FACIAL EXPRESSION RECOGNITION BASED ON CULTURAL PARTICLE SWAMP OPTIMIZATION AND SUPPORT VECTOR MACHINE

A. B. Adetunji[1], J. P. Oguntoye[1*] O. D. Fenwa[1] and E. O. Omidiora[1]

[1]*Department of Computer Science and Engineering*

Faculty of Engineering & Technology,

Ladoke Akintola University of Technology (LAUTECH), Nigeria

ABSTRACT

Facial expressions remain a significant component of human-to-human interface and have the potential to play a correspondingly essential part in human-computer interaction. Support Vector Machine (SVM) by the virtue of its application in a various domain such as bioinformatics, pattern recognition, and other nonlinear problems has a very good generalization capability. However, various studies have proven that its performance drops when applied to problems with large complexities. It consumes a large amount of memory and time when the number of dataset increases. Optimization of SVM parameter can influence and improve its performance.Therefore, a Culture Particle Swarm Optimization (CPSO) techniques is developed to improve the performance of SVM in the facial expression recognition system. CPSO is a hybrid of Cultural Algorithm (CA) and Particle Swarm Optimization (PSO). Six facial expression images each from forty individuals were locally acquired. One hundred and seventy five images were used for training while the remaining sixty five images were used for testing purpose. The results showed a training time of 16.32 seconds, false positive rate of 0%, precision of 100% and an overall accuracy of 92.31% at 250 by 250 pixel resolution. The results obtained establish that CPSO-SVM technique is computational efficient with better precision, accuracy, false positive rate and can construct efficient and realistic facial expression feature that would produce a more reliable security surveillance system in any security prone organization.

Keywords: Cultural Algorithm, Particle Swarm Optimization, Culture Particle Swarm Optimization, Support Vector Machine.

INTRODUCTION

Facial expression recognition is an important aspect of biometrics that has attracted the attentions from several researchers due to its importance in achieving a smart human-machine interface. Facial expressions are one of the most powerful, natural and immediate means for human being to communicate their emotions and intensions (Ghimire, Jeong, Lee & Park, 2016) (Qayyum, Majid, Anwar & Khan, 2017).Information that has to do with human behaviour pertaining to facial expression can predict the mental state of a man. Since Human face plays a key role in relational intelligent interaction and affective computing (Priyanka, Kesari, Ligendra & Nazil, 2013).

Utilizing human face as a key to security, biometric face recognition technology has gotten noteworthy consideration because of its potential for a wide variety of both in law enforcement and non-law enforcement (Aluko, Omidiora, Adetunji & Odeniyi, 2015). With respect to various security challenges and the increasing criminal activities in the world today; it is necessary to implement an improved technique to recognise facial expressions

in human. Since whatever affects the heart sometimes show on the face; a reliable emotion perception scheme is required in order to translate human expression and behavioural changes into useful commands to control systems. Emotion recognition is a challenging task because humans do not always express themselves by words and gestures (Qayyum *et. al,* 2017). Facial expression is useful in applications such as access control, human–computer interaction, production control, e-learning, fatigue driving detection, and emotional robot (Ghimire *et. al.,* 2016) (Tang & Chen, 2013). Several techniques such as Hidden Markov Models ...work (ANN), ...Machine (SVM) etc. has been adopted to recognise facial expressions in human.

Support Vector Machine among other algorithms has a very good generalization capability and dynamic classification scheme which makes it suitable for facial expression recognition. Various studies have shown that the performance of SVM drops with increasing data samples (Abdulameer, Abdullah & Othman, 2014). Also, SVM consumes large amount memory and time we applied to problems with large complexities. Nevertheless,

adequate parameter selection for SVM will not only reduce its computational burden but also increase its generalization capability and accuracy.

Manekar and Waghmare (2014) suggested a novel approach to improving the accuracy of SVM using the hybrid cultural algorithm. Therefore, this study focuses on the use of Cultural Particle Swamp Optimization (CPSO) to optimise the parameter of SVM in facial expression recognition.

Particle Swarm Optimization

The Particle Swarm Optimization (PSO) is an optimization technique of the swarm intelligence paradigm(Windisch, Wappler & Wegener, 2007). PSO is a population-based optimization technique inspired by the behaviour of schools of fish, herds of animals or flocks of birds (Eberhart & Kennedy, 1995). In PSO the system is initialized with a population of random solutions called particles. These particles move through the problem space in search of the global minima or maxima. Each particle keeps track of its past best performance/fitness and its neighbours (Specified proximity radius) best performance to decide on its next move. Also, the swarm is aware of the global best achieved by all the particles. At each iteration, the particles will update its velocity and position using the equation (1) & (2) respectively.

$$V_i = \omega V_i + c_1 r_1 \cdot (pBest - x_i) + c_2 r_2 \cdot (gBest - x_i) \quad (1)$$

$$x_i = x_i + \delta V_i \quad (2)$$

Cultural Algorithms

Cultural Algorithms are a class of computational models derived from observing the cultural evolution process in nature(Yan, Wu, Zhang, Chen, Luo, & Li, 2012). The Cultural Algorithm is a dual inheritance system that characterizes evolution in human culture at both the macro-evolutionary level, which takes place within the belief space, and at the micro-evolutionary level, which occurs at the population space. In CAs the characteristics and behaviours of individuals are represented in the Population Space. This representation can support any population-based computational model such as Genetic Algorithms, Evolutionary Programming, Genetic Programming, Differential Evolution, Immune Systems, among others(Jin & Reynold 1999). The framework of the basic CA component and its relationships are depicted in Figure 1. Adding a central knowledge (belief space) to any search evolutionary Algorithm like Evolutionary Programming (EP), Genetic Programming (GP), Genetic Algorithm (GA), Particle Swamp Optimization (PSO), Ant Colony Optimization (ACO) etc. becomes a Cultural Algorithm.
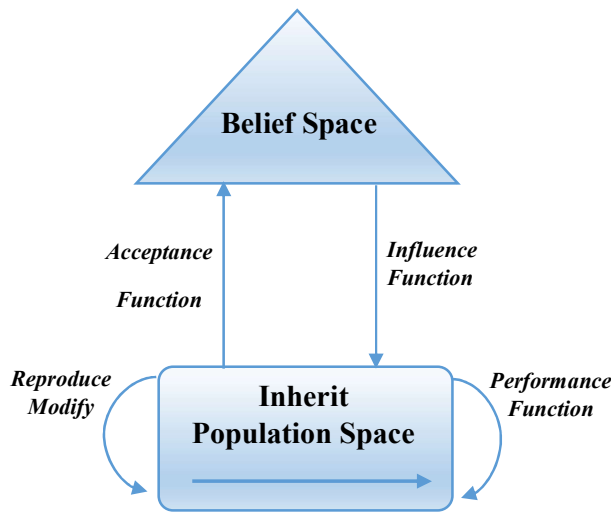

Figure 1: Cultural Algorithm Framework

Culture Particle Swarm Optimization

Culture Particle Swarm Optimization (CPSO) is a hybrid of Cultural Algorithm (CA) and Particle Swamp Optimization (PSO). Introducing Particle Swarm Optimization (PSO) into the model of the Cultural algorithm (CA) results into the Culture Particle Swarm Optimization (CPSO) techniques

(Manekar and Waghmare, 2014). Therefore, the CPSO has an advantage of fast convergence ability of PSO and the global optimizing ability of CA. The idea behind the construction of CPSO is to increase the diversity of the particle swarm with the aim of improving its global optimizing ability through the evolution of the population space and the knowledge space.

Support Vector Machine (SVM)

Support Vector Machines are a maximal margin hyperplane classification method that depends on results produced from statistical learning theory to ensure a very high generalization performance. Kernel functions are used to efficiently map input data that may not be linearly separable to a high dimensional feature space where linear methods can then be applied (Cristianini & Shawe-Taylor, 2000).

Related Works

Susskind, Littlewort, Bartlett, Movellan & Anderson (2007) proposed a Support Vector Machine (SVM) based facial expression recognition system. This study compared the human judgment and the performance of the SVM model for facial expression recognition. The results obtained showed the recognition rates for the human and SVM model were 89.2% and 79.2% respectively. Geetha, Ramalingam & Palanivel (2009). proposed real-time facial expression system using Support Vector Machine (SVM). The application of SVM on two expression class (neutral and smile) gave a recognition rate of 98.5%. However, the training and recognition times are high.

Samad and Hideyuki (2011) applied Support Vector Machine (SVM) on FEEDTUM (Facial Expressions and Emotions from Technical University of Munich) database to achieve a recognition rate of 91.7%. Edge based feature extraction technique with Gabor filter was used to extraction facial features while PCA was used for

feature dimensionality reduction before the application of SVM.

Tang and Chen (2013) combines curvelet transform with improved Support Vector Machine (SVM) based on Particle Swarm Optimization (PSO) for facial expression recognition. Seven different expressions were recognized and an average recognition rate of 94.94% was achieved for JAFFE database.

Adeyanju, Omidiora & Oyedokun, (2015) analysed the Performance of different Support Vector Machine kernels (Radial Basis Function, Linear Function, Quadratic Function and Polynomial Function) for face emotion recognition. A local African database of 714 face emotion images consisting of seven facial expression taken twice from 51 persons was used. The results obtained using the SVM multi-class classification scheme reveals that the Quadratic Function SVM kernel performs best for face emotion recognition with an average accuracy of 99.33%. However, despite the good performance achieved with higher dimensions the computation time is High.

MATERIALS AND METHODOLOGY
In this study, a black African local database (LOCDAT) which consist of 240 facial expression images were taken with a digital camera with a default size of 1200 x 1600 pixel. The acquired database is made of six facial expression images from forty subjects. The original face images were downsized into a 100 x 100, 150 x 150, 200 x 200 and 250 x 250 pixels.
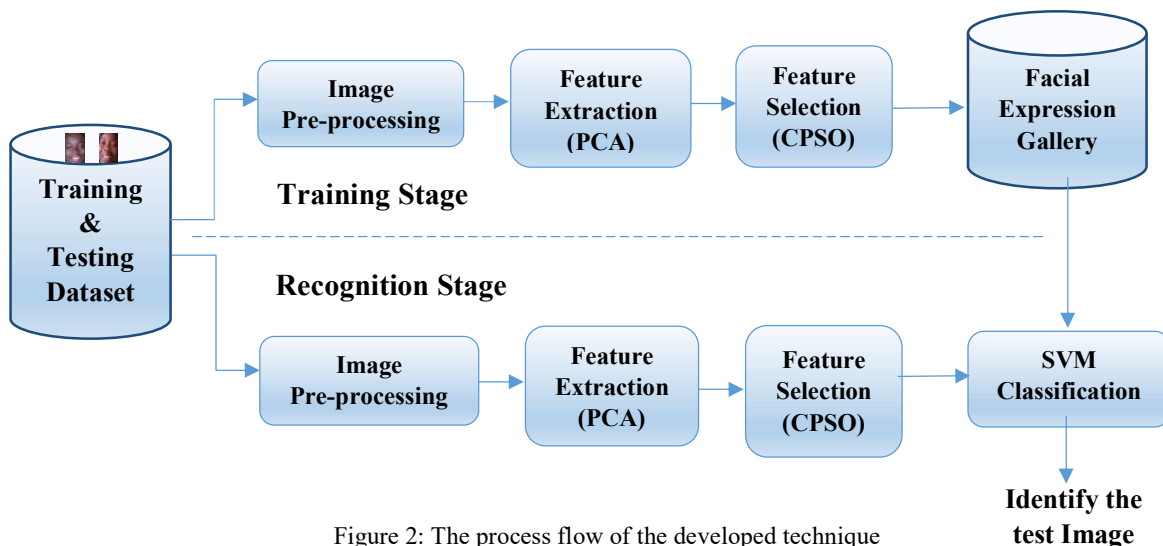

Figure 2: The process flow of the developed technique

One hundred and seventy-five (175) of those images were used for training and Sixty-five (65) images were used for testing the system. The acquired images were passed into the system for pre-processing to manage time and memory space by converting the coloured images into grayscale as well as histogram equalization for image enhancement.

The flow diagram in figure 2 above shows the process flow of the training and testing phase. The training phase consists of the trained face gallery, the feature extraction, and selection component that produces the feature vectors which serves as the template used for matching. PCA is the feature extraction component while CPSO is the feature selection component from which an optimized feature subset was obtained. The optimized feature subset was saved in the database for comparison. Similarly, the testing phase also consists the input test image which also undergoes the pre-processing stage after which PCA was used for dimension reduction and extraction of facial features.

CPSO was used to perform the selection of the optimum feature from the entire facial feature extracted by PCA and is fed to SVM for classification. The result of classification is compared with trained feature stored in the library for recognition. The final output of the system based on the recognition of the test images was displayed.

PCA for Feature Extraction

PCA is a statistical strategy for discovering relationship between features in data. At the point when utilized on facial images the subsequent images are frequently referred to as Eigenfaces. PCA is utilized for lessening dimensionality of data by eradicating out trivial information from the dataset and is as often as possible utilized as a part of both image processing and machine learning (Lemley, Abdul-Wahid, Banik & Andonie 2016).

Principal Component Analysis (PCA) was employed in this study to extract features and reduces the dimension sizes of images to form Eigenfaces. The resultant feature representation presents a suitable platform for selecting the optimal feature subsets.

The CPSO for Feature Selection

The optimal feature subset selection was done by applying CPSO algorithm on the image data matrix from the feature extraction stage. CPSO automatically chose parameter for SVM to reduce the numbers of features in the entire features extracted by PCA with minimum error(Manekar and Waghmare, 2014). The process flow of CPSO is shown below in figure 3. The algorithm is described step-by-step as follows.

Step 1*:* **Initialization of parameter and generation of particles**
Particles (P_i) with random positions and velocities are created within the range [0,1].

Step 2*:* Creation of initial belief space
Belief space (B_s)was initially created as an empty set.

Step 3: Fitness Computation of each particle P_i
The fitness value of each of the particles was computed using (3); to evaluate the performance of each particle.

$$F = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\emptyset_i - \bar{\emptyset}_i\right)^2} \qquad (3)$$

Where $\emptyset_i$represents the ith model output; $\bar{\emptyset}_i$represents the ith desired output, and N represents the number of input data.

Step 4: Determine and update the current local best position $L_{Best}$and global best position G_{Best} using equation (4)$L_{Best}(t + 1) =$
$\begin{cases} P_i(t) \ , & \text{if } F\big(P_i(t)\big) < F\big(L_{Best}(t)\big) \\ L_{Best}(t), & \text{if } F\big(P_i(t)\big) \geq F\big(L_{Best}(t)\big) \end{cases}$
$G_{Best}(t + 1) = \arg\min_{L_{Best}} F\big(L_{Best}(t + 1)\big), 1 \leq i \leq I.(4)$
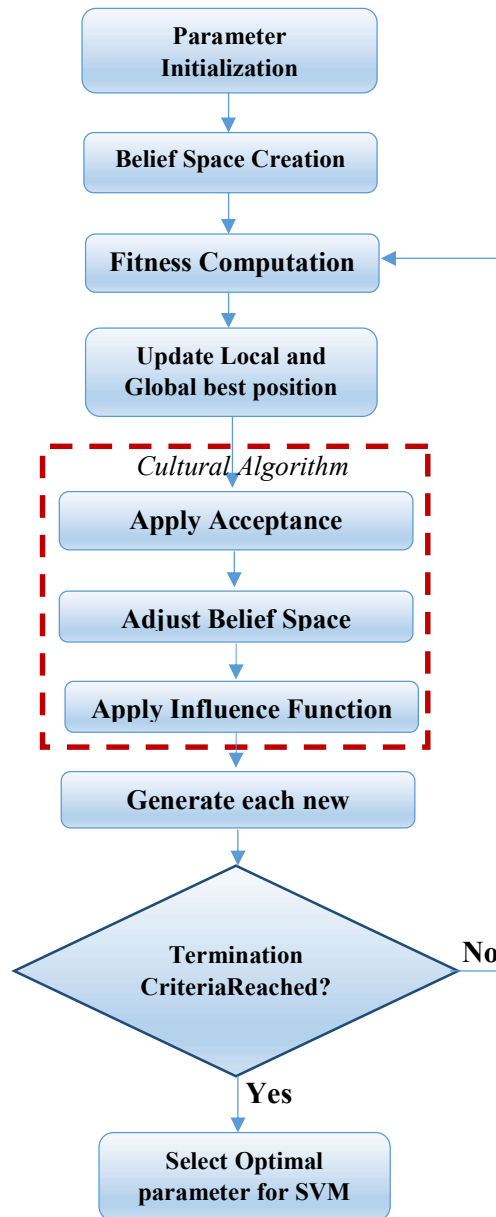
Figure 3: The process flow of CPSO

Step 5: Apply the acceptance function and adjust the belief space B_s.

Equation (5) determines the number of particles that will be used to adjust the belief space while equation (6) determines the interval of the belief space.

$$N_{accepted} = n\% \times I + \frac{n\%}{t} \times I$$

(5)

Where $n\%$ is a parameter that is set by the user, I is the number of particles, and t represents the tth generation.

$$IB_s = [l_w, u_p] = \{p|l_w \leq p \leq u_p, p \in 3i\}$$

(6)

Where l_w is the lower bound on belief space B_s and u_p is the upper bound on belief space B_s. l_w and u_p are determined using (7):

$$l_w = \begin{cases} P_i \text{ if } p_i \leq l_w \\ l_w, \quad \text{otherwise} \end{cases} \quad u_p = \begin{cases} P_{i,} \text{ if } P_i \geq u_p \\ u_p, \quad \text{otherwise} \end{cases}$$

(7)

Step 6: Apply influence function to generate new particle swarm.

Based on the updated L_{Best}, G_{Best}, l_w and u_p adjust the position of the particle swarm using an influence function (8) to change the direction of each particle in solution space and to avoid being easily trapped at a local optimum. Update the velocity and position of each particle using

equation (9) and (10) to generate new particle swarm.

$$P_i(t) = \begin{cases} P_i(t) + |R() \times (u_p - l_w)| \text{if} P_i < l_w \\ P_i(t) - |R() \times (u_p - l_w)| \text{if} P_i > u_p \end{cases} (8)$$

$$V_i(t+1) = w \times V_i(t) + c_1 \times R() \times [L_{Best}(t+1) - P_i(t)] + c_2 \times R() \times [G_{Best}(t+1) - P_i(t)](9)$$

$$P_i(t+1) = P_i(t) + V_i(t+1) \qquad (10)$$

Where c_1 and c_2 signify the acceleration coefficients; w controls the magnitude of $V_i(t)$ and R() are random numbers uniformly distribution in the range [0, 1] at each iteration.

Step 7: Convergence
If the maximum iteration times have reached, then go to Step 8, else return to Step 3.

Step 8: Select Optimal Parameter
Select the best global position P_i of the particle swarm.

Classification Using SVM
The Selected best global position (P_i) of the particle swarm trained the SVM with the detected feature subset mapped by P_i and modelled with the optimized parameters C and σ using equation (11).

$$\min \quad \frac{1}{2}\|P_i\|^2 + C\sum_{i=1}^{N}\xi_i$$

$$Such \; that \quad \sum_{i=1}^{N}P_i x_i \geq \left(\frac{1-\xi_i}{y_i}\right) - b$$

$i = 1, 2, \ldots, N, \; \xi_i \geq 0, \; i = 1, 2, \ldots, N, (11)$

Equation (12) was applied to obtain the final classification of each case:

$$y_i = arg \; max_{k(1 \ldots k)}(P_i^T y_i(x_i) + b_i) \quad (12)$$

Where N is the size of the dataset, C is the cost function. I , ξ are the slack variables, x and b is an offset scalar.

The SVM classification schemes used in this study are binary classification and multiclass classification. The binary classification schemes involve one against one i.e. the dataset were classified in pairs while the multiclass classification involves one against all i.e. multiple binary classifications.

Implementation in MATLAB
The implementation tool used was MATLAB R2012a version on Windows 7 Ultimate 64-bit operating system, Intel®Pentium® CPU T4500@2.30GHZ Central Processing Unit, 4GB Random Access Memory and 500GB hard disk drive.The model experimented by taken into consideration the facial expression recognition in 100 x 100, 150 x 150, 200 x 200 and 250 x 250-pixel resolution. An interactive Graphic User Interface (GUI) application was developed with a real-time database consisting of 240 facial expression images from 40 persons. The performance of the techniques on trained and recognized faces was measured against recognition accuracy, precision, false positive rate and computation time.

RESULTS AND DISCUSSION
The results obtained by the CPSO-SVM techniques with respect to the metrics previously mentioned were evaluated as follows. Table I presents the average training time at different resolutions for CPSO-SVM and SVM model after four trials in each case.

The results shown in Table I reveals that the training time increases with increase in the features of the training set. The CPSO-SVM model trains much faster than the SVM model. Therefore, the CPSO-SVM model is less computationally expensive compared to the SVM model. Figure 4 shows the graph of average training time against the dimension size.

Table I: Average training time at different resolutions for CPSO-SVM and SVM model

Dimension Size	CPSO + SVM (seconds)	SVM (seconds)
100 by 100	5.61	8.65
150 by 150	8.28	15.78
200 by 200	12.96	21.52
250 by 250	16.32	31.70

Table II presents the performance of CPSO-SVM and the SVM in terms of the previously mentioned performance metrics. The study reveals that at 250 x 250-pixel resolution with threshold value 0.8; the

CPSO-SVM model generated a false positive rate of 0.0%, the precision of 100% and accuracy of 90.7% at 6.66 seconds in SVM binary classification scheme. Also, the CPSO-SVM model generated a

false positive rate of 0.0%, the precision of 100% and accuracy of 92.31% at 27.71 seconds in SVM multiclass classification scheme.

It can be inferred from the table II that the CPSO-SVM model gave an increased 1.54% recognition accuracy, 12% precision, 7.69% specificity and a decreased FPR of 7.69% over the SVM model in

SVM binary classification scheme. Similarly, the CPSO-SVM also gave an increased 1.54% recognition accuracy, 10.53% precision and a decreased FPR of 4.55% over the SVM model in SVM multiclass classification scheme. The CPSO-SVM model is less computationally expensive in terms of recognition time compared to the SVM model.
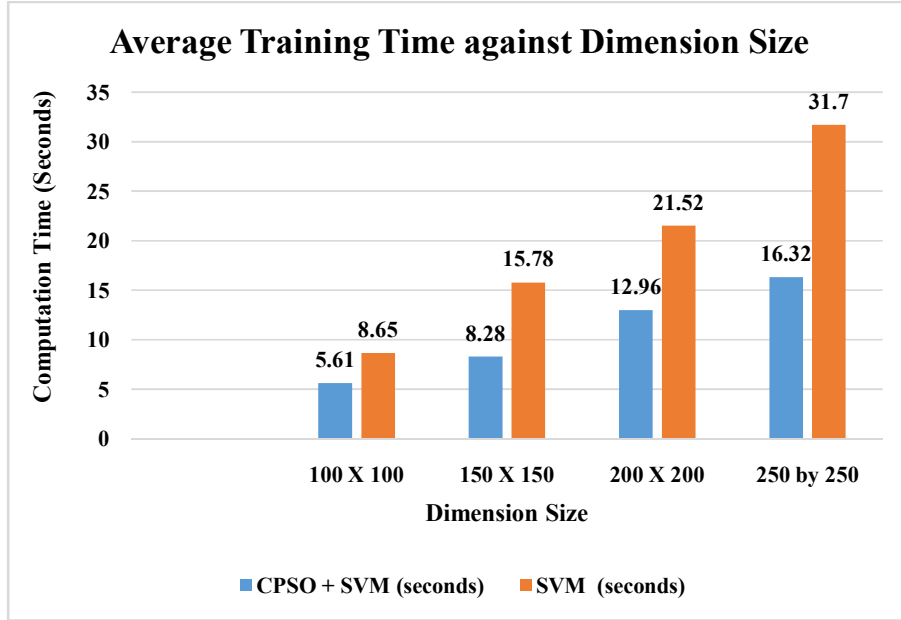

Figure 4: Shows the graph of average training time against the dimension size

Zhou, Bai, Tian & Zhang (2008) stated that CPSO will achieve faster convergence speed and better precision when applied to optimize the parameters of SVM. The results presented above establishes the fact that the application of CPSO achieved a faster convergence speed, better precision and improves the accuracy of SVM.

Table 2: Recognition Results based on performance metrics

Table

Type	Method	FPR (%)	Precision (%)	Accuracy (%)	Recognition Time (sec)
Binary Classification	CPSO-SVM	0.00	100	90.77	6.66
	SVM	7.69	88.00	89.23	23.26
Multiclass Classification	CPSO-SVM	0.00	100	92.31	21.20
	SVM	4.55	89.47	90.77	27.71

III

presents the recognition accuracy for each expression at threshold value of 0.8 and 250 x250 pixel resolution for both CPSO-SVM and SVM model. The result obtainable from table shows that the CPSO-SVM model have an average recognition accuracy of 92.31% while the SVM model has 89.23%.

Table III: Facial Expression Recognition rate

Expression	CPSO + SVM (%)	SVM (%)
Anger	92.31	87.69
Disgust	89.23	86.15
Fear	90.77	89.23

Happiness	95.38	92.31
Sadness	92.31	89.23
Surprise	93.85	90.77
Average	**92.31**	**89.23**

A t-test values was measured between the facial expression recognition rate of CPSO-SVM and SVM for each expression. The paired t-test analysis conducted between CPSO-SVM and SVM reveals that there no much distinction in the test result with mean difference of 3.0783 (i.e. $\mu = 3.0783$). Nevertheless, the result confirmed that the developed technique CPSO-SVM is statistically significant at $P < 0.05; P = 0.01$ with $t\ value = 7.742$. The t-test result further validates the fact the CPSO-SVM outperformed the SVM.

CONCLUSION

The experimental results demonstrate that the developed technique (CPSO-SVM) outperforms the existing SVM model in terms of recognition accuracy, precision, false positive rate, training time and recognition time. Also, the statistical analysis conducted on the recognition accuracy per expression further presumes that the proposed model is statistically significant. These reveal that the developed technique can construct efficient and realistic facial expression feature to realise good accuracy, computation efficiency, and robustness. Therefore, it would produce a more reliable security surveillance system in any security prone organization.

REFERENCES

Abdulameer M. H., Abdullah H. S. & Othman A. Z. (2014). Support Vector Machine Based on Adaptive Acceleration Particle Swarm Optimization. *Hindawi Publishing Corporation, the Scientific World Journal,* 1-8.

Adeyanju, I. A., Omidiora, E. O., & Oyedokun, O. F. (2015). Performance evaluation of different support vector machine kernels for face emotion recognition. In *SAI Intelligent Systems Conference (IntelliSys)IEEE,* 804-806.

Aluko J. O., Omidiora E. O., Adetunji A. B. & Odeniyi O. A. (2015). Performance Evaluation of Selected Principal Component Analysis-Based Techniques For Face Image Recognition. *International Journal of Scientific & Technology Research*, 4 (1), 1-7.

Cristianini, N. & Shawe-Taylor, J. (2000). An Introduction to Support Vector Machines. Cambridge: Cambridge University Press.

Eberhart R. C., and Kennedy J. (1995). A new optimizer using particle swarm theory. In Proceedings of the 6th International Symposium on Micromachine Human Science, 39–43.

Geetha, A., Ramalingam V., Palanivel S. (2009). Facial expression recognition, A real-time approach. *Expert Systems with Applications*, 36(1), 303-308.

Ghimire, D., Jeong, S., Lee, J., & Park, S. H. (2016). Facial expression recognition based on local region specific features and support vector machines. *Multimedia Tools and Applications*, 1-19.

Jin, X., & Reynold, R. G. (1999). Using Knowledge-Based System with Hierarchical Architecture to Guide the Search of Evolutionary Computation, In: Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence, 29–36.

Lemley, J., Abdul-Wahid, S., Banik, D., & Andonie, R. (2016). Comparison of Recent Machine Learning Techniques for Gender Recognition from Facial Images. 27th *Modern Artificial Intelligence and Cognitive Science* (*MAICS*) *conference 2016*. 97–102.

Manekar and Waghmare (2014). Improving Accuracy of SVM Using Hybrid Cultural Algorithm. *International Journal Computer Technology & Applications (IJCTA)* 5(3), 1194-1197.

Priyanka T. M, Kesari V., Ligendra K. V., & Nazil P. (2013). Facial Expression Recognition Using Data Mining Algorithm, *Journal of Economics, Business and Management,* 1(4), 343-346.

Qayyum, H., Majid, M., Anwar, S. M., & Khan, B. (2017). Facial Expression Recognition Using Stationary Wavelet Transform Features. *Mathematical Problems in Engineering*, *2017*.

Samad, Rosdiyana, & Hideyuki Sawada (2011). Edge based Facial Feature Extraction Using Gabor Wavelet and Convolution Filters. In *MVA*, 430-433.

101

Susskind, J.M., Littlewort G., Bartlett M. S., Movellan J., Anderson A. K. (2007). "Human and computer recognition of facial expressions of emotion", *Neuropsychologia,* 45, 152-162.

Tang, M., & Chen, F. (2013). Facial expression recognition and its application based on curvelet transform and PSO-SVM. *Optik-International Journal for Light and Electron Optics*, *124*(22), 5401-5406.

Windisch, A., Wappler, S., & Wegener, J. (2007). Applying particle swarm optimization to software testing. Proceedings of the 9th annual conference on Genetic and evolutionary computation, pp. 1121-1128.

Yan, X., Wu, Q., Zhang, C., Chen, W., Luo, W., & Li, W. (2012). An Efficient Function Optimization Algorithm based on Culture Evolution. *International Journal of Computer Science Issues*, *9*(5), 11-18.

Zhou J., Bai T., Tian J. & Zhang A. (2008). The study of SVM optimized by Cultural Particle Swarm Optimization on Predicting Financial Distress. *IEEE conference 2008.*