ISSN:0975-9646

Review on: Emotion Detection

Navleen kaur[#1], Madhu Bahl[*2],Harsimran Kaur[#3]

[#]*Computer Science Departmaent, Punjab Technical University, Landran,Distt.Mohali,Punjab,India*
[*]*Computer Science Department, Landran,Distt.Mohali*

***Abstract* - Emotions are most widely represented with eye and mouth expressions. The proposed system uses colour images and it is consisted of three modules. The first module implements skin detection, using Markov random fields' models for image segmentation and skin detection. A set of several coloured images with human faces have been considered as the training set. A second module is responsible for eye and mouth detection and extraction. The specific module uses the HLV colour space of the specified eye and mouth region. The third module detects the emotions pictured in the eyes and mouth, using edge detection and measuring the gradient of eyes' and mouth's region figure.**

1. INTRODUCTION

Human emotion detection from image: First, it takes an image, then by skin colour segmentation, it detects human skin colour, then it detect human face. Then it separates the eyes & lip from the face. Then it draws bezier curve for eyes & lips. Then it compares the bezier curve of eyes and lips to the bezier curves of eyes & lips that are stored in the data base. Then it finds the nearest bezier curse from the data base & gives that data base stored bezier curve emotion as this image emotion. Provides high spectral quality of the fused satellite images. However, the fused image by Wavelets has much less spatial information than those by the Brovey, IHS, and PCA methods. The spatial information of fused image is an important factor as much as the spectral information in many remote sensing applications. For emotion detection of an image, we have to find the Bezier curve of the lip, left eye and right eye. Then we convert each width of the Bezier curve to 100 and height according to its width. If the person's emotion information is available in the database, then the program will match which emotion's height is nearest the current height and the program will give the nearest emotion as output. If the person's emotion information is not available in the database, then the program calculates the average height for each emotion in the database for all people and then gets a decision according to the average height. For face detection, first we convert binary image from RGB image. For converting binary image, we calculate the average value of RGB for each pixel and if the average value is below than 110, we replace it by black pixel and otherwise we replace it by white pixel. By this method, we get a binary image from RGB image. Then, we try to find the forehead from the binary image. We start scan from the middle of the image, then want to find a continuous white pixels after a continuous black pixel. Then we want to find the maximum width of the white pixel by searching vertical both left and right site. Then, if the new width is smaller half of the previous maximum width, then we break the scan because if we reach the eyebrow then this situation will arise. Then we cut the face from the starting position of the forehead and its high will be 1.5 multiply of its width. In the figure, X will be equal to the maximum width of the forehead. Then we will have an image which will contain only eyes, nose and lip. Then we will cut the RGB image according to the binary image. Then we find the starting high or upper position of the two eyebrows by searching vertical. For left eye, we search w/8 to mid and for right eye we search mid to w – w/8. Here w is the width of the image and mid is the middle position of the two eyes. There may be some white pixels between the eyebrow and the eye. To make the eyebrow and eye connected, we place some continuous black pixels vertically from eyebrow to the eye. For left eye, the vertical black pixel-lines are placed in between mid/2 to mid/4 and for right eye the lines are in between mid+(w-mid)/ 4 to mid+3*(w-mid)/ 4 and height of the black pixel-lines are from the eyebrow starting height to (h- eyebrow starting position)/4. Here w is the width of the image and mid is the middle position of the two eyes and h is the height of the image. Then we find the lower position of the two eyes by searching black pixel vertically. For left eye, we search from the mid/4 to mid - mid/4 width. And for right eye, we search mid + (w-mid)/ 4 to mid+3*(w- mid)/ 4 width from image lower end to starting position of the eyebrow. Then we find the right side of the left eye by searching black pixel horizontally from the mid position to the starting position of black pixels in between the upper position and lower position of the left eye. And left side for right eye we search mid to the starting position of black pixels in between the upper position and lower position of right eye. The left side of the left eye is the starting width of the image and the right side of the right eye is the ending width of the image. Then we cut the upper position, lower position, left side and the right side of the two eyes from the RGB image. For lip detection, we determine the lip box. And we consider that lip must be inside the lip box. So, first we determine the distance between the forehead and eyes. Then we add the distance with the lower height of the eye to determine the upper height of the box which will contain the lip. Now, the starting point of the box will be the ¼ position of the left eye box and ending point will be the ¾ position of the right eye box. And the ending height of the box will be the lower end of the face image. So, this box will contain only lip and may some part of the nose. Then we will cut the RGB image according the box.

In this project a method for face detection/segmentation in color images has been implemented. The algorithm begins by modeling human skin color in a suitable chrominance space using a database of skin pixels. Two methods of

modeling the skin color have been implemented. In the first method the skin color distribution is modeled as a unimodal or single component Gaussian. In the second method a Gaussian mixture model is used. Similarly a non-skin or background model is built using a database of non-skin pixels. These two models are used in computing the probability of each pixel in an input colour image to represent skin. Thus a Skin Probability image is obtained in which the gray level of each pixel represents the probability of the corresponding pixel in the input image to represent skin (scaled by a constant factor). The skin probability image is then analysed using a set of connected operators. The result is a set of connected components that have a high probability of representing a face. Finally a normalized area operator is used to retain only those components that are sufficiently large in size in comparison to the largest face component detected. The areas lying within the bounding boxes of these connected components, in the input image, are faces. They are segmented out from the image. The details of the algorithm are explained in further sections.

A. Skin color modeling using a unimodal Gaussian

The inspiration to use skin color analysis for initial classification of an image into probable face and non-face regions stems from a number of simple but powerful characteristics of skin color. Firstly, processing skin color is simpler than processing any other facial features. Secondly, under certain lighting conditions, color is orientation invariant. The major difference between skin tones is intensity eg. due to varying lighting conditions and different human races [3]. The color of human skin is different from the color of most other natural objects in the world. An attempt to build a comprehensive skin and non-skin models has been done in [4].

One important factor that should be considered while building a statistical model for colour is the choice of a Colour Space. Segmentation of skin coloured regions becomes robust if only the chrominance component is used in analysis. Therefore, the variations of luminance component are eliminated as much as possible by choosing the CbCr plane (chrominance components) of the YCbCr colour space to build the model. Research has shown that skin colour is clustered in a small region of the chrominance space [4]. The distribution of a set of sample training skin pixels in the CbCr plane is given in the figure below (Fig.1).

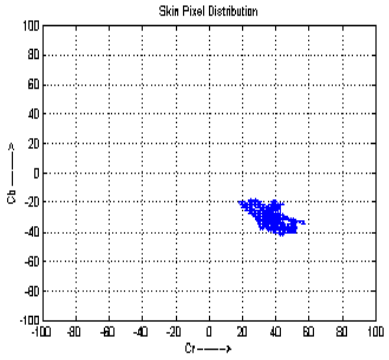


Fig. 1 Skin Pixel Distribution

The above figure shows that the colour of human skin pixels is confined to a very narrow region in the chrominance space. Motivated by the results in the figure, the skin colour distribution in the chrominance plane is modelled as a unimodal Gaussian [3]. A large database of labelled skin pixels is used to train the Gaussian model. The mean and the covariance of the database characterize the model. Images containing human skin pixels as well as non-skin pixels are collected. The skin pixels from these images are carefully cropped out to form a set of training images. Let $c = [Cb\ Cr]^T$ denote the chrominance vector of an input pixel. Then the probability that the given pixel lies in the skin distribution is given by:

$$p(c/skin) = \frac{\exp\left[-\frac{1}{2}(c-\mu_s)^T \Sigma_s^{-1}(c-\mu_s)\right]}{2\pi\sqrt{|\Sigma_s|}} \quad (1.1)$$

Where $\boldsymbol{\mu}_s$ and $\boldsymbol{\Sigma}_s$ represent the mean vector and the covariance matrix respectively of the training pixels. Thus the mean and the covariance have to be estimated from the training data to characterize the skin colour distribution as a unmoral Gaussian. This model is used to obtain the Skin Probability image of an input colour image as described in Section 5.

B. Skin colour modelling using a Gaussian Mixture Model

In the previous section, modeling of skin color using a unimodal Gaussian was considered. The reason for using a unimodal Gaussian was the localization of skin color to a small area in the CbCr chrominance space. Though the skin color values are distributed in a localized area in the chrominance space, the histogram (see Fig. 5) of the data available shows randomly distributed peaks in that region. Hence a Gaussian with a single mean may not provide a good approximation of the underlying distribution function. A mixture model consisting of a number of Gaussian components can better approximate such a distribution. In the theory of density estimation, mixture models [6] were developed to combine the advantages of both parametric and non-parametric methods of density estimation. Parametric methods estimate a density function for a given data set by calculating the parameters of a standard density function that approximately fits the given data. Parametric models allow the density function to be evaluated very quickly for new values of input data. On the other hand, non-parametric methods fit very general forms of density function to the given data. In the non-parametric method, the density function can be represented as a linear combination of kernel functions with each kernel centered on each data point [6]. This makes the number of variables in the model to grow proportional to the amount of training data. Hence evaluation of density function for new values of input data becomes computationally expensive. Mixture models provide a trade-off between the two and the method could be called semi-parametric density estimation [6].

In the modeling of skin color using a multimodal Gaussian, the probability of each color value, given it is a skin color,

is a linear combination of its probabilities calculated from the *M* Gaussian components. Thus the probability of a pixel $c = [Cb\ Cr]^T$, given it is a skin pixel, is:

$$p(c/skin) = \sum_{j=1}^{M} p(c/j).P(j) \tag{1.2}$$

where, *M* is the number of Gaussian components in the mixture model

$P(j)$ is the weighting function for the j^{th} component. It is also called the prior probability of the data point having been generated from the component *j* of the mixture.

$$p(c/j) = \frac{\exp\left[-\frac{1}{2}(c-\mu_j)^T \Sigma_j^{-1}(c-\mu_j)\right]}{2\pi\sqrt{|\Sigma_j|}} \tag{1.3}$$

where $\boldsymbol{\mu_j}$ is the mean and $\boldsymbol{\Sigma_j}$ is the covariance matrix of the j^{th} component

Note that the priors are chosen to satisfy :

$$\sum_{j=1}^{M} P(j) = 1$$

$$0 \le P(j) \le 1 \tag{1.4}$$

Hence the parameters to be estimated from the given data are the number of components *M*, mean vectors $\boldsymbol{\mu_j}$,Covariance matrices $\boldsymbol{\Sigma_j}$,and the prior probabilities $P(j)$, $j = 1$ to *M*, i.e. for each of the *M* components. A way to decide the number of components is to observe the histogram of the data and choose *M* depending upon the number and location of the peaks in the histogram. In this project, the number of components is decided automatically by a constructive algorithm [5] using the criteria of maximizing likelihood function. The details of the algorithm are described later. Once the number of components *M* is decided, the parameters, viz. the mean, the covariance, and the prior probability of each component have to be calculated from the given data. A number of procedures have been developed for determining the parameters of a mixture model from a given dataset. One approach is to maximize a likelihood function of the parameters for the given set of data. The negative log-likelihood for the dataset is given by:

$$\boldsymbol{E} = -\ln \boldsymbol{L}$$

$$= -\sum_{n=1}^{N} \ln p(c_n) = -\sum_{n=1}^{N} \ln\left\{\sum_{j=1}^{M} p(c_n/j).P(j)\right\} \tag{1.5}$$

which can be regarded as an error function. Note that *N* is the number of data point's c_n. Maximizing the likelihood $\boldsymbol{L}$ is equivalent to minimizing the error function $\boldsymbol{E}$. A special case of Maximum Likelihood (ML) techniques is the Expectation Maximization (EM) algorithm. This algorithm has been used to determine the parameters of the mixture model that best fit the data in the ML sense.

The EM algorithm [6] begins by making some initial guess for the parameters of the Gaussian mixture model, which shall be called the *'old'* parameter values. The new parameter values are evaluated using the following equations. This gives a revised estimate for the parameters which shall be called the *'new'* parameter values. The update equations move the parameters in that direction that minimizes the error function $\boldsymbol{E}$ for the data set. In the next iteration the *'new'* parameter values become the *'old'* ones and the process is repeated until convergence of the error function.

The change in the error function $\boldsymbol{E}$ is given by:

$$\Delta = \boldsymbol{E}^{\text{new}} - \boldsymbol{E}^{\text{old}} = -\sum_{n=1}^{N} \ln\left(\frac{p^{new}(c_n)}{p^{old}(c_n)}\right) \tag{1.6}$$

where $p^{new}(c_n)$ represents the probability density evaluated using the *'new'* values for the parameters and $p^{old}(c_n)$ represents the density evaluated using the *'old'* values for the parameters. Minimizing $\boldsymbol{E}^{\text{new}}$ w.r.t the *'new'* parameter values [6] the following update equations are obtained for the parameters of the mixture model:

$$\mu_j^{new} = \frac{\sum_{n=1}^{N} P^{old}(j/c_n)c_n}{\sum_{n=1}^{N} P^{old}(j/c_n)} \tag{1.7}$$

$$\Sigma_j^{new} = \frac{\sum_{n=1}^{N} P^{old}(j/c_n).(c_n-\mu_j^{new}).(c_n-\mu_j^{new})^T}{\sum_{n=1}^{N} P^{old}(j/c_n)} \tag{1.8}$$

$$P(j)^{new} = \frac{1}{N}\sum_{n=1}^{N} P^{old}(j/c_n) \tag{1.9}$$

where,

$$P^{old}(j/c_n) = \frac{p^{old}(c_n/j).P(j)^{old}}{p^{old}(c_n)} = \frac{p^{old}(c_n/j).P(j)^{old}}{\sum_{i=1}^{M} p^{old}(c_n/i).P(i)^{old}} \tag{1.10}$$

Note that the superscript *'old'* refers to the quantities evaluated using old parameter values, similarly for the superscript *'new'*.

In order to determine the number of Gaussian components (model order) required to model the skin data, a standard technique known as *cross validation* [5] is used. In this technique the available data is divided into independent training and validation sets. A number of models of different order are trained, using the training data, so as to minimize the error function described before (Expectation Maximization). The error function is computed for validation data also using the Expectation Maximized parameters for each model order. Among these models, the one with the lowest error for the validation set is considered to exhibit the best generalization and its order is taken to be

optimal. A constructive scheme involving splitting up of components and monitoring generalization ability is employed. The available data set is partitioned into disjoint training and validation sets. The algorithm begins with a small model order typically one. Model order is then adapted by iteratively applying EM and splitting components. The likelihood for the validation set is computed after every iteration. The optimal model order corresponds to the peak in this function over time. The following sub-sections describe the methods of splitting components and automatic model order selection.

Splitting components

For each component j, the total responsibility r_j for a data set is defined as:

$$r_j = \sum_{n=1}^{N} P(j / c_n) = \sum_{n=1}^{N} \frac{p(c_n / j).P(j)}{\sum_{i=1}^{M} p(c_n / i).P(i)} \qquad (1.11)$$

The component k with the lowest total responsibility for the validation set is selected for splitting:

$$k = \arg\min_j (r_j) \qquad (1.12)$$

This component k is split to generate two new components with means μ_{new1} and μ_{new2} ,and covariance matrices $\boldsymbol{\Sigma}_{new1}$ and $\boldsymbol{\Sigma}_{new2}$ as follows:

$$\mu_{new1} = \mu_k + \frac{e_1}{2} u_1$$

$$\mu_{new2} = \mu_k - \frac{e_1}{2} u_1 \qquad (1.13)$$

$$\sum_{new1} = \sum_{new2} = \sum_k$$

Where e_1 is the largest eigen value of the covariance matrix $\boldsymbol{\Sigma_k}$ and u_1 is the corresponding eigenvector. The directions of the eigenvectors of covariance matrix represent the directions of maximum variance. Hence the new mean points are determined along the direction of the maximum variance on either side of the old mean.
Also the prior probabilities for the new components are assigned as:

$$P_{new1} = P_{new2} = \frac{P_k}{2} \qquad (1.14)$$

Automatic Model Order Selection

Let i denote the iteration, M_i the number of components at the *ith* iteration and $\boldsymbol{L}_i$ the likelihood for the validation set w.r.t the model at iteration i. The initial number of components can be taken to be $M_0 = 1$. The algorithm [5] for model order selection can be outlined as follows:

1. Apply Expectation-Maximization for model with M_i components.
2. Compute $\boldsymbol{L}_i$ for validation set.
3. Save model
4. Find component j with the lowest total responsibility
5. Split component j
6. Restart from step 1 with $M_{i+1} = M_i + 1$ and $i = i + 1$.

The above sequences of steps are repeated until M_i reaches a desired value (10 in this implementation). The peak in the Likelihood function for the validation data corresponds to the optimal model order. The parameters of this optimal order model are used to calculate the probability *p(c/skin)* as in eq. (1.1).

Skin Probability image

Once the skin color is modeled using either a unimodal or a multimodal Gaussian, it can be used to calculate the probability of an input pixel representing skin, i.e, $p(skin/c)$, where c is the input color value. As evident from the previous sections, the Gaussian model(unimodal or multimodal) can be used to evaluate the probability of a color value given it is a skin color, i.e., *p(c/skin)*. This is again used to compute the required probability $p(skin/c)$ using the Bayes' formulation [3]:

$$p(skin/c) = \frac{p(c/skin).P(skin)}{p(c/skin).P(skin) + p(c/non-skin).P(non-skin)} \qquad (1.14)$$

To calculate the above probability for each input pixel, the skin and the non-skin classes are assumed to occur with equal probability [4]. Hence

$$P(skin) = P(non\text{-}skin) = 0.5 \qquad (1.15)$$

which gives,

$$p(skin / c) = \frac{p(c / skin)}{p(c / skin) + p(c / non - skin)} \qquad (1.16)$$

To obtain the probability, *p(c/non-skin)*, a similar Gaussian model is built for non-skin pixels also which is called the non-skin or the background model. The background or the non-skin color is modeled as a unimodal Gaussian here, to reduce the computational complexity of skin probability calculation. A multimodal Gaussian could have been assumed for non-skin color modeling also, but here, a unimodal Gaussian gave satisfactory results at the same time reducing the computational overhead. Finally, given an input colour image, the two conditional probabilities and the above ratio are computed pixel-by-pixel to give the probability of each pixel representing skin given its chrominance vector c. This results in a gray level image where the gray value at a pixel gives the probability of that pixel representing skin. This is called the Skin Probability image given by:

$$skin\_prob(i,j) = a.p(skin/c_{ij}) \quad (1.17)$$

where a is a suitable scaling factor and c_{ij} is the chrominance value of pixel *(i,j)*. Here a is chosen to be 255 so that the highest probability value results in a gray level of 255 in the Skin Probability image.

Face Detection and Segmentation using Connected Component Operators

Connected component operators are non-linear filters that eliminate parts of the image, while preserving the contours of the remaining parts. This simplification property makes them attractive for segmentation and pattern recognition applications. The skin probability image obtained in the previous section may contain high gray levels at non-skin regions where the background colour resembles the colour of skin. Also there will be bright regions corresponding to other parts of the human body where the skin may be exposed. These regions have to be eliminated from being considered as probable face candidate regions. As a first step, gray level 'open' operation is performed. This operation involves gray level erosion followed by gray level dilation using the same structuring element. Erosion removes small and thin isolated noise-like regions that have very low probability of representing a face. Dilation preserves those regions that are not removed during erosion. Hence, the effect of using area open is removal of small but bright regions of the skin probability image. This is followed by gray level 'close' operation. Closing is dilation followed by erosion using the same structuring element. The dilation during close operation enhances small regions of low intensity that may lie within large regions of high intensity in the skin probability image. Hence, during the thresholding step that follows, holes are not created within large high probability regions with a small gray level depression inside their periphery. These depressions may be caused due to bad lighting conditions or the skin model may fail to give a high probability in those regions. The erosion (of close operation) removes the extra pixels that may be added, during the previous dilation operation, as high probability pixels around existing regions. A smaller structuring element is used for close operation so that a large area of pixels around existing regions is not enhanced [3]. This image is then threshold into a binary image for further shape analysis. A threshold of 60 is chosen here so that large areas of relatively smaller gray levels that remain after open/close operation are not excluded from shape analysis. The connected components are labelled and isolated and shape analysis is done separately on each connected component. A hierarchy of 3 shape based connected operators [3] is used for deciding whether a component represents a face or not. These simple but effective operators rely on the combinations of the pixel area (A), perimeter (P) and the bounding box dimensions (Dx, Dy) of the connected components. Hence these have to be computed only once for the three operators. Finally a normalized area operator is used that rejects connected components that have face-like shape but have pixel area less than a certain fraction of that of the largest face component detected. The choice of this operator is based on observations made on a number of images containing multiple faces. The following is a description of the various operators used [3]:

Compactness:

Compactness (C) of a connected component is defined as the ratio of its area to the square of its perimeter.

$$Compactness = \frac{A}{P^2} \quad (6.1)$$

This criterion is maximized for circular objects. Faces are nearly circular in shape and hence face components exhibit a high value for this operator. A threshold is fixed for this operator based on the observations on various face components. If a particular component shows a compactness value greater than this threshold its is retained for further analysis, else discarded.

Solidity:

Solidity (S) of a connected component is defined as the ratio of its area to the area of the bounding box.

$$Solidity = \frac{A}{Dx.Dy} \quad (1.18)$$

The solidity also assumes a high value for face components. If the solidity of a component is lesser than a threshold value, it is eliminated, otherwise retained for further analysis.

Orientation:

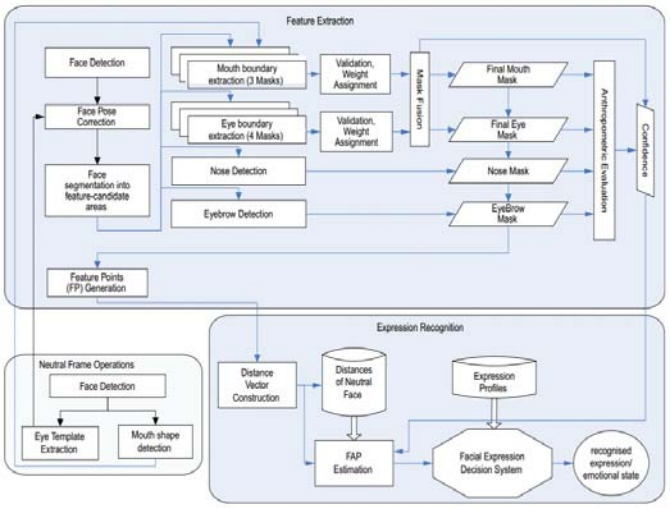
Orientation (O) is nothing but the aspect ratio of the bounding box surrounding the component.

$$Orientation = \frac{Dy}{Dx} \quad (1.19)$$

Normally, face components have orientation within a range. This range is found out based on observations on a number of images. If a component's orientation falls out of this range, the component is eliminated.

Normalized Area:

It is the ratio of the area of the connected component to that of the largest component that remains after the application of the above three operators. In images containing multiple faces it is assumed that the smallest face component has an area that is not less than a certain fraction of the largest face component. The connected components that remain after the application of all the above operators contain faces. The regions in the original image that lie within the bounding boxes of these connected components are faces and are segmented out. The segmented faces can be input to an application that requires isolated faces for further processing.



CONCLUSION:

This research work presents a comprehensive study and analysis of system to find the Emotion Recognition. Hybrid classifier often involves the method of specifying a set of simple rules and a method of iteratively applying those rules. The hybrid classifier approach for the proposed system is found hopefully to be more efficient than the existing digital image processing techniques, since it presents a multiple classifier system for efficient facial feature extraction to increase recognition accuracy.

REFERENCES

1. PNAS Plus: Compound facial expressions of emotionPNAS 2014 ; published ahead of printMarch 31, 2014
2. Lyusin, Dmitry and Ovsyannikovw, Victoria, A New Videotest for measuring Emotion Recognition Ability (February 12, 2014). Higher School of Economics Research Paper No. WP BRP 16/PSY/2014.
3. P. Marasamy, S. Sumathi‟‟ “ Automatic Recognition and Analysis of Human Faces and Facial Expression by LDA using Wavelet Transform”‟‟ International Conference on computer Communication and Informatics ICCCI 2012.
4. S. Logesh, S. Arun Bharathi, P.V.S.S.R. Chandra Mouli ‘’An efficient and robust face detection method using Neuro-Fuzzy Approach‟‟‟‟ School of Computing Science and Engineering VIT University, Vellore, India 2011 International Conference on Image Information Processing (ICIIP 2011).
5. Segmentation and Template „‟ College of Information Science and Engineering, Wuhan University of Science and Technology Wuhan, China E-mail:chenaiping00@163.com 2010 Second International Workshop on Education Technology and Computer Science.
6. Ming-Yuan Shieh,Choung Ming Hsieh, Jian-Yuan Chen,Jeng –Han Li, “ PCA and LDA Based Fuzzy Face Recognition System” SICE Annual Conference Taipei, Taiwan-2010