National Conference on Recent Innovations in Computer Science & Communication Engineering, *July 2016* *ISBN: 978-93-86005-02-1*

Gray Scale Image Recognition using Finite State Automata

Rajib Saha*, Abhishek Bal and Moumita Bose

Department of Computer Science and Engineering, RCC Institute of Information Technology, Kolkata – 700015, West Bengal, India;
rajib.saha@rcciit.org, abhisheknew1991@gmail.com, bose.moumita5@gmail.com

Abstract

In image processing, processed images of faces can be seen as vectors whose components are the brightness of each pixel. The dimension of this vector space is the number of pixels. The eigenvectors of the covariance matrix associated with a large set of normalized pictures of faces are called eigen faces. They are very useful for expressing any face image as a linear combination of some of them. In the facial recognition branch of biometrics, eigen faces provide a means of applying data compression to faces for identification purposes. Research related to eigen vision systems determining hand gestures has also been made This paper is about human face recognition using finite automata. Face recognition involves matching a given image with the database of images and identifying the image that it resembles the most. In this paper, face recognition is achieved using IMED (Image Euclidean Distance) and Frechet distance and tested using standard database. Euclidean distance uses the prior knowledge that pixels located near one another have little variance in gray levels, and determines the relationship between pixels only according to the distance between pixels on the image lattice. In many applications, however, we are only interested in face images. Therefore, more prior knowledge can be obtained from these images to determine the relationship between pixels.

Keywords: Eigen Faces, Euclidean Distance, Face Images, Finite Automata, Normalized Pictures

1. Introduction

Facial recognition (or face recognition) is a type of biometric (Biometrics is the science and technology of measuring and analyzing biological data) software application that can identify a specific individual in a digital image by analyzing and comparing patterns.

Many applications for face recognition have been envisaged, and some of them given below. Commercial applications have so far only scratched the surface of the potential. Installations so far are limited in their ability to handle pose, age and lighting variations, but as technologies to handle these effects are developed, huge opportunities for deployment exist in many domains. The Interactive Face Recognition

*Author for correspondence

can benefit the areas of: Law Enforcement, Airport Security, Access Control, Driver's Licenses and Passports, Homeland Defense, Customs and Immigration and Scene Analysis.

Face recognition as a biometric derives a number of advantages from being the primary biometric that humans use to recognize one another. Some of the earliest identification tokens, i.e., portraits, use this biometric as an authentication pattern. Furthermore it is well-accepted and easily understood by people, and it is easy for a human operator to arbitrate machine decisions - in fact face images are often used as a human-verifiable backup to automated fingerprint recognition systems.

Because of its prevalence as an institutionalized and accepted guarantor of identity since the advent of photography, there are large legacy systems based on face images-such as police records, passports and driving licences-that are currently being automated. Face recognition has the advantage of ubiquity and of being universal over other major biometrics, in that everyone has a face and everyone readily displays the face. (Whereas, for instance, fingerprints are captured with much more difficulty and a significant proportion of the population has fingerprints that cannot be captured with quality sufficient for recognition).

Face region has number of advantages. Firstly, it is non intrusive. Whereas many techniques require the subject's cooperation and awareness in order to perform identification or verifications such as looking into eye scanner or placing their hand on fingerprint reader. A system based on face recognition could be more safe, cheap and easy to use.

Over the years, face recognition has gained rapid success with the development of new approaches and techniques. Due to this success, the rate of successful face recognition has been increased to well above 90%. Despite of all this success, all face recognition techniques usually suffer from common challenges of image visibility. These challenges are lighting conditions variations, skin variations and face angle variations[2].

Figure 1. Example of gray scale image.

A digitized image of the finite resolution *m x n* consists of *m x n* pixels each of which takes a Boolean value (1 for black, 0 for white) for bi-level image, or real value (practically digitized to an integer value 0 and 256) for a gray-scale image shown in Figure 1.

Here we will consider square images of resolution 2^n x 2^n (typically $6 \leq n \leq 11$). In order to facilitate the application of finite automata to image description we will assign each pixel at 2^n x 2^n resolution a word of length *n* over the alphabet $\Sigma = \{0, 1, 2, 3\}$ as its address. A pixel at 2^n x 2^n resolution corresponds to a sub square of size 2-n of the unit square. We choose ε as the address of the whole unit square. Its quadrants are addressed by single digits as shown in Figure 2 on the left. The four sub-square of the square with address *w* are addressed w0, w1, w2 and w3, recursively. A necessary condition for black and white multi-resolution image to be represented by a regular set, is that is has only a finite number of different sub images in all the sub squares with addresses from Σ^*. Therefore, images that can be perfectly (i.e., with infinite precision) described by regular expressions (finite automata) are images of regular or fractal character. Self-similarity is a typical property of fractals. Any image can by approximated by a regular expression (finite automaton), however, an approximation with a smaller error might require a larger automaton. Figure 2 shows the concept behind the process.

Let us consider square images of resolution 2^n x 2^n and assign each pixel at 2^n x 2^n resolution a word of length n over the alphabet $\Sigma=\{0,1,2,3\}$ as its address. A pixel at 2^n x 2^n resolution corresponds to a sub square

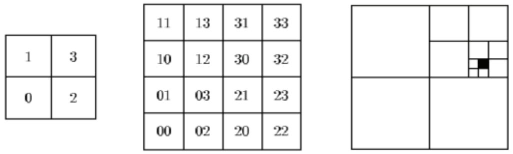

Figure 2. Concept of finite automata.

of size 2^n of the unit square[1]. Quadrants are addressed by single digits as shown in Figure above on the left. Address of the entire sub square (pixels) of resolution 4 x 4 are shown in Figure above and middle. The sub square (pixel) with address 3203 is shown on the right of Figure above.

In order to specify a black and white image of resolution 2^m x 2^m, we need to specify a Boolean function $\Sigma m \rightarrow \{0,1\}$, or alternately we can specify just the set of pixels which are black, i.e., a language $L \subseteq \Sigma m$. Frequently, it is useful to consider multi resolution images, that is images which are simultaneously specified for all possible resolution, usually in some compatible way. (We denote Σm the set of all words over Σ of the length m, by Σ* the set of all words over Σ).

In our notation a bi-level multi-resolution image is specified by a language $L \subseteq \Sigma^*$, $\Sigma = \{0, 1, 2, 3\}$, i.e., the set of addresses of all the black squares, at any resolution.

Let us consider a Boolean function like $\Sigma m \rightarrow \{0,1\}$ and specify the set of pixels which are black, i.e., a language $L \subseteq \Sigma m$. A bi-level multi-resolution image is specified by a language $L \subseteq \Sigma^*$, $\Sigma = \{0, 1, 2, 3\}$, i.e., the set of addresses of all the black squares, at any resolution. Let the original image represented as language is L where 128*128 images represented as L1, L2, L3 and L4. By applying concatenation operation we get L31and L24. Similarly, we can apply this procedure on consecutive images.

2. Literature Review

In[1] proposed finite automata as a tool for specification and compression of gray-scale image. This work describes, what are interests points in pictures and idea if they can hang together with resultant finite automata.

In[10] introduce the Weighted Finite State Automata (WFA) as a tool for image specification and loss or loss-free compression. This paper describe how to compute WFA from input images and how the resultant automaton can be used to store images (or to create image database) and to obtain additional interesting information usable for image indexing or recognition. Next, we describe an automata composition technique. This work also describes an automata composition technique and presents a possible way of storing automata in persistent storage.

In[12] evaluate various face detection and recognition methods, provide complete solution for image based face detection and recognition with higher accuracy, better response rate as an initial step for video surveillance. Solution is proposed based on performed tests on various face rich databases in terms of subjects, pose, emotions, race and light.

3. Proposed Work

3.1 Proposed Recognition Method

This paper proposed face recognition that is using IMED (Image Euclidean Distance) and Frechet distance with the help of finite state automata and tested using standard database and another method of face recognition (with noise) that is using Mahalanobish distance Frechet distance with the help of finite state automata and tested using standard database.

This paper is about human face recognition in image files. Face recognition involves matching a given image with the database of images and identifying the image that it resembles the most. The aim of this paper is to successfully demonstrate the human face recognition using on IMED using different distance classifiers. This paper can be extended to Expression detection also. The steps of the proposed method for noisy and noiseless image are explained in algorithm section.

3.2 Algorithm

3.2.1 Recognition without Noise

Step1: Read the image.
Step2: Get the dimensions of the image.
Step3: Divide the image up into 4 blocks.
Step4: Change this value into 128,64,32,16,8,4,2 and 1(Rows in block).
Step5: Change this value into 128,64,32,16,8,4,2 and 1(Columns in block).
Step6: Figure out the size of each block.
Step7: Pre allocate a 3D image.
Step8: Now scan though, getting each block and putting it as a slice of a 3D array.
Step9: Determine starting and ending rows.
Step10: Determine starting and ending columns.
Step11: Extract out the block into a single subimage.
Step12: ImageArray = grayImage(row1:row2, col1:col2)
Step13: TwoBlock = grayImage(row3:row4, col3:col4)
Step14: Specify the location for display of the image.(change display location as required).
Step15: Show second block. (2,2 then 4,4 then 8,8 then 16,16 then 32,32)change this as required.
Step16: Make the caption the block number.
Step17: Now image3D is a 3D image where each slice or plane is one quadrant of the original 2D image.
Step18: Load test block and calculate the distance of that block.
Step19: Load another block and calculate the distance of that block.
Step20: Calculate *Euclidean distance* between two distances (sqrt of sum(sistance1-distance2)^2).
Step21: If the test image block is match with equivalent image then the result is successful and can show the image.

3.2.2 Recognition with Noise

Step1: Read the image.
Step2: If psnr is greater than 30 then divide the image into 8*8 blocks.
Step3: If psnr is greater than 30 then divide the image into 8*8 blocks.
Step4: Store the block into database 1.
Step5: Apply mahalanobis distance formula in 8*8 blocks.
Step6: Calculate mean values from database 2.
Step7: Find the min mean.
Step8: Find the index no of min mean.
Step9: Display the image if it is match with equivalent image.
Step10: If psnr is greater than 15 then divide the image into 8*8 blocks.
Step11: Store the block into database 3.
Step12: Apply fréchet distance formula in 16*16 blocks.
Step13: Calculate mean values from database 4.
Step14: Find the min mean.
Step15: Find the index no of min mean.
Step16: If psnr is less than 15 then input new image.

3.3 Experimental Result

In this proposed work matlab software to implement this project. Matlab (matrix laboratory) is a multi-paradigm numerical computing environment.

Figure 3 explains the face recognition for noiseless image and Figure 4 explains the recognition for noisy. The comparative evolution results of proposed algorithm are shown in Figure 5. The success rate for the proposed method is better than the other present methods. From the evolution result of proposed method it is shown that proposed method gives better recognition result then noisy image.

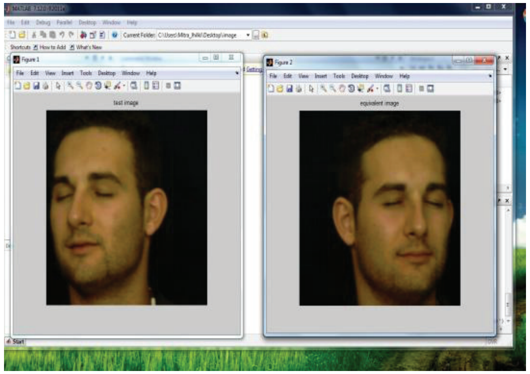

Figure 3. Recognition without noise.

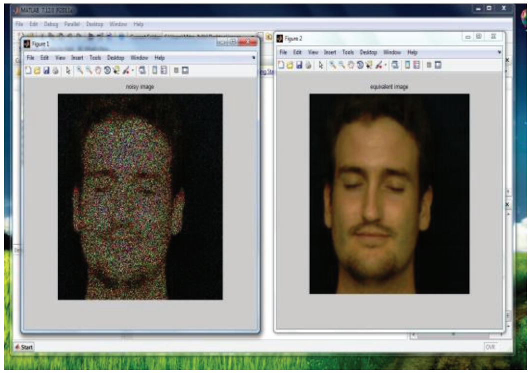

Figure 4. Recognition with noise.

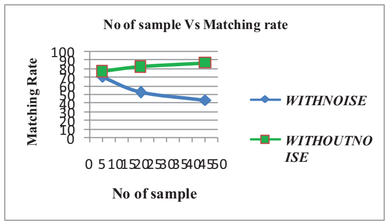

Figure 4. Evolution results for with noise and without noise image.

4. Conclusion and Future Work

In this paper face recognition is achieved using IMED (Image Euclidean Distance) and Frechet distance and tested using standard database. Euclidean distance uses the prior knowledge that pixels located near one another have little variance in gray levels, and determines the relationship between pixels only according to the distance between pixels on the image lattice. The experimental result shows that proposed method is good enough for face recognition for noiseless as well as noisy image.

Future enhancements are extension of image database and generate enough amounts of test data and use it to develop and test a general pattern identification technique and to represent every image by that pattern to meet the end goal which is image recognition. Another future enhancement could be to combine both video face detection and face recognition methods to not only detect a face in a video, but also recognize matching face in the test database.

5. Acknowledgment

The authors are very much grateful to the Department of Computer Science and Engineering for giving the opportunity to work on Gray Scale Image Recognition using Finite State Automata. Both the authors sincerely express his gratitude to Dr. Arup Kumar Bhaumik, Principal of RCC Institute of Information Technology College for giving constant encouragement in doing research in the field of image processing.

6. References

1. Mindek M. Finite state automata and image recognition. Snasel V, Pokorny J, Richta K, editors. Dateso; 2004. p. 141–51. ISBN: 80-248-0457-3.

2. Turk MA, Pentland AP. Face recognition using eigen faces. IEEE Proc of Computer Vision and Pattern Recognition; 1991 Jun. p. 586–91.
3. Pham-Ngoc P-T, Huynh Q-L. Robust face detection under challenges of rotation, pose and occlusion; 2010 Jan 26. Available from: http://www.fas.hcmut.edu.vn/webhn10/Baocao/PDF/YVSM_PhuongTrinh.pdf
4. Beymer D. Face recognition under varying pose. IEEE Conf on Comp Vision and Pattern Recognition; 1994. p. 756–1.
5. Jolliffe IT. Principal component analysis springer series in statistics; 2002. p. 1–9. DOI: 10.1007/0-387-22440-8_1.
6. Brunelli R, Poggio T. Face recognition: Features versus Templates. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1993; 15(1042–52).
7. Dowson D, Landau B. The Frechet distance between multivariate normal distributions. Journal of Multivariate Analysis. 1982; 12:450–5.
8. Bhattacharya AK. On a measure of diverg. between two statistical populations defined by their probability distribution. Bull Calcutta Math Soc. 1943; 35:99–110.
9. Comaniciu D, Ramesh V, Meer P. Kernel-based object tracking. IEEE Trans on Pattern Analysis and Machine Intelligence. 2003; 25:564–77.
10. Mindek M, Burda M. Image storage, indexing and recognition with finite state automata. IAENG International Journal of Computer Science. 2007; 33(1).
11. Kondor R, Jebara T. A kernel between sets of vectors. Proc Int Conf on Machine Learning, ICML 2003; 2003.
12. Ahmad F, Najam A, Ahmed Z. Image-based face detection and recognition. IJCSI International Journal of Computer Science Issues. 2012 Nov; 9(6).