Facial Expression Recognition Using Uniform Local Binary Pattern with Improved Firefly Feature Selection

Abdulla M. K. Elmadhoun, and Md. Jan Nordin

Department of Computer Science, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi, Malaysia

***Abstract–Facial* expressions are essential communication tools in our daily life. In this paper, the uniform local binary pattern is employed to extract features from the face. 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 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 firefly where the great deluge algorithm (GDA) has been integrated. The great deluge is a local search algorithm that helps to enhance the exploitation ability of the firefly algorithm, thus preventing being trapped in local optima. The improved firefly algorithm has been employed in a facial expression system. Experimental results using the Japanese female facial expression database show that the proposed approach yielded good classification accuracy compared to 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.**

***Index Terms*—Facial expression recognition, feature selection, firefly algorithm, optimization.**

I. Introduction

Automated analysis of facial expressions has been gaining momentum in the field of computer vision over the past few years. Interestingly, facial expressions contribute a significant part of the nonverbal communication between human beings (Khatri, et al., 2014). Developing automated systems to recognize human emotions with good accuracy and speed under different imaging variations such as illumination and scale have been gaining considerable attention (Khatri, et al., 2014; Jamshidnezhad and Nordin, 2013). The successful application of the texture descriptors motivated the use of local binary pattern (LBP) for face representation (Ojala, et al., 2002; Hamid and Nordin, 2016). However, the essential problem of using local descriptors is the high dimensionality of the data. High dimensional data can decrease the speed and accuracy of the classifier (Bereta, et al., 2013; Alsalibi, et al., 2015). High dimensionality not only effects recognition accuracy but also imposes computational constraints (Alsalibi, et al., 2017). Hence, employing an optimization feature selector is essential for eliminating redundant and irrelevant features so as to provide highly discriminating feature representation.

Firefly algorithm is a metaheuristic algorithm inspired by the social behavior of a group of fireflies. It was introduced by Yang in 2010 (Yang, 2010). During the optimization process, the algorithm attempts to move the fireflies as inspired by the interaction of real fireflies. As each firefly produces light based on the phenomenon of bioluminescence, certain suggestions are made in the algorithm. In principle, each firefly will be exploring and searching for other fireflies and preys randomly. However, the main limitation of the classical firefly algorithm is the risk of being trapped in local optima due to the loss of population diversity during the optimization process (He and Huang, 2017).

Hence, the aim of this paper is to propose an automatic facial expression recognition system using uniform LBP descriptor and a modified firefly optimization algorithm. Approximately, the improved version of the firefly algorithm has been 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.

The rest of this paper is organized as follows: Section 2 provides a brief description of the classical firefly algorithm and the modified firefly algorithm. The application of the improved firefly algorithm in facial expression recognition is presented in Section 3. Experimental results and discussions are presented in Section 4. Finally, Section 5 concludes the paper and gives some suggestions for the future work.

II. firefly algorithm

A brief description of the classical firefly algorithm and the modified firefly algorithm is presented in the following subsections.

ARO-The Scientific Journal of Koya University
Volume VI, No.1 (2018), Article ID: ARO.10378, 10 pages
DOI: 10.14500/aro.10378
Received 04 February 2018; Accepted: 24 March 2018
Regular research paper: Published 09 April 2018
Corresponding author's, e-mail: p74470@siswa.ukm.edu.my

 http://dx.doi.org/10.14500/aro.10378

A. The classical FA algorithm

Firefly Algorithm is a metaheuristic algorithm which simulates the social behavior of a group of fireflies. The search pattern of FA is determined by the attractions among fireflies, whereby a less bright firefly moves toward a brighter firefly. However, the flashing lights depend on some physics factors. One of these factors is the light intensity *I* which decreases when the distance *r* increases. 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 is 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.

In FA algorithm, the light intensity I of a firefly at a location x is associated with the value of objective function. In addition, it decreases as the distance r increases, so the expression of light intensity is as presented in equation 1:

$$I(r)=I_0e-\Upsilon r^2 \quad (1)$$

Where, I_0 is the light intensity of the source and Y is the fixed light absorption coefficient. The attractiveness β is proportional to the light intensity I(r). Thus, it can be defined by (2) as follows:

$$\beta=\beta_0e^{-\Upsilon r2}) \quad (2)$$

Where, the parameter β_0 denotes the attractiveness at the distance r=0. Each firefly X_i is compared with all other fireflies X_j, where J ≠ i. If X_j is brighter (better) than X_i, X_i will be attracted to and move toward X_j. The movement of X_i can be defined by equation 3.

$$r_{ij}=\|X_i-X_j\|=\sqrt{\sum_{d=1}^{D}\left(X_{id}-X_{jd}\right)^2} \quad (3)$$

Note that r_{ij} is the distance between X_i and X_j, *D* is the dimension of the problem. When firefly X_i is attracted to another firefly X_j, the movement from firefly X_i to firefly X_j is defined as follow:

$$X_{id}(t-1)=X_{id}(t)+\beta_0e-\Upsilon rij^2(X_{jd}(t)-X_{id}(t)+\alpha€i) \quad (4)$$

Where $€_i$ is a random number uniformly distributed in the range [−0.5, 0.5] and α ∈ [0,1] is the step factor.

B. Improved FA algorithm

In this section, the improved version of the FA algorithm is presented. The classical FA algorithm still encounters serious problems in large-scale databases. In such cases, the key shortcomings of the FA are the risk of being trapped in local minima (premature convergence) and the slow convergence rate. In this context, striking a balance between exploration and exploitation (intensification) is essential to cope with such limitations. The classical FA algorithm mimics the social behavior of fireflies based on the flashing and attraction that typically involves interaction between different fireflies in the swarm. To enhance the performance of such interactions and to improve the trajectories of fireflies, the great deluge local search algorithm has been hybridized with FA.

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 and Ngnotchouye, 2017; Mistry, et al., 2017). Mainly modifications have focused on the light intensity and attractiveness factors. For example, (Tilahun and 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 the best performance by improving the brightness of firefly using m uniform random vector. However, 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, using Merkle-Hellman knapsack cipher algorithm. 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 the genetic algorithm for solving this problem.

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 searches 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.

C. The GDA

GDA is a variant of simulated annealing local search algorithm (Yang, 2010) known for its ability to maintain the diversity of the population and avoid the trap in local minima. The distinguishing characteristic of GDA is its deterministic level-based acceptance criterion (He and Huang, 2017). Besides the acceptance criterion, GDA requires only one parameter (called decay rate) to be determined to control the acceptance of the nonimproving solutions, which is the reason behind the selection of GDA in this work. In addition to accepting improving solutions, keeping a maximization problem in mind, GDA can accept a nonimproving solution given that its fitness is greater than or equal to a dynamically updated value (called the level).

D. GD-FA procedure

Hybridizing GD algorithm with firefly algorithm can prevent the later from being stuck in local optima, especially after the longtime of generations. Furthermore, it can help the firefly algorithm to obtain the best accuracy in less number of function evaluations. This improvement can be done by applying the GD algorithm for each candidate solution (firefly) in the current generation. Each firefly performs a local search, and the generated solution will replace the old solution based on the acceptance criteria. This can be performed by sending the firefly values to the GDA algorithm as initial solutions. Basically, the GDA algorithm starts by generating new neighbors by changing random number of bits from binary 0 to 1 or vice versa. Then, it calculates the accuracy (Fitness) of the new generated solution. After that, GDA compares the fitness of the new solution with the local best, if the new solution is better than the local best, the algorithm will replace the local best and current solution with the new solution. Besides that, the value of the water level will be linearly increased using $\Delta\beta$. The GDA process will be iterated until the termination condition is encountered which is set to be reaching the maximum number of iterations. However, if the local best is better than the new solution, the algorithm will decide whether to accept the new solution as a current solution or not according to the water level value. If the fitness of the new solution is greater than the water level, then it will accept the new solution, otherwise, the algorithm will start generating a new solution and continue the evolution process. After applying the local search for all fireflies in the generation, the firefly will continue processing the fireflies by applying the movement equation. Fig. 1 shows a step by step summary of the GD-FA algorithm.

The procedure of GD-FA algorithm can be summarized as follows:

1. Initialize the first generation of fireflies randomly.
2. Send each firefly values to GDA in the generation.
    a. The current solution and local best will be equal to the solution sent from FA.
    b. Calculate the accuracy (fitness) of the current solution.
    c. Generate new solution from current solution by changing a random number of bits.
    d. Calculate the accuracy (fitness) of the new solution.
    e. If new solution is better than local best, replace local best and current solution with new solution.
    f. If local best is better than the new solution and a new solution is greater than the water level, then replace current solution with new solution.
    g. If neither e nor f, then start again from c.
    h. Update water level.
    i. If stop criteria have been met send back the local best to FA.
3. Find the local best in the generation.
    If the local best is better than the global best, update the global best. Note that, initially, the value of the Global best is equal to zero.
4. Apply movement to all fireflies, and create a new generation.
5. Fix invalid values in each firefly to be binary 0 or binary 1 in the new generation.
6. Redo 2–6 until it reaches to a maximum number of generations.

III. Application of improved firefly algorithm for facial expression recognition system

Typically, the facial expression recognition process consists of three main phases which are (face detection, feature extraction, and classification) for static images. In the face detection phase, the input image is processed to extract the face region from the original image. Thereafter, some preprocessing steps are applied such as converting the image to grayscale image, resizing the cropped face and preparing it for feature extraction. The output of the feature extraction process is a representation of the face image in a feature vector with length equals to the number of features extracted from the image. After applying the face detection and feature extraction to all images in the dataset, a dataset containing all features of all faces in the dataset has been generated, which will be ready to be trained using one of the training algorithms such as support vector machine, k-nearest neighbor, neural networks, and other training algorithms which also called the classification process.

Interestingly, it is possible to add other phases to the recognition process pipeline to improve the accuracy and performance of the overall model. In this paper, the feature selection process is added after feature extraction process to select the most important features from the extracted feature space and exclude the redundant and noisy features that could negatively affect the classification accuracy, where those excluded features will be considered as noise. The feature selection process can be performed by many techniques, mostly by artificial intelligence methods which search for the best combination of features to achieve the best accuracy.

A. Face detection stage

The detection of faces in images is the first step in any typical facial recognition system. The main purpose of face detection is to detect the presence of faces in the image and their corresponding location. For this purpose, the well-known Viola–Jones face detector algorithm has been used to detect the face region due to its accuracy, speed, and straightforward implementation in MATLAB using the computer vision toolbox. First, the face and eyes will be detected by Viola–Jones face detector; each eye will be framed by square where the center of the square will be the center of the eyes. Then, by detecting the center of the eyes the face will be normalized by making the distance between the eyes equals 55 pixels. Based on that, the face will be cropped and resized to 110 × 150 pixels. Fig. 2 shows an illustrative example of the face detection process.

B. Feature extraction stage

In this stage, informative features are extracted from face images. Bearing in mind that not all pixels in an image contain reliable features, extracting meaningful features sufficient to represent the face is an essential step. The extracted features should minimize intraclass variations caused by different illumination conditions while maximizing interclass variations.

LBP has been first proposed by Ojala, et al. and since then it has become one of the most widely used local descriptors

http://dx.doi.org/10.14500/aro.10378

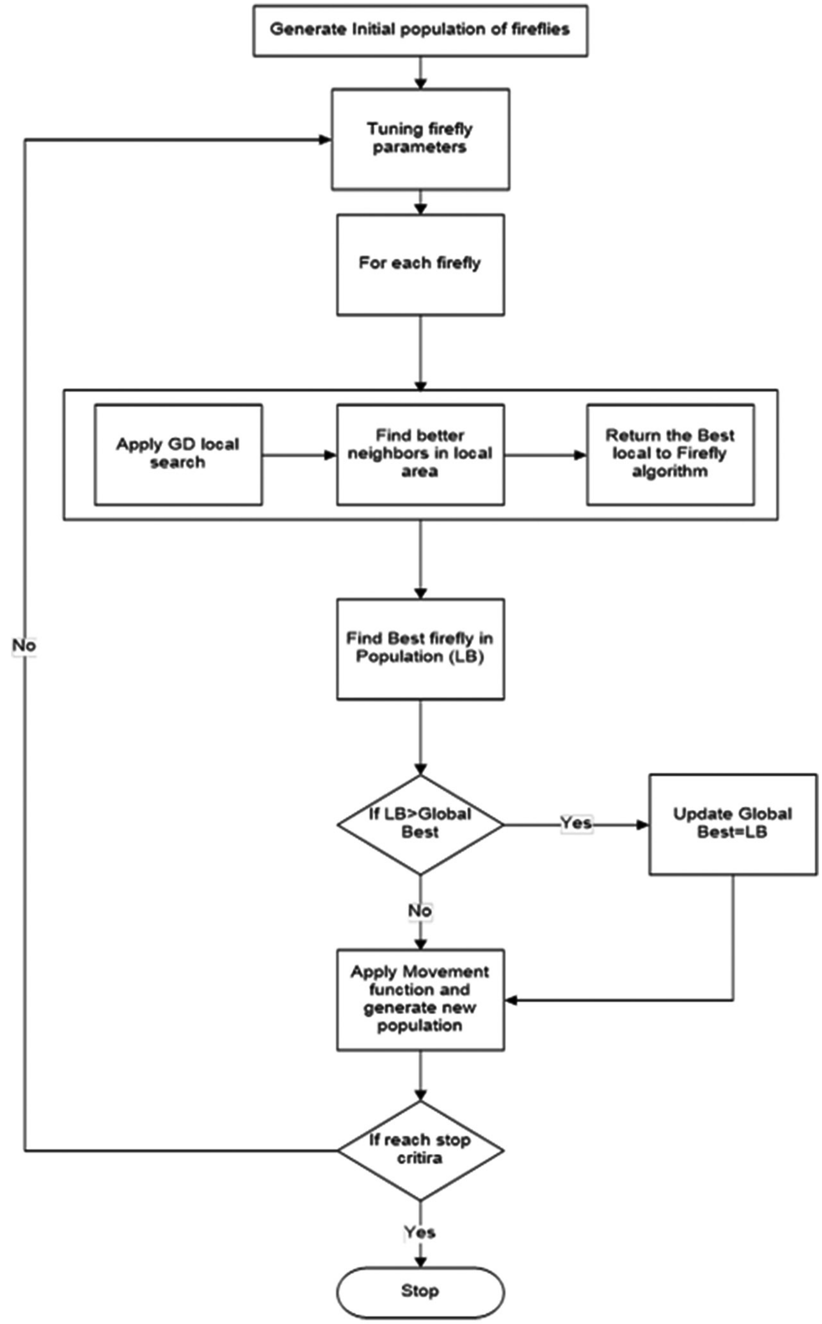

Fig. 1. GD-FA process

due to its computational simplicity and invariance to different lighting conditions (Ojala, et al., 2002). Although the standard LBP is relatively insensitive to monotonic gray level changes, its main limitation is the sensitivity toward noisy pixels wherein the value of the pixels can be easily affected by the erroneous surrounding pixels (Bereta, et al., 2013). In this phase, after detecting the face region and cropping it, the system will extract the features from the face image using uniform LBP (uLBP) as illustrated in 5–8. Following the literature, for extracting the uLBP features from the face, the face will be divided into 42 regions, each region with a size of 18x21 pixels, and then feature extraction will be applied in each region to generate 59 features by applying uLBP (5-8). In the end, all regions will be combined and concatenated together in one feature vector. Note that, this combined feature vector includes 2478 features for representing one face image. By applying the same process to all Japanese Female Facial Expression (JAFFE) dataset, a features matrix with a size of 213 × 2478 will be created and prepared for feature selection process.

$$LBP_{p,r}(Z_c) = \sum_{p=0}^{p-1} \delta\left(g_p - g_c\right) 2^p \quad (5)$$

$$U\left(P\right)=P\left(P-1\right)+2 \quad (6)$$

$$h_l = \sum B\left(L\left(Z_c\right)=l\right), l = 0, \ldots\ldots, \left(\left(U\left(P\right)+1\right)-1\right) \quad (7)$$

$$B(x)\begin{cases}1 & \text{if} \times \text{ is true} \\ 0 & \text{Otherwise}\end{cases} \tag{8}$$

Fig. 3 shows a sample of feature extraction process applied to a face image.

C. Feature selection stage

In pattern recognition context, the term of facial feature selection refers to the mining mechanism that looks for a subset of features from the feature pool that is sufficient to maximize the interclass and minimize the intraclass discrimination between different classes. Feature selection not only reduces the feature vector dimensionality but also reduces the computational cost and improves the classification accuracy (Alsalibi, et al., 2015; Alsalibi, et al., 2017). Therefore, feature selection became a fundamental step in facial expression recognition systems for a better feature representation. In this stage, to further select the optimal set of discriminative facial features through the extracted feature space, firefly algorithm has been used. Here, the problem of feature selection is formulated as an optimization problem where the aim is to maximize the classification accuracy while minimizing the cardinality of the feature subset.

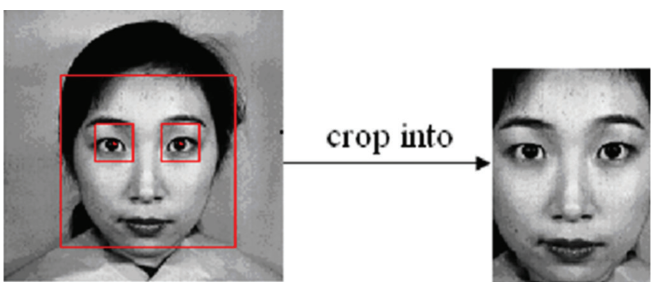

Fig. 2. (a and b) Face detection example

Feature selection in this research is a process in which the system will try to minimize the number of features extracted by uLBP by finding the most relevant and important features from the faces as well as eliminating redundant features that could reduce the recognition accuracy which is like noisy features in the dataset to achieve better recognition accuracy. In this research, a bioinspired binary search algorithm called FA-GD is used to select features of all images which were extracted by uLBP to reach better accuracy.

FA-GD feature selection process

The first step in the feature selection process is to prepare the FA-GD algorithm by initializing the required parameters, those parameters can be initialized by defining:

1. Number of generations that the firefly algorithm will generate after applying the movement function.
2. Number of fireflies in each generation.
3. The initial value of attractiveness, light intensity, and randomization parameters.

Next, the FA-GD algorithm starts to randomly generate the initial generation of fireflies by creating a sequence of binary codes with length 2478 for each firefly as a vector. In this vector, the bits with value one represent that the particular features are being selected for training and testing, whereas, the bits with value (0) represent that those features will be dropped and not included in the training and testing process. Fig. 4 shows an illustrative example of the binary encoding structure of the firefly. Each firefly in the initial population will be evaluated by applying a 10-k cross-validation using k-nearest neighbors algorithm (k-NN) to obtain the accuracy of each firefly, where the accuracy of each firefly typically represents the fitness value of each firefly individually.

However, this process can be done by rebuilding the dataset which was generated from the feature extraction

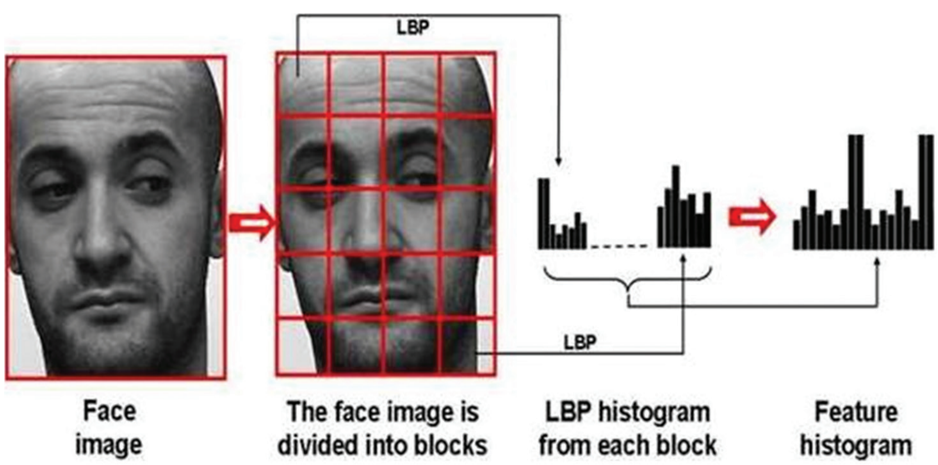

Fig. 3. Feature extraction example

f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	...	fn
1	0	1	1	1	0	0	1	1	0	...	0

Fig. 4. FA-GD solution structure

 http://dx.doi.org/10.14500/aro.10378

process by removing all the features from the dataset in which the firefly have the binary value of 0, and keeping all the features in the dataset in which the firefly have the binary value of 1. For example, if the firefly has the binary value of 0 at index 50, this means that all the features in index 50 in the whole dataset will be removed. Thus, the new generated dataset will have the same number of instances as the original dataset, but it will have a different number of features depending on the number of binary 1 and binary 0 in the firefly chromosome. For instance, if the firefly chromosome has 1000 binary (1) values, then the new rebuilt dataset will have the dimensions of 213 × 1000, where 213 is the number of instances and 1000 is the number of features which represent the firefly. This process will be applied for all fireflies individually before evaluation, so each firefly will have a different dataset structure depending on number and position of binary 0 and binary 1 in each firefly. Fig. 5 shows an illustrative example of rebuilding a new dataset for each firefly in the initial generation.

After evaluating all fireflies in the initial generation, the algorithm will find the best firefly that achieved the best accuracy and stores it as the global best, and then the algorithm will apply the movement as in 4 for all fireflies in the generation to create a new generation. As a result of applying the movement equation to each firefly, the new firefly may contain invalid values as a real number with decimal point, to fix this problem, a logistic sigmoid transformation function (Rathipriya, et al., 2011) is applied to all values in the firefly to fix the values and set it to be binary 0 or binary 1. By applying this transfer process to all fireflies after their movement, the new generation of fireflies is ready to be evaluated and continue with the typical evolution process of firefly algorithm.

D. Classification stage

The fourth phase of the proposed expression recognition model is the classification phase. As mentioned previously, in feature extraction, one feature vector is extracted from each face image. Next, relevant and important features are selected. Then, all the feature vectors are fed to the classifier. In this paper, K-nearest neighbor (k-NN) has been used to

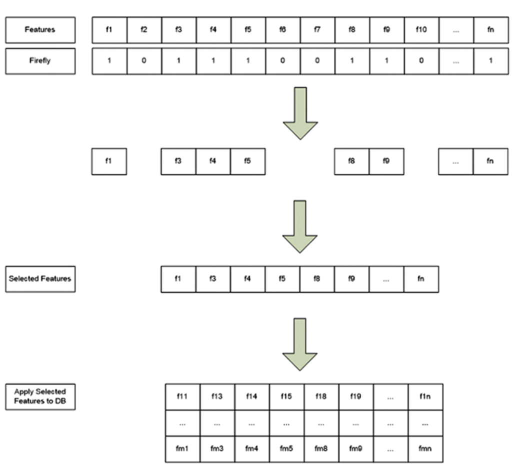

Fig. 5. Sample of building a new dataset based on selected features

classify the emotions for recognition. K-NN is a simple, fast method and it shows a good recognition results in other face and expression recognition systems as documented in the literature. The method will be used as a fitness function in the feature selection process, where the accuracy result will be the fitness value in training. Furthermore, it will be used as a classification for testing. The Euclidean distance as shown in 9 has been used as measure function in k-NN classification to calculate the distance between the features. K-NN algorithm works by calculating the minimum distance from the query instance to the training samples to determine the K-nearest neighbors. After gathering the K-nearest neighbors, a simple majority of these K-nearest neighbors are taking to be the prediction of the query instance.

$$(p,q)=\sqrt{(p_1-q_1)^2+(p_2-q_2)^2+\ldots+(p_n-q_n)^2} \tag{9}$$

IV. Experimental results and discussion

This section demonstrates the experimental evaluation of the proposed approach and its impact on improving the overall performance of the proposed recognition model. Basically, one of the databases that researchers often use for facial expression recognition is the JAFFE database (Lanitis, et al., 1995). This database contains 213 frontal facials images corresponding to 10 Japanese females with 7 different expressions (Happy, sad, angry, disgust, surprised, natural, and fear). The size of each image in the database is 256x256 pixels, all available in grayscale. Fig. 6 shows a sample of four female subjects from the JAFFE database. The proposed facial expression recognition system is coded in MATLAB 2015 under Windows OS platform on an Intel Core i7 2.4GHz processor with 8GB of RAM.

In all the experiments, the proposed firefly algorithm has an adaptive parameter control strategy based on Chaos maps. To demonstrate the effect of the number of fireflies on the performance of the system, the maximum number of generations is fixed, and the number of fireflies is varied from 10 to 50. Along similar lines, to investigate the effect of varying the number of generations on the performance of the system, the number of fireflies is fixed, and the maximum number of generations is varied from 25 to 100 as will be shown in the following experiments. Note that in all experiments, 10-k fold cross validation has been performed for classification. In the first set of experiments, the effect of the number of fireflies, number of generations, and number of GD iterations has been investigated. In the next experiment, the classical firefly algorithm will be compared with the improved firefly (GD-FA) in a statistical sense to prove that the proposed firefly algorithm significantly outperforms the conventional firefly in the feature selection task. Thereafter, the convergence behavior trends and computational time of both algorithms have been shown. Finally, the proposed facial expression system will be compared with other state-of-the-art approaches using the JAFFE dataset.

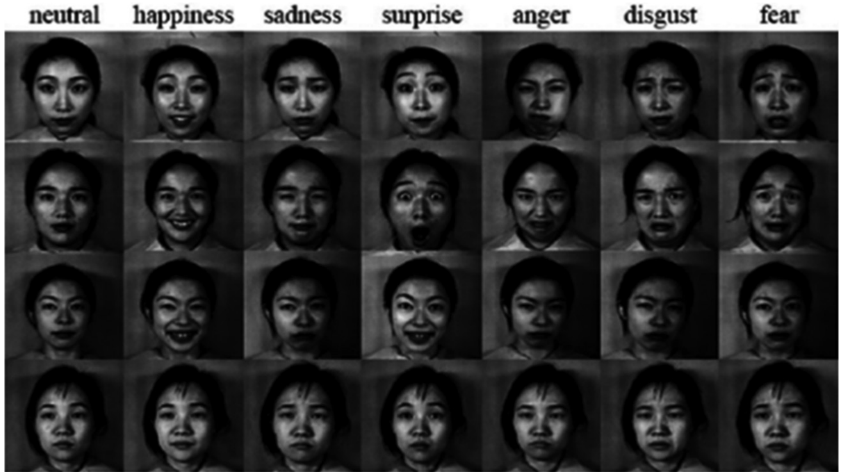

Fig. 6. Sample images from Japanese Female Facial Expression database

A. A. Effect of the number of fireflies

Basically, varying the number of fireflies in the initial population plays a significant impact on the performance of the firefly algorithm. Hence, in this experiment, to demonstrate the effect of the number of fireflies on the classification accuracy of the facial expression recognition system, the number of generations will be fixed. For each generation number from 25 to 150, the number of fireflies will be varied from 10 to 50 fireflies. The mean accuracy, best accuracy, standard deviation, and number of selected features will be reported for each combination of those parameters.

The results of this experiment are tabulated in Table I, where best results are shown in bold font. From the results shown in Table I, it can be seen that the best result has been obtained when using 50 fireflies and 150 generations where classification accuracy of 94.923% has been yielded. Furthermore, the number of selected features was 1254, which is sufficient to get good classification accuracy. Note that the original number of features was 2478 and reduced to about 50% when using the feature selection phase.

TABLE I
Mean, best accuracy, SD, and number of the selected features for the FA-GD algorithm on Japanese female facial expression

Gen. No.	No. Firefly	Mean	Best	SD	Selected feature
25	10	92.394	93.427	0.577	1262
	20	93.239	94.366	0.63	1233
	30	93.568	94.366	0.544	1218
	40	93.944	94.836	0.562	1235
	50	94.131	95.305	0.507	1266
50	10	93.380	94.366	0.643	1275
	20	93.615	93.897	0.328	1237
	30	94.131	94.836	0.456	1244
	40	94.366	95.305	0.495	1251
	50	94.319	94.836	0.604	1225
100	10	93.568	94.366	0.445	1206
	20	93.850	94.836	0.562	1217
	30	94.695	95.305	0.317	1260
	40	94.923	95.775	0.604	1266
	50	94.789	95.305	0.267	1246
150	10	94.923	94.366	0.411	1239
	20	94.319	94.836	0.267	1234
	30	94.695	95.305	0.387	1275
	40	94.742	95.305	0.431	1235
	50	94.923	95.775	0.373	1254

SD=standard deviations

B. B. Effect of the number of GD iterations

In this experiment, to figure out the optimal value of the GD iteration, it will be varied from 10 to 50 iterations while fixing the number of fireflies and the number of firefly generations. The standard deviation, mean, and best accuracy, as well as the number of selected features, are reported. Note that, the number of selected features indicates the subset of features that have been selected by FA-GD at the end of iterations. From Table II, it can be seen that the maximum mean accuracy (96.2%) has been obtained when using 50 generations, 50 fireflies, and 20 GD iterations. Furthermore, when the number of fireflies is small (Dueck, 1993), the accuracy cannot be improved due to the lack of exploration.

C. Performance and convergence analysis

Convergence is an important metric for meta-heuristic optimization algorithms to indicate how fast the algorithm can reach to the optimum solution. The convergence metric is usually used to investigate whether the optimization algorithm can achieve and maintain a proper balance between exploration and exploitation during the search process so that it can avoid being stuck in local optima. As can be depicted from Fig. 7, the firefly algorithm suffers from the premature convergence problem as it gets stuck in local optima at generation 40. From generation 40 to 150, the algorithm cannot get out from the local optimum value. In this experiment, the average accuracy reported was 93.4%, where the accuracy of each facial expression is shown in Table III.

As can be deduced from Table III, the facial expression model shows good performance in detecting the natural, anger, disgust, and surprise expressions. However, lower performance has been reported for the fear and sad expressions where classification accuracy of 90.6% and 77.4% has been reported, respectively.

http://dx.doi.org/10.14500/aro.10378

TABLE II
Mean, best accuracy, SD, and number of the selected features for the FA-GD

No. FA generations	No. firefly	No. GD iterations	Mean	Best	SD	Selected features
25	10	10	94.178	94.836	0.396	1246
		20	94.225	94.836	0.387	1229
		30	94.225	95.305	0.736	1251
		40	94.648	95.305	0.594	1208
		50	94.085	94.836	0.505	1254
	20	10	94.554	95.305	0.396	1242
		20	94.883	96.244	0.643	1255
		30	94.836	95.775	0.495	1297
		40	95.164	95.775	0.445	1205
		50	94.883	95.775	0.467	1204
	30	10	94.742	95.775	0.431	1229
		20	95.305	95.775	0.383	1247
		30	95.070	95.775	0.332	1242
		40	95.258	96.244	0.467	1266
		50	95.681	96.714	0.577	1253
	40	10	94.930	95.775	0.533	1270
		20	95.540	95.775	0.247	1254
		30	95.305	95.775	0.221	1235
		40	95.493	96.244	0.454	1265
		50	95.587	96.244	0.396	1262
	50	10	95.211	95.775	0.431	1268
		20	95.446	95.775	0.544	1239
		30	95.258	95.775	0.346	1247
		40	95.587	96.244	0.396	1227
		50	95.681	96.714	0.533	1232
50	10	10	94.131	94.366	0.332	1260
		20	94.507	95.305	0.588	1277
		30	94.554	95.305	0.396	1243
		40	94.836	95.305	0.313	1245
		50	94.883	95.305	0.346	1245
	20	10	94.883	95.775	0.517	1226
		20	95.493	96.224	0.551	1249
		30	95.352	96.244	0.517	1203
		40	95.493	96.244	0.454	1276
		50	95.352	96.244	0.467	1217
	30	10	95.857	95.775	0.328	1261
		20	95.493	95.775	0.328	1255
		30	95.399	96.244	0.370	1231
		40	95.775	95.775	0.242	1236
		50	95.493	95.775	0.242	1232
	40	10	95.455	96.242	0.387	1233
		20	95.631	96.712	0.497	1245
		30	95.871	96.712	0.370	1236
		40	95.771	96.242	0.383	1233
		50	95.921	96.712	0.445	1244
	50	10	95.777	96.712	0.495	1239
		20	96.200	96.71	0.411	1231
		30	95.966	96.711	0.454	1239
		40	95.922	96.711	0.445	1232
		50	95.963	96.711	0.396	1234

SD=standard deviations

In what follows, the same experimental procedure has been applied to the improved firefly algorithm to evaluate its performance and convergence trend. The convergence behavior and the confusion matrix of the improved firefly algorithm when using 20 fireflies and 50 generations are shown in Fig. 8 and Table IV, respectively. An average classification accuracy of 95.4% has been reported.

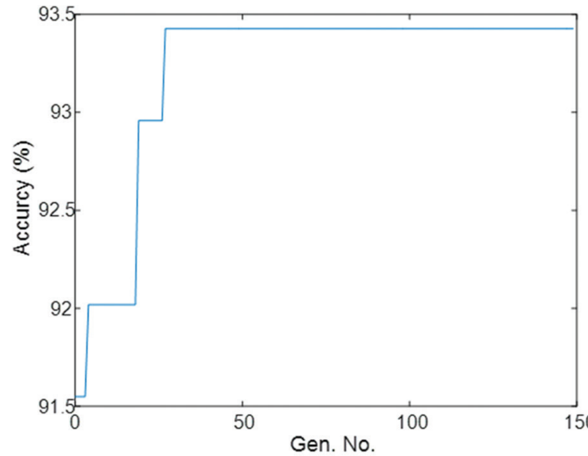

Fig. 7. Convergence behavior of the firefly algorithm during the first 150 generations and using 20 fireflies algorithm on Japanese female facial expression

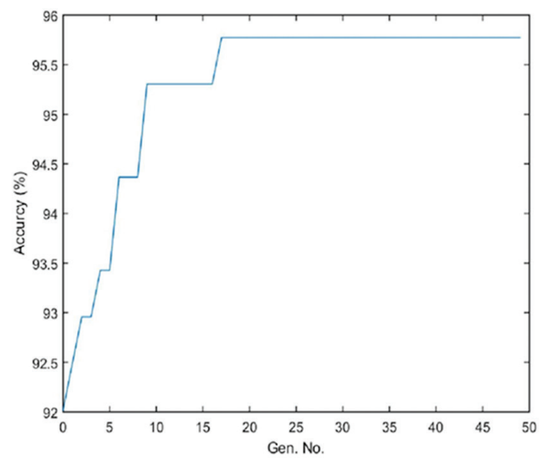

Fig. 8. Convergence behavior of FA-GD algorithm using 20 fireflies

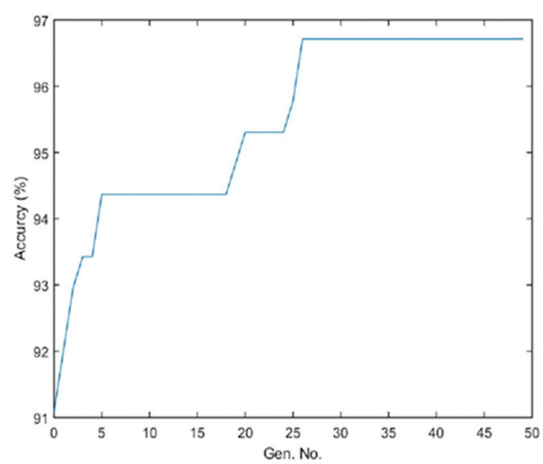

Fig. 9. Convergence behavior of the FA-GD algorithm using 50 fireflies

The convergence behavior trend and the confusion matrix of the improved firefly algorithm when using 50 fireflies and 50 generations are shown in Fig. 9 and Table V, respectively.

An average classification accuracy of 96.7% has been reported. The obtained results show that the performance and convergence behavior of the improved firefly algorithm is better than the classical firefly algorithm. The incorporation of the Great Deluge algorithm helps in enhancing the exploitation ability of the firefly algorithm and achieving a better balance between exploration and exploitation and hence preventing the premature convergence problem.

TABLE III
Confusion matrix using the k-NN classifier (150 generations and 10 fireflies)

	Anger	Disgust	Fear	Happy	Natural	Sad	Surprise
Anger	96.67	0	0	0	0	0	0
Disgust	0	96.55	3.13	0	0	3.23	0
Fear	0	3.45	90.63	0	0	6.45	0
Happy	0	0	0	96.77	0	3.23	0
Neutral	0	0	3.13	3.23	100	9.68	0
Sad	3.33	0	3.13	0	0	77.42	0
Surprise	0	0	0	0	0	0	96.67s

TABLE IV
Confusion matrix of the FA-GD algorithm (50 generations and 20 fireflies)

	Anger	Disgust	Fear	Happy	Natural	Sad	Surprise
Anger	96.67	0	0	0	0	0	0
Disgust	0	96.55	3.13	0	0	3.23	3.33
Fear	0	0	93.75	0	0	3.23	0
Happy	0	0	0	93.55	0	3.23	0
Natural	3.33	0	0	0	100	9.68	0
Sad	0	3.45	3.13	6.45	0	80.65	0
Surprise	0	0	0	0	0	0	100

TABLE V
Confusion matrix using the improved firefly algorithm, (50 generations and 50 fireflies)

	Anger	Disgust	Fear	Happy	Natural	Sad	Surprise
Anger	100	0	0	0	0	0	0
Disgust	0	96.55	3.13	0	0	3.23	0
Fear	0	0	93.75	0	0	3.23	3.33
Happy	0	0	0	90.32	0	3.23	0
Natural	0	0	0	3.23	100	3.23	0
Sad	0	3.45	3.13	6.45	0	87.10	0
Surprise	0	0	0	0	0	0	96.67

D. Comparative evaluation with the state-of-the-art studies

In this section, the proposed facial expression recognition system will be compared with state-of-the-art methods that adopted similar protocols using the JAFFE database. To establish a fair comparison with the previous studies, the same experimental procedure has been used where a 10-fold cross validation has been considered. The performance of the proposed approach is compared with recent state-of-the-art benchmark methods as listed in Table VI. The best results and the standard deviations (SD) for different methods with the corresponding reduced dimension are listed in Table VII. From the results in Table VII, it can be seen that the proposed approaches outperformed most of the compared methods, except for the Gabor-SRC method. However, the number of features used in Gabor-SRC approach is approximately twice the number of features in our proposed approach.

V. Conclusion

This paper proposed an efficient mechanism to improve the recognition accuracy of facial expression recognition systems. First, uLBP features were extracted from face images. Second, to further enhance the feature descriptor, an enhanced version of firefly algorithm, called FA-GD was proposed to select the most discriminative and robust facial features. As the dimensionality of the extracted features is relatively high, FA-GD has been used to select the optimal set of facial features and eliminate redundant and noisy features to boost the recognition performance and speed up the computations. Several experimental evaluations have been carried out using the JAFFE database. Results 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.

VI. Acknowledgment

The researchers wish to thank Universiti Kebangsaan Malaysia, for supporting this work by research grant DIP-2016-018.

TABLE VI
Keys to comparative approaches on the Japanese female facial expression database

Key	Method	Evaluation procedure	Reference
DKLLE	Discriminant kernel locally linear embedding	10-fold cross-validation	(Zhao and Zhang, 2012)
LLE	Locally linear embedding		(Zhao and Zhang, 2012)
LDA	Linear discriminant analysis		(Belhumeur, et al., 1996)
PCA	Principle component analysis		(Turk and Pentland, 1991)
P-LBP	Patch-based LBP		(Happy and Routray, 2015)
Gabor-SRC	Gabor feature and SRC Classifier		(Lu, et al., 2015)
SVM-FA	SVM based improved FA		(Mistry, et al., 2017)

TABLE VII
The best accuracy (std) of different state-of-the-art methods on the Japanese female facial expression

Method	LDA	PCA	LLE	SLLE	DKLLE	P-LBP	Gabor-SRC	SVM-FA	Proposed (firefly)	Proposed (FA-GD)
Dimension	6	20	80	30	40	2478	2560	50-65	1242	1230
Accuracy (%)	80.81±3.6	78.09±4.2	75.57±3.8	78.57±4.0	84.06±3.8	91.8	97.68	87.75	94.8±0.75	96.7±0.54

References

Alsalibi, B., Venkat, I., Subramanian, K., Lutfi, S.L. and Wilde, P.D., 2015. The impact of bio-inspired approaches toward the advancement of face recognition. ACM Computing Surveys (CSUR), 48(1), p. 5.

Alsalibi, B., Venkat, I. and Al-Betar, M.A., 2017. A membrane-inspired bat algorithm to recognize faces in unconstrained scenarios. Engineering Applications of Artificial Intelligence, 64, pp.242-260.

Bereta, M., Pedrycz, W. and Reformat, M., 2013. Local descriptors and similarity measures for frontal face recognition: A comparative analysis. Journal of Visual Communication and Image Representation, 24(8), pp.1213-1231.

Belhumeur, P.N., Hespanha, J.P. and Kriegman, D.J., 1996. Eigenfaces vs. Fisherfaces: Recognition using Class Specific Linear Projection. *In: European Conference on Computer Vision*. Springer, Berlin, Heidelberg, pp.43-58.

Dueck, G., 1993. New optimization heuristics: The great deluge algorithm and the record-to-record travel. Journal of Computational Physics, 104(1), pp.86-92.

Happy, S.L. and Routray, A., 2015. Robust Facial Expression Classification using Shape and Appearance Features. *In Advances in Pattern Recognition (ICAPR), 2015 Eighth International Conference on*. IEEE, pp.1-5.

Hamid, A.A.K.A. and Nordin, M.J., 2016. Radius based block LBP for facial expression recognition. international information institute (Tokyo). Information, 19(9B), p. 4197.

He, L. and Huang, S., 2017. Modified firefly algorithm based multilevel thresholding for color image segmentation. Neurocomputing, 240, pp.152-174.

Jamshidnezhad, A. and Nordin, M.J., 2013. Bee royalty offspring algorithm for improvement of facial expressions classification model. International Journal of Bio-Inspired Computation, 5(3), pp.175-191.

Khatri, N.N., Shah, Z.H. and Patel, S.A., 2014. Facial expression recognition: A survey. International Journal of Computer Science and Information Technologies (IJCSIT), 5(1), pp.149-152.

Lanitis, A., Taylor, C.J. and Cootes, T.F., 1995. Automatic face identification system using flexible appearance models. Image and Vision Computing, 13(5), pp.393-401.

Lu, X., Kong, L., Liu, M. and Zhang, X., 2015. Facial expression recognition based on gabor feature and SRC. *In*: *Chinese Conference on Biometric Recognition*. Springer, Cham, pp.416-422.

Mistry, K., Zhang, L., Sexton, G., Zeng, Y. and He, M., 2017. Facial expression recognition using firefly-based feature optimization, *2017 IEEE Congress on Evolutionary Computation (CEC)*, San Sebastian, pp.1652-1658.

Ojala, T., Pietikainen, M. and Maenpaa, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7), pp.971-987.

Palit, S., Sinha, S.N., Molla, M.A., Khanra, A. and Kule, M. 2011. *A Cryptanalytic Attack on the Knapsack Cryptosystem using Binary Firefly Algorithm.* Vol. 2. International Conference on Computer and Communication Technology (ICCCT), pp.428-432.

Rathipriya, R., Thangavel, K. and Bagyamani, J., 2011. Binary particle swarm optimization based biclustering of web usage data. arXiv preprint arXiv:1108.0748. International Journal of Computer Applications, 25, pp. 975-8887.

Tilahun, S.L. and Ngnotchouye, J.M.T., 2017. Firefly algorithm for discrete optimization problems: A survey. KSCE Journal of Civil Engineering, 21(2), pp.535-545.

Tilahun, S.L., Ngnotchouye, J.M.T. and Hamadneh, N.N., 2017. Continuous versions of firefly algorithm: A review. Artificial Intelligence Review, 2017, pp.1-48.

Tilahun, S.L. and Ong, H.C., 2012. Modified firefly algorithm. Journal of Applied Mathematics, 2012, pp.12.

Turk, M. and Pentland, A., 1991. Eigenfaces for recognition. Journal of Cognitive Neuroscience, 3(1), pp.71-86.

Wang, H., Zhou, X., Sun, H., Yu, X., Zhao, J., Zhang, H. and Cui, L., 2017b. Firefly algorithm with adaptive control parameters. Soft Computing, 21(17), pp.5091-5102.

Yang, X.S., 2010, Firefly algorithms for multimodal optimization. *In: International Symposium on Stochastic Algorithms.* Springer, Berlin, Heidelberg, pp.169-178.

Zhao, X. and Zhang, S., 2012. Facial expression recognition using local binary patterns and discriminant kernel locally linear embedding. EURASIP Journal on Advances in Signal Processing, 2012(1), p. 20.