A PROPOSED RDIT ALGORITHM FOR ANALYSIS OF FACE VERIFICATION IN VIDEOS IN IMAGE DATABASE FOR CRIME INVESTIGATION

V.S. Manjula,
Asst. Professor,
Dept. of Computer Application,
St. Joseph' College ,
Chennai- 600 122, INDIA.

Lt. Dr. S. Santhosh Baboo,
Reader, P.G & Research,
Dept. of Computer Application,
D.G. Vaishnav College,
Chennai-106. INDIA.

Abstract
In the face recognition most researches has considered about face detection and identification or a face detection and tracking, but best of our knowledge very few researches has focused on the face detection, identification and tracking. The combination of these three mechanism are been used in many real time applications .Here we consider one of the application as theft detection in real time which proposes a new mechanism named as FACE RDIT (Face Rectangular Detection, Identification and tracking) algorithms. Also here we include some preprocessing algorithms like gray scale conversion and histogram equalization for image enhancement process. Apart from this we also derive a new algorithm which is based on rectangular features of the face and multi linear training. Also the proposed technique has four main steps like training, detection, identification and tracking by using this proposed method we train the unauthorized users photo and then make the detection of the faces in the group of frames and then identify the particular personality from the group of images. Finally it tracks the identified thief in next following video frames by our proposed algorithm. Thus the effectiveness of the proposed methods is well supported by both detailed analysis and extensive experimentation on a face verification problem.

***Keywords*:** RDIT, face Rectangular Detection, histogram equalization, rectangular features.

1. Introduction

The Face detection is one of major research area in computer vision. The task of face detection is so trivial for the human being, yet it still remains a challenging and difficult problem to enable a computer to do face detection. Many difficult problem are caused by a diversity of variations, such as human races, illumination, facial expression, contrast between face and background, face regions overlapped one another and orientation of the face. The efficient detection of human faces in images, however, is fundamental in a variety of applications requiring intelligent human computer interaction[1].

We proposed used mechanism to several real time applications. we can consider the thief detection in the video frame .base on that we can propose the new mechanism named as FACE RDIT (Face Rectangular Detection, Identification and tracking).here we utilize some preprocessing algorithms like gray scale conversion and histogram equalization for image enhancement. Than we derive the new algorithm based on rectangular features of the face and multi linear training .in the proposed technique have four main steps they are training, detection, identification and tracking .that is we can train the thief photo image, than detect the faces in the video frame and than identify the thief face [2, 3]. Finally track that identified thief in next following video frames by our proposed algorithm. Face detection and tracking find applications in areas like video structuring, indexing, and visual surveillance and form active areas of research. If the application is to identify an actor in a video clip or to find a particular shot in the video sequence in which the actor is playing, then faces are the most important "basic units"[4] . This requires detection and tracking of a face through a sequence [5, 6].

The two approaches to handle these issues could be frame based detection, that is, to detect faces in each frame without taking into account the temporal information or integrated detection and tracking in which the face is detected in the first frame and tracked through the sequence. The frame-based approach completely overlooks the fact that the frames are contiguous in the sequence. In the second approach, tracking and detection are independent and information from only one source is used at a time, causing a loss of information[7]. This motivated us to develop a novel approach that integrates detection and tracking into a unified framework—the temporal approach. It uses the temporal relationships between the frames to detect

multiple human faces in a video sequence, instead of detecting them in each frame independently. Alternatively, face tracking and detection can be combined by detecting facial features like lips, mouth, nostrils, and eyes and by tracking them through the sequence, but this imposes the constraint that these features would need to be visible and, therefore, only frontal views of the face can be handled [8, 9].

2. Our Work in Perspective

The features of our approach which distinguish it from existing approaches are simultaneously detection and tracking information at each time step and is therefore able to handle changes in imaging conditions (face scale, lighting, and orientation) and changes in image content (the complexity of the background and the number of faces), as has been shown in the experiments. It is able to improve over detection results of the existing detectors. Handling pose change is a challenging problem for any detector. Most of the tracking approaches suffer from the problem of manual initialization. We avoid this by using detection for initializing the tracker. The detection information is integrated at each time step as the parameters are being propagated, that is, the probabilities are accumulated over time. This causes the algorithm to continuously detect faces even in frames where the frame-based detector fails. The detection information provides knowledge of the appearance of new faces to the trainning, which can be readily incorporated whenever they appear by a process of updating.

2.1. Training

In the training process the user will start entering the username along with a frame number as a reference for the training data sets. Secondly we check the trained data after the processing starts.Apart from this here we introduce a rectangle based approach which is used for face detection. In this proposed approach the face is been spited into different rectangle portions of features and are stored in the database for further verifications. In the Initial stage the system is been processed with basic preprocessing steps like Grey Scale Conversion and Histogram Equalization for the Color Images. Figure 1.describes about the Proposed Architeture for the face detection and tracking approach.

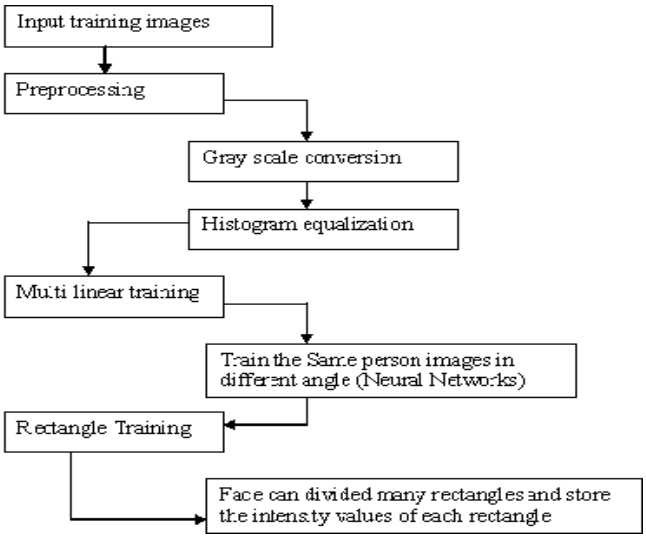


Figure 1.Proposed Architeture

The tainning approach is done by neural network which is of a back-propagation is adopted as the second part of our face detection system. In our neural system, the input vector of the network is been projected with weights of the face space and we choose hundreds of face images, which include different faces with different posses. The input images for processing are taken from different databases like ORL dataset, the MIT CMU face database, and the Wide World Web to produce different face blocks. The training face images consist of all combination of input datasets from different datasets like ORL face images, MIT_CMU images, and randomly selected WWW images. In addition, fifty eigenvectors with the largest associated eigenvalues are adopted as the face space. The projection weights of the face and non-face blocks are computed and used as the positive and negative training vectors of the neural network, respectively. In our system, the neural network consists of a hidden layer with nine hidden nodes.

Backpropagation Neural Network

- Set all weight to random value range from -1.0 to 1.0.
- Set an input pattern (binary values) to the neurons of the net's input layer.
- Active each neuron of the following layer:
  - Multiply the weight values of the connections leading to this neuron with the output values of the preceding neurons.
  - Add up these values.
  - Pass the result to an activation function, which computes the output value of this neuron.
- Repeat this until the output layer is reached.
- Compare the calculated output pattern to the desired target pattern and compute a square error value.

- Change all weights values of each weight using the formula:
  Weight (old) + Learning Rate * Output Error * Output (Neuron i) * Output (Neuron i + 1) * (1 – Output (Neuron i + 1))
- Go to the first step.
- The algorithm end, if all output pattern match their target pattern.

2.2 Image Preprocessing

Face image is what is at the input of the system. The expected result of preprocessing stage is an image which contains only the significant features of the face. To avoid variations which present in the input image we will attempt to build an algorithm of averaging the details so that only main features remain. Thus the face regions of different images which were taken in dissimilar conditions, e.g. light, background, and face to camera distance, may cause varied appearances of faces. In order to have the property of the scale invariance, the input image is resized into the image pyramid. We partition each sub-sampled image into 21 x 21 pixel blocks at every position of the image, and normalize the image blocks to have zero mean and unit variance.

2.2.1 Gray scale conversion

Grayscale images are distinct from one-bit bi-tonal black-and-white images, which in the context of computer imaging are images with only the two colors, black, and white (also called *bilevel* or *binary images*). Grayscale images have many shades of gray in between. Grayscale images are also called monochromatic, denoting the absence of any chromatic variation (i.e., one color). Grayscale images are often the result of measuring the intensity of light at each pixel in a single band of the electromagnetic spectrum (e.g. infrared, visible light, ultraviolet, etc.), and in such cases they are monochromatic proper when only a given frequency is captured. The original image contains about 1crore and seventy lakhs colors (since the input image has 2 power 24 bit colors). This original 24 bit image will be converted into gray scale image. The gray scale image contains 2 power 16bit (65536) colors. This color conversion is performed by using the pixel values of the original image.

We take an RGB color image as input

$$(R_i, G_i, B_i) \in [0.1]^3$$

And produce a grayscale image as output

$$T \in [0.1]$$

To avoid gamma correction issues.

2.2.2. Histogram equalization

However it can also be used on color images by applying the same method separately to the Red, Green and Blue components of the RGB color values of the image. Still, it should be noted that applying the same method on the Red, Green, and Blue components of an RGB image may yield dramatic changes in the image's color balance since the relative distributions of the color channels change as a result of applying the algorithm. However, if the image is first converted to another color space, Lab color space, or HSL/HSV color space in particular, then the algorithm can be applied to the luminance or value channel without resulting in changes to the hue and saturation of the image.

Consider a discrete grayscale image, and let n_i be the number of occurrences of gray level i. The probability of an occurrence of a pixel of level i in the image is

$$P(x_i) = n_i / n$$

Where, $i \in 0 \ldots L - 1$

L → the total number of gray levels in the image

n → the total number of pixels in the image
P → in fact the image's histogram and Normalized to [0, 1]
x → an occurrence of a pixel in the image

$$C(i) = \Sigma^{i}_{j=0} P(x_j)$$

Where ***C*** *→ the cumulative distribution function corresponding to p,*
We would like to create a transformation of the form that will produce a level Y for each level x in the original image, such that the cumulative probability function of Y will be linearized across the value range. The transformation is defined by:

$$Y_i = T(x_i) = c(i)$$

Y_i → *linearized transformation of pixels*
Notice that the T maps the levels into the domain of 0...1. In order to map the values back into their original domain, the following simple transformation needs to be applied on the result.

$$y'_i = y_i \cdot (Max - min) + Min$$

y'_i →*equalized image*
Max →Maximam value pixel in the image
Min → Minmam value pixel in the image

2.2.3. Multi linear training

we present a neural network-based algorithm to detect upright, frontal views of faces in gray- 1scale images. The algorithm works by applying one or get a representative sample of images which contain faces. Each network is trained to set as training progresses [21]. The algorithms makes the output efficient when number of input data training gets increased and the size of the training set are higher . Thus the Multi linear training results to produce a proper detection that are obtained with significant improvement with good accuracy of the detector.

$$\text{Train1} \rightarrow I_i (N_i)$$

Where, I_i → *Intensity,*
N_i →*Different angle image for single person ,*
Train1 → Multi linear training

2.2.4. Rectangle facial training

The first step in our work is to look for face candidates. We use the feature-based method, and the point of the view is from the characteristics human face, one is intensity; the other is symmetry. One feature measures the difference in intensity between the region of the eyes and a region across the upper cheeks. The feature capitalizes on the observation that the eye region is often darker than the cheeks. The other feature compares the intensities in the eye regions to the intensity across the bridge of the nose. Here we propose a three rectangle features which are used to look for face candidates, which are based on the idea mentioned above. In our method, a rectangular window is scanned on the input image to look for face candidates by using three rectangle features. Fig. 2 shows the rectangle features. The parameter ie. the intensity values are been gathered from these rectangles of the image features and are stored in the database for trainning [7].

$$Rect_{si} = Rect_f (w*h) / n_i$$

$$\text{Train2} \rightarrow I_i (Rect_{si})$$

Where, $Rect_{si}$ →*Small rectangle ,* $Rect_f$ →*Full face Rectangle ,* w → *Rectangle width ,* h → *Rectangle height ,* n_i → *Number of Rectangle , Train2 → Rectangle facial training*

3. Proposed Detection Identification And Tracking Algorithm (PDIT)

In face detection the Skin color provides the major role in extracting good information from the face area. The use of color information can simplify the task of face localization in complex environments [7].Several studies show that the major difference is not intensity but color itself. Many researchers have proposed various skin detection techniques based on different color space models such as HSV, YCbCr and YIQ. Human skin color, though it differs widely from person to person, is distributed over a very small area on the CbCr plane [8, 9]. This model is robust against different types of skin, such as those of people from Europe, Asia and Africa. So we use the YCbCr color model for the detection of skin color in this paper.

3.1 Face Detection

Given as input an arbitrary image, which could be a digitized video signal or a scanned photograph, determine whether or not there are any human faces in the image, and if there are, return an encoding of the location and spatial extent of each human face in the image. [3]

$GRect_{si}$=$GRect_f$ (I_i>equalized value)

$SRect_{si}$ =$GRect_{si}$ (w*h) / n_i

If I_i ($SRect_{si}$) = Train2

This rectangle is in face

Else

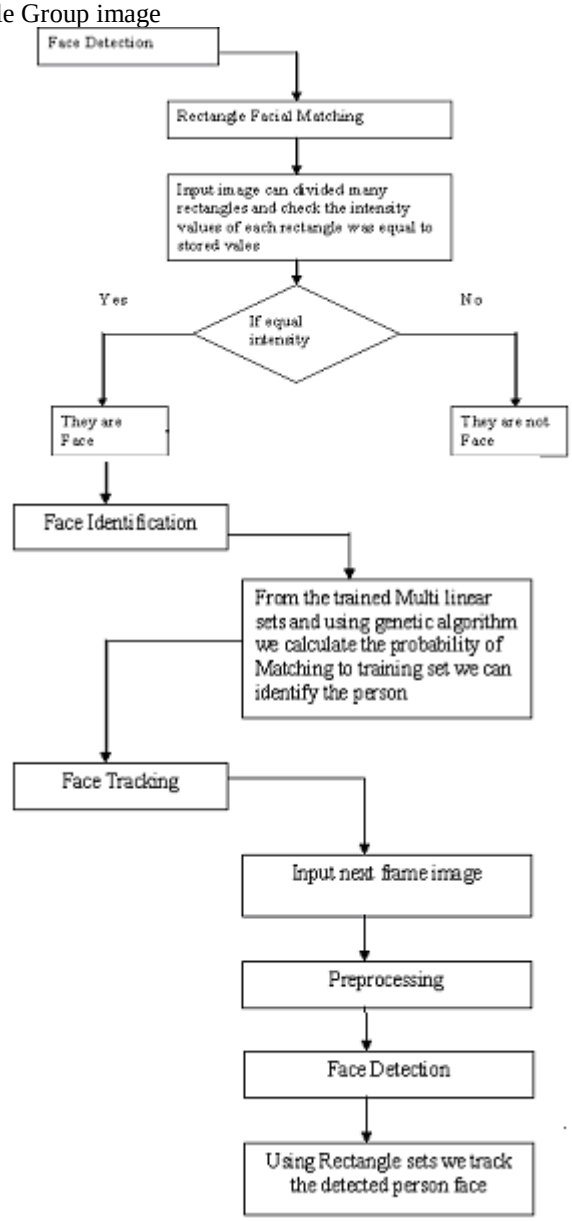
This rectangle not in face

Where

$GRect_{si}$→Face rectangle Group image

$GRect_f$→Full input Group mage Rectangle

$SRect_{si}$ →Small rectangle Group image



3.2 Face Identification

All face images are been splitted into triangle blocks and are given to the neural network. However, this might be still exist some face blocks in the face candidates from the network outputs. We design a simple face verification method to remove these false detection blocks. The main idea of the verification scheme is to consider the distribution of the edge points from the general facial features. Each candidate block is transformed into edge map by the Sobel edge detector method in order to detect the face edges.

The face identification can be used for number of frames or group of frames collecting and compare to training processs.In the training process the face features are been extracted and stored in database, for the face identication and matching. Each image in the database is been matched with the features that are been extracted from the input image and is done by the STRect algorithm. This algorithm stores face rectangle features from different images.

If I_i ($GRect_{si}$) = Train1
Face Identified who is that person
STRect= $GRect_{si}$
Else
No one Identified
Where
STRect→Store the Identified face rectangle

3.3 Face Tracking & Recognisation

In dynamic scenes, tracking is used to follow a face through the sequence. In order to incorporate the face changes over time, in terms of changes in scale, position and to localize the search for the face, it is essential to exploit the temporal correspondence between frames. Tracking exploits the temporal content of image sequences. Face tracking can be divided into two categories 1) head tracking and 2) facial feature tracking. Feature tracking methods track contours and points [12] or follow eyes and mouth [11], and require independent trackers for each feature. Head tracking methods use the information from the entire head and can be region-based [11], color-based [24], or shape-based [5]. Color-based approaches are not robust to lighting changes and approaches that use information from the entire head are, in general, unable to handle s. Tracking involves prediction and update for which filters like Kalman filter and Condensation filter have been used. Tracking approaches can also be model-based, for example, using statistical models or exemplar-based (although only specific features of the face, e.g., lips have been tracked). A combination of feature and head tracking methods, together with filtering, has tried to eliminate the problems of the individual approaches. The tracker of Burchfield simultaneously exploits the elliptical contour fitted to the face and the color information enclosed. This approach can handle out-of plane rotations and occlusions but is unable to handle multiple faces and requires manual initialization. The framework of can be used to track multiple faces, but it does not permit the addition of new faces. Raja et al. combined motion detection with an appearance-based face model. Multiple people tracking were performed using multiple Kalman filters.

If I_i ($NFRect_{si}$) = STRect
Identified person was present in the frame
Else
Identified person was not present in the frame
Where
NF → Next Frame of Group image (applied preprocessing and Face Detection)
$NFRect_{si}$ → Next Frame Face rectangles

4. Experiments and Results

Different real color images containing multiple faces with various sizes and different lighting conditions are employed to test the efficiency of the proposed approach. The appearance of the skin color can be changed due to different lighting conditions. In order to reduce the effect of illumination, we adopt gray

world method [10] to perform light compensation. The corrected red, green and blue color components were then nonlinearly transformed in the YCbCr color space. The skin tone pixels are detected using Cb and Cr components in the YCbCr color space. Let the thresholds be chosen as [Cb1, Cb2] and [Cr1, Cr2], a pixel is classified to have skin tone if the values [Cb, Cr] fall within the thresholds. Each pixel in Cb and Cr components which does not meet a certain threshold range [Cb, Cr] is set to zero. We can see the skin segmented result on a Real color image by YCbCr in Fig. 4.1,4.2,4.3 a

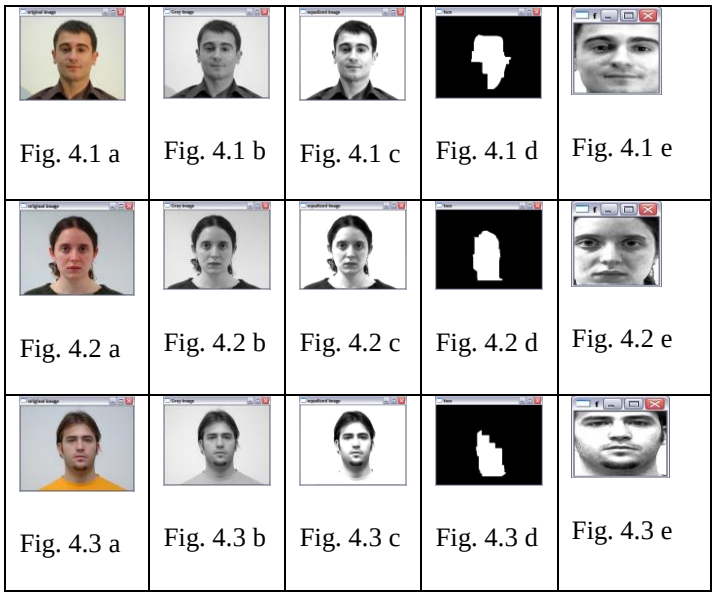


Fig. 4.1 a	Fig. 4.1 b	Fig. 4.1 c	Fig. 4.1 d	Fig. 4.1 e
Fig. 4.2 a	Fig. 4.2 b	Fig. 4.2 c	Fig. 4.2 d	Fig. 4.2 e
Fig. 4.3 a	Fig. 4.3 b	Fig. 4.3 c	Fig. 4.3 d	Fig. 4.3 e

Fig. 4.1,4.2,4.3 Comparison of (a)original (b)Grey Scale (c) Histogram equalization Image (d) Skin Color Extraction (e) Face Extraction

As is evident from these results, our algorithm demonstrates exceptional performance. As the quantitatively assessing the performance in practical application is a complicated issue because of the ideal images and are normally unknown at the receiver end. So here we use the following method for experiments. An original image is applied with Grey scale conversion is represented in Fig. 4.1,4.2,4.3 b and are transformed into the Histogram equalization and is represented in Fig. 4.1,4.2,4.3 c and by taking skin color extraction is done according to proposed algorithm which is described in 4.1,4.2,4.3 d. In this algorithm, In Fig. 4.3 a, we show the results of applying the number of trained input vs. accuracy and in Fig. 4.3 b it describes about the number of trained input vs. speed of detection rate of the face images.

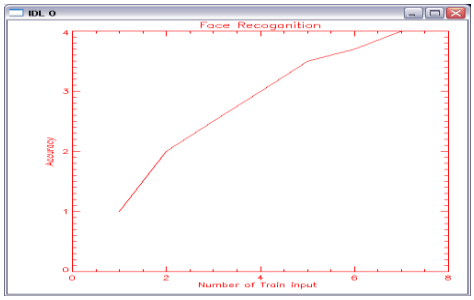


Fig. 4.3 a number of trained input vs. accuracy of detection

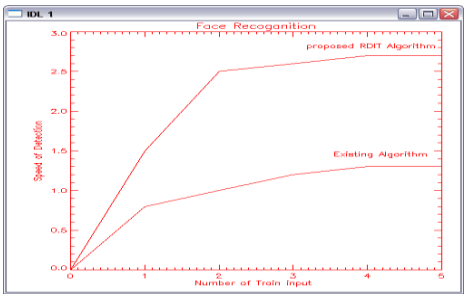


Fig. 4.3 b number of trained input vs. speed

5. Conclusion

In this paper, a new proposed RDIT algorithm is applied to Face detection & tracking and to overcome the existing issue. we can consider the thief detection in the video frame .base on that we can propose the new mechanism named as FACE RDIT (Face Rectangular Detection, Identification and tracking).here we utilize some preprocessing algorithms like gray scale conversion and histogram equalization for image enhancement. Than we derive the new algorithm based on rectangular features of the face and multi linear training .in the proposed technique have four main steps they are training, detection, identification and tracking .that is we can train the thief photo image, than detect the faces in the video frame and than identify the thief face. Finally track that identified thief in next following video frames by our proposed algorithm.

References

1. Yepeng Guan, Lin Yang "unsupervised face detection based on skin color and geometric information"

2. R.-L. Hsu, M. Abdel-Mottaleb, A.K. Jain, "Face detection in color images", IEEE Trans. PAMI, 2002, 24(5), pp. 696–706

3. H. Wang, S.F. Chang, "A highly efficient system for automatic face region detection in MPEG videos", IEEE Trans. Circuit Systems for Video Technology, 1997, 7(4), pp. 615–628

4. C. Garcia, G. Tziritas, "Face detection using quantized skin color regions merging and wavelet packet analysis", IEEE Trans. Multimedia, 1999, 1(3), pp. 264–277

5. A. Rizzi, C. Gatta, and D. Marini, "Color correction between gray world and white patch", IS&T/SPIE Electronic Imaging 2002. The human Vision and Electronic Imaging VII Conference, 4662, San Jose, 2002, pp. 367-375

6. A. Guetta, M. Pare, and S. Rajagopal, "Face Detection. EE368 Final Project", 2003

7. I. Pitas, A. Karasaridis, Multichannel transforms for signal/image processing, IEEE Trans. Image Processing, 1996, 5 (10), pp. 1402–1413.

8. C.-C. Chiang, W.-K Tai, et al, "A novel method for detecting lips, eyes and faces in real time", Real-Time Imaging, 2003, 9(4), pp. 277-287.

9. H. Kruppa, B. Schiele, "Using Local Context to Improve Face Detection", In Proc. of the British Machine Vision Conference (BMVC'03), Norwich,

10. P. Viola, M. J. Jones, "Robust real-time face detection", International Journal of Computer Vision, 2004, 57(2), pp. 137-154.

11 L. Mostafa and S. Abdelazeem, "Face Detection based on Skin Color using Neural Networks", in Proc. of the first ICGST International Conference on Graphics, Vision and Image Processing GVIP '05, Cairo, Egypt, pp. 53-58.