IOSR Journal of Dental and Medical Sciences (IOSR-JDMS)
e-ISSN: 2279-0853, p-ISSN: 2279-0861.Volume 17, Issue 4 Ver. 4 (April. 2018), PP 35-44
www.iosrjournals.org

Reducing the Computational Cost of SVM in Face Recognition Application Using Hybrid Cultural Algorithm

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

[1]*(Departmentof Computer Science and Engineering, Faculty of Engineering & TechnologyLadoke Akintola University of Technology (LAUTECH), Nigeria)*

Abstract: *The computational cost in terms of computation time and memory utilization of various techniques in the development of face recognition applications is a major challenge. The fundamental properties of high dimension exhibited by face images leads to superfluous information that instigates computational burden in term of processing speed and memory usage.Support Vector Machine despite being a wonderful face recognition technique due its generalization capability and high theoretical background of classification accuracy consumes large amount of time and memory: a major setback in its implementation in real world applications. Consequently, a hybrid cultural algorithm is proposed to reduce the computational cost of SVM in face recognition applications. Cultural Algorithm optimises the parameter of SVM to lessen the computation requirement. The experimental results reveal that the proposed technique does not only improves the efficiency of SVM but also makes it less computationally expensive in terms of both computation time and memory utilization.*

Keywords –*Computation Cost, Cultural Algorithm, Culture Particle Swamp Optimization, Feature Selection, Support Vector Machine*

I. Introduction

Face recognition is an effective research area which has provoked the interest of researchers from psychology, neuroscience and image processing to computer vision. Furthermore, it is also regarded as an important biometric technique that identify people by “who they are” and not by “what they have” or “what they know” [1]. Similarly,face recognition is referred to as a machine learning process where the high dimensional features of subject are received to provide the identity of subject[31]. It is an active area mainly due to increasing security demands and its potential commercial and law enforcement applications. The last decade has shown dramatic progress in this area, with emphasis on such applications as human-computer interaction (HCI), biometric analysis, content-based coding of images and videos, and surveillance [2]. Computational models of face recognition are interesting because they can contribute not only to theoretical knowledge but also to practical applications [29]. The rapid development of face recognition systems is due to a combination of factors: (1) Active development of algorithms, (2) The availability of a large databases of facial images, and (3) A method for evaluating the performance of face recognition algorithms [5].

Face recognition is a pattern recognition problem with high space and time complexity due to the fact that face images have an inherent property of high dimension; pixels are highly correlated, leading to redundant information which causes computational burden in terms of processing speed and memory utilization [4]. Several techniques have been developed and adopted with the aim of developing a robust and a computationally efficient face recognition system. Consequently, these techniques are not only computationally expensive but require a high degree of correlation between the test and training images, and do not perform effectively under large variations in pose, scale and illumination, etc. [3].

There is always issues with computational cost most especially when the database contains a large number of images. If it took too much time to recognize face when database is large, it is discouraged from further development. Due to high complexity and computation cost it is hard to apply most of these technologies in a real-time face recognition system. Most of the prevailing methods have high computational complexity and the re-rendering image has poor quality. So, there are many issues to be solved before it can be applied to commercial or law enforcement applications [6].

Support Vector Machine (SVM) by the virtue of its strong theoretical foundations and a good generalization capability performed effectively in face recognition applications. SVM is a machine learning tool that is based on the idea of large margin data classification. Standard implementations, though provide good classification accuracy, are slow and do not scale well. Hence, they cannot be applied to large-scale data mining applications that involves images. They typically need large number of support vectors. Therefore, the training as well as the classification times are high [7]. Furthermore, SVM can also be viewed as a way to train polynomial neural networks or Radial Basis function classifiers [6]. Training the SVM involves solving a quadratic optimization problem which requires the use of optimization routines from numerical libraries. This step is computationally intensive, can be subject to stability problems and is non-trivial to implement [8].

Cultural algorithms (CAs) are a class of evolutionary algorithms (EAs) that were developed based on the concept of cultural evolution [9]. The use of knowledge by CA makes it to have an advantage of hybrid approach to problem solving, fast convergence speed and the global optimizing ability than other evolutionary algorithms. Since Cultural Algorithm supports hybrid approach to problem solving; this paper proposes a hybrid cultural algorithm involving Particle Swamp Optimization(PSO) i.e. Culture Particle Swamp Optimization (CPSO) to reduce the computational cost of SVM.

II. Literature Survey

2.1 Cultural Algorithms

Cultural Algorithms are computational models of cultural evolution. It provides a framework to accumulate and communicate knowledge so as to allow self-adaptation in an evolving model and also encompasses the scales of the other approaches so should be able to solve their types of problems as well. The cultural algorithm components consist of a belief space and a population space which interact through a communication protocol [12]. The population space provides a fundamental base in which individuals reside and interact and also could accommodate any population-based evolutionary computation model such as Genetic Algorithms, Evolutionary Programming, Genetic Programming, Differential Evolution, Immune Systems [10]. The belief space is a data vault where the individual can keep their experience for alternate individual to learn them by implication. Adding a central knowledge (belief space) to any search evolutionary Algorithm like Evolutionary Programming (EP), Genetic Programming (GP) etc. becomes a Cultural Algorithm. Figure 1 depict the framework of Cultural Algorithm. Introducing PSO to search the population space of Culture Algorithm form a hybrid Cultural Algorithm named Culture Particle Swamp Optimization (CPSO). This combine the fast convergence speed of PSO with the global optimizing ability of CA.

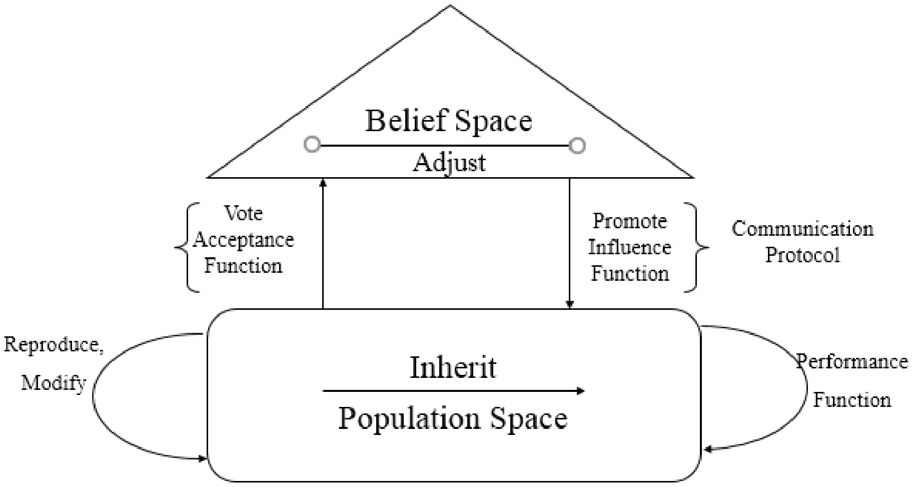


Figure 1: Framework of Culture Algorithm

2.2 Support Vector Machine

Support Vector Machines (SVM) are classification and regression methods which have been derived from statistical learning theory [14]. The concept is based on optimal linear separating hyperplane that is fitted to the training patterns of two classes within a multi-dimensional feature space. The optimization problem that has to be solved relies on structural risk minimization and is aiming at a maximization of the margins between the hyperplane and closest training samples. Support vector machine method classifies both linear as well as non-linear data. It transforms the data into higher dimension. The SVM finds hyperplane using support vectors and margin define by support vector. The data transform into dimension equals to the number of attribute in data. Hyperplane with maximum margin is classifying the data with high accuracy. There is high classification accuracy of support vector machine [11].

Given a training set of instance-label pairs $(x_i, y_i), i = 1,\ 2,\ .\ .\ ., l$, where $x_i \in R^n$ and $\in y\ \in \{1, -1\}^l$, the support vector machines (SVM) ([13][14]) require the solution of the following optimization problem [15]:

$$\min_{w,b,\xi} \frac{1}{2}\boldsymbol{w^T w} \ + C\sum_{i=1}^{l} \xi_i$$
$$Subject\ to \quad y_i(\boldsymbol{w^T}\phi(x_i) + b) \geq 1 - \xi_i$$

$\xi_i \geq 0.$ (1)

From above the training vectors x_i are mapped into a higher dimensional space by the function ϕ. SVM finds a linear separating hyperplane with the maximal margin in this higher dimensional space. $C > 0$ is the penalty parameter of the error term. Moreover, $K(x_i, x_j) \equiv \phi(x_i)^T\phi(x_j)$ is called the kernel function. Despite the fact that new kernels are being proposed by new researchers, SVM uses for basic kernels [15]:

- ✓ Linear : $K(x_i, x_j) = x_i^T x_j$ (2)
- ✓ Polynomial: $K(x_i, x_j) = \left(\gamma x_i^T x_j + r\right)^d, \gamma > 0.$ (3)
- ✓ Radial Basis Function (RBF) : $K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right), \gamma > 0.$ (4)
- ✓ Sigmoid: $K(x_i, x_j) = tanh\left(\gamma x_i^T x_j + r\right)$ (5)

Where $\gamma, r,$ and d are kernel parameters.

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 [16]. Previous research work from [18] [17] showed optimal performance with the polynomial kernel, which lead to this experiment adjusting the degree of the polynomial to try to increase accuracy. For the binary classification case, the optimal hyperplane was a line, independent of the polynomial degree.

The biggest difficulties in setting up the SVM model are choosing the kernel function and its parameter values. If the parameter values are not set properly, then the classification outcomes will be less than optimal [19]. In complex classification domains, some features may contain false correlations, which impede data processing. Moreover, some features may be redundant, since the information that they add is contained in other features. Redundant features can lengthen the computational time, influencing the classification accuracy. Hence, the classification process must be fast and accurate using the minimum number of features, which is a goal attainable through the use of feature selection. Feature selection has been applied to enhance classification performance, and to reduce data noise [20–22].

If the SVM is adopted without feature selection, then the dimension of the input space is large and non-clean, lowering the performance of the SVM. Thus, the SVM requires an efficient and robust feature selection method that discards noisy, irrelevant and redundant data, while still retaining the discriminating power of the data. Features extracted from the original data are adopted as inputs to the classifiers in the SVM [28].

2.3 Related Works

Getta *et al.* [23] 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.

Abdulaeer et al. [26] proposed a novel technique based on adaptive accelerated particle swarm optimization (AAPSO) to optimise the parameter of SVM in both face and iris recognition application. The proposed technique is aimed at overcoming the computational cost in terms of processing speed and memory utilization. The performance of the system was evaluated against the existing PSO-SVM technique. The result obtained shows that shows that the proposed technique (AAPSO-SVM) outperforms the existing PSO-SVM. Furthermore, the result of the t-test values for the recognition accuracy measured between AAPSO-SVM and PSO-SVM face recognition techniques shows that the proposed technique (AAPSO-SVM) is statistically significant and with the t-test result P< 0.05 at P = 0.0265. The AAPSO uses less computational time to perform the optimization process compared with the conventional PSO.

Xiao-ming [24] applied magnitude of Gabor and PCA on ORL database to extract facial features and SVM for classification. The SVM technique reported recognition rate of 99.5% i.e. 0.005 error rate. Similarly, Li Xianwei [25] achieved a recognition rate of 85% i.e. 0.25 error rate on the ORL database with the application of PCA for feature extraction and SVM for the classification of the features. Also, Amjath Fareeth Basha [27] proposed a face recognition system based One-Dimensional (1D) Continuous wavelet transform (CWT) and SVM. The proposed technique achieved percentage recognition accuracy of 98 % on the ORL database. Despite the good classification performance SVM in [24] [25] [27] the major drawback in the application of the above techniques on ORL data is the fact they it computationally expensive in terms of both training and recognition time as well as memory utilization.

III. Methodology

The steps in the development of the technique in this study include; image acquisition, image pre-processing, feature extraction, feature selection and parameter optimization, and feature Classification. The scheme of the proposed techniques is shown in Figure 2 below.

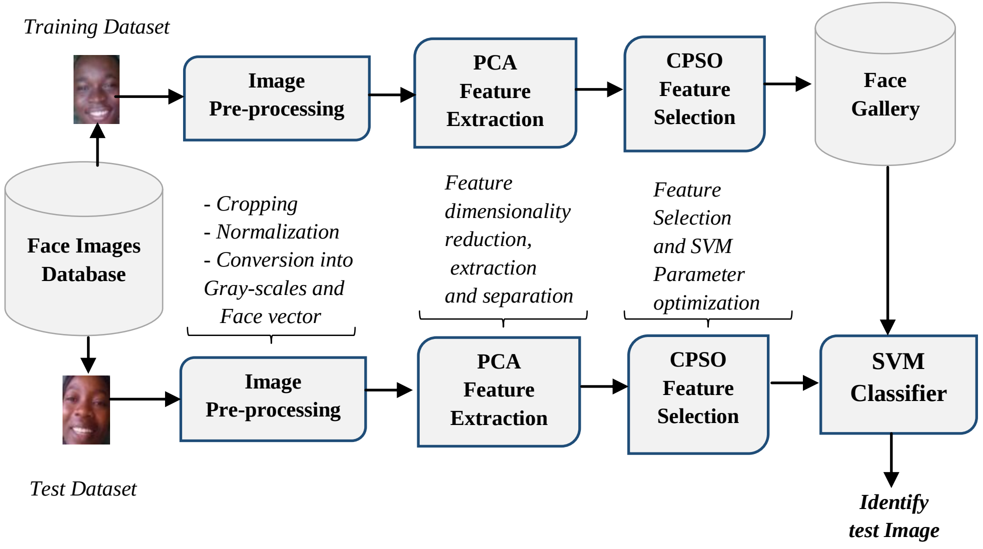
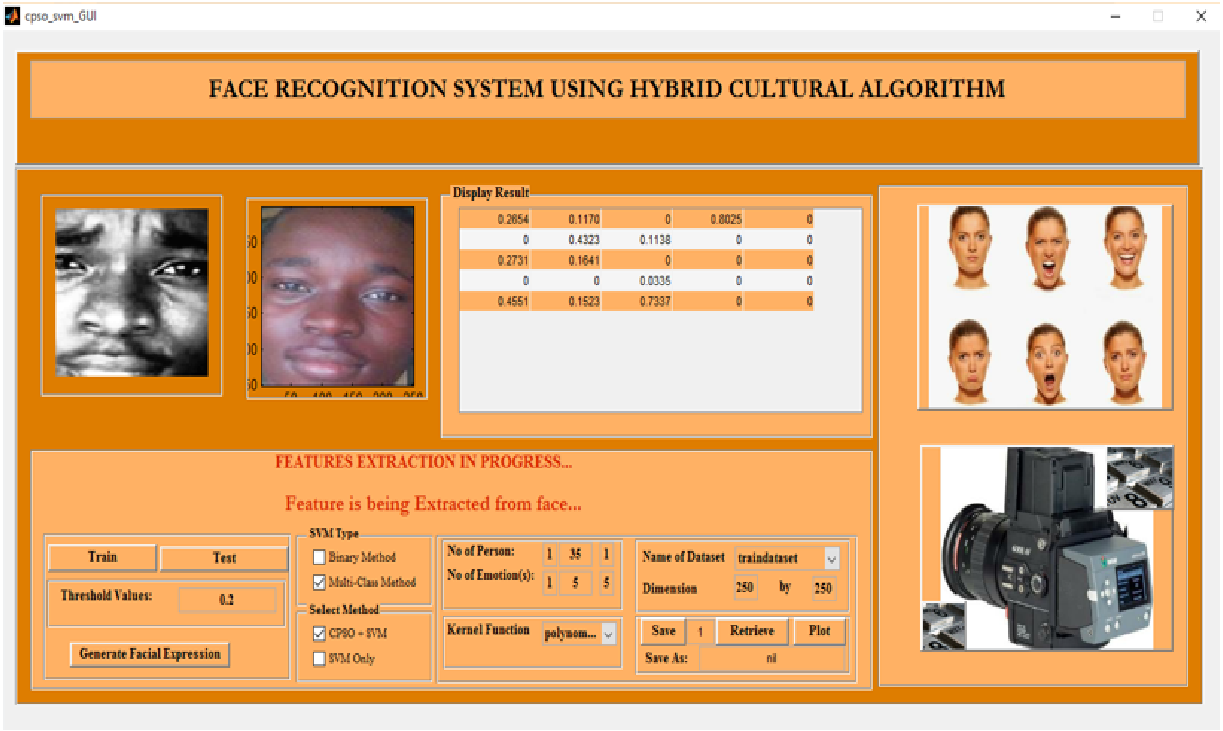


Figure 2: Scheme of the Proposed Techniques

Figure 3 depict an interactive Graphic User Interface (GUI) application developed with a real-time database using 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 performance of the techniques on trained and recognized faces was measured against computation time (training and recognition) and memory utilization.



FACE RECOGNITION SYSTEM USING HYBRID CULTURAL ALGORITHM

Display Result

0.2654	0.1170	0	0.8025	0
0	0.4323	0.1138	0	0
0.2731	0.1641	0	0	0
0	0	0.0335	0	0
0.4551	0.1523	0.7337	0	0

FEATURES EXTRACTION IN PROGRESS...

Feature is being Extracted from face...

Threshold Values: 0.2

SVM Type
□ Binary Method
☑ Multi-Class Method

Select Method
☑ CPSO + SVM
□ SVM Only

No of Person: 1 35 1
No of Emotion(s): 1 5 5

Kernel Function polynom...

Name of Dataset traindataset
Dimension 250 by 250

Save 1
Save As: nil

Figure 3: Graphic User Interface (GUI) Application

3.1 Image Acquisition

The images used in this study involves a black African Local Database (LOCDAT). This database has 240 facial expression images taken with a Canon digital camera of default size 1200 x 1200. The face images were downsized into a 100 x 100, 150 x 150, 200 x 200 and 250 x 250 pixel. The training phase used One hundred and seventy-five (175) of those images while the testing phase used Sixty-five (65) of the images.

3.2 Image Pre-Processing

The acquired images were pre-processed by cropping the region of interest, conversion of the coloured image to grayscale i.e. from 3D to 2D for time and space management and normalized using histogram equalization for enhancement.

3.3 Feature Extraction

Principal Component Analysis (PCA) is a statistical technique that can be utilized for prediction, removal of redundancy, data compression, feature extraction, image recognition etc [30]. PCA serves as a feature extraction component employed in this study to extract facial features that are critical for classification. It was applied to achieve feature dimensionality reduction to form Eigen faces and face vector from the pre-processed image.

3.4 Feature Selection

CPSO was applied to select optimal feature sub set from the entire features extracted by PCA to further reduce its dimension. The application of CPSO optimises the parameter of SVM to increase it convergence speed and minimise its error. The algorithm steps for CPSO is described step-by-step as follows [32].

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 $\boldsymbol{P_i}$

The fitness value of each of the particles was computed using (6); to evaluate the performance of each particle.

$$F = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\emptyset_i - \overline{\emptyset}_i\right)^2} \qquad (6)$$

Where $\emptyset_i$represents the *i*th model output; $\overline{\emptyset}_i$represents the *i*th desired output, and *N* represents the number of input data.

Step 4: Determine and update the current local best position $\boldsymbol{L_{Best}}$and global best position $\boldsymbol{G_{Best}}$ using equation (7)

$$L_{Best}(t+1) = \begin{cases} P_i(t) \ , & \text{if } F\big(P_i(t)\big) < F\big(L_{Best}(t)\big) \\ L_{Best}(t), & \text{if } F\big(P_i(t)\big) \geq F\big(L_{Best}(t)\big) \end{cases}$$

$$G_{Best}(t+1) = \arg\min_{L_{Best}} F\big(L_{Best}(t+1)\big),\ 1 \leq i \leq I. \qquad (7)$$

Step 5: Apply the acceptance function andadjust the belief space $\boldsymbol{B_s}$.

Equation (8) determines the number of particles that will be used to adjust the belief space while equation (9) determines the interval of the belief space.

$$N_{accepted} = n\% \times I + \frac{n\%}{t} \times I \qquad (8)$$

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

$$IB_s = \left[l_w, u_p\right] = \left\{p | l_w \leq p \leq u_p, p \in 3i\right\} \qquad (9)$$

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 (10):

$$l_w = \begin{cases} P_i \text{if} p_i \leq l_w \\ l_w, \quad \text{otherwise} \end{cases} u_p = \begin{cases} P_{i,} \text{if} P_i \geq u_p \\ u_p, \quad \text{otherwise} \end{cases} \qquad (10)$$

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 (11) 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 (12) and (13) to generate new particle swarm.

$$P_i(t) = \begin{cases} P_i(t) \ + \left|\mathrm{R}() \ \times \left(u_p - l_w\right)\right| \mathrm{if} P_i < l_w \\ P_i(t) \ - \left|\mathrm{R}() \ \times \left(u_p - l_w\right)\right| \mathrm{if} P_i > u_p \end{cases} \qquad (11)$$

$$V_i(t+1) = w \times \ V_i(t) + c_1 \times \mathrm{R}() \times [L_{Best}\,(t+1) - P_i(t)] + c_2 \times \mathrm{R}() \times \ [G_{Best}\,(t+1) - P_i(t)](12)$$

$$P_i(t+1) = P_i(t) + V_i(t+1) \qquad (13)$$

Where $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$ 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.

3.5 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 (14).

$$\min \ \frac{1}{2}\|P_i\|^2 + C\sum_{i=1}^{N}\xi_i \qquad Such\ that \ \sum_{i=1}^{N} P_i x_i \ \geq \left(\frac{1 - \ \xi_i}{y_i}\right) - b$$

$$i = 1,\ 2,\ .\ .\ .,N,\ \ \xi_i \geq 0,\ \ i = 1\ ,\ 2,\ .\ .\ .\ ,\ N, \qquad (14)$$

Equation (15) was applied to obtain the final classification of each case:

$$y_i = arg\ max_{k(1.\ .\ .k)}(P_i^T y_i(x_i) + \ b_i) \qquad (15)$$

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.

IV. Results and Discussion

Table I below shows the average training time for the proposed technique after four trials for each dimension sized. The results obtained reveals that the computation time spent increases as the dimension size of the images increases, which implies that the time consumed depends on the features in the training set. The result shows that the CPSO-SVM model is less computationally expensive in terms of training time compared to the SVM model.

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

Dimension Size	Technique	Time1(s)	Time2(s)	Time3(s)	Time4(s)	Average Time (seconds)
100 by 100	CPSO-SVM	5.68	5.63	5.57	5.55	**5.61**
	SVM	8.72	8.68	8.61	8.59	**8.65**
150 by 150	CPSO-SVM	8.42	8.33	8.23	8.13	**8.28**
	SVM	15.92	15.84	15.73	15.63	**15.78**
200 by 200	CPSO-SVM	12.98	12.97	12.96	12.94	**12.96**
	SVM	21.88	21.61	21.42	21.16	**21.52**
250 by 250	CPSO-SVM	16.35	16.34	16.32	16.28	**16.32**
	SVM	32.16	32.11	31.53	30.99	**31.70**

Figure 3 shows the graph of average training time against the dimension size. Regression analysis shows that there is high correlation between the computation time and the dimension size. The relationship between the average training time (T_t) and the dimension size (dm) is found to be linear with a high correlation coefficient for both CPSO-SVM model and existing SVM model as shown in equation (16) and (17) respectively.

$$T_t = 0.0002dm + 3.7663 \qquad R^2 = 0.98 \qquad (16)$$

$$T_t = 0.0004dm + 5.0367 \qquad R^2 = 0.99 \qquad (17)$$

Table II shows the computation (Recognition) time obtained at 250 by 250- pixel resolution with respect to the threshold values of 0.2, 0.4, 0.6 and 0.8 for both binary and multiclass SVM classification scheme. The result obtained also reveals that the proposed technique is less computationally expensive in terms of recognition time in both classification scheme.

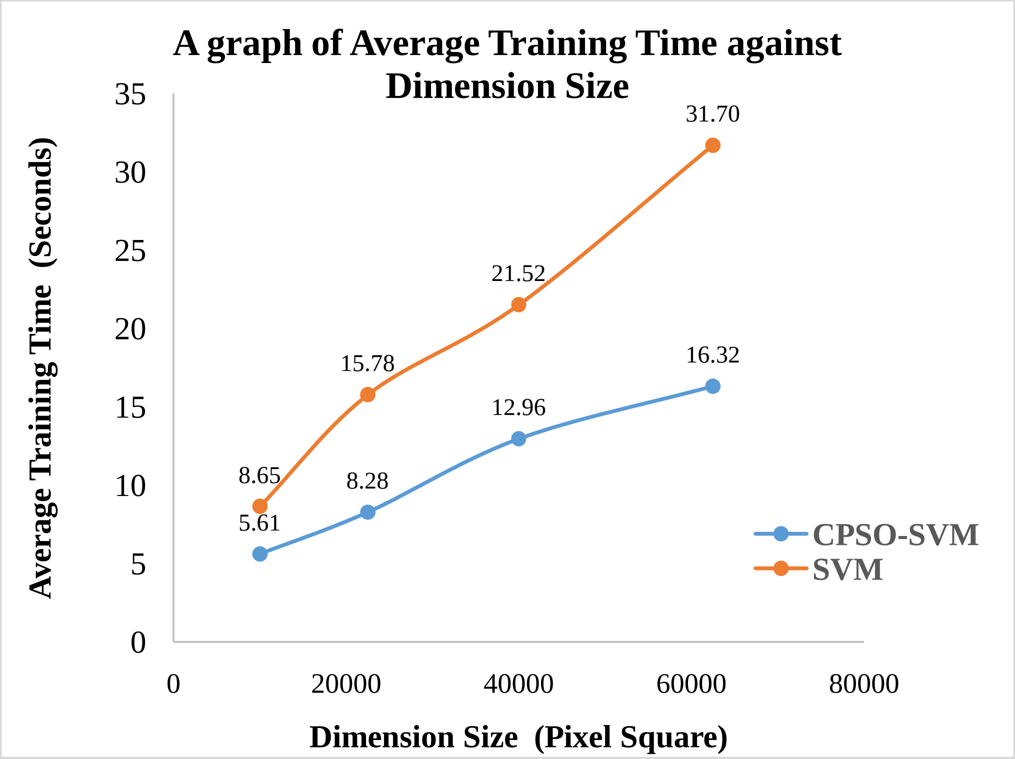


Figure 3: A graph showing the relationship between Average training time (seconds) and Dimension size (pixel square).

Table II: Average Recognition Time at 250 by 250-pixel Resolution

250 by 250	Binary Classification Scheme		Multi-Class Classification Scheme	
Threshold value	**CPSO-SVM**	**SVM**	**CPSO-SVM**	**SVM**
0.2	33.28	42.46	16.94	48.50
0.4	17.06	28.67	21.81	33.49
0.6	15.52	44.64	41.38	51.24
0.8	6.66	23.26	21.20	27.71
Average Time (seconds)	**18.13**	**34.76**	**25.33**	**40.24**

Figure 4 and 5 shows the graphs which depicts the relationship between the average recognition time and the threshold values for both binary and multiclass classification schemes respectively.

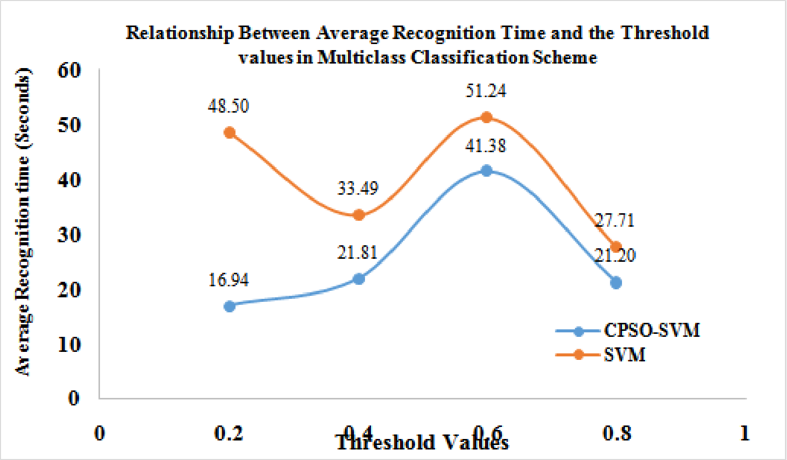


Figure 4: A graph showing the relationship between average recognition time and threshold values in SVM multiclass classification scheme.

From the graph in Figure 4; the regression analysis reveals that the relationship between the average recognition time (T_R) and the threshold values (*th*) is found to be polynomial of the third order with a high correlation coefficient for both CPSO-SVM model and SVM model multiclass classification scheme as shown in equation 18 and 19 respectively.

$$T_R = -1134.2th^3 + 1544.7th^2 - 584.83th + 81.189 \qquad R^2 = 0.96 \qquad (18)$$

$$T_R = -1542th^3 + 2259.8th^2 - 999.14th + 170.27 \qquad R^2 = 0.93 \qquad (19)$$

Similarly, the graph in Figure 4.3 shows the relationship between the average recognition time (T_R) and the threshold values (*th*) and is also found to be polynomial of the third order with a high correlation coefficient for both CPSO-SVM model and SVM model binary classification scheme as shown in equation 20 and 21 respectively.

$$T_R = -458.51th^3 + 733.8th^2 - 393.02th + 86.205 \qquad R^2 = 0.99 \quad (20)$$

$$T_R = -1397.8th^3 + 2049.2th^2 - 907.07th + 153.09 \qquad R^2 = 0.95 \quad (21)$$

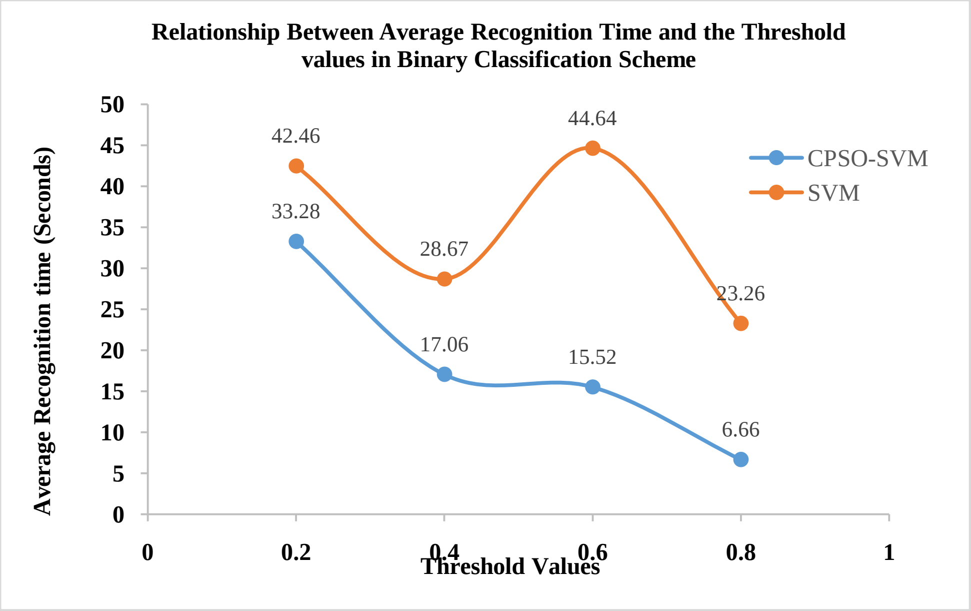


Figure 5: A graph showing the relationship between Average recognition time and threshold values in SVM binary classification scheme

It could be inferred from the result obtain that the SVM based on the kernel function usedin both cases has a time complexity of $O(n^3)$. Therefore, by the virtue of the proposed technique (i.e. CPSO-SVM) the computational burden of SVM is reduced. A t-test value was measured between the average recognition time of CPSO-SVM and SVM obtained for both classification schemes. The paired t-test analysis conducted reveals that CPSO-SVM was statistically significant at $P < 0.05; P = 0.002$ with $Mean\ difference = -15.765, df = 7\ and\ t\ value = -4.714$. The mean difference and t-value being negative assert the fact the application of CPSO along with SVM is efficient to reduce the computational cost of SVM.

Table III: Memory Utilization at 250 by 250-pixel Resolution

Algorithm	Used (MB)/ 4096 MB	Available (MB)/4096MB	% Physical Memory Used
CPSO-SVM	**2872**	**1143**	**70**
SVM	**3551**	**424**	**87**

Table III shows the memory utilization of the CPSO-SVM and SVM. The memory utilization of developed technique was monitored using the window resource monitor based on the amount of image dataset trained.The experiment conducted also shows that CPSO-SVM averagely used 2872MB which is 70% of the physical memory while SVM averagely used 3551MB which is 87% of the physical memory. Therefore, the computational cost of SVM in terms of memory utilization is greatly reduced averagely by a difference of 17%.

V. Conclusion

The experimental results obtained using the proposed method helps in the reduction of both the storage and computation time cost of SVM in face recognition application. Therefore, the proposed technique is less computationally expensive in terms of computation times and memory utilization. It can be adopted to recognise human face in security surveillance systems or other related systems. It is recommended that other evolutionary search algorithm such as Ant Colony Optimization (ACO), Evolutionary Programming (EP), Genetic Programming (GP), Differential Evolution (DE), Artificial Immune Systems (AIS), should be introduced into the model of cultural algorithm instead of PSO to determine its performance on SVM. Also, the hybrid cultural algorithm should be hybridized with SVM to determine its computational efficiency on face recognition systems.

References

[1]. E. O. Omidiora, A. O. Fakolujo, R. O. Ayeni and I. A. Adeyanju, Optimised Fisher Discriminant Analysis for Recognition of Faces Having Black Features. Journal of Engineering and Applied Sciences, 3 (7), 2008, 524-531.

[2]. A. S. Abdallah, A. L. Abbott and M. A. El-Nasr, A new face detection technique using 2D DCT and self-organizing feature map. In Proc. of World Academy of Science, Engineering and Technology, 21, 2007, 15-19.

[3]. C. Beumier and M. Acheroy, Automatic Face Recognition, in Proceedings symposium IMAGING. Eindhoven, The Netherlands, 2000, 77-89.

[4]. R. S. Babatunde, S. O. Olabiyisi, E. O. Omidiora and R. A. Ganiyu. Local Binary Pattern and Ant Colony Optimization Based Feature Dimensionality Reduction Technique for Face Recognition Systems. British Journal of Mathematics & Computer Science, 11(5), 2015, 1-11.

[5]. A. S. Tolba, A. H. El-Baz, and A. A. El-Harby. Face recognition: A literature review. International Journal of Signal Processing, 2(2), 2006, 88-103.

[6]. L. Wei-Yang, Face Recognition: A Literature Survey. (ECE 738 Course Project, 2013)

[7]. S. N. Jayenthi, Efficient Classification Algorithms using SVMs for Large Datasets. Supercomputer Education and Research Center, 2007.

[8]. J. C. Platt, 12 fast training of support vector machines using sequential minimal optimization. Advances in kernel methods, 1999, 185-208.

[9]. A. Keramati, M. Hosseini, M. Darzi, and A. Liaei, (2011, October). Cultural algorithm for feature selection. In IEEE: Data Mining and Intelligent Information Technology Applications (ICMiA), 2011, 71-76.

[10]. X. Jin, and R. G. Reynold, 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, 1999, 29–36.

[11]. J. Cervantes, L XiaoOu and Y. Wen, SVM Classification for Large Data Sets by Considering Models of Classes Distribution, Artificial Intelligence - Special Session, Sixth Mexican International Conference; MICAI 2007.

[12]. R. G. Reynolds, An Overview of Cultural Algorithms, Advances in Evolutionary Computation, McGraw Hill Press, 1999.

[13]. B. E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, ACM Press, 1992, 144-152.

[14]. V. Vapnik. The Nature of Statistical Learning Theory. (Springer-Verlag, New York, 1995)

[15]. H. Chih-Wei, C. Chih-Chung, and L. Chih-Jen (2003). A Practical Guide to Support Vector Classification. National Taiwan University, Taipei 106, Taiwan http://www.csie.ntu.edu.tw/~cjlin. (Last updated: May 19, 2016)

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

[17]. S. S. Keerthi, C. J. Lin, Asymptotic behaviors of support vector machines with Gaussian kernel, Neural Computing. 15, 2003, 1667–1689.

[18]. T. Joachims. Text categorization with support vector machines: learning with many relevant features. In Claire N´edellec and C´eline Rouveirol, editors, Proceedings of ECML-98, 10th European Conference on Machine Learning, Springer Verlag, Heidelberg, DE, 1998, 137–142.

[19]. K. Morik, P. Brockhausen, and T. Joachims. Combining statistical learning with a knowledge-based approach - a case study in intensive care monitoring. In Proc. 16th International Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA, 1999, 268–277.

[20]. K. Lee, V. Estivill-Castro, Feature extraction and gating techniques for ultrasonic shaft signal classification, Appl. Soft Comput. 7, 2007, 156–165.

[21]. G. Valentini, M. Muselli, F. Ruffino, Cancer recognition with bagged ensembles of support vector machines, Neurocomputing 56, 2004, 461–466.

[22]. J. Wang, X. Wu, C. Zhang, Support vector machines based on k-means clustering for real-time business intelligence systems, Int. J. Business Intell. Data Mining 1, 2005, 54–64.

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

[24]. X Wang, C. Huang, G. Ni and J. Liu, Face Recognition Based on Face Gabor Image and SVM, in 2nd International Congress Image and Signal Processing, 2009, 1 - 4.

[25]. L. Xianwei, C. Guolong, Face Recognition Based on PCA and SVM, In IEEE conference Photonics and Optoelectronics, 2012, pp.1 - 4.

[26]. M. H. Abdulameer, H. S. Abdullah and A. Z. Othman. Support Vector Machine Based on Adaptive Acceleration Particle Swarm Optimization. Hindawi Publishing Corporation, the Scientific World Journal, 2014, 1-8.

[27]. A. F. Basha and G. S. Jahangeer, Face Gender Image Classification Using Various Wavelet Transform and Support Vector Machine with various Kernels, International Journal of Computer Science Issues, 9(2), 2012, 150-157.

[28]. L. Shih-Wei, Z. Lee, S. Chen and T. Tseng. Parameter determination of support vector machine and feature selection using simulated annealing approach. In: Applied Soft Computing 8, 2008, 1505–1512.

[29]. E. Puja and R. Rajput (2016)"Feature Extraction in Face Recognition using SVM, Skin Fusion and Edge Detection Technique" International Journal of Innovative Research in Computer and Communication Engineering 4(10), 2016, 18889- 18898.

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

[31]. A. D. Gadekar, S. S. Suresh.Face Recognition Using SIFT- PCA Feature Extraction and SVM Classifier. IOSR Journal of VLSI and Signal Processing (IOSR-JVSP), 5(2), 2015, 31-35.

[32]. A. B. Adetunji, J. P. Oguntoye, O. D. Fenwa and E. O. Omidiora. Facial Expression Recognition Based On Cultural Particle Swamp Optimization and Support Vector Machine.LAUTECH Journal of Engineering and Technology, Ogbomosho, 2017.

A. B. Adetunji "Reducing the Computational Cost of SVM in Face Recognition Application Using Hybrid Cultural Algorithm."IOSR Journal of Dental and Medical Sciences (IOSR-JDMS), vol. 17, no. 4, 2018, pp 35-44.