Advanced Research in Electrical and Electronic Engineering
Print ISSN: 2349-5804; Online ISSN: 2349-5812 Volume 1, Number 5 (2014) pp. 9-14

http://www.krishisanskriti.org/areee.html

Face Detection and Recognition Using K-means and Neural Network Methods

Ashoka S.B.

Dept of Computer Science and Applications, Bangalore UniversityBangalore-56

Abstract: **Face recognition is one of the most successful applications of image analysis and understanding and has gained much attention in recent years. The aim of this paper is detecting and recognizing human faces in crowded image. 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. 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: **Backpropogation, Bhattacharya distance, Color quantization, feature extraction, Skin segmentation.**

1. INTRODUCTION

A Facial recognition system is a computer application for automatically identifying or verifying a person from a digital image or a video source. One of the ways to do this is by comparing selected facial features from the image and a facial database. It is typically used in security systems and can be compared to other biometrics such as fingerprint or eye iris recognition system.

We propose a novel algorithm for automatically detecting human faces in digital still color images, under nonconstrained 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.

Color segmentation of these I frames was performed at MPEG macro-block level(16 16 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. It performs first color clustering of the original image, in order to extract a set of dominant colors and quantize the image according to this reduced set of colors. Then, a chrominance-based segmentation using an improvement of the method is performed.

A merging stage is iteratively applied on the set of homogeneous skin color regions in the color quantized image, in order to provide a set of candidate face areas, without scanning all the different possible aspects ratios and the possible aspects ratios and the possible positions into binary segment areas. This improvement leads to a better precision in location the faces and helps in segmenting the faces from the background, especially when parts of the surrounding background have a color that may be classified as skin color.

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 .

2. COLOR QUANTIZATION USING K-MEANS ALGORITHM

Vector quantization is applied to quantize the image colors to a reduced set of so-called dominant colors. The basic principle is to map a color vector x onto another color vector y_i, y_i being a codebook vector and representing a color cluster C_i in color space. y_i is a dominant color and is the mean value of color vectors belonging to $C_{i,}$ the initial codebook vectors y_i are first determined by color histogram calculation.

The finale set of codebook vectors is obtained using an iterative algorithm that originates from pattern recognition, The **K-means algorithm**. Also known as the Linde-Buzo-

Gray algorithm, which has been first applied to vector quantizer design for signal compression tasks. At each iteration each color vector(pixel) is assigned to the closest color cluster according to its distance t the mean color vector of the cluster, and the mean color vector of the set pixels which have been associated to one color cluster is updated. The algorithm stops after a given number of iterations.

Then, color cluster are merged according to a predefined maximum distance. Finally, each pixel in the image receives the value the value of the mean color vector of the cluster it has been assigned to the resulting image is therefore quantized according to the set of dominant colors.

3. SKIN COLOR SEGMENTATION

The purpose of skin color segmentation is to segment the quantized color image according to skin color characteristics. Two color models are used. The YCbCr model is naturally related to MPEG and JPEG coding. The HSV(Hue, Saturation, Value) model is naturally in computer graphics and is considered by many to be more intuitive to use, closer to how an artist actually mixes colors.

3.1 YCbCr Color space

In the case of the YCbCr color space, we noticed that the intensity value Y ha little influence on the distribution in the CbCr plane and that sample skin colors from a small and very compact cluster in the CbCr plane. The envelop of the skin color subspace in YCbCr is estimated, using the intensity value in order to cope with strong lighting variations. Planner approximation is used to approximate the skin color borders. The following is used to skin color segmentation.

cb=0.148*I(:,:,1)-0.291*I(:,:,2)+ 0.439*I(:,:,3) +128;
cr=0.439*I(:,:,1)-0.368*I(:,:,2)- 0.071*I(:,:,3)
+ 128; (1)
If 140<cr<165 and 140<cb<196 and 0.01< hue<0.1

3.2 Merging the Nearest Color Region

After the pre-processing the segmented image is merged using the region adjacency graph to find the candidate face region. The skin colour regions merging algorithm starts by computing a region adjacency graph, where each node represents a homogenous skin colour region of the quantized image.

The criterion of the connectivity is based on a minimum distance computed between the associated bounding boxes of each homogenous region. Evaluating connectivity using a distance the vertices of the bounding boxes is much faster than estimating the connectivity in the pixel domain. The distance is calculated by formula:

$$\mathrm{sqrt}((R_1-R_2)^2+(G_1+G_2)^2+(B_1+B_2)^2). \quad (2)$$

Let C_0 be the set of homogenous skin colour regions Ri i=[1,….N]. First, the criterion of connectivity is applied to build the region adjacency graph.

The merging criterion among the bounding boxes of adjacent regions R_i and R_i is based on a maximum allowed distance $D_r(R_i,R_i)$ which encodes colour dissimilarity in the bounding boxes. For that reason the color dissimilarity measure has been chosen to be the following:

$$D_r(R_i, R_j) = \alpha\,|H_i - H_j| + \beta\,|S_i - S_j| + \gamma\,|V_i - V_j|$$

Where (Hi, Si, Vi) is the avergage skin color vector in the bounding box of the region Ri. The value of α has to be choose much bigger than the values of β and γ in order to take into account mainly Hue differences.

The set C1 is obtained by merging the compatible adjacent of C0. Then, each new set of merged Ck(k=[1…K] is obtained by merging iteratively the set of merged region Ck-1 with the original set C0 in the following way(note that a region adjacency graph is computed before each new iteration):

$$\forall k \in [1\cdots K],\quad \forall \mathcal{R}_i \in \mathcal{C}_0,\quad \forall \mathcal{R}_j \in \mathcal{C}_{k-1}$$
$$\mathcal{C}_k = \bigcup \mathcal{R}_i \cup \mathcal{R}_j \quad (3)$$

In the Current implementation, iterative merging is performed with a maximum number of iterations K=3 as we want to avoid time-consuming combinatory. We also noticed that the merging process provides the expected candidate face areas within these three iterations. Finally, the candidate face area set CF contains the original regions set C_0 and the interactively built sets of merged region C_k(k=[1…K]) i.e. $CF=U^k_{k=0}\, C_k$..

4. FINDING THE POSSIBLE FACE AREA

The discrete wavelet series for a continuous signal s(t) is defined by the following equations:

$$c_{n,k} = \frac{1}{2^{n/2}} \int s(t)\psi^*(2^{-n}t - k)\, dt. \quad (4)$$

The series use a dyadic scale factor , n being the scale level and k being the localization parameter. The function ψ(t) is the mother wavelet which satisfies some admissibility criteria and ensures a complete, non-redundant, and orthogonal representation of the signal.

The discrete wavelet transform (DWT) results from the above series, and is equivalent to the successive decomposition of the signal by a pair of filters *h(.)* and *g(.)* as shown in fig:

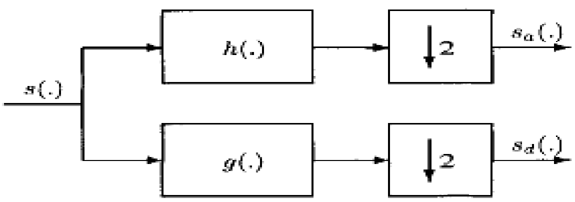

Figure 1:Dyadic Decomposition of Signal

The low-pass filter *h(.)* provides the approximation of the signal at coarser resolutions. The High-pass filter *g(.)* provides the details of the signal at coarser resolutions.

In classical wavelet decomposition, the image is split into an approximation and details. The approximation is then split into a second-level of approximation and details. For a *n*-level decomposition, the signal is decomposed in the following way:

$$A_n=[h_x *[h_y*A_{n-1}]_{\downarrow 2,1}]_{\downarrow 1,2}$$
$$D_{n1}=[h_x *[h_y *A_{n-1}]_{\downarrow 2,1}]_{\downarrow 1,2}$$
$$D_{n2}=[g_x *[h_y *A_{n-1}]_{\downarrow 2,1}]_{\downarrow 1,2}$$
$$D_{n2}=[g_x *[h_y *A_{n-1}]_{\downarrow 2,1}]_{\downarrow 1,2} \quad (5)$$

Where * denotes the convolution operator, $\downarrow 2,1$ ($\downarrow 1,2$) sub sampling along the rows (columns) and $A_0=I(x,y)$ is the original image. A_n is obtained by low-pass filtering and is the approximation image at scale n. The detail images D_{nj} are obtained by bandpass filtering in a specific direction (i=1,2,3…) for vertical, horizontal and diagonal directions respectively and thus contain directional details at scale n. The original image is thus represented by a set of sub images at several scales : $\{A_n, D_{nj}\}$. This results in a wavelet decomposition tree as shown in following fig.

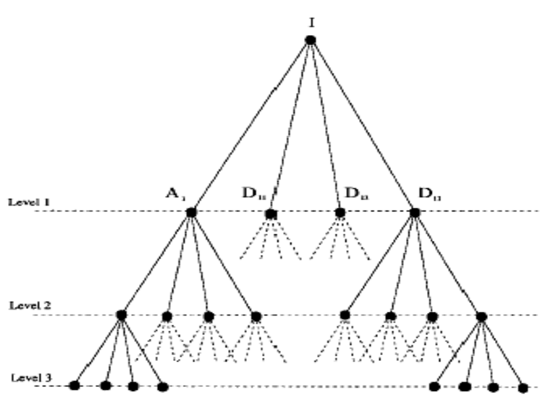

Figure 2: Wavelet packet tree

By Considering each R_i area as a bounding box, and by dividing it into four parts: a left top part, a right top part a left bottom part and a right bottom part all of equal size, where the wavelet packet analysis is performed by extracting moments from wavelet coefficients in these areas, we obtain information about the face texture, related to different facial parts like the eyes, nose and mouth, including facial hair.

Therefore, from the top and bottom areas, we extract the corresponding standard deviations $\sigma_{(top1)}$, $\sigma_{(top2)}$, $\sigma_{(bottom1)}$ and $\sigma_{(bottom2)}$ of the wavelet coefficients in the approximation image of the selected level of decomposition.

5. 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 coefficient can be used to determine the relative closeness of the two samples being considered. It is used to measure the separability of classes in classification.

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. Due to the design of the filters, the mean values are zero valued for the detail images.

The resulting distance *D* between two feature vectors V_k and V_l is

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

Therefore, Classification is performed by evaluating the distance *D* from R_i each feature vector V_k to the prototype feature vector V_l of the corresponding size category. Each R_i feature vector has to be classified as face or non-face according to distance *D* to the average prototype vector of the corresponding size category. R_i is classified as a face area, if is below threshold T_{HD} and rejected otherwise.

6. 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

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

7. 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.

Back-Propagation is a multilayer feed forward, supervised learning network based on gradient descent learning rule. Being gradient descent method it minimizes the total squared error of the output computed by the net. The aim is to train the network to achieve a balance between the ability to respond correctly to the input patterns that are used for training and the ability to provide good response to the input that are similar.

7.1. 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*).

4. 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). This is done by taking the errors from the output neurons and running them back through the weights to get the hidden layer errors. For example if neuron A is connected as shown to B and C then we take the errors from B and C to generate an error of A.
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.

8. SYSTEM DETAILS

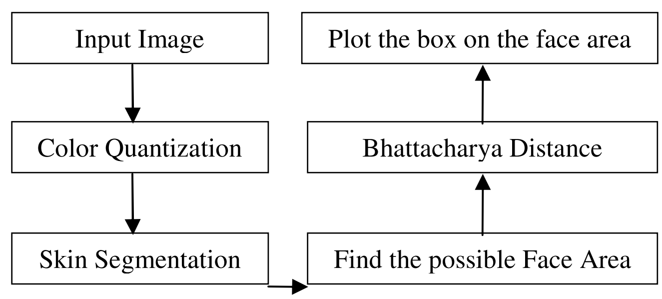

Figure 3: Structure of Face Detection

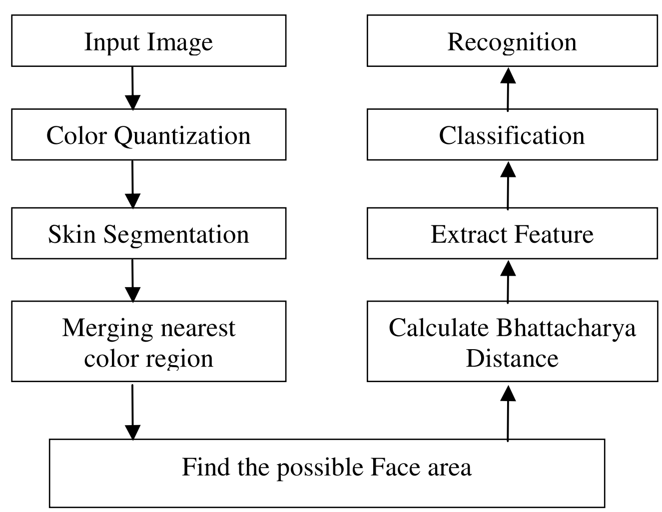

Figure 4: Structure of Face Recognition

The various parameters assumed for Back propagation algorithm are as follows:
No. of Input unit=1 feature matrix

Accuracy=0.001
Learning rate=0.4
No. of epochs=200
No. of hidden neurons=70
No. of output unit=1

The Neural Network is trained using different set of database like the database1 consists of 20 images of 5 persons, database2 40 images of 20 persons and the training time is noted down to calculate the network efficiency.

The images consisting of different no. of faces are taken to test the face detection algorithm. Through this test the efficiency noted is 82.8%. The database used to test the face detection algorithm consist the face taken in the complex background. The following table shows the efficiency of the face detection

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	

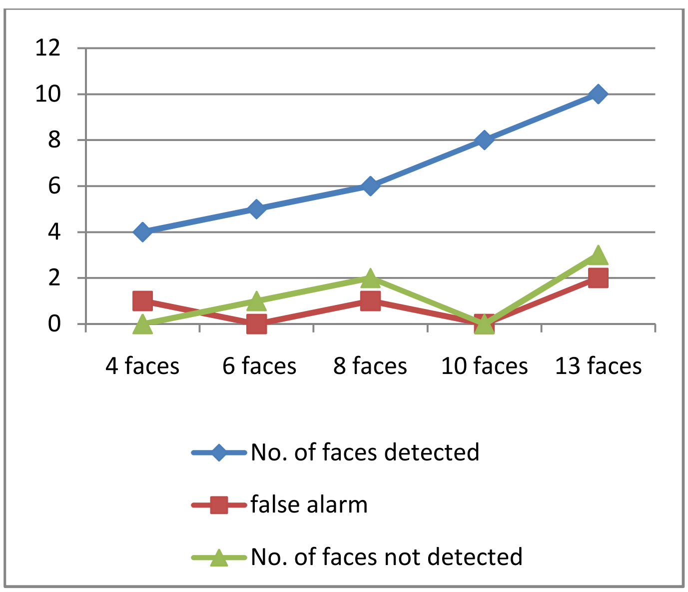

Figure 5: Graph of face detection

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	

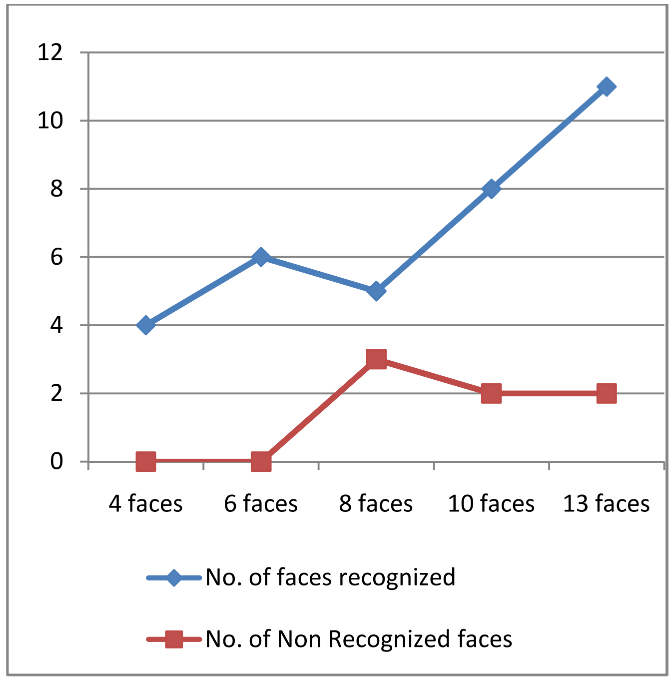

Figure 8: Graph of Face Recognition

9. CONCLUSION

The major focus of this paper is that the face recognition system has to be develop a pre-attentive pattern recognition capability that does not depend on having three-dimensional information or detailed geometry; the goal of the paper is to develop a computational model of face recognition that is fast, reasonably simple, and accurate in constrained environment.

REFERENCES

[1] M. Kirby and L. Sirovich, "Application of the Karunen-Loeve procedure and the characterization of human faces," *IEE Trans, Pattern Anal, Machine Intell,* vol.12, pp. 103-108, Jan 1990..

[2] G. Yang and T.S Huang. "Human face detection in a complex background," *Pattern Recognit,* vol.27, no. 1, pp.55-63, 1994.

[3] J.A. Richards, *Remote sensing Digital Image Analysis,* Springer, Berlin, 1993.

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

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

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

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

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

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

[10] R. Chellappa, C.L. Wilson and S.Sirohey, "Human and Machine Recogition of Faces: A Survey", *Proceedings of the IEEE,* 83(5), pp. 703-740, May 1995.

[11] 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.*