Position Determination and Face Detection Using Image Processing Techniques and SVM Classifier

Gogula Suvarna Kumar[1], P.V.G.D. Prasad Reddy[2],
Sumit Gupta[1], and Ravva Anil Kumar[1]

[1] Department of Computer Science Engineering
Maharaj Viajayam Gajapathi Raj College of Engineering
Vizianagarm, Andhra Pradesh, India
{gsk,prasadreddy.vizag,sumit108,ravvaanilkumar}@gmail.com
[2] Department of Computer Science and Systems Engineering
Andhra University Vishakapatnam, India

Abstract. In this paper, an improved algorithm for detecting the position of a person in controlled environments using the face detection algorithm is proposed. This algorithm ingeniously combines different face detection, occlusion detection algorithms and SVM classifier. A class room environment with thirty students is used along with some constraints such as position of the camera being fixed in a way that covers all the students, the static student's position and the class environment with the fixed lighting conditions. The students are treated as classes in this technique. For every class, a set of 6 attributes are derived and updated in a database. The image is given as an input to the face detection algorithm to detect some of the faces. Some faces are not detected because of occlusion, so an occlusion detection technique is implemented to detect all the occluded faces. In the training phase, a set of four images with the entire thirty students taken in four different days is used. Therefore a database of total 120 set of records with 6 attributes is used.

Keywords: Face Detection, Occlusion Detection, SVM, EMD.

1 Introduction

In the last decade, it can be observed that many algorithms were developed in image processing for face recognition [1] and face detection [2] but there was no algorithm for detecting a position in an image. The position detection in the controlled environment can be used in many environments where the positions are fixed. Some environments like seminar halls, conference halls, Auditoriums etc. And we can apply the same algorithm for some environments where the positions are reserved. With the help of this position detection algorithm many applications can be developed like automatic attendance system [3] in controlled environments. This technique can also be used for improving the face detection technique although no face detection technique will give the 100 % accuracy in its respective detection. Some of the faces in an image are blurred, not clear, occluded, the normal face detection algorithm will not detect those faces but this technique detects most of the faces in an image. First

S.C. Satapathy et al. (Eds.): Proc. of the Inter. Conf. on Inform. Systems, AISC 132, pp. 227–236.
springerlink.com

the image is given as an input to the basic face detection algorithm followed by giving the acquired result of the above as an input to the occlusion detection [4] and then this technique is applied.

A large number of face detection algorithms are derived from neural network approach, algorithmic approach [5] and some image morphological techniques [6]. However most of the works concentrate on single face detection, with some constrained environments. In this proposed technique, a face detection algorithm by using local SMQT features and split up snow classifier [7] and an occlusion detection algorithm is used and from the result obtained the attributes are derived and updated in the database. A face detection algorithm is used which is implemented using the Local SMQT Features and split up Snow Classifier.

2 Existing System

Many face detection algorithms are derived. Some of the face detection algorithms use neural network approach, algorithmic approach and some image morphological techniques. With the help of these existing face detection algorithms, 80% of accuracy is obtained. When the result is given as an input to the occlusion detection algorithm, the accuracy is improved to 95% accuracy. Then the above acquired result is given as an input to the proposed technique. Face detection algorithm using Local SMQT Features and Split up Snow classifier is explained in the next paragraph.

Illumination and sensor variation are major concerns in visual object detection. It is desirable to transform the raw illumination and sensor varying image so the information only contains the structures of the object. Some techniques previously proposed to reduce this variation are Histogram Equalization (HE), variants of Local Binary Patterns (LBP) [8] and the Modified Census Transform (MCT) [9]. HE is a computationally expensive operation in comparison to LBP and MCT, however LBP and MCT are typically restricted to extract only binary patterns in a local area. The Successive Mean Quantization Transform (SMQT) [10] can be viewed as a tunable tradeoff between the number of quantization levels in the result and the computational load.

The SMQT uses an approach that performs an automatic structural breakdown of information. Let x be one pixel and $D(x)$ be a set of $|D(x)| = D$ pixels from a local area in an image. Consider the SMQT transformation of the local area.

$$\text{SMQT (local)}:D(x)\text{->}M(x) \tag{1}$$

These properties are desirable with regard to the formation of the whole intensity image $\mathrm{I}(x)$ which is a product of the reflectance $\mathrm{R}(x)$ and the illuminance $\mathrm{E}(x)$. Additionally, the influence of the camera can be modelled as a gain factor g and a bias term b. Thus, a model of the image can be described by

$$I(x) = gE(x)R(x) + b \tag{2}$$

The SNoW learning architecture is a sparse network of linear units over a feature space [9]. One of the strong properties of SNoW is the possibility to create lookup-tables for classification. Consider a patch$W$ of the SMQT features$M(\mathbf{x})$, then a classifier.

$$\theta=\sum_{x\epsilon W} h_x^{nonface}\left(M(X)\right)-\sum_{x\epsilon W} h_x^{face}\left(M(x)\right) \tag{3}$$

can be achieved using the nonface table $h_x^{nonface}x$, the face table h_x^{face} and defining a threshold for θ. Since both tables work on the same domain, this implies that one single lookup-table.

$$h_x = h_x^{nonface} - h_x^{face} \tag{4}$$

3 Proposed System

This proposed technique was implemented with some attributes derived from an image and the framework is shown in Figure-1. The attributes are derived for every face. In this technique, the faces are treated as a single class. For every single class, six attributes are derived. These attributes have been derived for 4 different images taken in four different days. The attributes are length and height of the face, position of the face in co ordinates, and the number of horizontal lines and vertical lines that are passed from a single class. The framework of the proposed system is shown in the figure 1. In the first step the a sample image, figure 2 is given as an input to the face detection algorithm [2] whose result is shown in figure 3, here some of the faces that are not detected are the ones that are occluded [4], this is shown in figure 4. To detect the occluded faces, the above obtained result is given as an input to the occlusion detection algorithm procedure. The output of the occlusion detection algorithm has been shown in figure 5. Here the overview of the database used in this technique is explained.

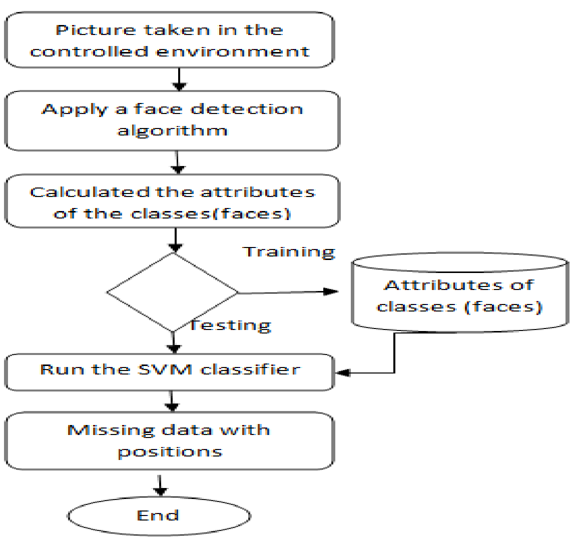

Fig. 1. The Framework of the proposed system

Fig. 2. A Sample image in a dataset

Fig. 3. The output of the detection algorithm

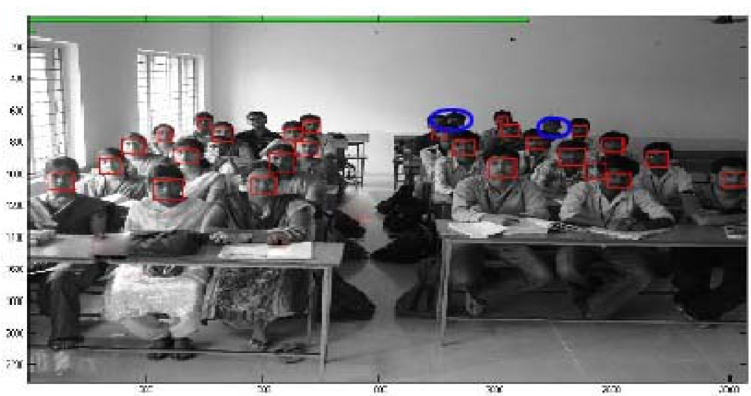

Fig. 4. The occluded faces are highlighted and shown

The dataset of four different days of 30 different classes are updated and maintained in the database followed by the training phase those attributes are used for finding the results. In this technique a closed circuit camera is used to take the pictures. The lighting conditions should be constant.

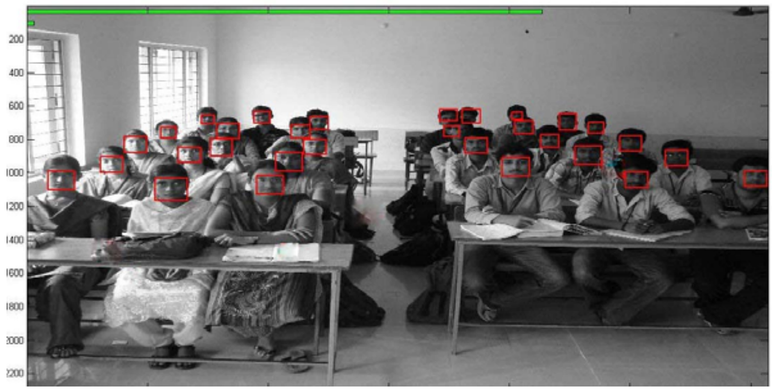

Fig. 5. The result of the Occlusion Detection

Now the process is explained in terms of consecutive steps. In the first step, the image of the classroom is taken and given as an input to the face detection for detecting the faces and creating the bounding boxes around the faces. The sample training image is shown in the figure 2. The training is given with four different images taken in four different days and a single sample image is shown in figure 2.

3.1 First Step

In the first step, a line equation [13] is calculated by using all the points which are selected from all the four training sets of images. The line equation is drawn separately for the rows and columns. A set of five points are taken from every single column of every image and with the help of those 20 points that were selected from those five images, a slope is calculated. The slope is used to plot a curve [13] on the column of every image and this curve will pass almost all the faces in a column. This line is plotted on the first column and the same process and is used to plot all the curves on all columns of an image. The same process is applied for all the rows in the image to draw the curves on all the rows. Therefore a grid is created on the image. The output of the first step is shown in step 4. A grid is formed on all the train images. Two attributes are derived from all the train images with the help of the grid on it. The attributes are the number of horizontal and vertical lines that pass through the faces. The data is derived for all the thirty classes in all the four training images. This will be used for the training stage. Proceeding to the second step, two attributes are updated in the database. The below equations are used to find the best curve which passes through all the faces in the rows and columns.

The equation 5 is used for finding the curve fitting for all the points which are selected in this step. The output along with the grid is shown in figure 6. The below line equation has been used to find the best fitting curve which passes through all the faces in a row and as well as the columns. The line equation p(x) determines the best fit curve by using all the coordinates of faces in the row and column.

$$P(x) = p_1 x^n + p_2 x^{n-1} + p_2 x^{n-2} + p_3 x^{n-3} + \cdots + p_n x + p_{n+1} \tag{5}$$

Fig. 6. A grid on the image with horizontal and vertical lines

3.2 Second Step

The second step is used to find the length and height [13] of the faces. Lengths and heights are calculated for all the classes in the image. The calculated lengths and heights of all the faces are updated in the database for training. The respective process is now used to find the lengths and heights of all the four training images. The below geometrical equation is used to find the length and height of each and every class.

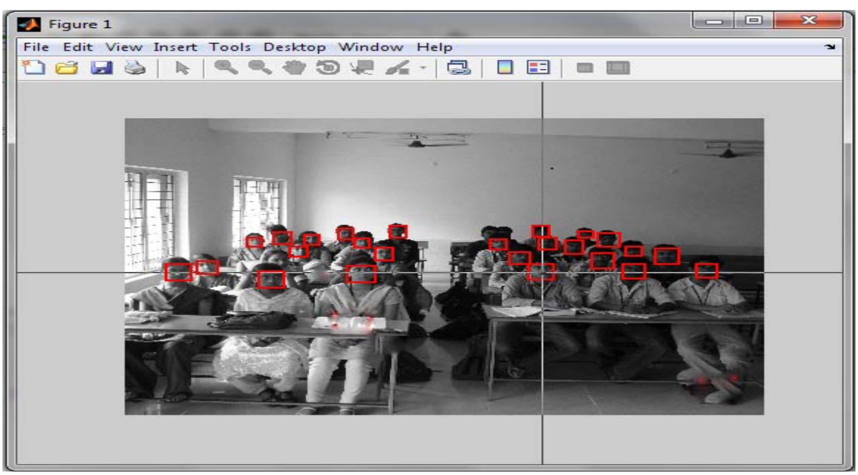

Fig. 7. A snapshot to show how the coordinates are selected

In the second step, all the co ordinates [13] are selected for calculating the length and height of every class. This process has been shown in the figure 7. The coordinates need to be selected for calculating the lengths and heights of the classes. Firstly, all the x coordinates are selected followed by the selection of y coordinates and finally, the z coordinates are selected. Now, with the help of the equations 6 and 7, the lengths and heights of all the classes are derived and updated. The lengths (between x and y), heights (between x and z), positions coordinates, and the number of horizontal lines and vertical lines are passed through the single class. The mathematical equations for calculating the length and height of the class are shown. The respective points are selected from the classes. Three different points are picked from the three different vertex points. The equation 6 and 7 has been used find the length and height of the classes respectively. The equations are derived from the basic co ordinate geometry. The equations are listed below and numbered.

$$L=\sqrt{((x(i+1)-x(i))^2+(y(i+1)-y(i))^2)} \tag{6}$$

$$H=\sqrt{((x(j+1)-x(j))^2+(y(j+1)-y(j))^2)} \tag{7}$$

3.3 SVM Classifier

SVM is a classifier derived from statistical learning theory by Vapnikand Chervonenkis [12]. SVMs are introduced by Boser, Guyon and Vapnikin COLT-92. It was initially popularized in the NIPS community but now is an important and an active field of every Machine Learning research.

Main features:

- By using the kernel trick, data is mapped onto a high-dimensional feature space without much of the computational efforts;
- Maximizing the margin achieves better generation performance;

SVM requires that each data instance is represented as a vector of real numbers. Hence, if there are categorical attributes, we have to convert them into numeric data. We recommend using m numbers to represent an m-category attribute. Only one of the m numbers is one, and others are zero. For example, a three- category attribute such as red, green, blue can be represented as (0,0,1), (0,1,0), and (1,0,0). Scaling before applying SVM is very important. The main advantage of scaling is to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges. Another advantage is to avoid numerical difficulties during the calculation.

4 Experimentations and Results

This technique is implemented in MATLAB and some part of the technique using the c language. This new technique was implemented and run over a classroom database. The database consists of classroom images of 20 different days. The total classes (faces) present in a single image are 30. In the entire 20 days database, the positions of the thirty classes are fixed and. And a graphical representation of an accuracy of

Table 1. Comparisons of Face Detection algorithms

Number of faces = 30	Skin color based algorithm	Face detection SMQT with split up snow classifier	Face detection with Occlusion detection	Combination of all the algorithm
False Detection Rate (%)	12.5%	16%	11%	3.5%
False Dismissal Rate (%)	20%	24%	8%	2%
Accuracy	67.5	60%	81%	94.5%

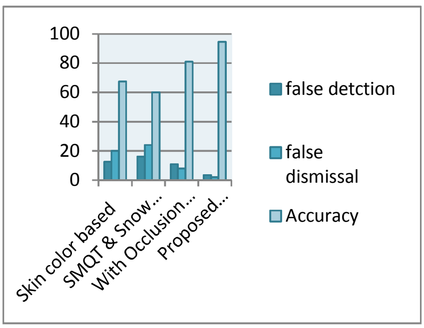

Fig. 8. The Bar Chart to show the comparison of the proposed detection algorithm and the existing algorithms

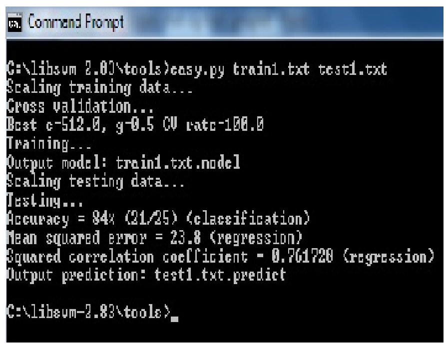

Fig. 9. The training of the SVM

Fig. 10. The output of the Classifier missing data was plotted

these algorithms is shown in figure 8. Here the result is created in a text file named predict. This file will give the missing values in the training image. This resultant text file is used to highlight the missing classes in the test image.

Figure 9 shows the accuracy of the System. The missing data is plotted with a blue plot boxes and this has shown in figure 10. The class should be in the intersection of both horizontal and vertical lines, if it is not there then it is missing class, which is marked by the classifier.

5 Conclusions and Future Works

In this paper, a novel technique for detecting missing faces and properly mapping them to specific individuals has been presented. A system based on horizontal and vertical depth lines along with position coordinates has been used as input for SVM Classifier. The results are promising and a good performance is observed in spite of large number of faces and poor illumination conditions.

Also, the applications of the proposed methodology can be extended to various environments where the seating arrangements are fixed like air travel, train travel, seminar halls, laboratories, etc., and advanced curves can be employed for the same.

References

[1] Turk, M., Pentland, A.: Eigenfaces for Recognition. Journal of Cognitive Neuroscience 3(1), 71–86 (1991)

[2] The Face Detection Homepage, `http://home.t-online.de/home/Robert.Frischholz/index.html`

[3] Kawaguchi, Y., Shoji, T., Lin, W., Kakusho, K., Minoh, M.: Face Recognition based Attendance System, `http://www.mm.media.kyoto-u.ac.jp`

[4] Lawrence Zitnick, C., Kanade, T.: A Cooperative Algorithm for Stereo Matching and Occlusion Detection. CMU-RI-TR-99-35

[5] Vijaya Lakshmi, H.C., Patil Kulakarni, D.: Segmentation algorithm for multiple face detection in color images with skin tone regions using color spaces and edge detection techniques. International Journal of Computer Theory and Engineering, 1793–8201 (2010)

[6] Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 2nd edn. Pearson Education, India (2002)

[7] Nilsson, M., Nordberg, J., Claesson, I.: Face Detection Using Local SMQT Features and Split up SNOW Classifier 20(12), 1222–1239 (2006)

[8] Lahdenoja, O., Laiho, M., Paasio, A.: Reducing the feature vector length in local binary pattern based face recognition. In: IEEE International Conference on Image Processing (ICIP), vol. 2, pp. 914–917 (September 2005)

[9] Froba, B., Ernst, A.: Face detection with the modified census transform. In: Sixth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 91–96 (May 2004)

[10] Nilsson, M., Dahl, M., Claesson, I.: The successive mean quantization transform. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 4, pp. 429–432 (March 2005)
[11] Yang, M.-H., Kriegman, D., Ahuja, N.: Detecting faces in images: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 24(1), 34–58 (2002)
[12] Osuna, E., Freund, R., Girosi, F.: Training support vector machines:an application to face detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 1997), pp. 193–199 (1997)
[13] `http://www.mathopenref.com`