JSMO
DYPCOE

JOURNAL OF SIMULATION MODELING & OPTIMIZATION

Available online at www.jsmo.co.in

A Parallel Algorithm for Recognition of Facial Expressions Using Multi-class SVM Classifier

***Narendra Kumar Rout*[a]*, Santosh Kumar Moharana*[b] *, Saswati Sahoo*[c]*, Elina Pattnaik*[d]**

[a,b,c,d]*Department of Computer Science and Engineering, Gandhi Engineering College, Bhubaneswar.*

ABSTRACT

This paper presents a novel facial expression recognition system for detection and classification of the seven basic facial expressions from nearly frontal face images. Facial expressions are identified by observing the changes in one or more distinct features such as opening and closing of eyes, raising the eyebrows, tightening the lips or certain combinations of them. In this work, we have used eleven feature points that represent and identify through monitoring the movement of eyes, eyebrows and lips as well as provide measurements of the distinct features responsible for each of the seven basic human facial expressions. By measuring the degree of displacement of these eleven feature points from a non-changeable rigid point, a Feature vector composed of eleven features is obtained. The obtained feature vectors are used for training Multi Class SVM classifier, so that it can classify facial expressions when given to it in the form of a feature vector. In this work, a parallel computing algorithm has been proposed for selection and extraction of facial feature points from images, and also training and testing the classifier, that can execute on a cluster of workstations. The developed Facial Expression Recognition System has been tested on a JAFFE database and comparative study was made with others. Experimental results show that our methods outperform others.

Keywords: Multi-class SVM, JAFFE Database, Feature extractions, Mapping, MPI..

1. Introduction

Facial expression plays a principal role in human interaction and communication since it contains critical and necessary information regarding emotion. Understanding the person's activity may be possible through person's facial expressions such as anger, disgust, fear, happiness, sadness, surprise and neutral. Facial expressions can be recognized by humans with no errors and no delay, the task of automatically recognizing different facial expressions in human-computer environment is significant and challenging.

Considerable amount of work done by different researches for classification of facial expressions. Since JAFFE[12] database is commonly used in measuring the performance of facial expression recognition systems, we concentrate on applying our system on this database and perform comparisons with other systems. Sohail et al[8] developed automatic facial expressions using k-NN classifier with eleven feature points on JAFFE database and achieved an average 90.76% successful classification rate. They also tested using Neural Network and Naïve Bayes Classifiers, achieved 83.19% and 84.05% respectively. Kharat et al[9] utilised DCT, Fast Fourier Transform(FFT), and Singular Value Decomposition (SVD) in the extraction of whole facial features and then feed the features into an SVM for human emotion recognition. Sun et al[10] used histogram sequence of local Gabor binary patterns. The Gabor Coefficients Map (GCM) are extracted by convolving the face image with the multi-scale and multi-orientation Gabor filters. Then, the local binary pattern operator is performed on each GCM to extract the local Gabor binary pattern. Next, the face image is described using the histogram sequence of all these local Gabor binary patterns. Finally, the multi-class Support Vector Machine (SVM) for facial expression recognition. Their recognition rate is 97.28. Lyons et al[3] made use of Gabor filter at different scales and orientations, and applied 34 fiducial points for each convolved image to construct the feature vector for representing each facial image. After that, PCA is applied to reduce the dimensionality of feature vectors, and LDA is used to identify seven different facial expressions. Their recognition rate is 92% for JAFFE database. Zhang et al[2] adopted Gabor wavelet coefficients and geometric positions to construct the feature vector for each image and applied two-layer perceptron to distinguish seven different facial expressions. Their recognition rate is 90.1%. Buciu et al[6] tested different feature extraction methods such as Gabor filter and ICA combined with SVM using three different kernels:

* *Corresponding author.* Tel.: +917978907914
E-mail address: [1]narendra.rout@gec.edu.in

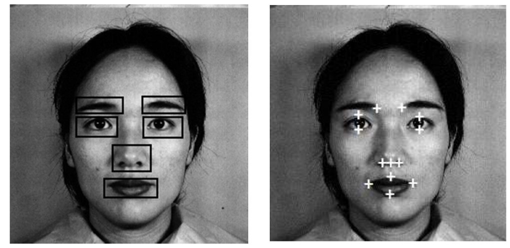
linear, polynomial and radial basis function, to check which combination can produce the best result. From their experiments, the best recognition rate is 90.34% by using Gabor wavelet at high frequencies combined with the polynomial kernel SVM at degree 2. Dubuisson et al[4] combined the sorted PCA as feature extractor with LDA as the classifier to recognize facial expressions. Their recognition rate is 87.6%. Shinohara and Otsu24[7] used Higher-order Local Auto-Correlation (HLAC) features and LDA to test the performance. Unfortunately, their system is unreliable since the correct rate is only 69.4%.

In this paper, we have been proposed a parallel algorithm for detection and classification of seven facial expressions using multi-class SVM classifier on JAFFE database. Facial expressions are classified using eleven feature points that identify the principle muscle actions as well as provide measurements of the discrete features responsible for each of the seven human expressions. Introducing parallelism in the image-processing algorithm may reduce the high computational cost. A significant number of works has been done for achieving parallelism in different image processing tasks under different architecture [14]. According to Amdahl's law [16], the time complexity of feature extraction process can be reduced to (n/p) + c by using p number of simultaneous symmetric processors. Here c is communication overhead between processors and n is the number of images in each expression. In image processing, data parallelism is less complex to implement as compare to task parallelism [13]. The parallel algorithm has been implemented using MPI to estimate the time complexity and efficiency.

The subsequent sections of this paper is organised as follows. Section II describes the feature selection and extraction. Section III describes the support vector machine. Section IV describes parallel computing paradigm and mapping. Section V then describes computer algorithm. Section VI shows the experimental results. Finally, Section VII draws conclusions.

2. Feature Selection And Extraction

The salient properties of the facial expression have to be passed through multi-class SVM classifier as feature vectors during both the learning and classification stages. For detection and classification of facial expressions, we have selected only eleven feature points that are present in the eyebrows, eyes, nose and mouth regions as shown in Fig. 1(a) and Fig. 1(b).A set of Haar-like features method is used to detect face and above regions and OpenCV is implemented for the detection of face and those regions. Using these eleven feature points a set of eleven distances are obtained such as $\{D_1, D_2, D_3, D_4, D_5, D_6, D_7, D_8, D_9, D_{10}, D_{11}\}$ [1] [8]. This feature vector represents one of the seven facial expressions, and is used as a set of features by the classifier for training as well as classifying facial expression.



(a) (b)

Fig. 1: (a) Localization of Facial Feature Regions and (b) Feature points used for capturing facial expressions

The task of feature point selection and extraction has been done in two stages. In first stage feature regions of human face are selected using an anthropometric face model based technique. In this method, eyes, eyebrows, nose and mouth regions of the face images are isolated. In second stage hybrid image processing technique is applied over these feature regions for detecting the eleven feature points. Finally, the feature vectors are calculated based on Euclidean distance formula for 2D plane.

The process for extracting the eleven feature points are described below:

The steps are

I. Detect face region from an image.

II. If face region is detected, then crop face; else return.

III. Detect eye region and identify the mid points of the upper and lower eyelids of left and right eye, such as (X_{lue},Y_{lue}), (X_{lle},Y_{lle}) for left eye and (X_{rue},Y_{rue}), (X_{rle},Y_{rle}) for right eye.

IV. Detect eyebrow region and identify inner corners of left and right eyebrow, such as (X_{leb},Y_{leb}) and (X_{reb},Y_{reb}) for left and right inner eyebrow corners respectively.

V. Detect nose region and identify nostrils, then identify midpoint of nostrils such as (X_{mn},Y_{mn}).

VI. Detect mouth region and identify left and right corner of mouth and mid points of upper and lower lips such as (X_{lmc},Y_{lmc}), (X_{rmc},Y_{rmc}) for left and right corners of mouth and (X_{mul},Y_{mul}), (X_{mll},Y_{mll}) for mid points of upper and lower lips.

VII. Calculate distances using Euclidean distance for 2-D plane for

$D_1((X_{reb},Y_{reb}), (X_{mn},Y_{mn}))$,
$D_2((X_{leb},Y_{leb}),(X_{mn},Y_{mn}))$,
$D_3((X_{reb},Y_{reb}), (X_{leb},Y_{leb}))$,
$D_4((X_{lue},Y_{lue}), (X_{lle},Y_{lle}))$,
$D_5((X_{rue},Y_{rue}), (X_{rle},Y_{rle}))$,
$D_6((X_{rmc},Y_{rmc}),(X_{mn},Y_{mn}))$,
$D_7((X_{lmc},Y_{lmc}), (X_{mn},Y_{mn}))$,
$D_8((X_{mul},Y_{mul}), (X_{mn},Y_{mn}))$,
$D_9((X_{mll},Y_{mll}), (X_{mn},Y_{mn}))$,
$D_{10}((X_{mul},Y_{mul}), (X_{mll},Y_{mll}))$,
$D_{11}((X_{rmc},Y_{rmc}), (X_{lmc},Y_{lmc}))$.

3. SUPPORT VECTOR MACHINE

An SVM maps the input space to a higher dimensional feature space and constructs a hyper plane to separate class members from non-members. It is a useful classification technique based on statistical learning theory [9]. Many hyper planes separate the data. However, only one achieves maximal margin for avoiding the possible misjudgement while determining which type of the new data will belong to. Let $S = \{(x_i, d_i) \mid i = 1, 2, \ldots , N\}$ be a set of N training patterns, where $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ belongs to R^n denotes an input vector in the input space and d_i belongs to $\{-1, 1\}$ represents the label class of x_i. Assume S be a linear separable. A linear SVM seeks an optimal separating hyper plane with maximum margin so that the linear SVM has good generalization. Assume the pair (w_o, b_o) is an optimal solution for a corresponding separating hyper plane for S, the linear SVM can be performed by the decision function f defined as $f(x)=sgn(w_0x^t+b_0)$ where sgn(.) stands for the sign function.

A. Multi-class SVM

Support Vector machine is basically designed for binary classification i.e. two class classifier. In order to use for multi class classification we have to follow following methods on SVM.

a) One-Against-All(OvA)

For the N-class problems (N>2), N two-class SVM classifiers are constructed. The i^{th} SVM is trained while labeling the samples in the i^{th} class as positive examples and all the rest as negative examples. In the recognition phase, a test example is presented to all N SVM's and is labeled according to the maximum output among the N classifiers. The disadvantage of this method is its training complexity, as the number of training samples is large. Each of the N classifiers is trained using all available samples.

b) One-Against-One(OvO)

This algorithm constructs N*(N-1)/2 two-class classifiers, using all the binary pair-wise combinations of the N classes. Each classifier is trained using the samples of the first class as positive examples and the samples of the second class as negative examples. To combine these classifiers, the Max Wins algorithm is adopted. It finds the resultant class by choosing the class voted by the majority of the classifiers. The number of samples used for training of each one of the OvO classifiers is smaller, since only samples from two of all N classes are taken in consideration. The lower number of samples causes smaller nonlinearity, resulting in shorter training times.

We implemented One-against-one (OvO) method which is more efficient than One-against-all on SVM[5] for classifying 7 classes in our facial feature extraction phase along with we used linear kernel on each SVM. We trained total 21 SVM's for 7 classes (ANGRY, DISGUST, FEAR, HAPPY, NEUTRAL, SAD, SURPRISE) as shown in Fig. 2.

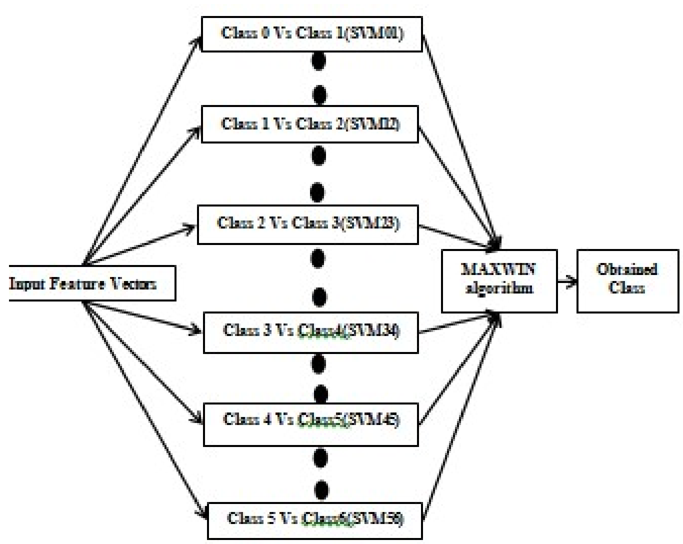


Fig. 2: Architecture of Multi Class SVM Classifier.

4. PARALLEL COMPUTING PARADIGM AND MAPPING

Parallel computing is a way of computing many calculations simultaneously and operating on the principle that large problems can often be divided into smaller ones which are solved concurrently. Through parallel computing the application programs run efficiently, reliably and quickly. The parallel programing can be implemented using MPI, Open MPI, Parallel Virtual Machine (PVM),

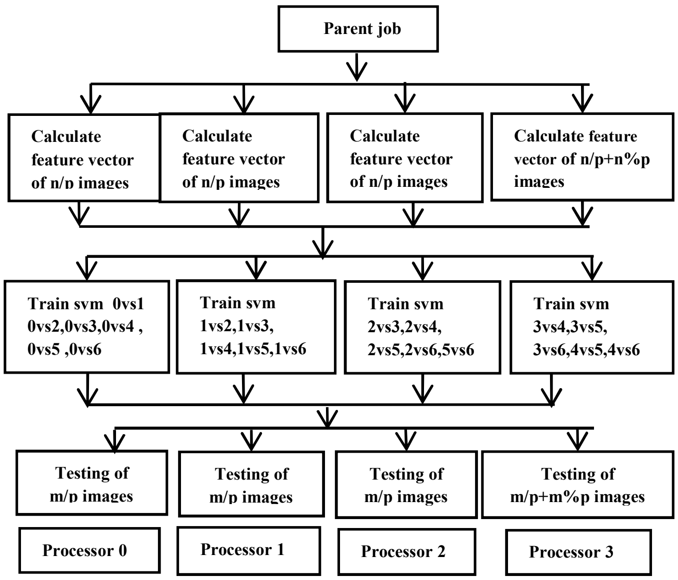


Fig. 3 Mapping Process

and CUDA(Compute Unified Device Architecture). In this work the parallel algorithm was implemented using Message Passing Interface (MPI)[15]. The MPI is a communication system that was designed to supply programmers with a standard for distributed memory parallel programming that is portable and usable on a variety of platforms.

The process of assigning task to the processor in a cluster of workstations is called mapping. The objectives of mapping are to maximize processor utilization and minimize inter processor communication. We divide the task into three parts and each part of the task to be executed parallel.

First part, we have to generate the feature vectors of all images of seven expressions. So we assign the task to each process to calculate feature vectors. Let n be the number of train images of each expression, p be the number of processor and d be the number of images assign to each processor to calculate feature vector such as for (p-1) processors, d=n/p images and for remaining one processor, d=(n/p + n%p) images. Those processors complete earlier shall have to wait for other processors to complete. This is done by MPI Barrier library.

Second part, we are using 21 SVM classifier to training our system and generating 21 support vectors. These tasks are assigned to processors as follows. Let us consider the number of processors is 4 and feature vectors of seven expressions with index 0, 1, 2, 3, 4, 5 and 6. Then, each processors train SVM with pair of feature vectors.

Processor 0: 0vs1, 0vs1, 0vs3, 0vs4, 0vs5, 0vs6
Processor 1: 1vs2, 1vs3, 1vs4, 1vs5, 1vs6
Processor 2: 2vs3, 2vs4, 2vs5, 2vs6, 5vs6
Processor 3: 3vs4, 3vs5, 3vs6, 4vs5, 4vs6

Again, those processors completed their task earlier shall have to wait for other processors to complete.

Third part, The large number of images to be tested using SVM classifier are distributed to processors in cluster of computers. Let m be the number of test images, p be the number of processor and t be the number of images to be tested in each processor, therefore t=m/p test images assign to (p-1) processors and t=(m/p + m%p) test images assign to another one processor. In this part, each processor determines feature vector of images, matches feature vector with all support vectors and declare class of each image. The Fig. 3 have been summarized the mapping process.

4. Computer Algorithm

In this section, we are presenting two algorithms: sequential algorithm for SVM training and testing and parallel algorithm for same.

A. Sequential Algorithm

We have been considered n facial images of each expression as training data and m facial images as testing data. The sequential algorithm for SVM training and testing describe as follows:

The steps are:

1. Calculate feature vectors for all training images (n). Store feature vector of each expression separately.
2. Determine support vectors by training multi class SVM classifier that takes a pair of feature vectors.
3. Continue step-2 for 21 times.
4. For testing a set of new images (m).
   a. Calculate feature vectors of all m new images.
   b. Matches feature vector of an image with all support vectors.
   c. Then, by using MAXWIN algorithm, determine class of facial expressions of that new image.
   d. Repeat step (b) and step (c) until all new images have been covered.

B. Parallel Algorithm

Let p be the number of processors, n be the number of training images of each expression and m be the number of testing images. The parallel algorithm for SVM training and testing describes as follows:

The steps are:

1. Initiate MPI.
2. Determine number of images (d) of each expression to be processed for training by each processor.
3. Each processor calculates feature vectors of d training images of each expression.

4. Repeat step 3 until all the expressions have been covered.
5. MPI Barrier.
6. Store feature vectors separately expression wise.
7. Each processor determines support vector by training SVM classifier which takes a pair of feature vectors for a pair of expressions.
8. Repeat step (6) until all pairs have been covered.
9. MPI Barrier.
10. Determine number of images (t) each processor will process for testing images.
11. Each processor does as follows:
    a. Calculate feature vector of t new images.
    b. Matches feature vector of an image with all support vectors.
    c. Determine class of facial expression of that new image by using MAXWIN algorithm.
    d. Repeat step (b) and (c) for t times.
12. MPI Finalise.

6. Experimental Results and Comparison

As specified earlier, the multi-class SVM classifier technique has been used for the recognition part of our facial expression classification system. Our system has been tested using Japanese Female Facial Expression (JAFEE) Database which contain 213 images each representing seven different facial expressions, posed by 10 Japanese female models. Fig. 4(a-g) represents seven facial expressions of one Japanese Female model. We used 66% images of each expression as training data and 34 % images of each expression as testing data. Our system provides the recognition rate of 100% on training images successfully and recognition rate of 98.59% on test images successfully. Obtained results are summarized in table I.

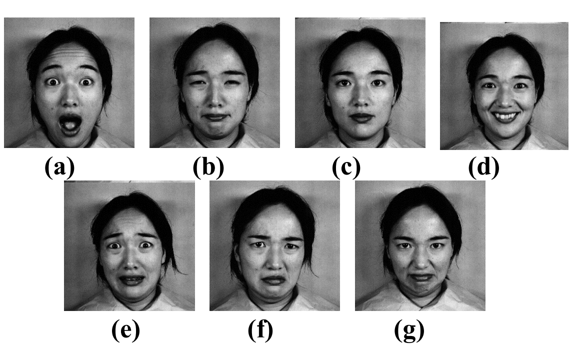


Fig. 4: (a) Surprise, (b) Sadness, (c) Neutral, (d) Happy, (e) Fear, (f) Disgust and (g) Angry.

TABLE I
TESTING PREDICTION

Predicted							Correct Class
Anger	Disgust	Fear	Happy	Sadness	Surprise	Neutral	
30	0	0	0	0	0	0	Anger
0	29	0	0	1	0	1	Disgust
0	0	30	0	0	0	0	Fear
0	0	0	30	0	0	0	Happy
0	1	0	0	29	0	0	Sadness
0	0	0	0	0	30	0	Surprise
0	0	0	0	0	0	29	Neutral
100	96.67	100	100	96.67	100	96.67	Correct rate

Performance of our multi-class SVM classifier based facial expression recognition system has been compared with others K-NN classifier, Back-Propagation Neutral network classifier, naïve Bayes classifier, few other SVM classifiers, PCA and LDA classifiers on JAFEE database as shown table II.

TABLE II
COMPARISON RESULTS

Author/ Method	Accuracy for testing images on JAFEE database
Sohail[8] Neural Network	83.19%
Sohail[8] Naïve Bayes	84.05%
Sohail[8] K-NN	90.76%
Sun[10]	97.28%
Kharat[9]	94.29%
Lyons[3]	92%
Zhang[2]	90.1%
Buciu[6]	90.34%
Dubuisson[4]	87.6%
Shinohara and Otsu[7]	69.4%
Our system	98.59%

Apart from this, the parallelization of the algorithm increases the performance of the system. The aim of parallelization is to gain speed up. Speed up is the ratio between sequential execution time and parallel execution time. The below table III presents the result of parallelization of our algorithm. This was done using dual core system.

TABLE III
RESULTS OF PARALLEL AND SEQUENTIAL EXECUTION

Sequential run time(in seconds)	Parallel run time (in seconds)	Speed up
13.304	8.090	1.644

7. Conclusion

A parallel algorithm was designed to recognize facial expressions using multi-class SVM classifier. First, it detects and extract eleven feature points from face region . Second, the multi class SVM classifier is trained to predict the facial expressions according to these features. Finally our experimental results showed an average successful recognition rate of 98.59%. Besides this, the parallelization of the algorithm was fully distributed to all available processors. Individual processor performs same task on different images. The parallel algorithm was implemented using MPI libraries on cluster of workstations with speed up of 1.644. The recognition rate of facial expression can be improved by adding more feature points. Load balancing and job mapping scheme for parallelization can be re-evaluated as part of future work to enhance the performance of the system.

REFERENCES

[1] A.S.M. Sohail, P. Bhattacharya: Localization of Facial Feature Regions Using Anthropometric Face Model. In: First International Conference on Multidisciplinary Information Sciences and Technologies (2006).

[2] Z. Zhang, M. Lyons, M. Schuster and S. Akamatsu, Comparison between geometry based and Gabor-wavelets-based facial expression recognition using multi-layer perceptron, Proc. IEEE Int. Conf. Automatic Face and Gesture Recognition (1998), pp. 454–459.

[3] M. Lyons, J. Budynek and S. Akamatsu, Automatic classification of single facial images, IEEE Trans. Patt. Anal. Mach. Intell. 21 (1999) 1357–1362.

[4] S. Dubuisson, F. Davoine and M. Masson, A solution for facial expression representation and recognition, Sign. Process.: Imag. Commun. 17 (2002) 657–673.

[5] Chih-Wei Hsu and Chih-Jen Lin-A Comparison of Methods for Multiclass Support Vector Machines, IEEE Transactions on Neural Networks, Vol. 13, No. 2, March 2002.

[6] I. Buciu, C. Kotropoulos and I. Pitas, ICA and Gabor representation for facial expression recognition, Proc. IEEE Int. Conf. Image Processing (2003), pp. 855–858.

[7] Y. Shinohara and N. Otsu, Facial expression recognition using Fisher weight maps, Proc. IEEE Int. Conf. Automatic Face and Gesture Recognition (2004), pp. 499–504.

[8] A.S.M. Sohail,P. Bhattacharya: Classification of Facial Expressions Using K-Nearest Neighbor Classifier. Springer-Verlag Berlin Heidelberg 2007.

[9] G. U. Kharat and S. V. Dudul, “Human emotion recognition system using optimally designed SVM with different facial feature extraction techniques,” WSEAS Transactions on Computers, vol. 7, no. 6, pp. 650–659, 2008.

[10] X. Sun, H. Xu, C. Zhao, and J. Yang, “Facial expression recognition based on histogram sequence of local Gabor binary patterns,” in 2008 IEEE Conference on Cybernetics and Intelligent Systems, pp. 158–163.

[11] F. Y. Shih, C. F. Chuang & P. S. P. Wang, Performance Comparisons of Facial Expression Recognition in JAFFE database, International Journal of Pattern Recognition and Artificial Intelligence Vol. 22, No. 3 (2008) 445–459.

[12] JAFFE. (2008, August). The Japanese female facial expression(JAFFE) database. Available at http://www.kasrl.org/jaffe.html.

[13] Shankar Ramaswamy, Sachin Sapatnekar and Prithviraj Banerjee, “A Framework for Exploiting Task and Data Parallelism on Distributed Memory Multicomputers“, IEEE Transactions On Parallel And Distributed Systems, Vol. 8, No. 11, November 1997.

[14] Cristina, Nicolescu, Pieter, Jonker ”A data and task parallel image processing environment”, Elsevier Science, Parallel Computing 28, page 945–965, 2002.

[15] J Quinn, “Parallel programming in c using MPI and OpenMP”. Mark D. Hill, Michael R. Marty, “Amdahl’s Law in the Multicore Era”, IEEE Computer Society, July 2008.