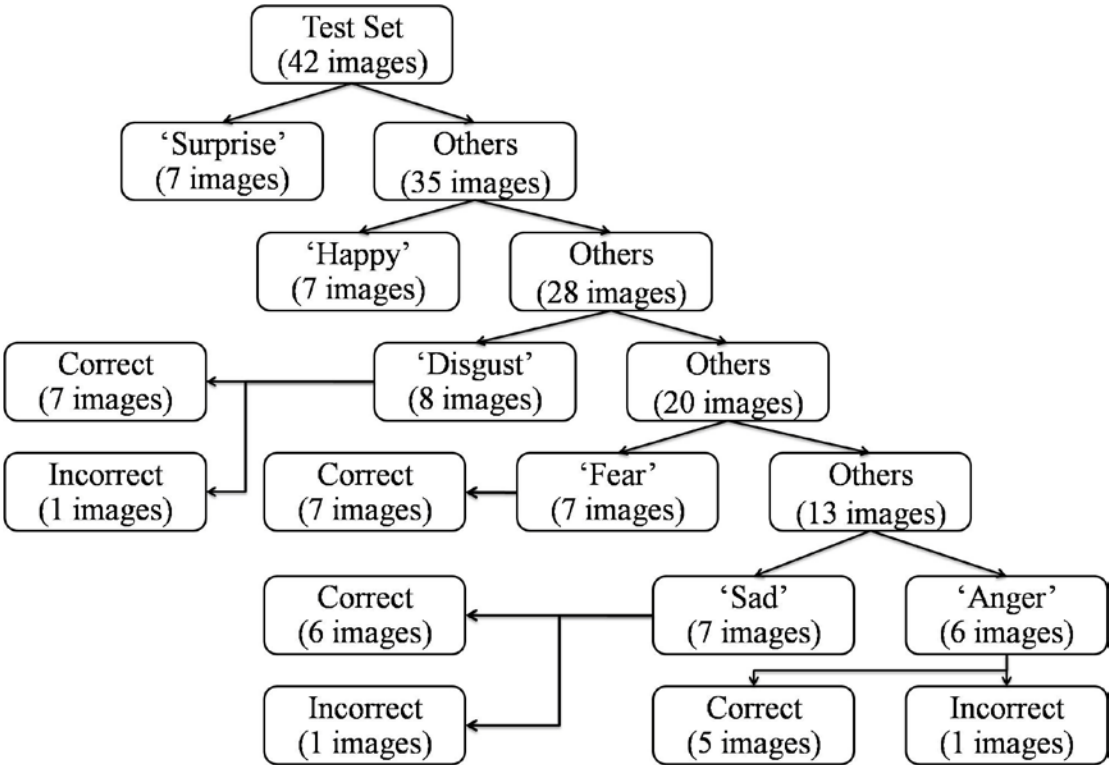


Figure 5.4: MLP (Angle features) with no validation

5.1.5 MLP (Graph-based features) with cross validation

In case of MLP (Graph-based features) with cross validation, the experimental results in the first four levels are totally correct while the misclassifications of two images from the 'Anger' class are occurred in the last level.

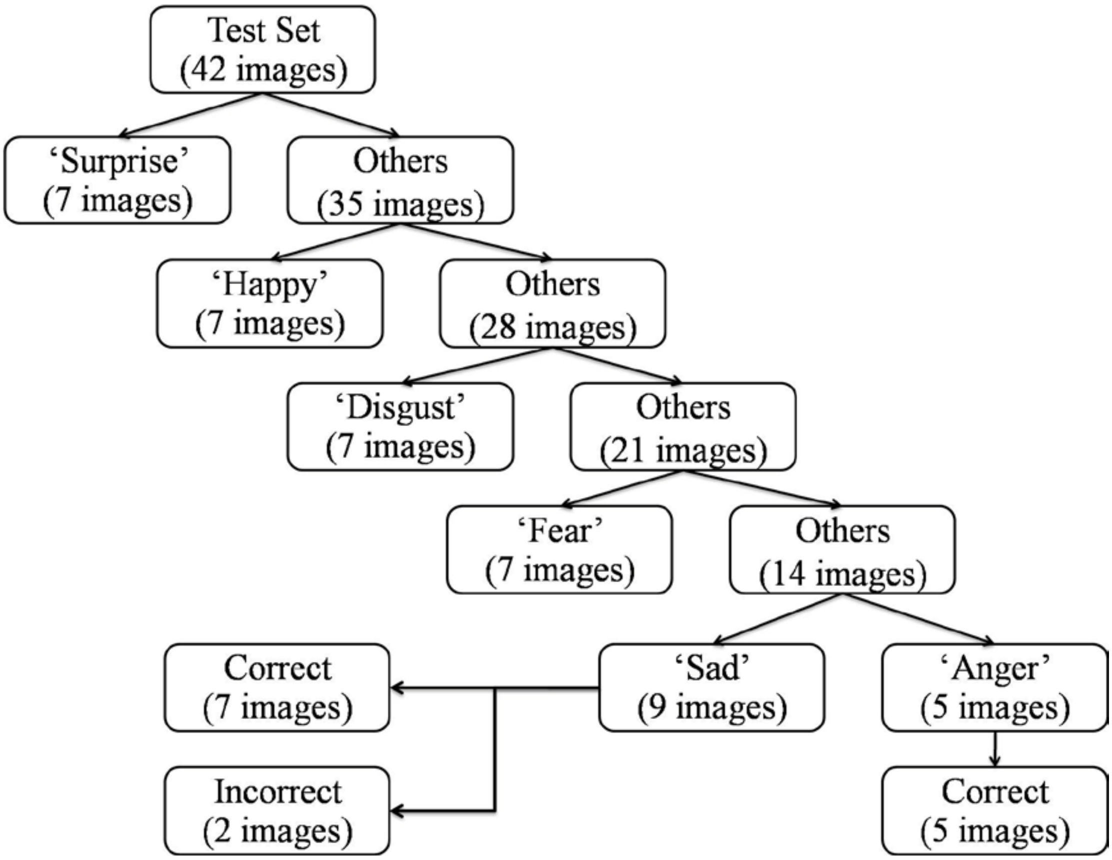


Figure 5.5: MLP (Graph-based features) with cross validation

5.2 Accuracy Evaluation

The diagonal entries show percentage of correct classification for each class while off-diagonal entries depict the misclassification. To compare with the modern existing techniques [11], the accuracy is computed by the equation (5.1).

$$\%\ accuracy = \frac{Number\ of\ \ correct\ classifications}{Number\ of\ \ all\ classifications} \times 100 \qquad (5.1)$$

Table 5.1 has represented the percentage of accuracy of the proposed recognition system under five conditions.

In the aspect of validation, the experimental results from the MLP (Graph-base features) without validation and MLP (Graph-based features) with cross validation can achieve the higher performance than MLP (Graph-base features) with validation. This is because dividing images for the validation is required, thus the number of images for training in the recognition system with validation will be less than the model without the validation, and the reason why the model with cross validation can also obtain the high performance in spite of having the requirement in dividing the images for validation is that this model employs the images required for validation in training process so all images are trained and tested in this model.

In the aspect of features, the experimental results have shown that MLP (Graph-base features) without validation can achieve the highest performance (95.24%), while the performances of MLP (angle features) without validation and MLP (Graph-based features + angle features) without validation are 92.86% and 88.10%, respectively. From these results, they have illustrated that Graph-base features contains the feature vectors that are more relevant for the facial expression recognition than the angle features. Nevertheless, even though both of Graph-base features and angle features can acquire the high accuracy, the combination of these two features does not lead to the higher performance. This is because the combination will impact on the distribution of the data so the recognition system will be hard to distinguish the emotional expressions.

Table 5.1: The percentage of accuracy of the proposed recognition system under 5 conditions

Method	Percentage of Accuracy
MLP (Graph-based features) without validation	95.24
MLP (Graph-based features) with validation	92.86
MLP (Graph-based feature + feature angle) without validation	88.10
MLP (angle features) without validation	92.86
MLP (Graph-based features) with cross validation	95.24

Table 5.2: Confusion matrix of MLP (Graph-base features) without validation

	Surprise	Sad	Happy	Fear	Disgust	Anger
Surprise	100	0	0	0	0	0
Sad	0	85.71	0	14.29	0	0
Happy	0	0	100	0	0	0
Fear	0	0	0	100	0	0
Disgust	0	0	0	0	100	0
Anger	0	14.29	0	0	0	85.71

Table 5.3: Confusion matrix of MLP (Graph-based features) with cross validation

	Surprise	Sad	Happy	Fear	Disgust	Anger
Surprise	100	0	0	0	0	0
Sad	0	85.71	0	0	0	14.29
Happy	0	0	100	0	0	0
Fear	0	0	0	100	0	0
Disgust	0	0	0	0	100	0
Anger	0	14.29	0	0	0	85.71

Table 5.4: Comparison of accuracy of proposed scheme

Method	Percentage of Accuracy
Moment Invariants (MI)	73.81
Singular Value Decomposition (SVD)	76.19
MI + SVD	90.48
Graph-based Features with Neural Networks	95.24

CHAPTER VI

CONCLUSION

6.1 Conclusions

In this thesis, the high accuracy recognition system based on machine learning with a reasonable number of samples and a small amount of facial points are introduced. Graph-based features were developed by locating fourteen facial points that influence directly on the expression of emotions. Then, Euclidean distances between each point were computed and reduced to twenty one distances for describing geometric facial structure. For the recognition systems of six basic emotions, multilayer-perceptrons with back-propagation was adopted to recognize these graph-based features by constructing five neural networks in a form of binary tree. These neural networks were used to classified test set in each state of ‘Surprise’, ‘Happy’, ‘Disgust’, ‘Fear’, ‘Sad’, and ‘Anger’, respectively. The proposed method was evaluated with test set comprised of 42 images from six basic emotions and the results have shown that the proposed method can perform 95.24% accuracy. Two failure cases are from the misclassification from ‘Sad’ class to ‘Fear’ class and from ‘Anger’ class to ‘Sad’.

6.2 Suggestion

For this experiment, the proposed method can be divided into two parts: construction of facial features from face image and recognition system for classifying the emotional expressions. In part of construction the facial features, 14 points were proposed for measuring the facial movements, and then the Euclidean distance for all pairs of the fourteen points were computed. As a result, the 14x14 symmetric distance matrices were built. After consideration of the unnecessary computation in case of the source and the destination are the same points and the symmetry of the matrices, each matrix can be reduced to 91 feature distances. Then, they were normalized by the distance of the diagonal line across the face to decrease the image scaling errors, and only 21 feature distances that influence on the recognition of the emotional expressions were selected, as described in the research methodology of Chapter 4.

Nevertheless, the 21 feature distances were selected from the researcher's observation on the overview of this database. Therefore, the feature vectors were chosen without statistical information so these 21 features might not optimize the recognition systems. Although the experimental results have shown that the 21 feature vectors is good enough for the recognition system, the number of the feature vectors are directly impact on the recognition rate. Consequently, researches for the relevant features in addition to the appropriate number of the features are necessary for recognizing the facial expressions potentially.

Multilayer perceptron network with back-propagation learning algorithm was selected for recognizing the facial expressions in this thesis. This recognition system contains five neural networks for the classification of six basic emotions, and the bi-classification tree for the system is sorted by the easiness of distinction. Hence, an order of sorting might affect on the recognition rate in the classification process. Additionally, although the experimental results have shown that the recognition system can achieve the high accuracy in the recognition rate, the database that applied to the system has a small number of samples.

In summary, the future works that are necessary for the facial expression recognition are as follows:

1. Databases which have a large number of samples with various kinds of condition, such as gender, race, age, and brightness, are needed to improve the performance of the recognition.

2. The relevant features as well as the suitable number of the features are necessary for optimizing the facial expression recognition

3. The effect of the arrangement in bi-classification tree of the classification will be studied.

References

[1] Mehrabian, "Communication without words," **Psychology Today**, vol.2, no. 4, 1968.

[2] A. Ryan, J. Cohn, S. Lucey, J. Saragih, P. Lucey, F. D. la Torre and A. Rossi., "Automated Facial Expression Recognition System," **In Proceedings of the International Carnahan Conference on Security Technology**, pages 172–177, 2009.

[3] A. Vinciarelli, M. Pantic, and H. Bourlard; "Social Signal Processing: Survey of an Emerging Domain.," **Image and Vision Computing** 31(1):1743–1759, 2009.

[4] E. Vural, M. Cetin, A. Ercil, G. Littlewort, M. Bartlett, and J. Movellan, "Automated Drowsiness Detection for Improved Driver Saftey Comprehensive Databases for Facial Expression Analysis," **In Proceedings of the International Conference on Automotive Technologies**, 2008.

[5] P. Lucey, J. Cohn, S. Lucey, I. Matthews, S. Sridharan, and K. Prkachin, "Automatically Detecting Pain Using Facial Actions," **In Proceedings of the International Conference on Affective Computing and Intelligent Interaction**, pages 1–8, 2009.

[6] P. Ekman and W. V. Friesen, "Facial Action Coding System," **CA: Consulting Psychologist Press**, 1978.

[7] P. Lucey, J. 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," **Computer Vision and Pattern Recognition Workshops (CVPRW)**, 2010 IEEE Computer Society Conference, Page(s): 94 – 101, 2010.

[8] H. Zhao; T. Xue; L. Han, Facial complex expression recognition based on Latent Dirichlet Allocation, **Natural Computation (ICNC)**, 2010 Sixth International, Page(s): 1958 - 1960

[9] A. Metallinou, C. Busso, S. Lee, S. Narayanan, "Visual emotion recognition using compact facial representations and viseme information," **Acoustics Speech and Signal Processing (ICASSP)**, Acoustics Speech and Signal Processing

(ICASSP), 2010 IEEE International Conference on Issue Date: 14-19 March 2010 on page(s): 2474 – 2477, 2010.

[10] L. Oliveira, M. Mansano, A. Koerich, A. Britto Jr., "Selecting 2DPCA Coefficients for Face and Facial Expression Recognition," **Computing in Science & Engineering**, Volume: PP , Issue: 99, Page(s): 1, 2011.

[11] W. Guojiang, Y. Guoliang, F. Kechang, "Facial expression recognition using moment invariants and singular value decomposition," **Advanced Computer Control (ICACC)**, 2010 2nd International Conference on Volume: 4, Page(s): 132 – 135, 2010.

[12] L. Chen, T. Zhang, X. Gui and G. Yang, "Development of research on facial expression recognition," **Intelligent Control and Information Processing (ICICIP)**, page(s): 451 - 454, 2010

[13] T.Kanade, J.F.Cohn, Y. Tian, "Comprehensive Database for Facial Expression Analysis," **Automatic Face and Gesture Recognition. Proceedings**, Fourth IEEE International Conference on Digital Object Identifier, Page(s): 46 – 53, 2000

[15] M. H. Beale, M. T. Hagan, H. B. Demuth, **Neural Network Toolbox™ 7 User's Guide.** The MathWorks, Inc (2010).

[1o] S. Haykin, **Neural Networks: A comprehensive foundation**, E.d., Prentice Hall International, Inc (1999).

VITAE

Chaiyasit Tanchotsrinon was born in June , 1986, in Bangkok, Thailand. He obtained his Bachelor's Degree in Survey Engineering and Geographic Information from the Faculty of Engineering, Kasetsart University, Bangkean campus in 2008.