UNIFORM LOCAL BINARY PATTREN WITH IMPROVED FIREFLY FEATURE SELECTION FOR FACIAL EXPRESSION RECOGNITION

ABDULLA M.K. ELMADHOUN

UNIVERSITI KEBANGSAAN MALAYSIA

UNIFORM LOCAL BINARY PATTERN WITH IMPROVED FIREFLY FEATURE SELECTION FOR FACIAL EXPRESSION RECOGNITION

ABDULLA M.K. ELMADHOUN

DISSERTATION SUBMITTED IN PARTIAL FULFILMENT FOR THE DEGREE OF MASTER OF COMPUTER SCIENCE

FACULTY OF INFORMATION SCIENCE AND TECHNOLOGY
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

2017

POLA BINARI SETEMPAT SERAGAM DENGAN PEMILIHAN FITUR KELIP-KELIP DIPERBAIK UNTUK PENGECAMAN EKSPERISI WAJAH

ABDULLA M.K. ELMADHOUN

DISERTASI YANG DIKEMUKAKAN UNTUK MEMENUHI SEBAHAGIAN DARIPADA SYARAT MEMPEROLEHI IJAZAH SARJANA SAINS KOMPUTER

FAKULTI TEKNOLOGI DAN SAINS MAKLUMAT
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

2017

DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries which have been duly acknowledged.

14 Feb 2017 Abdulla M. K. Elmadhoun

P74470

ACKNOWLEDGEMENT

Completing my master degree is probably the most challenging goal of my first 30 years of my life. The best and worst moments of my journey in obtaining my master degree have been shared with many people who I would like to thank here.

First and foremost, I would like to thank my God, whose many blessings have allowed me to reach this stage of my life. Special thanks to my supervisor Assoc. Prof. Dr. Md. Jan Bin Nordin. He patiently provided the vision, guidance, encouragement and advise nessessary for me to procced through the master program and complete my dissertation.

I would also like to thank my parents, Dr. Mahmoud Elmadhoun -Rahmatullah Alaih- and Ibtisam Elmadhoun, who encouraged and supported me throughout my whole life and showed me the way to be a right man.

Also I would like to thank my wife Bisan Alsalibi for her support, help and she kept me going on when the coffee has lost its effect.

ABSTRACT

Facial expressions are essential communication tools in our daily life as they impart a person's emotional state as well as his intentions. This motivates efforts to empower machines to become affect-aware, such as the ability to automatically recognize human facial expressions. In this thesis, uniform local binary pattern is employed to extract features from the face to form the feature representation. However, this feature representation is very high in dimensionality. The high dimensionality would not only affect the recognition accuracy, but also can impose computational constraints. Hence, to reduce the dimensionality of the feature vector, the firefly algorithm is used to eliminate the noisy features and to select the optimal subset that leads to better classification accuracy. However, the standard firefly algorithm suffers from the risk of being trapped in local optima after a certain number of generations. Hence, this limitation has been addressed by proposing an improved version of the conventional firefly algorithm where the great deluge algorithm has been integrated. The great deluge is a local search algorithm that helps to enhance the exploitation ability of the firefly algorithm and thus preventing being trapped in local optima. To validate the proposed algorithm in terms of convergence behaviour, five benchmark test functions have been used. Thereafter, the improved firefly algorithm has been employed in a facial expression system to select the best discriminative features. Experimental results using the JAFFE database show that the proposed approach yielded good classification accuracy compared to other state-of-the-art methods. The best classification accuracy obtained by the proposed method is 96.7% with 1230 selected features, whereas, Gabor-SRC method achieved 97.6% with 2560 features.

ABSTRAK

Ekspresi wajah adalah alat komunikasi yang penting dalam kehidupan seharian kita kerana ianya boleh menggambarkan keadaan emosi dan niat seseorang. Ini mendorong usaha bagi memberi mesin keupayaan untuk mempunyai kesedaran, seperti keupayaan untuk secara automatik mengenali ekspresi wajah manusia. Dalam tesis ini, pola binari setempat seragam digunakan untuk mengeluarkan sifat dari wajah untuk membentuk perwakilan sifat. Tambahan pula, untuk mengurangkan kematraan vektor sifat, algoritma kunang-kunang digunakan untuk menghapus sifat-bising dan untuk memilih subset optimum yang membawa kepada ketepatan pengelasan yang lebih baik. Walau bagaimanapun, algoritma kunang-kunang standard mengalami risiko terperangkap dalam optima setempat selepas beberapa bilangan generasi. Oleh itu, pembatasan ini telah diselesaikan di dalam tesis ini dengan mencadangkan versi algoritma yang diperbaiki daripada algoritma kunang-kunang standard bersepadu dengan algoritma Great deluge. Algoritma Great deluge adalah algoritma carian setempat yang membantu untuk meningkatkan keupayaan eksploitasi algoritma kunang-kunang dan dengan itu menghalang ianya terperangkap dalam optima setempat. Untuk pengesahan algoritma yang dicadangkan dari segi tingkah laku tumpu, lima fungsi ujian penanda aras telah digunakan. Selepas itu, algoritma kunang-kunang diperbaiki telah digunakan dalam sistem ekspresi wajah untuk memilih sifat diskriminatif terbaik. Keputusan eksperimen menggunakan pangkalan data JAFFE menunjukkan bahawa pendekatan yang dicadang menghasilkan ketepatan klasifikasi yang lebih baik berbanding dengan kaedah terkini yang lain.

CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

LIST OF ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
AAM	Active Appearance Model
ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
AI	Artificial Intelligence
AIS	Artificial Immune System
AmQSO	Assembly of multi Quantum Swarm Optimization
AntRSAR	Ant Colony Optimization Rough Set Reduction
ASM	Active Shape Model
AU	Action Unit
BBO	Biogeography-Based Optimization
BeeRSAR	Artificial Bee Colon Rough Set Reduction
Bees	Multi-Objective Bees Algorithms
CEP	Classification Error Percentage
CK	Cohn-Kanade
CK	Cohn-Kanade
CK+	Extended Cohn-Kanade
CK+	Extended Cohn-Kanade
DE	Differential Evolution
DEED	Dynamic Economic Emission Dispatch
DG	Distributed Generation
DG	Great Deluge
DKLLE	Discriminant Kernel Locally Linear Embedding
EA-SAW	Evolutionary Algorithm-Simple Additive Weighting Algorithm
EDFA	Evolutionary Discrete Firefly Algorithm
EED	Environmental Economic Power Dispatch

ES	Eagle Strateg
ES	Evolutionary Strategy
FA	Firefly Algorithm
FA-GD	Firefly Algorithm-Greate Deluge
FACS	Facial Action Coding System
FER	Facial Expression Recognition
FMSO	Fast Multi-Swarm Optimization
Gabor-SRC	Gabor Wavelets-Sparse Representations Classifier
GenRSAR	Genetic Algorithm Rough Set Reduction
GPU	Graphical Processing Unit
HCI	Human Computer Interaction
HEA	Hybrid Evolutionary Algorithm
HEFA	Hybrid Evolutionary Firefly Algorithm
HMM	Hidden Markov Model
HSV	Hue Saturation Value
IEEE-RTS	Institute of Electrical and Electronics Engineers-Reliability Test System
JAFFE	Japanese Female Facial Expression
JSSP	Job Scheduling Problem
k-NN	k-Nearest Neighbor
LBP	Local Binary Patter
LDA	Linear Discriminant Analysis
LDP	Local Directional Pattern
LFW	Labeled Faces in the Wild
mCPSO	multi Charged Particle Swarm Optimization
MFA	Modified Firefly Algorithm
MMI	MMI Facial Expression Database
MMI	MMI Facial Expression Database
mQSO	multi-Quantum Swarm Optimization

NP-hard	non-deterministic polynomial-time hard
NSGAII	Non-Dominated Sorting Genetic Algorithm-II
OF	Optical Flow
OpenCV	Open Source Computer Vision Library
OR-Library	Operations Research-Library
OS	Operating System
P-LBP	Patch-Based Local Binary Pattern
PBIL	Probability-Based Incremental Learning
PCA	Principal Component Analysis
PDM	Point Distribution Model
POFA	Pictures of Facial Affect
POFA	Pictures of Facial Affect
PSO	Particle Swarm Optimization
PSO-CP	Particle Swarm Optimisation-Composite Particles
PSO-RSAR	Particle Swarm Optimization Rough Set Reduction
QAP	Quadratic Assignment Problem
RAM	Random Access Memory
RBF	Radial Basis Function
rSPSO	regression heuristic Standard Particle Swarm Optimisation
RST	Rough Set Theory
SAW	Simple Additive Weighting Algorithm
SEFNN	Structure Equivalent Fuzzy Neural Network
SGA	Stud Genetic Algorithm
SI	Swarm Intelligence
SPEA	Strength Pareto Evolutionary Algorithm
SPSO	Standard Particle Swarm Optimisation
std	Standard Deviations
SVM	Support Vector Machine

TSP	Traveling Salesman Problem
UC	Unit Commitment
UCAV	Uninhabited Combat Air Vehicle
UCI	University of California Irvine
uLBP	Uniform Local Binary Pattren
VEGA	Vector Evaluated Genetic Algorithm
VLBP	Volume Local Binary Pattern

CHAPTER I

INTRODUCTION

1.1 Facial Expression Recognition

Automated analysis of facial expressions has been gaining momentum in the field of computer vision over the past few years. Interestingly, facial expressions contributes a significant part of the non-verbal communication between human beings. Interpretation of these expressions gives a great deal of information about the thoughts and emotional state of a person. More importantly, the recognition of the user emotions has been given a great attention in a variety of Human Computer Interaction (HCI) studies. A vast majority of state-of-the-art studies have reported interesting findings pertaining to the benefits of detecting emotions. For instance, when the machine detects the students' state (puzzled, frustrating, boring) and sends it as a feedback to the teacher, the teacher can control the flow of the class and consequently improves the learning environment. Similarly, monitoring patients' states in tele-home care program and sending a feedback to the doctors in their offices increases the doctors' awareness. Furthermore, it can help in measuring the customers' satisfaction in business sector which can effectively replaces the traditional methods of measurements as surveys. Finally, adapting to the user's emotions and personality significantly increases the user satisfaction. User satisfaction is the ultimate goal of most of the researches in the HCI field.

Several psychological studies have been conducted to study how people belonging to different races interpret their emotions by the aid of facial expressions (Ekman & Friesen 1978). In their pioneer work, Ekman and Friesen have analyzed the muscles activities of the face wherein each muscle movement has been represented by an action unit (AU). Based on these activities, Facial Action Coding System (FACS) has been announced.

One facial expression corresponds to a combination of several AUs. On the basis of this analysis, they identified six common facial expressions that can be identified across cultures, namely: anger, disgust, fear, happiness, sadness and surprise. The major studies in facial expression analysis consider these six universal facial expressions.

Developing automated systems to recognize human emotions with good accuracy and speed under different imaging variations such as illumination and scale has been gaining considerable attention. The successful application of the texture descriptors motivated the use of Local Binary Pattern (LBP) for face representation. However, the essential problem encountered when using local descriptors is the high dimensionality of the data which might have a determinant effect on the speed and accuracy of the system (Bereta et al. 2013; Azazi et al. 2015; Alsalibi et al. 2017). Hence, the major aim of this thesis is to propose an automatic facial expression recognition system using uniform local binary pattern descriptor and a modified firefly optimization algorithm. Essentially, the improved version of the firefly algorithm will be used to select the optimal set of discriminating features so as to alleviate the cause of dimensionality associated with the use of uniform LBP descriptor.

1.2 Firefly Algorithm Background

In 2010, Yang (Yang 2010c) proposed a swarm intelligence algorithm called Firefly algorithm which is a bio-inspired, meta-heuristic and stochastic algorithm. It is considered as a bio-inspired algorithm because it is motivated by the behavior of fireflies flashing light. Furthermore, it has the concept of heuristics methods in the sense that it searches for solutions by using trial and error mechanism (Yang 2010c), and there is no guarantee to reach the optimal solution. Meta-heuristic means a 'higher level' so it can apply local search and randomization, in another word it can apply exploration and exploitation in the search space (Yang 2010c). The heuristic part in the algorithm focuses on creating new generations in the local search space and select then best solution in the generation, the randomization part will insure that the search will not stuck in local optima by creating a random jump for the fireflies. Finally, it is considered as stochastic because it contains the randomization part in its embedded search mechanism. This is the real reason behind its successful applicability on complex problems such as the NP-hard problems.

Two important components that the meta-heuristics depend on are exploration and exploitation (Crepinsek et al. 2011; Liu et al. 2013a). Basically, each meta-heuristic method has its own parameters to be tuned during the search process which in turn will change the way of exploration and exploitation. It is worthwhile to mention that achieving a balance between these two components is an essential factor of optimization approaches (Tashkova et al. 2012; Liu et al. 2013b,a).

The terms exploration (diversification) and exploitation (intensification) are often associated with the contribution of selection and genetic operators to search effectiveness. Exploration is the process of exploring entirely unknown regions of a search space. Exploitation is the process of intensifying a search within the neighborhood of previously visited regions. Exploitation is the effective use of current available information whilst exploration is the discovery of unknown information. Exploration not only enhances population diversity, but also helps avoid local minimal solutions to find better or near optimal solutions due to its randomization feature. On contrary, exploitation tries to improve solutions quality to ensure that solutions will converge to optimality (Liu et al. 2013a).

Although the firefly algorithm has been successfully applied in many applications, some drawbacks of the traditional FA have been reported in the literature. Firstly, traditional FA suffers from the risk of being trapped in local optima (Baykasoglu & Ozsoydan 2014). Sometimes it is unable to come out of that state. The parameters in the firefly algorithm are fixed and do not have any mechanism to remember the previous best situation of each firefly and this makes them move regardless of its previous better solution. Results from experiments running FA have shown that FA can either be mostly stuck into the local minimum or the results do not improved furthermore (stagnation). It seems that both phenomenons could be connected with the exploration and exploitation components of a process space (Fister et al. 2013b). Crepinsek et al. (2011) asserted that too much exploitation induces the stuck in local optima, while too much exploration slows down the convergence.

1.3 Research Questions

1. How to enhance the search capability of the fireflies in the firefly algorithm to reduce the risk of being trapped in local optima?
2. How to deal with the high dimensionality problem without losing the valuable features generated by uniform LBP?
3. How to apply the improved firefly algorithm for feature selection in the FER system?

1.4 Problem Statement

The use of uniform LBP partially overcomes some of the typical drawbacks in facial expression recognition systems such as illumination and pose (Luo et al. 2013). However, local descriptors such as LBP when concatenated lead to long and high dimensional feature vectors. Thus, feature selection is of significant importance as local approaches often generate a feature space with very high dimensionality, which greatly affect the accuracy of the classification process.

In this regard, we address this problem by integrating a feature selection phase to select the most optimal subset of discriminating features and exclude the noisy and irrelevant facial features. In particular, we employ a bio-inspired feature selection algorithm based on firefly optimization algorithm. However, the traditional firefly algorithm suffers from the risk of being stuck in local optima (Baykasoglu & Ozsoydan 2014; Fister et al. 2013b).

Integrating local search methods into global optimization algorithms such as firefly helps to enhance the exploitation ability of the algorithm and thus prevent the stagnation in local optima (Alsalibi et al. 2017).

1.5 Research Objectives

The thesis will focus on two essential parts; the first part is proposing an improved firefly algorithm by integrating a local search algorithm to improve the search capabilities of the fireflies in the search space. The second part is to build an automatic and person-independent Facial Expression Recognition (FER) based on uniform Local Binary Patterns (LBP) features and to use the first part of the research (improved firefly algorithm) as a feature selection to select the best discriminating features through the extracted features from 2D facial images to improve the accuracy of recognizing emotions from static facial images.

The research objectives of this thesis are as follows:

1. To improve the conventional firefly algorithm by integrating a great deluge local search algorithm to avoid the stuck in local optima.
2. To propose an automatic and person-independent facial expression recognition model with a reliable accuracy based on an improved firefly algorithm.
3. To investigate how the high dimensionality inherent within the use of LBP feature vector could be adequately reduced by exploiting an improved firefly optimization algorithm for the discriminative feature selection process.

1.6 Scope of the Research

This research focuses on building a facial expression recognition model using Uniform Local Binary Patterns as a feature extractor from 2D facial images, and improves this model by integrating a feature selection algorithm to improve the accuracy of emotion recognition.

The feature selection algorithm that will be used in the model is an improved firefly algorithm to optimize the features extracted by Local Binary Patterns. Also the research focuses on improving the firefly algorithm before integrating it to the model by hybridizing the algorithm with a local search algorithm to avoid being stuck in local optima.

The model will be used to recognize some types of emotions like (anger, disgust, fear happiness, sadness, surprise and natural) for static images. Also the model should achieve better accuracy in recognizing these emotions.

1.7 Organization of The Thesis

The remaining of the thesis is organized as follows: the literature review is presented in Chapter 2. The framework of the research methodology is given in Chapter 3. In chapter 4, the experimental results and the discussion are presented. Finally, Chapter 5 concludes the thesis and gives some future directions.

CHAPTER II

LITERATURE REVIEW

2.1 Introduction

When it comes to analyzing the facial expressions of emotions, the first question comes to the mind is "How facial expressions are perceived and processed to reflect a person's emotion?", i.e. what are the basic elements of expressions, how do they form a specific emotion, and how would they be classified?. This question first arose in psychology and different theories have been proposed to answer this question. A few of these theories have inspired the researchers in the field of computer science and established a strong foundation for facial expression analysis. This Chapter starts with a brief overview of facial expression recognition. Next, the state-of-the-art studies in 2D facial expression recognition with different approaches are discussed. Furthermore, a detalied overview of firefly algorithm and its varients is provided as well.

2.2 Facial Expression Recognition

Facial expressions are fundamental for social communication because they hold significant clues about emotions directly. Facial movements have several roles in the interaction and communication between human beings. Visible facial movements are used to enhance and influence the emotion from speech. Moreover, facial actions are activated for a short time when the emotion passes. Thus, detecting the facial expression is an intuitive way to recognize emotion.

The cross cultural study of (Donato et al. 1999) shows that some emotion expressions are universal for human beings regardless of their race and region. These emotions are happiness, sadness, anger, fear, disgust, and surprise. Each of the six basic emotions corresponds to a unique facial expression, and other emotional expressions may be culturally variable. Table 2.1 gives the description of facial expressions of the basic emotions.

For emotion recognition systems, facial expression analysis is considered to be a major indicator of a human affective state. Automatic emotion recognition from facial expression is inherently a multi-disciplinary enterprise involving different research fields (Ekman & Friesen 2002) including psychology, computer vision, feature data fusion, and machine learning. There are two main streams in current facial emotion recognition: the facial affect recognition and the facial muscle action detection. They came from two dominant facial expression analysis methods in psychological research: the message judgment method and the sign judgment method. The message judgment method is used to understand what underlies a displayed facial expression; while the sign judgment method is used to describe the shown behavior outside.

The facial action coding system (FACS) proposed by Ekman and Friesen is a commonly used vision-based tool to code human facial expression movements by their visual aspect on the face which belongs to the sign judgment method (Ekman & Friesen 1971). The FACS enables facial expression analysis through standardized coding of changes in facial motion in terms of atomic facial actions named Action Units (AUs). The FACS can decompose the facial muscular actions and key out the facial expression in AUs. Then, the AUs can be applied for any high-level decision making process including basic emotion recognition, various affective state recognition, and other complex psychological states. This system codes facial expressions manually by following a set of specific rules. So far, Ekman's basic facial emotion model and the FACS are the mainly used methods in vision-based facial emotion recognition. However, the inputs for the FACS are static images of facial expressions which result in a very time consuming process.

Table 2.1 Facial Expression Description of six Basic Emotions (Sumpeno et al. 2011).

No.	Emotion Name	Description of Facial expressions
1	Happiness	The eyebrows are relaxed.The mouth is open and the mouth corners upturned.
2	Sad	The inner eyebrows are bent upward. The eyes are slightly closed. The mouth is usually relaxed.
3	Fear	The eyebrows are raised ad pulled together. The inner eyebrows are bent upward. The eyes are open and tense.
4	Anger	The inner eyebrows are pulled downward and together. The eyes are wide open. The lips are tightly closed or opened to expose the teeth.
5	Surprise	The eyebrows are raised. The upper eyelids and the eyes are wide open. The mouth is opened.
6	Disgust	The eyebrows and eyelids are relaxed. The upper lip is raised and curled, often asymmetrically.

Ekman's (1997) pioneer work inspired many other researchers to analyze facial expressions via images and videos. The facial expressions were categorized by the facial feature tracking and facial movements measurement. Some survey works (Fasel & Luettin 2003; Pantic & Rothkrantz 2000, 2003) give a deep review of much research done in the field of automatic facial expression recognition. In general, there are four steps for recognizing emotion from facial expression: face detection, face tracking, feature extraction, and classification. Few facial emotion recognition systems can deal with both static images and image sequences. For example, the survey published by Pantic and Rothkrantz emphasized how to automatically analyze the facial expression (Pantic & Rothkrantz 2000). The face was detected using watershed segmentation with markers extracted on the Hue Saturation Value (HSV) color model algorithm. A point base face model is used, and the features are defined as some geometric relationships between the facial points or the image intensity in a small region defined relative to the facial points. Multiple feature detectors are used for each facial feature localization and model feature extraction. The reported recognition rate is 86% by using the rule-based classification method. In (Lyons et al. 2000), a set of multi-scale, multi-orientation Gabor filters were used to transform an image. Then an elastic graph matching method is used to obtain a registered grid. The grid was sampled and combined into one vector as features. When testing with a database of 193 images, 75% accuracy was achieved. By detecting the eye and lip position using low-pass filtering and edge detection method in (De Silva & Hui 2003), an average recognition rate of 60% was achieved. A bimodal system of facial emotion recognition has been proposed by (Wang & Guan 2008). An HSV color model was used to detect the face of the environment, and Gabor wavelet features were used to represent the facial expressions. The overall recognition rate is 82.84% by using multi-classifiers.

Though there are many different methods in facial emotion recognition, the common steps are face detecting, face tracking, feature extracting, and classification. The output is the emotion recognition result in the preselected basic emotions. Basically, the facial emotion recognition design in this dissertation also follows these common steps.A facial expression is a visible activation of a human's affective state, cognitive activity, temperament and personality, and psychopathology (Donato et al. 1999). Facial expressions together with other gestures can reveal non-verbal communication hints in the face-to-face interactions, which can help the listener to understand more about the intended meaning of the speaker's speech. FACS proposed by Ekman and Freisen (Ekman & Friesen 1978) can describe the distinguishable visual facial movements. Using the FACS, the parameters of action are designed to classify human emotions. From Mehrabian's research (Mehrabian et al. 1971), facial expression provides 55% of the effect of a message. The vocal part contributes 38% while the verbal part only contributes 7%. Facial expression applications are widely used, including image understanding, psychological studies, medicine, and image compression (Ostermann 1998).

Since facial expressions include a consequence of information about a person, they play an important role in the human computer interaction. Automatic emotion recognition from facial expression may act as a component of the natural human machine interface (van Dam 2000). This kind of interface can provide services requiring a good understanding of the emotional state of the user. For example, using this interface in some robots to recognize human expression can contribute more intelligent robots (Bruce 1993). Automatic facial expression analysis for behavioral science or medicine is another kind of application (Donato et al. 1999; Essa & Pentland 1997).

Many works have been done with automatic emotion recognition from images or videos of facial expression since 1990. In Mase (Kenji 1991), optical flow (OF) was used by the author to estimate facial muscle movements in order to recognize facial expressions. It was the first time to use image processing techniques in facial expression recognition. A flexible shape and appearance model for locating facial features, coding and reconstruction, recovering pose, recognizing gender and facial expression, and identifying individuals with an image was used in Lanitis, Taylor, and Cootes (Lanitis et al. 1995b).

Black and Yacoob (Black & Yacoob 1995) used a collection of local parameterized OF models to track and recover rigid and non-rigid facial motions. The image motion parameters were then used in a rule-based classifier to recognize the six universal facial expressions. Eye blinking and other simple head motions can also be recognized by their system. In Yacoob and Davis (Rosenblum et al. 1996) the authors computed the OF and used similar rules to Black and Yacoob (Black & Yacoob 1995) to classify the six basic emotions. OF regions on the face were computed in Rosenblum, Yacoob, and Davis (Yacoob & Davis 1996), and a radial basis function network architecture was used on the human emotion detection system. OF flow processing was also used in Essa and Pentland (Essa & Pentland 1997) as the basis to measure and classify facial emotions. In Otsuka and Ohya (Otsuka & Ohya 1997), the OF algorithm was first used to calculate a velocity vector. Then, a two dimensional Fourier transform is used for the vector at the eyes and mouth regions. Finally, Hidden Markov Models (HMMs) are used to recognize facial expression from image sequence for multiple persons. In their experiments, the proposed system can recognize basic emotions in nearly real time.

Although there are many achievements for emotion recognition from facial expression, there are still many difficulties and limitations in emotion recognition due to the complexity of emotion expression, especially when in a conversation. The advantages of recognizing emotion from facial expression are a) it is the most intuitive and natural way to observe human beings' emotional states; b) there are plenty databases available for research. One disadvantage is that it is misleading sometimes since there is no context information. Another one is that the recognition results depend heavily on the quality of the image or video.

Recently, many efforts have been done on implementing emotion recognition systems using both facial expressions and acoustic information. De Silva and Pei (De Silva & Ng 2000) proposed a bimodal emotion recognition system by combining audio and video information using a rule-based system. The authors described the use of statistical techniques and hidden Markov models (HMM) in the emotion recognition. The prosodic features in audio and maximum distances and velocities of video between six specific facial points were used. The performance of emotion recognition increased when using both ways together.

2.2.1 Factors of Facial Emotion Recognition

Vision based emotion recognition is mainly focused on facial expressions because of the significance of face in emotional expression and perception. Many approaches have been done in this area. However, there are still many challenges in the facial emotion recognition area. As we know faces are non-rigid and have different color and pose. Some of the facial features are not common and not suitable for pattern recognition. Lighting of the background and the illumination conditions can also change the overall recognition rate of facial expression. The above problems make the emotion recognition from facial expression more complicated.

Currently, many facial expressions recognition approaches are based on a two dimensional (2D) sparse data, such as 2D static images or 2D video sequences. However, there are mainly four steps for facial emotion recognition: face detection, face tracking, feature extraction and classification as shown in Figure 2.1. The input data for the recognition system can be video streams from a Web Camera or some

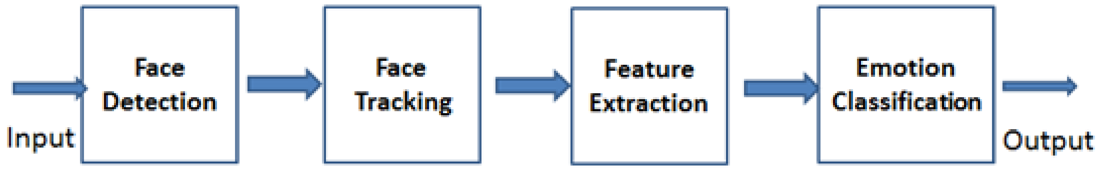

Figure 2.1 Facial emotion recognition.

recorded video clips, as well as some static facial images. After the 4 steps, the final decision of the facial emotion will be given as the output.

a. Face Detection

Though there are several types of input to the facial emotion recognition system, images containing faces are still essential to intelligent vision-based human computer interaction. Face detection tries to identify all image regions which contain a face regardless of its 3D position, orientation, and lighting conditions from a given image. The challenge here is that faces are non-rigid and have a high degree of variability in size, shape, color, and texture. A variety of face detection techniques have been developed. In the Yang, Kriegman, and Ahuja (Yang et al. 2002) survey, the authors classified the detection methods into four categories: Knowledge-based methods, Feature invariant approaches, Template matching methods, and Appearance-based methods.

Knowledge-based face detection methods are also called rule-based methods. These kinds of methods encode human knowledge of a typical face and are used mainly for face localization. Facial features will be extracted first from the input image, and then faces are identified based on the coded rules. The problem with this kind of method is that it is difficult to translate human knowledge in well-defined rules. Furthermore, it is difficult to extend this approach to detect faces in different poses. In Yang and Huang (Yang & Huang 1994), a hierarchical knowledge-based method was used by the authors to detect faces.

Compared to knowledge-based methods, the feature-based methods use the invariant face features to detect a face. These features are eyebrows, eyes, nose, mouth, and hairline which are extracted by using edge detectors. Then, based on the extracted features a statistical model is built to verify the existence of a face. Feature-based methods are widely used by many researchers (Graf et al. 1995; Sirohey 1998). The weakness for this kind of methods is that image features depends severely on the illumination, noise, and occlusion.

In template matching methods, a standard face pattern will be predefined with a function by experts. The correlation values for the face contour, eyes, nose, and mouth of a given image will be calculated with the standard patterns. Whether a face exists is dependent on these correlation values. This model is very easy to implement, but it is inefficient to deal with scale, pose, and shape for face detection. There are two types of templates in this kind of methods: the predefined templates (Sakai et al. 1969; Samal & Iyengar 1995; Tsukamoto et al. 1994)and deformable templates (da Vitoria Lobo & Kwon 1998; Lanitis et al. 1995a).

Finally, the models of appearance-based methods have been learned from a set of training images. They rely on techniques from statistical analysis and machine learning to find the relevant characteristics of face and non-face image. Many distribution models or discriminant functions are used to model the learned characteristics for face detection, including Eigenface (Turk & Pentland 1991), Gaussian distribution (Lew 1996), Neural Network (Rowley et al. 1998), SVM (Osuna et al. 1997), Naïve Bayes Classifier (Schneiderman & Kanade 1998), HMM (Rajagopalan et al. 1998), and Information-Theoretical Approach (Colmenarez & Huang 1997; Lew 1996). According to the survey by Zhang and Zhang (Zhang & Zhang 2010), the appearance-based methods are superior to the other method despite their computational load.

There are lots of challenges for face detection such as non-rigid structure, different illumination environment, size, shape, color, orientation as well as texture. Yang, Kriegman, and Ahuja (Yang et al. 2002) pointed out that gestures, presence or absence components, facial expression, partial or full occlusions, face orientation, and lighting conditions are the basic challenges in face detection.

Viola and Jones (Viola & Jones 2001) have proposed a frontal face detection system in grayscale images using the Harr Feature-based Cascade Classifiers. It can be used in a real-time face detection system. Now it is available for all the researchers in the Open Source Computer Vision Library (OpenCV) tool (ope 2014). Their work has been extended to handle multi-pose faces using skin-color cues to reduce the computation time and decrease the false detection rate, since the skin color is a useful cue for face detection under different illumination environments. Currently, the Viola and Jones face detecting method is still widely used by researchers.

b. Face tracking

Face tracking is used in the video frames to pass over the detected face forward or backward of each frame. Usually, two methods can be used in face tracking. The first one uses the face detector as a face tracker running at all the frames. For instance, the Viola and Jones (Viola & Jones 2001) face detector can be used as a face tracker running at each frame. The second approach is to develop a face tracker apart from the face detector. Active Shape Models (ASM) (Cootes et al. 1995) and Active Appearance Models (AAM) (Cootes et al. 2001) based tracker are widely used.

A face shape should be reconstructed to fit the target face image in the face tracking. For the ASM based method, manually labeled training images are used (Cootes et al. 1995). This approach will first search the salient points to fit the model in the image and then update these points at each frame. It is also called Smart Snake. The AAM also uses a training phase, but it uses both shape and appearance information of the target image (Cootes et al. 2001). ASM is faster than the AAM because AAM uses all information of the image. The original algorithms were used for the grayscale images, but both of them can be extended to the color image.

ASM generation and fitting. Point distribution models (PDM) are important in modeling of shapes. The statistical information of the training images is used to extract the mean and variance of the shape. Landmarks on the boundary are used when describing an object. A transformation is used to align all the images in the training process including translation, scaling, and rotation. Each image in the training database has been co-aligned with this transformation. When the algorithm converges, the relationship between the original shape α and the mean shape $\bar{\alpha}$ can be described as:

$$\alpha = \overline{\alpha} + T \times w \tag{2.1}$$

where T is the matrix containing the eigenvectors of the shape in its columns, and w is the variation weighting matrix for each of the eigenvectors. The generated ASM model must fit the targeting object. After initialization, landmarks are moved along the search path, and the model boundary is fit to the target object boundary (Sonka et al. 2014).

AAM generation and fitting. In AAM generation, not only a shape model, but also an appearance model was generated. As mentioned previously, the ASM model is generated only based on shape; while the AAM model is generated from both shape and appearance.

c. Feature Extraction

The extraction of facial features plays an important role in emotion recognition from facial expression. Among the facial features, eyes, nose, and lip are the most important features. Various approaches have been done to extract facial points (such as eyes and nose) from images and video sequence of faces. Also, feature extraction closely relates to the face detection, since in the face detection part some of the features have already been extracted.

Generally, there are four kinds of methods for facial feature extraction: geometry-based feature extraction, template-based feature extraction, color segmentation-based feature extraction, and appearance-based feature extraction. Geometry-based approaches extract features using geometric information such as relative positions and sizes of the face components like mouth, eyes, nose, and eyebrow, which can cover the variation in the appearance of the facial expression. The template-based feature extraction matches facial components to previously designed template using the appropriate energy functional. The color segmentation method uses skin color to segment the face, and the non-skin color region will be viewed as a selection for eyes and mouth. The appearance-based approach provides not only the simple components of the face, but also information about the texture of the face, such as wrinkles. To extract the feature vector, many methods can be used such as Gabor-wavelets, principal component analysis (PCA) and Local Binary Pattern (LBP).

Local Binary Patterns (LBP): The Local Binary Pattern (LBP) operator, originally introduced by Ojala et al. (Ojala et al. 1996), is a powerful way of describing the texture of an image. The simple LBP operator labels the pixels of an image by thresholding a 3×3 neighborhood of each pixel with the center value and considering the results as an 8-bit binary number or an LBP label for that pixel (as illustrated in Figure 2.2). These computed binary numbers can be used to represent different local primitives such as corners, curved edges, spots, flat areas, etc. (See Figure 2.3). The 256-bin histogram of the LBP labels computed over a region (or an image) is then used as a texture descriptor for that region (or an image).

The small 3×3 neighborhood limits the capturing of dominant features with large scale structures.Therefore, the basic LBP operator was extended to use neighborhoods of various sizes (Ojala et al. 2002).

The operator can be applied on circular neighborhoods around a pixel by bilinearly interpolating the pixel values allowing any radius value, R, and any number of equally spaced pixel samples in the neighborhood, (P) . This LBP operator can be

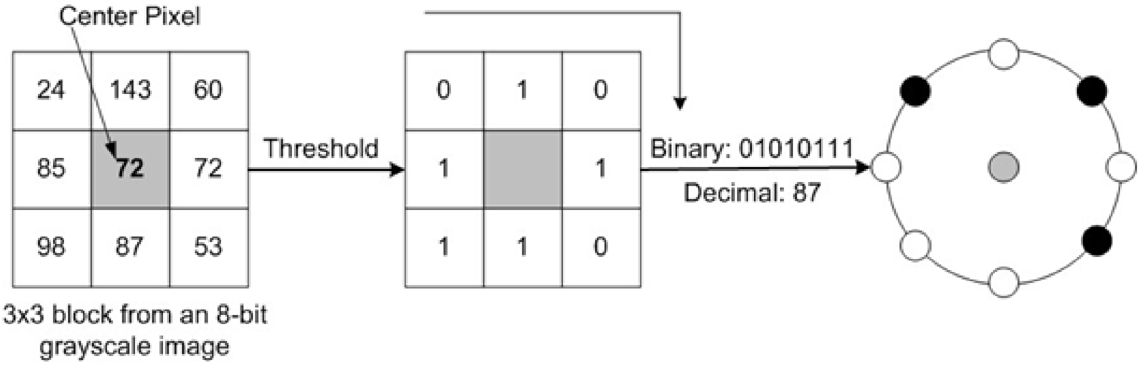

Figure 2.2 The basic LBP operator Ahonen et al. (2004).

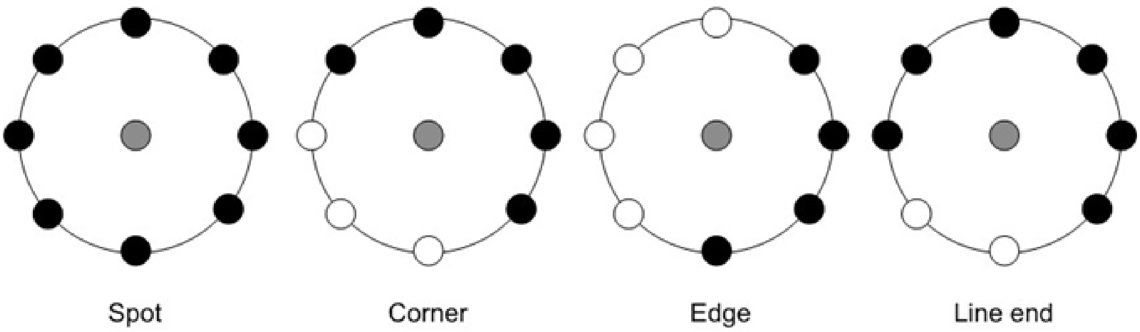

Figure 2.3 Example of texture primitives encoded by LBP (black circles represent zeros and white circles represent ones) (Hadid et al. 2004).

formalized as $LBP_{P,R}$ as shown in Figure 2.4 with different values of P,R. The value of the LBP code of a pixel z_c can then be formalized by:

$$LBP_{P,R}(z_c) = \sum_{p=0}^{P-1} \delta(g_p - g_c)2^p \tag{2.2}$$

where g_c corresponds to the gray value of the center pixel z_c, g_p refers to the gray values of P equally spaced pixels on a circle of radius R, and $\delta(x)$ defines a thresholding function as follows:

$$\delta(x) = \begin{cases} 1 & if\ x \geq 0 \\ 0 & otherwise \end{cases} \tag{2.3}$$

For P neighboring samples, the $LBP_{P,R}$ operator produces 2^P labels

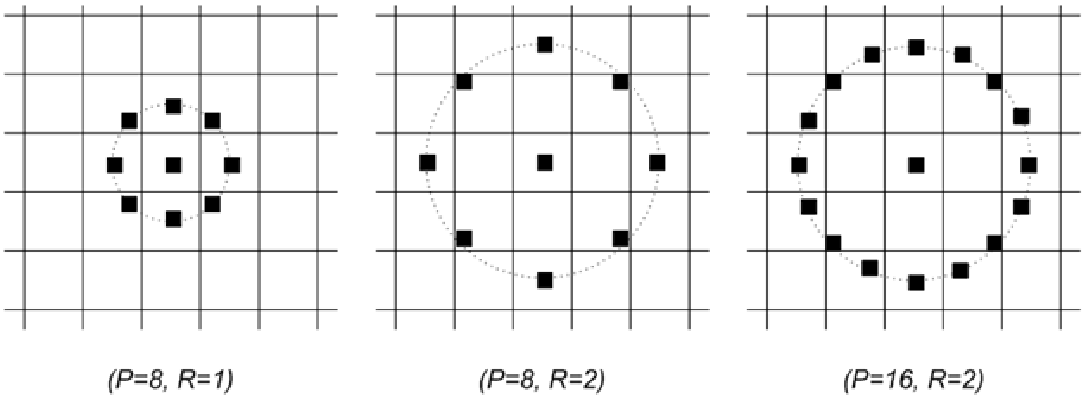

Figure 2.4 LBP circular neighborhood examples. From Left to Right: (8,1), (8,2), and (16,2) circular neighborhoods respectively (Ojala et al. 2002).

corresponding to the 2^P different output binary words. It has been shown that there is a subset of 2^P labels exists that encodes most of the texture information than the others (Ojala et al. 2002). This subset of fundamental patterns has been named as uniform patterns or $LBP_{P,R}^{u2}$. An LBP is called uniform if it contains at most two bitwise transitions from $0 \rightarrow 1$ or vice versa (e.g., 00000000, 00010000, 11100001). The rest of the patterns which have more than two bitwise transitions are accumulated into a single bin.

Experiments conducted in (Ojala et al. 2002) conclude that uniform patterns account for nearly 90% for all patterns in (8,1) neighborhood and about 70% in (16,2) neighborhood. The number of uniform patterns, $U(P)$, for P-pixel neighborhood can be calculated as follows:

$$U(P) = P(P-1) + 2 \tag{2.4}$$

Using Eq. (2.4), for an 8-pixel neighborhood we get 58 uniform patterns, yielding 59 total bins, and for a 16-pixel neighborhood we get 242 uniform patterns, thereby yielding 243 total bins. Once LBP labels have been calculated for each pixel, a histogram h_l of the labeled image $L(z_c)$ can be defined as:

$$h_l = \sum_{z_c} B(L(z_c) = l),\ l = 0, \ldots, ((U(P)+1)-1) \tag{2.5}$$

where $U(P)$ is the number of different labels produced by the LBP operator given by Eq. (2.4) and

$$B(x) = \begin{cases} 1 & if\ x\ is\ true \\ 0 & if\ x\ is\ false \end{cases} \tag{2.6}$$

Previously, it has been shown that facial images can be seen as a composition of micro-patterns which can be effectively detected by the LBP operator (Huang et al. 2011), especially applied towards face detection (Hadid et al. 2004; Itier & Batty 2009), face recognition (Ahonen et al. 2004; Hadid et al. 2004), and facial expression recognition (Shan et al. 2005, 2009). A LBP histogram computed over the entire eye region image encodes only the occurrences of the micro-patterns without any indication about their locations. To incorporate location information, local LBP histograms can be extracted by dividing the eye region image into M small regions, $R^0, R^1, \ldots, R^{(M-1)}$, and the LBP operator is applied to each individual region. Using Eq. (2.5), the regional LBP descriptor for a local region Rm can be written as in Eq. (2.7):

$$h^m = \left[h_{(m,0)}, h_{(m,1)}, \ldots, h_{(m,U(P))}\right]^T \tag{2.7}$$

where, $m = 0, 1, ..., M-1$, and T is the transpose operator. The individual histograms, h^m, are then concatenated together into a single column vector to generate a spatially enhanced histogram (using Eq.(2.7)) representing the signature of that image. The extracted feature histogram not only represents the texture at a local level, but also encodes the global shape/description of the eye region images.

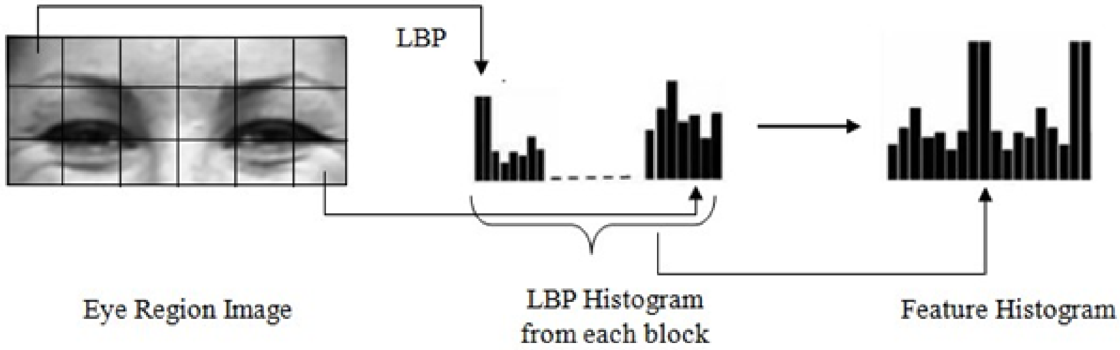

Figure 2.5 The eye region is divided into several small regions. The LBP histograms are extracted and concatenated from each region into a single feature histogram representing the signature of that image (Singh et al. 2015).

$$H = \left[(h^0)^T (h^1)^T \dots (h^{M-1})^T)\right]^T \tag{2.8}$$

LBP's tolerance to monotonic illumination variations and its computational simplicity have made it a popular technique for facial feature analysis in recent years (Huang et al. 2011), especially in facial expression recognition (Huang et al. 2011; Shan et al. 2009). Feng et al. (Feng et al. 2004) described a coarse-to-fine facial expression classification scheme using LBP. More precisely, at the coarse stage, a seven-class problem was first reduced to a two-class one, and later at the fine stage, a k-nearest neighbor (k-NN) classifier performed the final decision. Their approach produced 77% average recognition accuracy on JAFFE dataset. Later on (Feng et al. 2007), with the same facial description, a linear programming technique was applied for expression classification. A seven-class problem was decomposed into 21 binary classifications by using the one-against-one scheme. With this method, they obtained 93.8% accuracy on the JAFFE database.

Shan et al. (Shan et al. 2005, 2009) empirically evaluated LBP features with weighted chi-square template matching, SVMs, LDA, and Linear Programming classification techniques in regular and low-resolution images, concluding that LBP feature-based face representation outperformed Gabor features for FER (92.6% versus 89.8% on the Cohn-Kanade dataset using SVMs). He et al. (He et al. 2006) used LBP on four kinds of frequency images decomposed by Gabor wavelets for facial expression recognition, reporting increased performance than using LBP directly on the JAFFE dataset. Liao et al. (Liao et al. 2006) extracted LBP features in both intensity and gradient maps in order to consider multiple cues. They computed the Tsallis entropy of the Gabor filter responses as the first feature set and performed null-space LDA for the second feature set achieving 94.59% accuracy for images of 64×64 pixels, and 84.62% for 16×16 pixels on the JAFFE database using SVMs as the classification tool.

Some variants of LBP have also been used to classify facial expressions from static images and image sequences. In (Zhao & Pietikainen 2007), volume local binary patterns (VLBP) and local binary patterns on three-orthogonal planes (LBP-TOP) were employed on video sequences as a means to recognize facial expressions using dynamic texture. The evaluation was conducted over a range of image resolutions and frame rates and a recognition rate of 96.26% was achieved on the Cohn-Kanade database demonstrating that both approaches outperform other state-of-the-art methods. AdaBoost technique was used to learn the principal appearance and motion from the spatiotemporal descriptors. Recently, Jabid et al. (Jabid et al. 2010) proposed an alternative version of the LBP, called the local directional patterns (LDP) to extract facial features for facial expression recognition and reported better performance (96.4% on the Cohn-Kanade dataset using SVMs) than the LBP counterpart (Shan et al. 2005). LDP essentially computes the edge response values of a pixel in different directions to encode the image texture.

As described above, LBP and LBP variants have proved to be one of the most widely used feature representation techniques in recent years in the FER research community. These techniques have gained popularity due to their ability to effectively describe local characteristics of the facial image, their robustness to illumination changes, and their computational simplicity.

d. Feature Selection

Feature selection using bio-inspired algorithms is widely used for face and expression recognition. A comprehensive review of typical bio-inspired feature selection approaches that have been used in face recognition can be found in (Alsalibi et al. 2015).

Mistry et al. (2017) proposed a facial expression recognition system using modified Local Gabor Binary Patterns (LGBP) for feature extraction and a firefly algorithm for feature selection. In their work, the FA algorithm is used to reduce the dimensionality of the extracted facial features. The FA algorithm employs Levy distributions to further mutate the best solution identified by the FA to increase exploration in the search space to avoid premature convergence. The overall system was evaluated using three facial expression databases (i.e. CK+, MMI, and JAFFE). The authors claimed that their proposed system outperformed other heuristic search algorithms such as Genetic Algorithm and Particle Swarm Optimization. Along simlar lines, Zhang et al. (2016) proposed a facial expression recognition system using a variant of firefly algorithm for feature optimization. First of all, a modified Local Binary Pattern descriptor was proposed to produce an initial discriminative face representation. A variant of the firefly algorithm was proposed to perform feature selection. Simulated Annealing embedded with Levy flights was also used to increase exploitation of the most promising solution. Diverse single and ensemble classifiers were implemented for the recognition of seven expressions. Evaluated with frontal-view images extracted from CK+, JAFFE, and MMI, and 45-degree multi-view and 90-degree side-view images from BU-3DFE and MMI, respectively, their system achieves a good performance, and outperforms other state-of-the-art feature selection methods and related facial expression recognition models.

Apart from FA, several optimization algorithms have been used for feature selection in facial expression systems. For instance, Yogesh et al. (2017) proposed a new particle swarm optimization assisted Biogeography-based algorithm for feature selection. The obtained results convincingly proved the effectiveness of their proposed feature selection algorithm when compared to other metaheuristic algorithms (BBO and PSO). In another work, Mlakar et al. (2017) proposed an efficient feature selection system applied to a Facial Expression Recognition system. their system is capable of recognizing seven prototypical emotions including neutral expression based on a histogram of oriented gradient descriptor (HOG) and difference feature vectors. The emotion feature selection was carried out by using a modified multi-objective differential evolution algorithm. The number of used features was minimized, while the emotion recognition accuracy of the support vector machine classifiers was maximized simultaneously. This person-independent FER system with proposed feature selection was validated on three commonly used evaluation databases, where the mean emotion recognition rate was 98.37% on the Cohn Kanade database, 92.75% on the JAFFE database, and 84.07% on the MMI database, while the number of used features lowered up to 89% with respect to the original difference feature vector length.

e. Emotion Classification

There are many studies on classifiers of facial emotion recognition. For the static images, the facial expression will be classified based on the tracking result from each image. Bayesian network classifiers and Naïve Bayes classifiers are used generally to classify the facial emotions. While for the dynamic approach, the temporal pattern will be taken into account to the classifiers. k-Nearest Neighbor (KNN) classifier was proven to be suitable for facial expression recognition and have been widely used in this area (Wen & Zhan 2008; Sohail & Bhattacharya 2007). One of the advantages of using KNN is that the cost of learning process is zero and it is much faster than SVM classifier (Sohail & Bhattacharya 2007).

k-Nearest Neighbor Algorithm : k-Nearest Neighbor Algorithm (k-NN) (Mitchell 1997) uses two data sets: (1) training data set (to learn the patterns) and (2) test data set (to verify the validity of learned patterns. The training data contains emotion information. On the other hand, the test data set is similar to the training data set, except that it does not have the emotion information. In order to classify an

Table 2.2 Widely used facial expression databases in research. CK: Cohn-Kanade (Kanade et al. 2000), CK+: Extended Cohn-Kanade (Lucey et al. 2010), MMI (Pantic et al. 2005), JAFFE: Japanese Female Facial Expression (Lyons et al. 1999), POFA: Pictures of Facial Affect (Ekman & Friesen 1975).

Property	CK	CK+	MMI	JAFFE	POFA
No. of subjects	100	123	43	10	14
No. of images	-	-	>250	213	110
No. of videos	486	593	1280	-	-
Gray/Color	Gray	Gray	Color	Gray	Gray
Resolution	640x490	640x480	720x576	256x256	-
Pose	Frontal	Frontal	Frontal/Dual	Frontal	Frontal
FACS-coded	Yes	Some	Yes	No	-
Emotion-labelled	No	327	Some	Yes	-

instance of a test data into an emotion, k-NN calculates the distance between the test data and each instance of training data set: Let an arbitrary instance x be described by the feature$vector < a_1(x), a_2(x), , a_n(x) >$, where $a_r(x)$ is the r^{th} feature of instance x. The distance between instances x_i and x_j is defined as,

$$d(x_i, x_j) = \sqrt{\sum_n (a_r(x_i) - a_r(x_j))^2} \tag{2.9}$$

The algorithm then finds the k closest training instances to the test instance. The emotion with the highest frequency among k emotions associated with these k training instances is the emotion mapped to the test data.

2.2.2 Facial Expression Databases

Researchers in the field of facial expression recognition usually perform their evaluation and report results using a number of popular facial expression databases as listed in Table 2.2. These databases differ in terms of their stimuli and recording setup, actors, and the general pose of the actors. It is important to note that most of these databases contain posed expressions, in which subjects are asked to act out certain emotions.

2.3 Swarm Intelligence Algorithms

Swarm Intelligence (SI) is part of Artificial intelligence (AI) discipline which became more popular over the last ten years (Blum & Li 2008). Methods from Swarm Intelligence have been inspired from the behavior of swarms like ants, worms, bees, fireflies, birds, schools of fish and termites. Some individuals in these swarms have a unique behavior. The purpose of this behavior is to direct the swarms to their goals. Each individual in a swarm needs to interact with other individuals by simple interaction rules. For example, termites and worms are able to build nests, bees and ants use these behaviors to search for food as well. Ants interact with each other by using a chemical pheromone to lead for the shortest path between the nest and source of food. In bee colony, there is a bee called scouts. This bee works as informer for the other bees. One of the tasks of this bee is to search for the most promising areas of food then it will communicate with other bees by using the ‘waggle dance’, and this dance will inform the other bees the directions to the food source. furthermore, in bee colony after finding the new food source they continue collecting new information (exploration) and using the old information (exploitation) and communicate these information with each other (Beekman et al. 2008).

The swarms interact with the environment and get more local information to obtain food with a decentralized way, where each individual takes a decision about how to behave depending on these collected information.

(Beni & Wang 1993) probably was the first of using the term “swarm intelligence” as a research field that focuses on the collective behavior within self-organization and decentralized systems. This term has been used in cellular robotic systems which contains simple agents that manage themselves by interacting with other neighbors. Swarm intelligence methods have been used for optimization in many domains like telecommunications, load balancing, image processing. Some examples of swarm intelligence optimization methods are: ant colony optimization (ACO) (Dorigo & Di Caro 1999; Korošec et al. 2012), particle swarm optimization (PSO) (Kennedy & Eberhart 1999) and artificial bee colony (ABC) (Karaboga &

Algorithm II.1 Firefly algorithm

$t = 0; s^* = \phi; \gamma = 1.0;$ //initialize: generation number, best solution, attractiveness
$P^{(0)} = InitializeFA();$ //initialize a population
while (t<MAX_FES) do //MAX_FES: maximumn number of function evaluations
  $\alpha^{(t)}$ =AlphaNew(); //determine a new value of α
  EvaluateFA($P^{(t)}, f(s)$); //evaluate s according to $f(s)$
  OrderFA($P^{(t)}, f(s)$); //sort s according to $f(s)$
  s^* =FindTheBestFA($P^{(t)}$,$f(s)$); //determine the best solution
  $P^{(t+1)}$ =MoveFA($P^{(t)}$); //vary the attractiveness accordingly
  $t = t + 1;$
end while

Basturk 2007; Fister et al. 2012). In the last years, some promising optimization methods appear like Firefly algorithm (FA) (Yang 2008; Gandomi et al. 2013b; Yang 2013b; Sahab et al. 2013), cuckoo-search (Yang & Deb 2009) and bat algorithm (Yang 2010d). Also, a new generation of swarm intelligence optimization methods shown up in recent years like krill herd bio-inspired optimization algorithm (Gandomi & Alavi 2012), Grasshopper Optimisation Algorithm (GOA) (Saremi et al. 2017).

2.3.1 Firefly Algorithm (FA)

Firefly Algorithm is a metaheuristic algorithm inspired by the social behaviour of a group of fireflies. It was introduced by Yang (Yang 2008). During the optimisation process, the algorithm attempts to move the particles or fireflies as inspired by the interaction of real fireflies. Typically, fireflies use the flash light as a communication technique between each other. Flashing lights are used to attract other firefly for mating or warning other firefly from an enemy. However, the flashing lights depend on physics factors. One of these factors is the light intensity I which decreases when the distance r increases, as illustrated in the term $I \propto 1/r^2$. This rule inspired Yang to develop the firefly algorithm (Yang 2008). The brightness of each firefly is determined by the landscape of the objective function. Therefore, the variation of light intensity produced by each firefly in the search region is associated with the encoded objective function. As a result, Yang proposed the base of the FA algorithm as illustrated in Algorithm (II.1).

The FA can be characterized as follow:

- Fireflies are unisex, which means that one firefly can be attracted to other fireflies from same or different sex.
- There is a proportional relationship between the attractiveness and the brightness, and they both decrease as their distance increases. Thus for any two flashing fireflies, the less bright one will be attracted to the brighter one. In the case that there is no brighter one than a particular firefly, it will fly in a random manner.
- The brightness of a particular firefly is decided by the landscape of the fitness function.

The mapping of firefly algorithm to the optimization context can be represented as follows. Randomly generated feasible solutions are called fireflies which will be assigned with a light intensity based on their performance in the objective function. This intensity will be used to compute the brightness of the firefly, which is directly proportional to its light intensity. For minimization problem a solution x with smallest functional value will be assigned with highest light intensity. Once the intensity or brightness of the solutions are assigned each firefly will follow fireflies with better light intensity. For the brightest firefly since there is no other brighter firefly to follow it will perform a local search by randomly moving in its neighborhood.

FA algorithm starts with randomly initializing the first generation of fireflies. Then, the evolution process starts until it reaches the predefined maximum number of generations or the maximum number of function evaluations. In the begining of the FA process, the algorithm calculates ‘AlphaNew’ as illustrated in Algorithm II.1 to change the initial value of parameter α, this part is optional for the algorithm. Then it will start to evaluate the fireflies by applying them to the objective function. Then, the algorithm starts sorting the fireflies based on their fitness values. After that, the algorithm selects the best firefly in the current generation, then it will start to perform a move of the firefly positions in the search space towards the more attractive fireflies.

To design the FA there are two issues that need to be handled. The first is the light intensity and the second one is the attractiveness. The solution s is represented by the firefly light intensity I, the proportional relationship between the light intensity and the fitness function for the solution s is $I(s)\alpha f(s)$, while light intensity $I(r)$ changes according to Eq. (2.10):

$$I(r) = I_0 e^{-\gamma r^2} \tag{2.10}$$

Where I_0 is the light intensity of the source, and γ is a fixed light absorption coefficient. The attractiveness β is proportional to the light intensity $I(r)$. Thus, it can be defined by Eq.(2.10):

$$\beta = \beta_0 e^{-\gamma r^2} \tag{2.11}$$

Where β_0 is the attractiveness when $r = 0$. The distance between two fireflies s_i and s_j can be measured by Euclidean distance as shown in Eq.(2.12).

$$r_{ij} = \|s_i - s_j\| = \sqrt{\sum_{k=1}^{n} (s_{ik} - s_{jk})^2} \tag{2.12}$$

Where n is number of dimensions of the problem. When firefly s_i is attracted to another firefly s_j, the movement from firefly s_i to firefly s_j will be as follow:

$$s_i = s_i + \beta_0 e^{-\gamma r_{ij}^2}(s_i - s_j) + \alpha \varepsilon_i \tag{2.13}$$

Where ε_i is a random number. The movement equation contains three parts, the first part is the current position of the firefly i, the second part is the attractiveness to another firefly, and the third part is the randomization which have the parameter α and

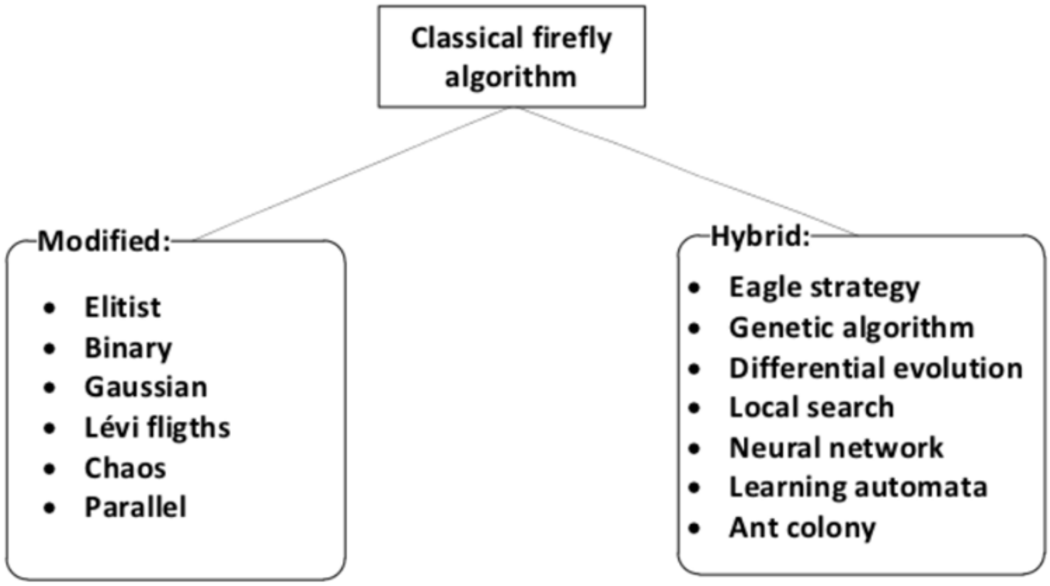

Figure 2.6 Taxonomy of firefly algorithms (Fister et al. 2013a).

a random number between [0,1]. The search algorithm will act as a blind search with random walks when $\beta_0 = 0$. While the parameter γ has an effect on the convergence speed, and also the value of parameter γ can be a value from the interval $\gamma \in [0, \infty)$, this value depends on the problem (Fister et al. 2014).

Although the FA algorithm has been successfully used in many applications, it has some disadvantages. Sometimes the firefly algorithm get stuck in local optima while it perform the local search and it cannot get rid of it (Fister et al. 2014). Also, the fixed parameters used in the FA algorithm is another disadvantage of the algorithm. Firefly algorithm cannot memorize the previous solutions, so it may move to same solutions which have been tested before. In this case it will waste more time to calculate solutions that already have been calculated before, and also this will reduce the chances to find the global optimum solution (Fister et al. 2013a).

Firefly algorithm is a global problem solver, it has been hybridized with other optimization algorithms, machine learning, heuristics and other techniques. This hybridization helps to enhance the performance of the search algorithm. The varients of firefly algorithm can be divided into two classes: modified and hybrid, as shown in Figure 2.6.

a. Modified Firefly Algorithms

In the past few years, FA algorithm has been modified in many ways to improve the search capabilities of the algorithm (Tilahun et al. 2017; Tilahun & Ngnotchouye 2017). Mainly modifications have focused on the light intensity and attractiveness factors. For example, Luleseged et al. (Tilahun & Ong 2012) have modified the FA algorithm by modifying the random part in the movement formula. If there is a firefly in the current best position and there is no improvement, this may reduce the brightness. Hence, they modified it by moving the firefly to other directions to achieve best performance by improving the brightness of firefly using m-uniform random vector. But if there is no direction that the firefly can move to, the firefly stays in the same position. This modification has been tested using seven benchmark functions, and the modified FA algorithm obtained better results than the classical FA algorithm.

In (Palit et al. 2011), Palit introduced the binary firefly algorithm to find the plain text from cipher text, by using Merkle-Helman knapsack cipher algorithm (Forouzan 2008). In their work, a new representation of the problem was considered using the firefly algorithm. The result of the FA algorithm was compared with GA, and they found that the binary firefly is better than genetic algorithm for solving this problem.

Another modification was made by Falcon et al. (Falcon et al. 2011) by using binary firefly for solving system-level fault diagnostic. In their work, an adaptive light absorption coefficient, binary encoding and problem-specific knowledge have been used. The results of binary FA algorithm were compared with artificial immune system (AIS) (Yang et al. 2008) and particle swarm optimization (PSO) (Falcon et al. 2010). The comparison showed that binary firefly algorithm is much better than the other compared algorithms in terms of speed and memory.

Chandrasekaran et al. (Chandrasekaran & Simon 2012) have implemented binary FA algorithm to solve network and reliability constructed unit commitment (UC) problem (Carrion & Arroyo 2006). The modified FA algorithm was tested using 10 unites of IEEE-RTS. The results of the algorithm was better than other algorithms reported in their paper.

Farahani (Farahani et al. 2011a) proposed a new modification of firefly algorithm by using Gaussian distribution. Gaussian distribution has been used to move all fireflies to global best in each generation by increasing convergence and speed. The new algorithm was tested with five standard functions. The results were better in accuracy and speed than the classical FA algorithm.

Yang (Yang 2011a) has studied the convergence and efficiency of three meta heuristics algorithms: simulated annealing (Bertsimas et al. 1993), particle swarm optimization (Beni & Wang 1993), and firefly algorithm. He analyzed the results of randomization methods, Gaussian random walk and L´evy flight. He concluded that the most important problem is how to balance between exploration and exploitation to get better results.

In (Yang 2010a), Yang modified the firefly algorithm movement formula using L´evy flights. After several experiments and numerical studies, the results conclude that L´evy flight FA is better than genetic algorithm and particle swarm optimization in both efficiency and accuracy rate.

In (Coelho et al. 2011), Coelho proposed a modification to firefly to improve the convergence by combining firefly algorithm with chaotic maps (Strogatz 2014). The main idea of using chaotic maps is to tune the random parameter α and light absorption coefficient γ in Eq.(2.13) to avoid being stuck in local optima. This new firefly algorithm has been tested with well-known benchmarks. The results were compared with other optimization algorithms. Results showed that the combined firefly algorithm with chaotic maps can achieve better performance than other optimization algorithms. On the other hand, Gandomi et al. (Gandomi et al. 2013a) proposed a new way to use chaos with firefly algorithm to increase the global search mobility for robust global optimization. The author proposed chaotic maps to tune the attractiveness parameter β_0 and light absorption coefficient α in Eq. 2.13. The new algorithm has been tested using 12 benchmark functions. The results showed that in majority of cases the new algorithm was better than the classical firefly algorithm.

Another development has been done to firefly algorithm by Subotic et al. (Subotic et al. 2012), by paralyzing the firefly algorithm, the parallelized algorithm has been tested with standard benchmarks functions, and the results showed that the paralyzed firefly algorithm is much faster and accurate than the classical algorithm. On the other hand, by using the concept of parallelism with firefly algorithm, Husselmann and Hawick proposed a modification to firefly algorithm to work with parallel graphical processing unit (GPU) (Husselmann & Hawick 2012). They tested the new firefly algorithm with standard benchmark functions. The results showed that the new firefly algorithm is much better in speed and accuracy than the classical firefly algorithm.

Yu et al. (2014) proposed a new adaptation strategy on the randomization parameter called wise step strategy for FA (WSSFA). In WSSFA, the wise step of the randomization is considered by taking the absolute distance between the firefly's global best position and best positions during the iteration process.

In anther work, Wang et al. (2016) proposed FA with random attraction (RaFA), which employs a randomly attracted model. In RaFA, each firefly is attracted to another randomly selected firefly. In order to enhance the global search ability of FA, a concept of Cauchy jump is utilized.

Recently, Wang et al. (2017b) presented a new adaptation mechanism for FAs' parameter called adaptive control parameters (ApFA). Comparative assessment in simulations of ApFA with standard FA and other variants of FA on benchmark functions have shown that ApFA outperformed those algorithms. In addition, Wang et al. (2017b) also proposed NSRaFA in which three neighborhood search and a new randomization model are employed to improve the exploration and exploitation abilities. The algorithm proposed is also capable of adjusting the control parameters automatically during the search process.

Table 2.3 Modified firefly algorithms (Fister et al. 2013a).

Topic	References
Elitist firefly algorithm	(Wang et al. 2012)
Binary represented firefly algorithm	(Palit et al. 2011; Falcon et al. 2011; Chandrasekaran & Simon 2012; Farahani et al. 2012)
Gaussian randomized firefly algorithm	(Farahani et al. 2011a; Yang 2011a)
Levy flight randomized firefly algorithm	(Yang 2011a, 2012, 2010a)
Chaos randomized firefly algorithm	(Coelho et al. 2011; Gandomi et al. 2013a)
Parallel firefly algorithm	(Subotic et al. 2012; Husselmann & Hawick 2012)

b. Hybrid Firefly Algorithms

Firefly algorithm can be hybridized with other algorithms to improve the results for solving problems. Firefly algorithm have the characteristics like multi-modality and good convergence speed which gives the firefly the ability to be hybridized with other algorithms.

In (Yang & Deb 2010), a new hybrid firefly algorithm with L´evy flight search has been proposed and called Eagle Strategy (ES). The eagle strategy algorithm was inspired by the behavior of eagles, the eagle fly in the space to search for prey in a random way like L´evy flights (Brown et al. 2007). When the eagle see the prey, it tries to catch it using all of it efficiency. The eagle strategy has two parts, the first one is a random search which works as L´evy flight and the other part is local search which works as firefly algorithm. This hybridized algorithm has been tested with Gaussian noise and Ackley function. The results of eagle strategy was better than particle swarm optimization in both efficiency and accuracy.

In (Luthra & Pal 2011), a hybridization of firefly algorithm with genetic algorithm has been proposed for cryptanalysis of the mono-alphabetic substitution cipher, for crossover operator the new algorithm used dominant gene crossover, and it used permutation as a mutation operator. After testing the new algorithm, it was clear that the algorithm is working better for large cipher text lengths, and if the algorithm will be applied to a short cipher lengths, more number of generations would be used to solve the problem.

Abdullah et al. proposed a new hybrid firefly algorithm (Abdullah et al. 2012) called ‘Hybrid Evolutionary Firefly Algorithm (HEFA). This new algorithm is a combination between the classical firefly algorithm and the evolutionary operations of the Differential Evolution (DE) method. This improvement made to improve the search algorithm to obtain better accuracy. The idea of this new algorithm is to split the generation to two sub-populations, the first sub-population will be handled by the firefly algorithm, and the second sub-population will be handled by the evolutionary operators of the differential evolution (Storn & Price 1997; Brest et al. 2006; Das & Suganthan 2011). The new algorithm has been tested to find the parameter values in biological model. The results of the Hybrid evolutionary firefly algorithm were much better than the results from the classical firefly, genetic algorithm, particle swarm optimization and evolutionary programming.

(Fister et al. 2012) hybridized the firefly algorithm with local search algorithm. The new algorithm has been applied to a graph 3-coloring problem, which is known as a combinatorial optimization problem (Iztok Fister 2013). The results of the new hybrid firefly algorithm using medium-scaled random graphs shows that it is better than other search algorithm like Hybrid Evolutionary Algorithm (HEA) (Galinier & Hao 1999), Tabucol (Hertz & de Werra 1987), and the evolutionary algorithm with SAW method (EA-SAW) (Eiben et al. 1998).

In speech recognition, firefly algorithm has been used to train parameters of Structure Equivalent Fuzzy Neural Network (SEFNN). FA has been used to improve the fuzzy neural networks as proposed by Hassanzadeh et al. (Hassanzadeh et al. 2012). The results of experiments showed that the new algorithm obtained higher accuracy than the classical fuzzy neural networks trained by particle swarm optimization.

Also the firefly algorithm has been used with back-propagation to train the feedforward neural network, this was applied by Nandy et al. (Nandy et al. 2012), the purpose of using firefly algorithm is to obtain more performance and increase the accuracy by training feedforward neural networks. The new algorithm has been tested with standard data sets. The results of the new algorithm were compared with genetic algorithm, and it showed that the new algorithm can obtain better accuracy and improve the speed to train the feedforward neural networks and also decrease the size of the feedforward neural network design.

In another work, (Hassanzadeh & Meybodi 2012) proposed to hybridize the firefly algorithm with cellular learning automata. Diverse solutions in the firefly algorithm was the main responsibility of the cellular learning automata, while the firefly algorithm improved the local search. The new algorithm was tested on five benchmarks functions, and the test results made a conclusion that the new algorithm can obtain the global optima and enhance the exploration of the classical firefly algorithm.

(Farahani et al. 2012) improved the firefly algorithm by learning automata. In theri work, they balanced the exploration and exploitation by hybridizing the firefly algorithm with genetic algorithm. These modifications have been tested with five benchmark functions. The results were compared with particle swarm optimization algorithm and classical firefly, and it showed that the new algorithms can be better than other algorithms. Guo et al. (2013) proposed a hybrid metaheuristic approach by hybridizing harmony search (HS) and FA, namely, HS/FA. HS/FA is used to solve function optimization. HS/FA combines exploration of HS with exploitation of FA. Rizk-Allah et al. (2013) presented hybridization between ant colony and firefly algorithm, named ACO-FA. The FA worked as a local search and the randomization parameter in FA is decreased gradually during the iteration process.

Later, Rahmani & MirHassani (2014) presented a hybridization with GA to solve discrete optimisation problem. The algorithm is applied to capacitate facility location problem (CFLP) which is a well-known combinatorial optimization problem. Tuba & Bacanin (2014) employed an improved seeker optimization algorithm (SOA) with firefly

Table 2.4 Hybrid firefly algorithms (Fister et al. 2013a).

Topic	References
Eagle strategy using Levy walk	(Yang & Deb 2010)
Genetic algorithm	(Luthra & Pal 2011; Farahani et al. 2012)
Differential Evolution	(Abdullah et al. 2012)
Memetic algorithm	(Fister et al. 2012)
Neural network	(Hassanzadeh et al. 2012; Farahani et al. 2012)
Cellular learning automata	(Hassanzadeh & Meybodi 2012; Farahani et al. 2012)
Ant colony	(Rajini & David 2011)

algorithm to build new hybrid FA algorithm. The approach uses either SOA or FA to enhance the exploitation search of the algorithm.

Rajan & Malakar (2015) presented a new hybrid algorithm combining Nelder–Mead (NM) simplex and Firefly Algorithm (FA). The NM simplex method is used to improve the exploitation section of FA and avoid premature convergence of FA. This algorithm is applied and demonstrated in solving power system ORPD problems. Sahu et al. (2015) combined Pattern Search (PS) and built a new hybrid method called hybrid Firefly Algorithm and Pattern Search (hFA-PS). The global exploration is done by FA and PS algorithm is used to enhance the local search. hFA-PS algorithm is used to optimise the scaling factors and PID controller gains for fuzzy PID controller of Load Frequency Control (LFC) of multi area power systems. The results outperform DE and a PSO variant. Recently, Gupta & Arora (2016) presented a new hybrid algorithm formulated by combining FA and social spider algorithm (SSA). The proposed algorithm is tested on various standard benchmark problems and then compared with FA and SSA. In another work, Nekouie & Yaghoobi (2016) proposed a new method to enhance firefly algorithm to solve multimodal optimization problems. The technique evolves in sub-population and utilizes a simulated annealing local optimization algorithm to increase search power, accuracy and speed of the algorithm.

A list of hybrid firefly algorithms is presented in Table 2.4.

2.3.2 Parameter Tunning of Firefly Algorithm

In the standard FA algorithm, the parameters in Eq. (4.13) are user defined and fixed. Similar to other metaheuristic algorithms, the performance of FA algorithm highly depends on the parameter values. It controls the degree of exploration and exploitation in the search process. In the basic FA algorithm, a firefly s_i moves towards a more promising solutions which helps the algorithm to move randomly and explore around better solutions. The effect of this random movement depends on the parameter α. If α is chosen to be large then the solution x_i will randomly jump away from the neighborhood and explore the solution space. Otherwise, its jump will be in the neighborhood and also may become negligible compared to the movement towards brighter fireflies. It can also dominate and move the solution out of the solution space if it is too large. To overcome this problem, different modifications have been proposed in the literature. Some studies purposed a modified α by assuming that the step length is a fixed value but not the upper bound (Farahani et al. 2011b,a).

In other works, parameter α has been modified by linearly decreasing its value according to the iteration number (Liu et al. 2015). Along similar lines, giving initial and final step length, an exponential decreasing step length is proposed in Shafaati & Mojallali (2012). Apart from given starting and final step length, there are some modification of parameter based on an initial step length only (Wang et al. 2012). In there work, α is made to be inversely proportional to the square of the iteration number where it decreases quickly. In (Selvarasu et al. 2013), the attraction step length has been updated by modifying γ and $\beta 0$.

Other modification on the attraction is done by using different chaotic mappings as presented in (Gandomi et al. 2013a), where twelve different chaotic maps are presented to make γ and β adaptive. Similar work is done in (Jansi & Subashini 2015) using chebyshev mapping and in (Khalil 2014) using sinusoidal maps. The attraction term is supposed to be influenced by the light intensity and also the distance between the fireflies along with the light absorbtion coefficient. A chaotic map and self-adaptive have been used to tune the randomization and absorption coefficients in the algorithm, the algorithm has been tested to solve a dynamic economic emission dispatch (DEED), the results of the algorithm has been evaluated using numerical simulation and it showed that firefly algorithm can perform efficiently to solve multi-objective problems (Niknam et al. 2012).

2.3.3 Applications of Firefly Algorithms

Firefly algorithm has been used in many research fields, like solving optimization problems and classification and also in engineering applications. Figure 2.7 shows the domains that firefly algorithm has been applied to. In the figure it is clear that firefly algorithm has been applied in to three types of problems: optimization, classification and engineering applications (Tilahun et al. 2017; Tilahun & Ngnotchouye 2017). For optimization, firefly algorithm has been applied in to continuous, multi-objective, combinatorial, dynamic and noisy, and constrained optimization. Also it was used as classifier in data mining, machine learning and neural networks. For engineering applications, firefly algorithm has been applied to industrial optimization, antenna design, image processing, sensor networks, business optimization, chemistry, civil engineering, semantic web and robotics. For classification, FA algorithm has been applied to feature selection problems.

a. Optimization

Continuous optimization. Many researchers published papers about firefly algorithm solving continuous problems like (Yang 2011a,b, 2012, 2010b,a; Tilahun et al. 2017; Tilahun & Ngnotchouye 2017; Wang et al. 2017b), most of the papers used well-known benchmarks. In Gandomi et al. (Gandomi et al. 2011), firefly algorithm has been used to solve continuous and discrete structural optimization problems. These problems discuss solving pressure vessel design, helical compression spring design, welded beam design, car side impact design, stepped cantilever beam design and reinforced concrete beam design. The experiments using firefly algorithm give better results than other optimization algorithms like genetic algorithm, differential evolution,

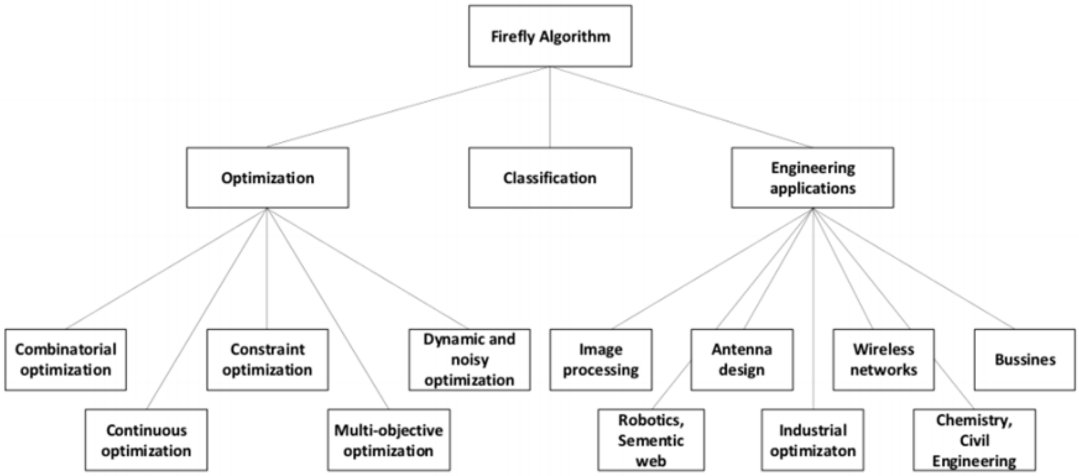

Figure 2.7 Taxonomy of firefly (Fister et al. 2013a).

particle swarm optimization and simulated annealing. Tilahun et al. (2017) have provided a detailed survey of continuous variants of firefly algorithm.

Combinatorial optimization In (Durkota 2009), Durkota used firefly algorithm to solve combinatorial optimization problems which called Quadratic Assignment Problem (QAP). The author changed the continuous functions to discrete functions by mapping from continuous to discrete for functions like movement, distance and attractiveness. The algorithm has been tested with 11 Quadratic Assignment Problems. The results of the algorithm showed that the algorithm achieve efficiency with simple problems and most times the algorithm stuck in local optima for hard problems.

A discrete firefly algorithm has been used to solve one of the NP-hard problems, which is flow shop scheduling problem. This experiment was proposed by Sayadi et al. (Sayadi et al. 2010). The results of using a well-known benchmarks problems shows that the discrete firefly algorithm achieved better success rates than the ant colony optimization algorithm.

On the other hand, firefly algorithm has been used to solve another NP-hard problem which is task graph scheduling problem (Hönig 2010). The benchmark that has been used in this problem was a 36,000 task problem. The results shows that firefly algorithm achieve better performance than other optimization algorithms used to solve the same benchmark.

Also evolutionary discrete firefly algorithm (EDFA) has been used to solve a symmetric traveling salesman problem (TSP) which is considered one of the NP-hard problems. This experiment has been proposed in (Jati et al. 2011). In their work, the inverse mutation has been used as a mutation operator, and the cities were represented in an array, the tour was presented in sequence of indices which points to the array. The results of the evolutionary discrete firefly algorithm shows that it was better than other optimization problems by using some of TSPLIB data sets. However, the EDFA algorithm sometimes stuck in local optima.

In (Khadwilard et al. 2011) work, authors solved the Job scheduling problem (JSSP), the paper perform the test in 5 data sets from OR-Library (Beasley 1990). The results in the paper shows that the firefly algorithm can achieve better solutions in some datasets. However, the author mentioned that the algorithm sometimes stuck in local optima.

A firefly algorithm has been used to solve path planning problem by Liu et al. (Liu et al. 2012). The algorithm has been used to improve the solution resulted by the classical firefly by designing adaptive random and absorption parameters. The results showed that there was an improvement in the classical firefly algorithm. Also (Wang et al. 2012) presented a modified firefly algorithm (MFA) to solve path planning problem for uninhabited combat air vehicle (UCAV). The modification is the higher fireflies exchange information with lower fireflies while the light intensity is updating. This modification improved the performance and results. The results of this algorithm were compared with original firefly algorithm and other optimization like ant colony optimization (ACO) (Dorigo & Di Caro 1999), biogeography-based optimization (BBO) (Simon 2008), differential evolution (DE) (Storn & Price 1997), evolutionary strategy (ES) (Fogel 1997), genetic algorithm (GA) (Goldberg 1989), probability-based incremental learning (PBIL) (Baluja 1994), particle swarm