ERUDITION ADAPTED MODEL FOR FACIAL APPEARANCE ANALYSIS AND GESTICULATION RECOGNITION

Karthika Rajarethinam[1],Shanmathy Arivinbam[2], M.R.Rajeswari[3],VeeralakshmiPonnuramu[4]

[1] *B.Tech, IT, Prince Shri Venkateshwara Padmavathy Engineering College, TN, India*

[2] *B.Tech, IT, Prince Shri Venkateshwara Padmavathy Engineering College, TN, India*

[3] *B.Tech, M.E, IT, Prince Shri Venkateshwara Padmavathy Engineering College, TN, India*

[4] *M.E, Ph.D , IT, Prince Shri Venkateshwara Padmavathy Engineering College, TN, India*

ABSTRACT

Image processing is the analysis and manipulation of a digital image, especially in order to improve its quality. This system is used to detect the human emotions from the input image. First an image is given as input,skin is segmented based on skin colour. The face region is then scanned by marking the edges of the face and the connected region is then cropped. From the cropped face image the eyes and the lips are separated. It draws Bezier curve for eyes and lips. Emotionof the Bezier curve is assigned as emotion of this image. The proposed approach in different applications: pain recognition and action unit detection using visual data and gestures classification using inertial measurements, demonstrating the generality of our method with respect to different input data types and basic classifiers. In terms of accuracy and computational time both with respect to user-independent approaches and to previous personalization techniques this project succeeds. This paper presents a framework for personalizing classification models which does not require labelled target data. It proposes a regression framework which exploits auxiliary annotated data to learn the relation between person-specific sample distributions and the parameters of the corresponding classifiers. Then, when considering a new target user, the classification model is computed by simply feeding the unlabelled sample distribution into the learned regression function.

Keywords :- *Bezier curve, binary conversion, Emotion detection,Sobel Filter;Image contrast, Skin color, Unlabeled dataset, Webcam User interface.*

1. INTRODUCTION

This paperdetects and recognises the face of the human. Then the emotions are classified based on human expressions. But generally a question arises that what is the difference between Face Recognition and Face Detection. But the answer is very simple. Face detection is a program that determines the locations of human faces in a digital image. Face recognition is a program that identifies a person in a digital image. First phase of face recognition is face detection. A facial expression is one or more motions or positions of the muscles beneath the skin of the face. According to one set of controversial theories, these movements convey the emotional state of an individual to observers. Facial expressions are a form of nonverbal communication. This Project used to detect human emotion from image.

In the previous work the image has been stored in the database, so the new image which is capture is been compared with the pervious image and gives the most nearest outcome. Just finding the similarity alone will not help in real world scenarios. For that case emotions are been detected. Still there are papers where emotions are also detected but those are from labelled data set only.

At the proposed system unlabeled data are used to detect the face and give the corresponding emotion of the image. It is the very first project where five emotions are detected. In this process various kernels are been used to check which kernel gives the best outcome. It uses the Bezier curve algorithm which yields an effective outcome regarding the personalized framework specifically in the case of gesticulation recognition. To increase the efficiency various edge detection techniques are also used so that noise, illusion, clarity, lightning, shadow effects are removed.

Our experimental evaluation is conducted on two datasets. First set is based on Pain recognition. We use the recent PAINFUL dataset, which collects videos of patients with shoulder injuries, and we devise a patch-based facial expression recognition approach based on Local Binary Pattern Histograms (LBPH) for better equalization of image contrast. Again, user-specific models are of utmost importance in this context as the way in which the patients spontaneously show pain varies considerably between different people. Second set is based on Action unit where the expression ofdifferent users has been analyses by using Extended Cohn Kanade (CK+) dataset.

This paper significantly improves the efficiency than the previous work. Specifically in the case (i) it's a novel transfer approach (ii) reduces the noise relates issues as it uses various edge detection techniques (iii) reduce the time complexity issue as the conditions are inbuilt (iv) reduces space complexity due to unlabeled dataset for reference and inbuilt condition (v) in the case of SVM out of boundary values are sometimes been taken into account, leads to wrong mismatch that's reduced using Bezier curve (vi) increased count of emotion for unsupervised learning approach.

The paper is organized as follows. Section II reviews the concept and work that needs to be understood before moving in-depth to the project. Section III introduces our approach, together with the considered application scenarios. It describes how the entire internal process works. The results of our experimental evaluation are presented in Sec. IV, and conclusions are drawn in Sec. V.

2. RELATED WORK

In this section we review the various concepts and approaches that are dealt in this systemto get the clear perspective. As far as we know, no previous work has been indulged with more than 4 emotions. OpenCVlibrary has been implemented for open source implementation.

2.1 Histogram Equalization

Histogram equalization is a technique for adjusting image intensities to enhance contrast.It is not necessary that contrast will always be increase in this. There may be some cases were histogram equalization can be worse. In those cases the contrast is decreased. To perform histogram equalization two steps are vital. 1) First have to calculate the Probability mass function value (PMF). 2) Then have to calculate the Cumulative distributive function (CDF). In the overall process image stress or any dimensions are not altered only the contrast of the image has been equalised. To transform the gray levels of the image so that the histogram of the resulting image is equalized to become a constant[3]. We also assume all pixels within the gray scale interval of the input image are mapped to the corresponding range of the output image. As the number of pixels being mapped remains unchanged. For the histogram of the output image to be equalized, it needs to be constant 1.

2.2 Kernel factors

The use of Kernels - also known as convolution matrices or masks - is invaluable to image processing. Techniques such as blurring, edge detection, and sharpening all rely on kernels[1] - small matrices of numbers - to be applied across an image in order to process the image as a whole. Each pixel is represented by a number - depending upon the image format these numbers can vary: for an 8 bit RGB image each pixel has a red, green, and blue component with a value ranging from 0 to 255. A kernel works by operating on these pixel values using straightforward mathematics to construct a new image. Let's take the above kernel and do some math: for each pixel, centre the kernel over the pixel, multiply the kernel values times the corresponding pixel values, and add the result - this final value is the new value of the current pixel. As each pixel is processed, a new image emerges based upon the calculated values. The new image is highly dependent upon the kernel used - each kernel has specific properties depending upon its values. Their types are as follows,

- ***Edge detection*:** This kernel detects edges within an image. If all pixel values are comparable, the resultant pixel value will be close to 0. However, edges - locations with extreme differences in pixel values - will result in values far from zero.
- ***Gaussian Blur:*** This kernel is similar to the blur kernel presented above, but is different in that it is dependent upon the Gaussian function [4]- a function which creates a distribution of values around the centre point. This results in a kernel in which pixels near the canter contribute more towards the new pixel value than those further away.
- ***Sharpening:*** This kernel sharpens an image - accentuating the edges of the image. Sharpening an image add contrast to edges, and a 3x3 version of this mask is similar to the edge detection kernel with a centre value of 5. This adds contrast around an edge by accentuating bright and dark areas.

- ***Unsharp Mask:*** Used to sharpen an image, this technique is based upon first creating a Gaussian blurred copy of the image[6]. This blurred copy is then subtracted from the original - pixels above a given threshold are sharpened by enhancing light and dark pixels.

Kernels can be of just about any size. More sophisticated kernels are typically larger, in fact many image processing software packages have options to customize a kernel. Customizing a kernel in this manner can be a time consuming, trial and error process. However this technique provides a great deal of flexibility in creating new ways to process an image, or fine-tuning older well established workflows.

Each kernel is used in different means yet all are useful in one or the other ways. The existing system uses Gaussian whereas the propose uses edge detection technique for better enhancement.

3. PROPOSED SYSTEM

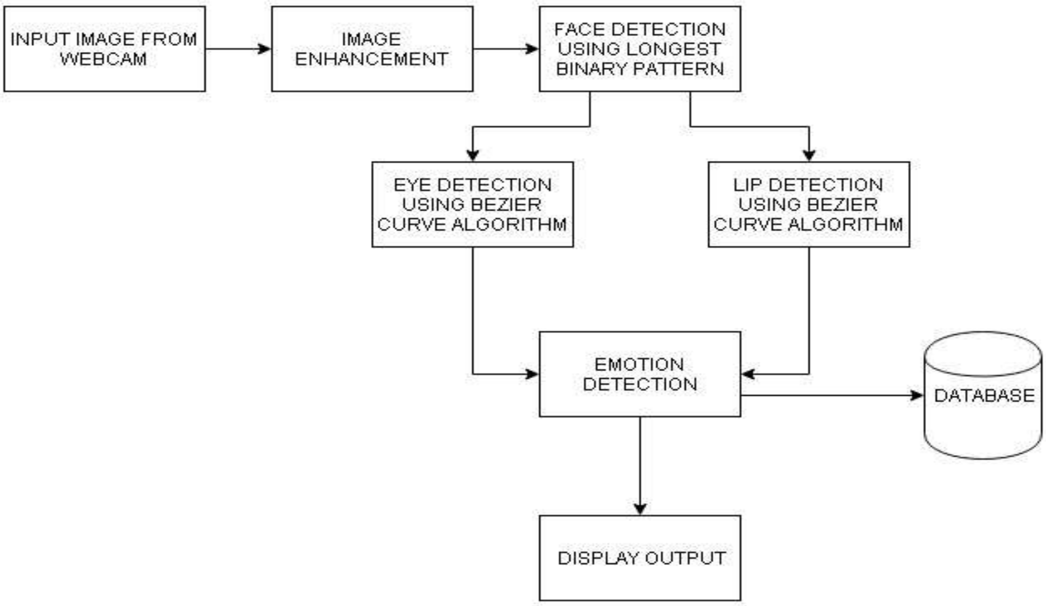

Fig-1 System architecture

3.1 INPUT FROM WEBCAM

For giving input to the system we use webcam. The webcam starts streaming and by clicking the capture button the required image is captured and it is stored in bin. The image is browsed from the desired directory and the image is processed.

3.2 IMAGE ENHANCEMENT

For skin colour segmentation, first we contrast the image. Then we perform skin colour segmentation. Then, we have to find the largest connected region. To check the probability to become a face of the largest connected region. If the largest connected region has the probability to become a face, then it will open a new form with the largest connected region. If the largest connected regions height & width is larger or equal than 50 and the ratio of height/width will be between 1 to 2, then it may be face.

Although skin colour appears to vary, we assume that there exists underlying similarities in the chromatic properties of all faces and that all major differences lie in intensity rather than in the facial skin colour itself. In this case, we have adopted to utilize a skin-color based approach using YCbCr colour model [7]. With the colour model, luminance information is represented by a single component, Y, and colour information is stored as two colour difference component, Cb and Cr [5].

$$Y = 0 + (0.299 \cdot R) + (0.587 \cdot G) + (0.114 \cdot B)$$
$$Cb = 128 - (0.168736 \cdot R) - (0.331264 \cdot G) + (0.5 \cdot B)$$

Cr=128+(0.5·R)−(0.418688·G)−(0.08131·B)
(1)

After converted colour model, an illumination calibration is pre-requisite during the pre-processing for the accurate face detection. Since the illumination condition is an important factor to effect on the performance of detection, we attempt pre-processing to equalize the intensity value in an image as follow;

Y0 = (y−min1 max1 −min1) ˙ (max2 −min2) +min2, i f(y ≤ Kl or Kh ≤ y) (2)

where min1 and max1 are minimum and maximum value of Y component on input image, min2 and max2 are the value of the transformed space, Kl = 30 and Kh = 220. The values of these experiential parameters are estimated from training data sets of skin patches of sample database. Histogram equalization enhances the performance which brightness is second into one direction,

Pk(rk) = nk n, 0 ≤ rk ≤ 1, k = 0,1,···,l −1
(3)

where l is the number of discrete values for the intensity, n is the number of total pixels in the image, rk is the k*th* intensity, and nk is the number of pixels which intensity is rk. Frequency cumulativeness is dependent on rk, thus, following Equation (3) can be used for the intensity equalization.

sk = k ∑ j=0 nk n = E(rk) = k ∑ j=0 Pr(rj) (4)

3.3 FACE REGION DETECTION

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[2]. 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. 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.

3.4 EYES DETECTION

For eyes detection, we convert the RGB face to the binary face. Now, we consider the face width by W. We scan from the W/4 to (W-W/4) to find the middle position of the two eyes. The highest white continuous pixel along the height between the ranges is the middle position of the two eyes.

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.

3.5 LIP DETECTION

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.

3.6 APPLY BEZIER CURVE ON LIPS

In the lip box, there is lip and may be some part of nose. So, around the box there is skin colour or the skin. So, we convert the skin pixel to white pixel and other pixel as black. We also find those pixels which are similar to skin pixels and convert them to white pixel. Here, if two pixels RGB values difference is less than or equal 10, then we called them similar pixel. Here, we use histogram for finding the distance between the lower average RGB value and higher average RGB value. If the distance is less than 70, then we use 7 for finding similar pixel and if the distance is greater than or equal 70 then we use 10 for finding similar pixel. So, the value for finding similar pixel depends on the quality of the image. If the image quality is high, we use 7 for finding similar pixel and if the image quality is low, we use 10.

So, in the binary image, there are black regions on lip, nose and may some other little part which have a little different than skin colour. Then we apply big connected region for finding the black region which contain lip in binary image. And we are sure that the big connected region is the lip because in the lip box, lip is the largest thing which is different than skin. Then we have to apply Bezier curve on the binary lip. For apply Bezier curve, we find the starting and ending pixel of the lip in horizontal. Then we draw two tangents on upper lip from the starting and ending pixel and also find two points on the tangent which is not the part of the lip. For the lower lip, we find two point similar process of the upper lip. We use Cubic Bezier curves for draw the Bezier curve of the lip. We draw two Bezier curve for the lip, one for upper lip and one for lower lip.

3.7 APPLY BEZIER CURVE ON EYES

For apply Bezier curve on eyes, first we have to remove eyebrow from eye. For remove eyebrow, we search 1stcontinuous black pixel then continuous white pixel and then continuous black pixel from the binary image of the eye box. Then we remove the 1st continuous black pixel from the box and then we get the box which only contains the eye.

Now, the eye box which contains only eye, has some skin or skin colour around the box. So, we apply similar skin colour like the lip for finding the region of eye. Then we apply big connect for finding the highest connected region and this is the eye because in the eye box, eye is the biggest thing which is not similar to the skin colour. Then we apply the Bezier curve on the eye box, similar to the lip. Then we get the shape of the eye.

Considering global shape information with the curve passing through the first and last control points [16]. If there are L+1 control points, the position is defined as $P_k: (x_k, y_k), 0 \leq k \leq L$ considering 2D shapes. These coordinate points are then blended to form P(t), which describes the path of Bezier polynomial function between P_0 and P_L

$$P(t) = \sum_{k=0}^{L} P_k BEZ_{k,L}(t) \quad (5)$$

where the Bezier blending function $BEZ_{k,L}(t)$ is known as the Bernstein polynomial, which is defined as

$$BEZ_{k,L}(t) = \binom{L}{k} t^k (1-t)^{L-k} \quad (6)$$

The recursive formula which are used to decide coordinate locations is obtained by

$$BEZ_{k,L}(t)=(1-t)\cdot BEZ_{k,L-1}(t)+t\cdot BEZ_{k,L-1}+t\cdot BEZ_{k-1,L-1}(t) \quad (7)$$

where $BEZ_{k,k}(t) = t^k$ and $BEZ_{0,k}(t) = (1-t)^k$. The coordinates of individual Bezier curve are represented by the following pair of parametric equations

$$x(t) = \sum_{k=0}^{L} x_k BEZ_{k,L}(t) \quad y(t) = \sum_{k=0}^{L} y_k BEZ_{k,L}(t) \quad (8)$$

3.8 DATABASE AND TRAINING

In our database, there are two tables. One table “Person” is for storing the name of people and their index of 5 kinds of emotion which are stored in other table “Position”. In the “Position” table, for each index, there are 6 control points for lip Bezier curve, 6 control points for left eye Bezier curve, 6 control points for right eye Bezier curve, lip height and width, left eye height and width and right eye height and width. So, by this method, the program learns the emotion of the people.

3.9 EMOTION DETECTION

For emotion detection of an image, we have to find the Bezier curve of the lip, left eye and right eye, refer Figure 1. Then we convert each width of the Bezier curve to 100 and height according to its width[5]. 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.

4. EXPERIMENTAL EVALUTION

The expressions such as smile, sad, surprise, normal and ambiguous are considered for the experiment of face recognition. All the face images are normalized using some parameters. The Bezier points are interpolated over the principal lines of facial features. These points for each curve form the adjacent curve segments. The distance is calculated based on the curve segments. Then, understanding and decision of facial emotion is chosen by measuring similarity in faces. The ground truth set for estimating the performance of the algorithm is provided with the categories in the experiments, which are correct if the decision is belonged to the correct category. Figure shows how to move the control points across different subjects (e.g., sad and smile) and to interpret a tracking facial features with Bezier curve and extracted feature points interpolation. To categorize facial emotion, we need first to determine the expressions from movements of facial control points

5. CONCLUSION

In this paper, a simple approach for recognition of the facial expression analysis is presented and implemented. The algorithm performs two major steps: one is a detection of facial region with skin color segmentation and calculation of feature-map for extracting two interest regions focused on eye and mouth. And the other is a verification of the facial emotion of characteristic features with the Bezier curve and sobel algorithm. Experimental results shows average successful ratio of 78.8% to recognize the facial expression, and this indicates the good performance and enough to applicable to mobile devices.

6. REFERENCES

[1]. Bettadapura, V. (2012). Face expression recognition and analysis: the state of the art. arXiv preprint arXiv:1203.6722.
[2]. Zhu, X., &Ramanan, D. (2012, June). Face detection, pose estimation, and landmark localization in the wild. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on (pp. 2879-2886). IEEE.
[3]. Rivera, A. R., Castillo, J. R., &Chae, O. O. (2013). Local directional number pattern for face analysis: Face and expression recognition. IEEE transactions on image processing, 22(5), 1740-1752.
[4]. Agrawal, S., & Khatri, P. (2015, February). Facial expression detection techniques: based on viola and jones algorithm and principal component analysis. In Advanced Computing & Communication Technologies (ACCT), 2015 Fifth International Conference on (pp. 108-112). IEEE.
[5]. Verma, A. & Sharma, L.K.(2013). Acomprehensive survey on human facial expression detection. International Journal of Image Processing (IJIP), 7(2), 171.
[6]. Li, X., Pfister, T., Huang, X., Zhao, G., &Pietikäinen, M. (2013, April). A spontaneous micro-expression database: Inducement, collection and baseline. In Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on (pp. 1-6). IEEE.
[7]. S. V. Tathe, A. S. Narote, S. P. Narote (December, 2016) Face Detection and Recognition in Videos India Conference (INDICON), 2016 IEEE Annual