A new Method for Face Recognition Using Variance Estimation and Feature Extraction

Walaa Mohamed¹, Mohamed Heshmat², Moheb Girgis³ and Seham Elaw⁴

¹, ², ⁴ Faculty of science, Mathematical and Computer Science Department, Sohag University, 82524, Sohag, Egypt.
³ Faculty of science, Department of Computer Science, Minia University, El-Minia, Egypt

Abstract: Face recognition has been one of the most studied topics in the last decades. Its research has increased in many applications in communications and automatic access control systems. In this paper, a new face recognition method based on OpenCV face detection technique is introduced. The goal of the propose method is to detect faces that exist in the image and try to locate each face in a previously prepared database using simple variance calculations and Euclidean distance of extracted facial features. The features under consideration are eyes, nose and mouse. Furthermore, a new method to extract facial features is developed based on feature location with respect to face dimensions. The proposed algorithm has been tested on various images and its performance is found to be good in most cases. Experimental results show that our method of detection and verification achieves very encouraging results with good accuracy and simple computations.

Keywords: face detection, facial features extraction, variance, RGB color space.

1. INTRODUCTION

The great progress of modern communication technology has led to the growing demands of image and video applications in medicine, remote sensing, security, entertainment and education. Fast and reliable face and facial feature detection are required abilities for any human computer interaction approach based on computer vision. One of the main challenging problems in building automated systems that perform face recognition and verification tasks is face detection and facial feature extraction. Though people are good at face identification, recognizing the human faces automatically by computer is very difficult. Face recognition has been widely applied in security system, credit-card verification, and criminal identifications, teleconference and so on. Face recognition is influenced by many complications, such as the differences of facial expression, the light directions of imaging, and the variety of posture, size and angle. Even to the same people, the images taken in a different surroundings may be unlike. The problem is so complicated that the achievement in the field of automatic face recognition by computer is not as satisfied as the finger prints [1].

A lot of methods have been proposed for face detection in still images that are based on texture, depth, shape and color information. Many methods depend on the observation that human faces are characterized by their oval shape and skin-color. The major difficulties encountered in face recognition are due to variations in luminance, facial expressions, visual angles and other potential features such as glasses, beard, etc. This leads to a need for employing several rules in the algorithms that are used to tackle these problems [2], [3], [4].

Nikolaidis et al. [2] proposed a combined approach for facial feature extraction and determination of gaze at direction that employs some improved variations of the adaptive Hough transforms for curve detection, minima analysis of feature candidates, template matching for inner facial feature localization, active contour models for inner face contour detection and projective geometry properties for accurate pose determination.

Koo et al. [5] suggested defining 20 facial features. Their method detects the facial candidate regions by Haar classifier, and detects eye candidate region and extracts eye features by dilate operation, then detect lip candidate region using the features. The relative color difference of a* in the L*a*b* color space was used to extract lip feature and to detect nose candidate region and detected 20 features from 2D image by analyzing end of nose.

Yen et al. [6] presented an automatic facial feature extraction method based on the edge density distribution of the image. In the preprocessing stage, a face is approximated to an ellipse, and a genetic algorithm is applied to search for the best ellipse region match. In the feature extraction stage, a genetic algorithm is applied to extract the facial features, such as the eyes, nose and mouth, in the predefined sub regions.

Gu et al. [1] proposed a method to extract the feature points from faces automatically. It provided a feasible way to locate the positions of two eyeballs, near and far corners of eyes, midpoint of nostrils and mouth corners from face image.

Viola and Jones [7]-[9] described a face detection
framework that is capable of processing images extremely rapidly while achieving high detection rates. There are three key contributions. The first is the introduction of a new image representation called the “Integral Image”, which allows the features used by their detector to be computed very quickly. The second is a simple and efficient classify, which is built using the AdaBoost learning algorithm to select a small number of critical visual features from a very large set of potential features. The third contribution is a method for combining classify in a “cascade”, which allows background regions of the image to be quickly discarded while spending more computation on the promising face-like regions.

Srivastava [10] proposed an efficient algorithm for facial expression recognition system, which performs facial expression analysis in a near real time from a live web cam feed. The system is composed of two different entities: trainer and evaluator. Each frame of video feed is passed through a series of steps, including Haar classifiers, skin detection, feature extraction, feature point tracking, creating a learned support vector machine model to classify emotions to achieve a tradeoff between accuracy and result rate.

Radha et al. [11] described a comparative analysis of face recognition methods: principle component analysis (PCA), linear discriminant analysis (LDA) and independent component analysis (ICA) based on curvelet transform. The algorithms are tested on ORL Database.

Kumar et al. [12] presented an automated system for human face recognition in a real time background world for a large homemade dataset of persons face. To detect real time human face AdaBoost with Haar cascade is used and a simple fast PCA and LDA are used to recognize the faces detected. The matched face is then used to mark attendance in the laboratory, in their case.

This paper presents a new face recognition method. The proposed method uses OpenCV [13] to load boosted haar classifier cascades, which allow an initial detection of the faces within a given image. Variance estimation of RGB components is computed to compare the extracted faces and the faces within the database used in comparison. In addition, a new facial feature extraction method is proposed based on feature location. Euclidean distance of facial features of the extracted faces from test image and faces extracted from the database after a variance test is used.

The rest of this paper is organized as follows. Section 2 describes the methodology of the proposed method with its stages: face detection, variance estimation, feature extraction, method representation and the proposed algorithm. Section 3 includes the results and method analysis. Section 4 draws the conclusion of this work and possible points for future work.

2. METHODOLOGY

Given an image that consists of many objects our goal is to detect humans faces, extract these faces and identify each face using a database of known humans. So, our algorithm is divided into three main steps, first: face detection, the proposed algorithm depends heavily on the open-source library OpenCV in this step. Second: facial feature extraction, an effective method to extract facial features like eyes, nose and mouth depending on their locations with respect to the face region is developed. Third: similar face identification or image searching; the goal of this step is to scan the database of known faces to find the most similar faces to the faces extracted from the test image in the first step.

2.1 Face Detection

The proposed method starts with a piece of code named facedetect.cpp. It detects all faces within a given image via haar classifier cascades and draws rectangles around them, as shown in figure 1.

![Figure 1 Face detection using OpenCV algorithm](Image 313x405 to 405x481)

Haar-like [9], [14], [15] feature classifier cascades are composed of multiple classifiers, or conditions, that are used to distinguish unique objects (i.e., face, eye, etc.) from anything that is not that object. Classifiers are a set of values representing sums of pixel brightness on a region-by-region basis. These specifically structured regions are called Haar-like features. Haar-like feature classifiers are created using the integral image which is an intermediate image representation that allows the features used by the detector to be computed very quickly. Rectangle features can be computed very rapidly using the integral image.

The integral image [9] at location \((x, y)\) contains the sum of the pixels above and to the left of \(x, y\) inclusive, as shown in the following figure:

![Figure 2 The integral image representation](Image 390x95 to 474x159)
in brightness of the corresponding pixel in the original image by finding the sum of the derivatives of the pixels above and to the left of the original:

\[ ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y'), \]  

(1)

where \( ii(x, y) \) is the integral image and \( i(x, y) \) is the original image.

2.2 Variance Estimation

Variance is a very light calculation and considered as an important constraint to prove similarity between two images. Let \( x \) be a vector of dimension \( n \), the variance of \( x \) can be calculated as follows:

\[ \text{var} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \overline{x})^2, \]  

(2)

where \( \overline{x} \) is the mean value of \( x \).

However, it is not necessary that the two images which have the same variance to be the same, because different images may have the same value of variance. So the variance is used at first to filter the codebook (database of faces) and extract faces that have the same or close value of variance of the input face image, then another test is applied to choose the most similar faces to this test face.

When working with 3-D images or RGB images the important problem appears is that there are three values for each pixel in the image, representing the red, green, and blue colors. To compute the variance of RGB image, the variance for each color is calculated separately. So there are three variance values, one for the red values, another for the green values and third for the blue values.

\[ v_{red} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \overline{x})^2, \quad v_{green} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \overline{x})^2, \quad v_{blue} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \overline{x})^2, \]  

(3)

To simplify the comparison, the average of the three variance values is computed as follows:

\[ \overline{v} = \frac{v_{red} + v_{green} + v_{blue}}{3}, \]  

(4)

2.3 Feature Extraction

The proposed method uses the face detector developed by Viola and Jones [7]-[9] to extract the face regions from experiment images. This step makes working on faces directly easier in searching the database for the most similar faces to the face which we are looking for.

OpenCV face detection code based on Viola and Jones’s algorithm depends on converting the image to grayscale and then applying face detection method.

In this portion of our work, the aim is to compare two color faces to detect the similarity between them. RGB (Red Green Blue) color space, which is used here, is an additive color system based on tri-chromatic theory. It is often found in systems that use a CRT to display images. The RGB color system is very common, and is being used in virtually every computer system as well as television, video etc [16].

![Figure 3 RGB color model](image)

In RGB color model, any source color (F) can be matched by a linear combination of three color primaries, i.e. Red, Green and Blue, provided that none of those three can be matched by a combination of the other two, see figure 3.

Here, \( F \) can be represented as:

\[ F = rR + gG + bB, \]  

(5)

where \( r, g \) and \( b \) are scalars indicating how much of each of the three primaries (R, G and B) are contained in \( F \). The normalized form of \( F \) can be as follows:

\[ F' = R' + G' + B' \]  

(6)

where

\[ R' = r/(r + g + b), \]

\[ G' = g/(r + g + b), \]

\[ B' = b/(r + g + b), \]  

(7)

A new facial feature extraction method based on feature location with respect to the whole face region is proposed in this paper. We try to locate eyes, nose and mouse in the faces extracted by using OpenCV [7]-[9] face detection algorithm. By detecting the candidate regions of left eye, right eye, nose and mouse, by training manually, then applying the obtained dimensions of each region on several other faces with the same size, the results were very good, as shown in figure 4.

Given a face image of 200 pixels height and 200 pixel width, after training with a lot of images, we found that the candidate region of eyes is located between rows 60 and 95, columns 25 and 80 for right eye and columns 115 and 170 for left eye. The candidate region for the nose is
located between rows 110 and 145 and columns 75 and 125 and the candidate region for the mouse is located between rows 145 and 185 and columns 60 and 135. When applying the dimensions obtained by training on many face images, we found that they were suitable for any face image with the same width and height, as shown in figure 5.

2.4 Method Representation

The proposed algorithm consists of four parts. Firstly, OpenCV face detection code is used to detect and extract faces within the input image. Secondly, variance estimation is applied to extract database images, which have a close variance value to the test image. Thirdly, the cut features method is used to extract facial features from the face images. Finally, Euclidean distance of facial features is computed by the following equation:

\[
d = \sqrt{\text{abs}(\text{test feature}[R] - \text{matched feature}[R])^2 + \text{abs}(\text{test feature}[G] - \text{matched feature}[G])^2 + \text{abs}(\text{test feature}[B] - \text{matched feature}[B])^2}
\]  

(8)

Figure 5 Examples of Feature extraction

to compare the similarity between features. The steps of the proposed algorithm are shown in figure 6.

3. RESULTS AND DISCUSSION

The experiments were performed on a computer with 2.20 GHz speed and 4 Gbyte RAM using several 200×200 color images containing humans and three codebooks of different sizes (7 images of two persons with different conditions, 10 different images (as shown in step 5 in figure 6), and 20 images include some images to the same persons). A sample of the used database images is shown in figure 7.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Read input image.</td>
</tr>
<tr>
<td>2</td>
<td>Apply face detection code on the input image and detect faces.</td>
</tr>
<tr>
<td>3</td>
<td>Extract each face as a separate image.</td>
</tr>
<tr>
<td>4</td>
<td>Read a test face from those extracted in the previous step.</td>
</tr>
<tr>
<td>5</td>
<td>Read codebook array (database of images)</td>
</tr>
<tr>
<td>6</td>
<td>Divide codebook into n images.</td>
</tr>
<tr>
<td>7</td>
<td>Calculate the variance of each image obtained in step 6, by using eq. (3), (4), and put variance values in an array.</td>
</tr>
<tr>
<td>8</td>
<td>Calculate variance of the test image (step 4) using eq. (3), (4).</td>
</tr>
<tr>
<td>9</td>
<td>Compare variance value of test image and each image in codebook and keep locations of the most similar images to test image, which satisfy the condition ((-100 \leq \text{variance difference} \leq 100)), in an array.</td>
</tr>
<tr>
<td>10</td>
<td>For i=1 to number of similar images which extracted from step 9, a) Extract facial features from each image according to location (right eye – left eye – nose – mouse). b) Calculate the Euclidean distance between the 3-arrays containing the RGB color values of each feature using Eq. (8).</td>
</tr>
</tbody>
</table>
11. Detect minimum distance \((d)\) and location of the image that has the minimum distance from step 10.

12. Display the best matched image from codebook.

13. Repeat the steps 4 to 12 for other faces in the input image.

**Figure 6** the proposed algorithm steps

Table 1 shows the results obtained using 20 different 200x200 RGB images and a database of RGB images which is shown in figure 7. The first column of the table shows the test face obtained by the face detection algorithm. The second column shows the range of the variance difference between test face and faces in database. The third column shows the similar images which returned after the variance test and their locations in the used database. The fourth column shows the facial features extracted from test image, and the last two columns show the best matched face and its location in the database after variance test.

For example, by using the proposed algorithm to locate the face #2 and finding the most similar face to it within the database using variance range close to 0, the right matched face is obtained. But there are some other face images belong to the same person with different conditions (close eyes and smile). When the variance difference range is extended, all the face images belonging to the same person are returned. The execution proceeds as follows: after the first test (variance test) with variance difference range equals 56 (experimentally obtained), the algorithm returned 3 locations of faces whose variance value close to the variance of the test face #2. In order to know which one of them is the same or the closest to the test face, the facial features of the test face and the facial features of the three face images are extracted and the Euclidean distance to their RGB components is calculated by Eq. (8). Then the face image with the minimum distance \((d)\) is considered as the best matched image and its location is returned.

To see the importance of the second test (feature similarity matching), consider some cases like image number #1, #6, #12 and #16, where all of them are images of the same person. In these cases, the test started with variance difference value close to zero and the algorithm worked well and the right image is returned. But when the variance difference is increased to get any other images for the same person, some images like the child image is returned because the variance of the child face image is close somehow to the variance value of some test face images under different conditions. But after applying the second test the right close face images are returned.

The search efficiency is evaluated by how many times the distance \((d)\) computations are performed on average compared to the size of the database. In the proposed method the total number of distance calculations is small, because it uses the variance test to find out the face images that have a close variance value to the input face image, then the distance computation is performed only on those images where their number is always small compared to the database size.

4. CONCLUSION

In this paper, a new method of face recognition based on variance estimation and facial feature cut is proposed. It can be used in face recognition systems such as video surveillance, human computer interfaces, image database management and smart home applications. The proposed algorithm has been tested using a database of faces and the results showed that it is able to recognize a variety of different faces in spite of different expressions and illumination conditions with light calculations. However, the deplorable failures of OpenCV face detection code in detecting zoomed images and rotated images indicates a need for further work. In addition, facial feature expressions and their effect on the recognition of humans need further investigation.
Table 1: Some results of the proposed algorithm starting with comparing each face obtained by OpenCV code and comparing it with the database images by variance estimation and Euclidean distance of features.

<table>
<thead>
<tr>
<th>Test image Size 200*200 pixels</th>
<th>Variance difference range</th>
<th>Similar images Returned after first variance test and there locations in database</th>
<th>Test image features</th>
<th>Best matched image after feature cut and Euclidean distance</th>
<th>Best matched image location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Face no.</td>
<td>Image</td>
<td>0</td>
<td>1</td>
<td>391</td>
<td>1,1,7,12,16</td>
</tr>
<tr>
<td>#1</td>
<td><img src="image3.png" alt="Image 3" /></td>
<td>0</td>
<td>2</td>
<td>56</td>
<td>2,6,11</td>
</tr>
<tr>
<td>#2</td>
<td><img src="image6.png" alt="Image 6" /></td>
<td>0</td>
<td>3</td>
<td>31</td>
<td>1,3</td>
</tr>
<tr>
<td>#3</td>
<td><img src="image9.png" alt="Image 9" /></td>
<td>0</td>
<td>4</td>
<td>45</td>
<td>4,20</td>
</tr>
<tr>
<td>#4</td>
<td><img src="image12.png" alt="Image 12" /></td>
<td>0</td>
<td>5</td>
<td>27</td>
<td>5,19</td>
</tr>
<tr>
<td>#5</td>
<td><img src="image15.png" alt="Image 15" /></td>
<td>0</td>
<td>6</td>
<td>56</td>
<td>2,6,11</td>
</tr>
<tr>
<td>#6</td>
<td><img src="image18.png" alt="Image 18" /></td>
<td>0</td>
<td>7</td>
<td>370</td>
<td>1,5,7,12,16,19</td>
</tr>
<tr>
<td>#7</td>
<td><img src="image21.png" alt="Image 21" /></td>
<td>0</td>
<td>8</td>
<td>447</td>
<td>8,10</td>
</tr>
<tr>
<td>#8</td>
<td><img src="image24.png" alt="Image 24" /></td>
<td>0</td>
<td>9</td>
<td>2216</td>
<td>8,9</td>
</tr>
<tr>
<td>#9</td>
<td><img src="image27.png" alt="Image 27" /></td>
<td>0</td>
<td>10</td>
<td>437</td>
<td>8,10</td>
</tr>
</tbody>
</table>
Table 1: Continued

<table>
<thead>
<tr>
<th>Image #</th>
<th>Test image Size 200*200 pixels</th>
<th>Variance difference range</th>
<th>Similar images Returned after first variance test and there locations in database</th>
<th>Test image features</th>
<th>Best matched image after feature cut and Euclidean distance</th>
<th>Best matched image location</th>
</tr>
</thead>
<tbody>
<tr>
<td>#11</td>
<td>0</td>
<td>11</td>
<td>11,12,13,14,15,18,20</td>
<td></td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>#12</td>
<td>0</td>
<td>12</td>
<td>12,13,14,15,18,20</td>
<td></td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>#13</td>
<td>0</td>
<td>13</td>
<td>13,14,15,18,20</td>
<td></td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>#14</td>
<td>17</td>
<td>14</td>
<td>14,15,16,17,18,20</td>
<td></td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>#15</td>
<td>0</td>
<td>15</td>
<td>15,16,17,18,20</td>
<td></td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>#16</td>
<td>0</td>
<td>16</td>
<td>16,17,18,20</td>
<td></td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>#17</td>
<td>0</td>
<td>17</td>
<td>17,18,19,20</td>
<td></td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>#18</td>
<td>0</td>
<td>18</td>
<td>18,19,20,21</td>
<td></td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>#19</td>
<td>0</td>
<td>19</td>
<td>19,20,21</td>
<td></td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>#20</td>
<td>0</td>
<td>20</td>
<td>20,21,22,23</td>
<td></td>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>
REFERENCES


AUTHORS

Walaa M. Abd-Elhafiez, received her B.Sc. and M.Sc. degrees from south valley university, Sohag branch, Sohag, Egypt in 2002 and from Sohag University, Sohag, Egypt, Jan 2007, respectively, and her Ph.D. degree from Sohag University, Sohag, Egypt. She authored and co-authored more than 29 scientific papers in national/international journals/conferences. Her research interests include image segmentation, image enhancement, image recognition, image coding and video coding and their applications in image processing.

Moahmed Heshmat, received his B.Sc. and M.Sc. degrees from south valley university, Sohag branch, Sohag, Egypt in 2002 and from Sohag University, Sohag, Egypt, 2010, respectively, and his Ph.D. degree from Sohag University, Sohag, Egypt and Bauhaus-University, Weimar, Germany. His research interests include computer vision, 3D data acquisition, object reconstruction, image segmentation, image enhancement, image recognition.

Seham Elaw, received her B.Sc. and M.Sc. degrees from south valley university, Sohag branch, Sohag, Egypt in 2004 and from Sohag University, Sohag, Egypt, Jan 2011, respectively. Her research interests include image segmentation, image enhancement, image recognition and their applications in image processing.