Dr. Hanumanthappa M et al. International Journal of Recent Research Aspects ISSN: 2349-7688, Vol. 5, Issue 2, June 2018, pp. 23-27

Face Detection and Recognition Using K-Means and Enhanced K-Means Algorithm with ANN

Dr. Hanumanthappa M[1] & Dr. Ashoka S B[*a]

[1]Professor, Department of Computer Science & Applications, Bangalore University, Bangalore-56
[*a] Assistant Professor, Dept of Computer Science [PG], Maharani's Science College for Women, Bangalore-01
(* Corresponding Author)

Abstract: **Face detection and recognition is one of the most successful applications of image analysis and understanding has gained much attention in recent years. The aim of this paper is detecting and recognizing human faces in crowded image and classification of facial expression from frontal facial images. Our approach of face detection and recognition is based on machine learning by implementing artificial neural network for gray scale basis with an extensibility of working with color images and then converting as gray scale label. Enhanced K-Means algorithm will classify the expressions into one of the expressions happy, sad and neutral. Expression classification will apply on the dataset of 200 images of KDEF (Karolinska Directed Emotional Face Database) database and expected to improve the performance of existing recognition system. A novel approach of building artificial neural network for face recognition is being applied by means of providing training to the neural network by extracted feature set rather than pattern recognition which reduces the overhead and complexity and makes the machine intelligent enough to sustain the fault tolerances.**

Keywords **– ANN (Artificial neural network), Back propogation, Bhattacharya distance, Color quantization, feature extraction, Skin segmentation, KDEF (Karolinska Directed Emotional Face Database).**

I. INTRODUCTION

We propose a novel algorithm for automatically detecting human faces in digital still color images and Facial expression recognition system solves the problem of face detection and feature extraction. Commonly three main steps are followed for expression recognition. First, detection of face boundary, second feature extraction and the last is facial expression recognition. Feature extraction referred to facial expression information, under non constrained scene condition, such as presence of a complex background and uncontrolled illumination, where most of the local facial feature based method are not stable. As a preliminary work a face detector which had been developed in order to index a huge amount of video and images data and to cope with high-speed requirements. Only I frames (Intraframe transform coding) were analyzed from MPEG streams as we wanted to avoid costly decompression. The features used to recognize the expressions are forehead, mid forehead, mouth and cheek.
Color segmentation of these I frames was performed at MPEG macro-block level (16x16 pixels). Skin color filtering was performed, providing a macro-block binary mask which was segmented into non-overlapping rectangular regions containing contiguous regions of skin color macro blocks (binary mask segments areas). Then, the algorithm searched for the largest possible candidate face areas and iteratively reduced their size in order to scan the entire possible scheme has been substantially enhanced. The proposed system is influenced by biological neural network behavior of identifying an object by means of supervised or unsupervised learning mechanism. An artificial neural network approach of parallel processing is being mimicked the way the biological neural networks works. In our method, a dedicated, suitable neural network is designed which is being tested with sample dataset and test dataset by means of extracted feature training provided to the neural network and make it intelligent to identify subject with the variance or tolerance for gray scale image as well as for color image. In this paper we mainly focused on face detection and face recognition in crowded place with the supporting tools of ANN to develop a computational model of face detection and recognition with fast, reasonably simple, and accurate in constrained environment. The result of this technique is feasible and effective for facial expression recognition.

II. METHODOLOGY

2.1 COLOR Quantization Using K-Means Algorithm And Enhanced K-Means Clustering Algorithm

K-Means Clustering Algorithm: K-Means Clustering algorithm is the most popular and simplest algorithm of clustering. It groups the input data points into the specified number of clusters based on the features. In the existing system, clusters are defined based on the expressions. K-Means algorithm is shown in the Algorithm

Input:
k // the number of clusters,
D= {d1, d2,…, dn} // set of n data items.
Output:
A set of k clusters.
Steps:
1. Randomly select k data points from D as the initial cluster centers;
2. (Re) assign each data point to the cluster, to which the data point is nearest;
3. Update the cluster means, i.e., calculates the mean value of the data points for each cluster;

4. Repeat, until no change.

"Enhanced K-Means Clustering algorithm" is to be used for Facial Expression Recognition.

Algorithm2: Enhanced K-Means Algorithm

Input:
k // the number of clusters,
D= {d1, d2,......,dn} // set of n data items.
Output:
A set of k clusters.
Steps:
Phase 1: Determine the initial centroids of the clusters by using Algorithm 3.
Phase 2: Assign each data point to the appropriate clusters by using Algorithm 4.

Algorithm3: Finding the initial centroids

Input:
D = {d1, d2,......, dn} // set of n data items
k // Number of desired clusters
Output:
A set of k initial centroids.
Steps:
1. Set m = 1;
2. First of all compute the distance between each data point and all other data- points in the set D;
3. Then find the closest pair of data points in the set D and create a new data-point set Am which contains these two data- points, delete these two data points from the set D;
4. After that find the data point in D that is closest to the new data point set Am, Add it to Am and delete it from D;
5. Repeat step 4 until the number of data points in Am reaches 0.75*(n/k);
6. If m<k and reaches a threshold, then at this point m = m+1, create another set data by follow the steps from 3-5
7. And now, for each data-point set Am (1<=m<=k) find the arithmetic mean of the vectors of data points.

Algorithm4: Assigning data-points to clusters

Input:
D = {d1, d2,......,dn} // set of n data items
C= {c1, c2,......,cn} // set of k centroids
Output:
A set of k clusters.
Steps:
1. Firstly compute the distance between data-point di (1<=i<=n) and all the centroids cj (1<=j<=k) as d (di, cj);
2. Now for each data-point di, find the closest centroid cj and assign di to cluster j.
3. Then for each cluster j (1<=j<=k), recalculate the centroids;
4. Repeat
5. for each data-point di,
a. Compute its distance from the centroid of the present nearest cluster;
b. If this distance is less than or equal to the present nearest distance, the data-point stays in the cluster;
c. Else for every centroid cj (1<=j<=k) compute the distance d (di, cj); End for;
6. Assign the data-point di to the cluster with the nearest centroid cj Endfor (step (2));
7. For each cluster j (1<=j<=k), Recalculate the centroids until the convergence criteria is met.

2.3 Bhattacharya Distance

The Bhattacharya distance measures the similarity of two discrete or continuous probability distributions. It is closely related to the Bhattacharya coefficient which is a measure of the amount of overlap between two statistical samples or populations.

The Bhattacharya distance is given as

$$\mathcal{B}_c = \frac{1}{c} \ln \left(\frac{\sigma_1^c + \sigma_2^c}{2\sqrt{\sigma_1^c \sigma_2^c}} \right) \quad (6)$$

According to the assumption that approximation images are distributed according to the Gaussian law, while the detail images are distributed according the Laplacian law. The resulting distance D between two feature vectors V_k and V_l is

$$\mathcal{D}(\mathcal{V}_k, \mathcal{V}_l) = \frac{1}{2} \sum_{i=0}^{3} \ln \left(\frac{\sigma_{ik}^2 + \sigma_{il}^2}{2\sigma_{ik}\sigma_{il}} \right) + \sum_{i=4}^{m+3} \ln \left(\frac{\sigma_{ik} + \sigma_{il}}{2\sqrt{\sigma_{ik}\sigma_{il}}} \right) \quad (7)$$

2.4 FEATURE EXTRACTION

After detecting the face area the feature vector of the face is extracted and facial feature is extracted using the following formula:

$$p[j] = \begin{cases} avg[j] & j \in [0, h) \\ ahg[j-h] & j \in [h, j-h+w) \end{cases} \quad (8)$$

where,

$$avg[i] = \sum_{y=0}^{h} f(i, y) \qquad i \in [0, w)$$

and

$$ahg[j] = \sum_{x=0}^{w} f(x, j) \qquad j \in [0, h) \quad (9)$$

2.5 Classification

A Successful face recognition methodology depends heavily on the particular choice of the feature used by the pattern classifier. The Back-Propagation is the best known and widely used learning algorithm in training multi layer perceptrons (MLP). The MLP refer to the network consisting of a set of sensory units(source nodes) that constitute the input layer, one or more hidden layers of computations nodes, and an output layer of computation nodes.

2.6 BACK-PROPAGATION ALGORITHM

In this System the neural neural network consists of 3 hidden layers and the back propagation algorithm is used to train the neural network. The algorithm works as in follows steps:

1. First apply the inputs to the network and work out the output- remember this initial output could be anything, as the initial weights were random numbers.
2. Nest Work out the error for neuron B. The error is *What you want—What you actually get*, in other words:

   ErrorB=OutputB(1-OutputB)(TargetB-OutputB)

   The "*Output(1–Output)*" term is necessary in the equation because of the sigmoid function- if we were only using a threshold neuron it would just be(*Target–Output*).
3. Change the weight. Let W+AB be the new (trained) weight and WAB be the initial weight.

   W+AB=W+AB+(ErrorB x OutputA)

   Notice that it is the output of the connecting neurons (neuron A) we use (not B). We update all the weights in the output layer in this way.
4. Calculate the Errors of the hidden layer neurons. Unlike the output layer we can't calculate these directly(because we don't have a target), so we *Back Propagate* them from the output layer (hence the name of the algorithm).

   ErrorA=OutputA (1-OutputA) (ErrorB WAB + WAC)

   Again, the factor "*Output (1-Output)*" is present because of the sigmoid squashing function.
5. Having obtained the Error for the hidden layer neurons now proceed as in stage 3 to change the hidden layer weights. By repeating this method we can train a network of any of layers.

III. System Details

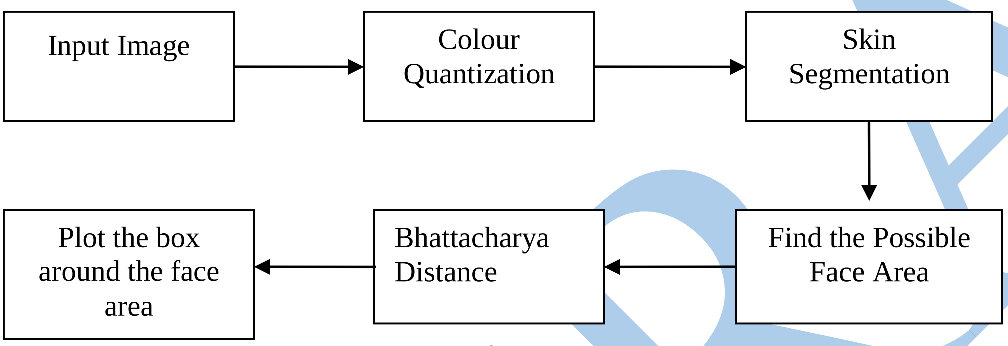

Fig 1: Structure of Face Detection

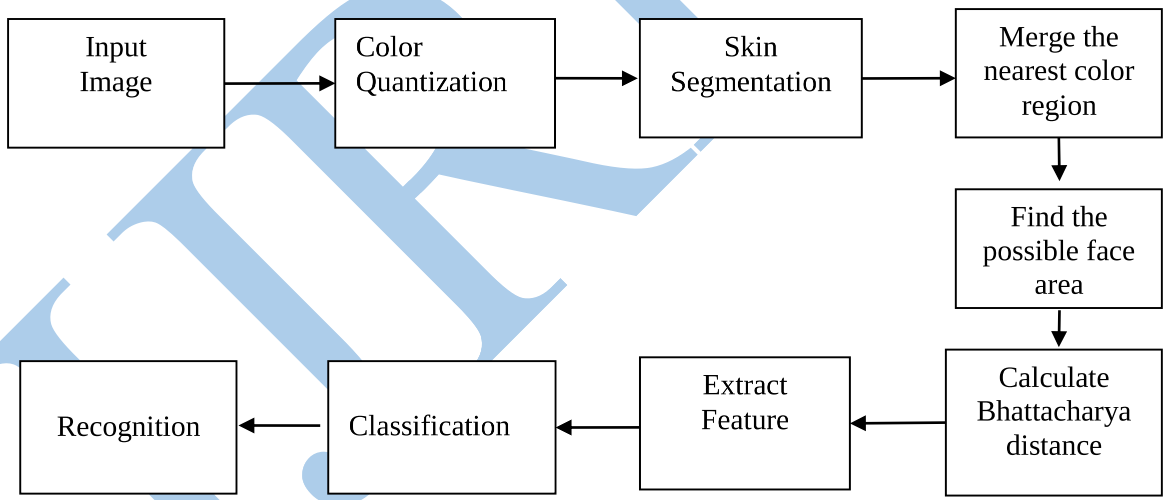

Fig 2: Structure of Face Recognition

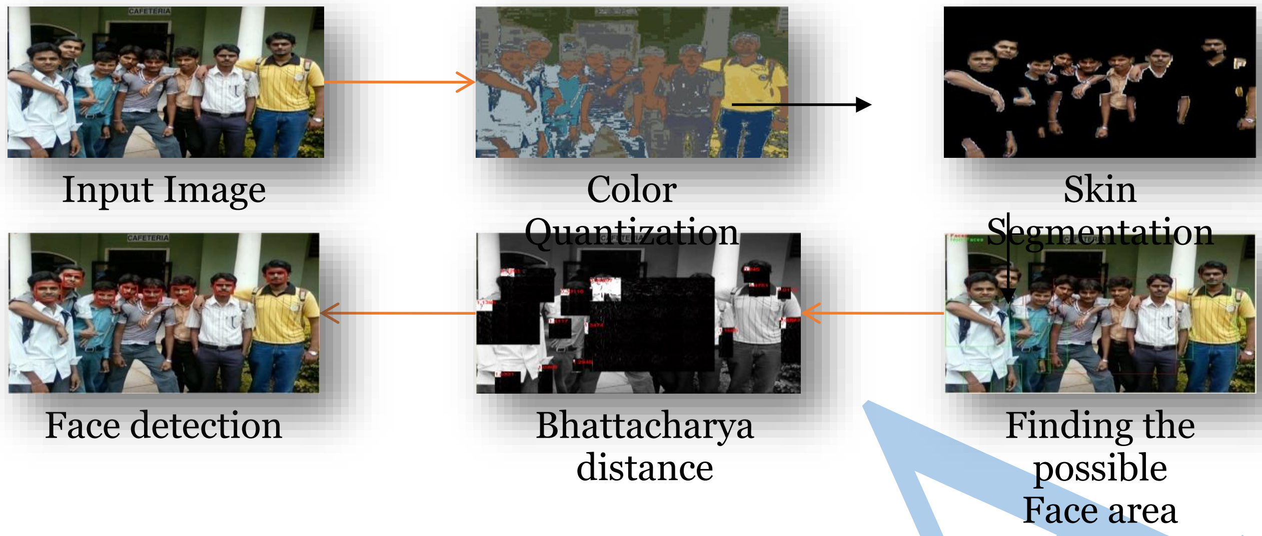

Fig. 3: Experimental observations of Face Detection.

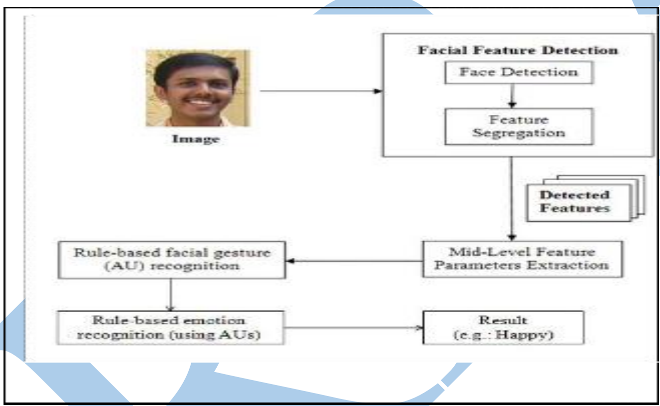

Fig.4 Block Diagram of the Facial Recognition System

Table 1: Efficiency table of face detection

No. of faces in input image	No. of face detected	No or false Alarm	No. of Undetected faces	Efficiency
4	4	1	0	82.8%
6	5	0	1	
8	6	1	2	
10	8	0	0	
13	10	2	3	

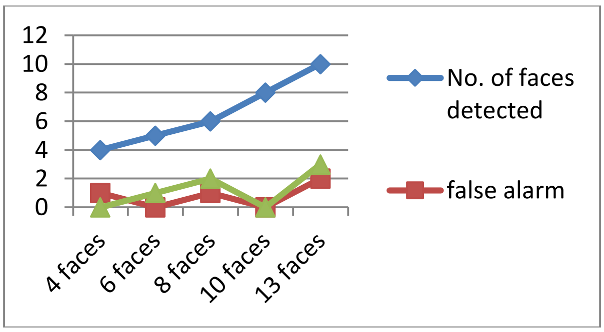

Fig. 5: Graph of face detection

Same Database is used to test both face detection and face recognition system. The efficiency of face recognition system is around 72.35%

Table 2: Efficiency Table of face recognition

No. of face input Image	No of faces recognized	No of Non recognized Faces	Efficiency
4	4	0	72.35%
6	6	0	
8	5	3	
10	8	2	
13	11	2	

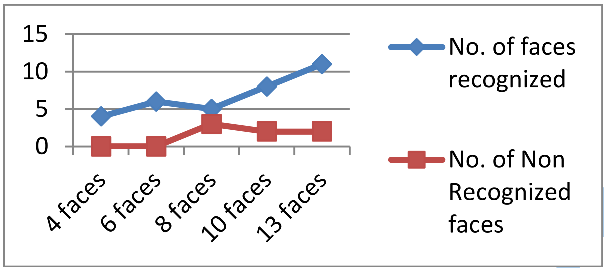

Fig. 6: Graph of Face Recognition

IV. CONCLUSION

The major highlights of this paper is that the face recognition system has to be developing a pre-attentive pattern recognition capability that does not depend on having three-dimensional information or detailed geometry and to develop a computational model of face recognition that is fast, reasonably simple, and accurate in constrained environment. The comparison between K-Means algorithm and enhanced K-Means algorithm in micro observation in experimental results shown the enhanced K-Means algorithm for facial expression recognition gives better result than K-Means algorithm and it is also effective so, it is expected that Enhanced K-Means algorithm will improve the performance with increases the rate of success for recognition. Challenging issues in future research is implementation of video surveillance concepts on detection and reorganizations.

REFERENCES

[1] Sun, L. Ding, X. Wang, and X. Tang. Deepid3(2015): Face recognition with very deep neural networks. CoRR , abs/1502.00873.

[2] Dipen Saini et al(2016) International Journal of Current Engineering and Technology, E- ISSN 2277 –4106, P-ISSN 2347 –5161, Vol.6, No.2.

[3] G. Yang and T.S Huang(1994). “Human face detection in a complex background,” Pattern Recognit, vol.27, no. 1, pp.55-63.

[4] Rafael C. Gonzalez and Richard E Woods, “Digital Image Processing using MATLAB7”, Pearson Education Asia.

[5] Christophe Garcia and Georgios Tziritas(September 1999), “Face Detection Using Quantized Skin Color Regions Merging and Wavelet Packet Analysis”, IEEE Trans, on Multimedia, vol.1, no. 3.

[6] Xudong Jiang and Alex kot(March 2008) “Eigenfeature Regularization and Extraction in Face Recognition” IEEE Tranc., on Pattern Analysis and Machine Intelligence, vol. 30, No.3.

[7] Yue-Hui Sun Ming-Hui Du(2006) “Face Detection using DT-CWT on Spectral Histogram” Machine Learning and Cybernetics, International Conference.

[8] W.J. Book(1990), Modelling design and control of flexible manipulator arms: A tutorial review, Proc. 29th IEEE Conf. on Decision and Control, San Francisco, CA, 500-506.

[9] A. Jain, R.P.W. Duin, J. Mao(Jan 2000), “Statistical pattern recognition : a review”, IEEE Trancations on pattern analysis and machine intelligence, 22(1), pp. 4-37.

[10] Yang, M.H., Ahuja, N., and Kriegman, D(2000). Face detection using mixtures of linear subspacings. Prodeedings fourth IEEE International Conference on Automatice Face and Gesture Recognition.

[11] Ashoka S.B (Novembe 2014) "face detection and recognition using k-means and neural network methods", Advanced research in electrical and electronic engineering.

[12] Abhjeet Sekhon et al(February-2017),International Journal of Computer Science and Mobile Computing, Vol.6 Issue.2, pg. 137 – 141.

