Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 5, May 2015, pg.608 – 616

RESEARCH ARTICLE

Comparative Study and Enhancement of Face Recognition

Name of Author = **Kavita Devi**
Name of Guide = **Mr. Tarun Dalal**

Abstract: This research mainly focus the recognize a person's identity is important mainly for security reason, but it could also be used to obtain quick access to medical, criminal, or any type of records. Solving this problem is important because it could allow personnel to take preventive action, provide better service - in the case of a doctor's appointment, or allow a person access to a secure area.

Index Terms— MATLAB, Image Processing, Eigen Vector, Eigen Values, Euclidean Distance, Signal System, Image Recognition, Image Segmentation

1. INTRODUCTION

The idea for this research came up while studying soft computing. Mainly research is based upon MATLAB comprising image processing. We have read different algorithms based on image processing like Adam's algorithm and fisher face algorithm etc. The images based on these algorithms as mentioned above had low quality and contrary to this context we tried to enhance the quality of the images.

Image Processing

In imaging science, **image processing** is any form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. Image processing usually refers to digital image processing, but optical and analog image processing also are possible.

MATLAB stores an intensity image as a single matrix, with each element of the matrix corresponding to one image pixel. The matrix can be of class double, in which case it contains values in the range [0,1], or of class uint8, in which case the data range is [0,255]. The elements in the intensity matrix represent various intensities, or gray levels, where the intensity 0 represents black and the intensity 1 (or 255) represents full intensity, or white.

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include: Math and computation.

MATLAB (**mat**rix **lab**oratory) is a multi-paradigm numerical computing environment and fourth-generation programming language. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, Fortran and Python.

Image Processing in MATLAB

The Image Processing Toolbox is a collection of functions that extend the capability of the MATLAB ® numeric computing environment. The toolbox supports a wide range of image processing operations, including:

- Geometric operations
- Neighborhood and block operations
- Linear filtering and filter design
- Transforms
- Image analysis and enhancement
- Binary image operations
- Region of interest operations

Many of the toolbox functions are MATLAB M-files, series of MATLAB statements that implement specialized image processing algorithms. You can view the MATLAB code for these functions using the statement:

```
type function_name
```

You can extend the capabilities of the Image Processing Toolbox by writing your own M-files, or by using the toolbox in combination with with other toolboxes, such as the Signal Processing Toolbox and the Wavelet Toolbox.

Images in MATLAB and the Image Processing Toolbox

The basic data structure in MATLAB is the array, an ordered set of real or complex elements. This object is naturally suited to the representation of images, real-valued, ordered sets of color or intensity data. (MATLAB does not support complex-valued images.)

MATLAB stores most images as two-dimensional arrays (i.e., matrices), in which each element of the matrix corresponds to a single pixel in the displayed image. (Pixel is derived from picture element and usually denotes a single dot on a computer display.) For example, an image composed of 200 rows and 300 columns of different colored dots would be stored in MATLAB as a 200-by-300 matrix.

This convention makes working with images in MATLAB similar to working with any other type of matrix data, and makes the full power of MATLAB available for image processing applications. For example, you can select a single pixel from an image matrix using normal matrix subscripting:

```
I(2,15)
```

This command returns the value of the pixel at row 2, column 15 of the image I.

Data Types

By default, MATLAB stores most data in arrays of class double. The data in these arrays is stored as double precision (64-bit) floating-point numbers. All of MATLAB's functions and capabilities work with these arrays. For image processing, however, this data representation is not always ideal.

The number of pixels in an image may be very large; for example, a 1000-by-1000 image has a million pixels. Since each pixel is represented by at least one array element, this image would require about 8 megabytes of memory.

In order to reduce memory requirements, MATLAB supports storing image data in arrays of class uint8. The data in these arrays is stored as 8-bit unsigned integers. Data stored in uint8 arrays requires one eighth as much memory as data in double arrays.

Because the types of values that can be stored in uint8 arrays and double arrays differ, the Image Processing Toolbox uses different conventions for interpreting the values in these arrays. (Non integer values cannot be stored in uint8 arrays, for example, but they can be stored in double arrays.) The next section discusses how the toolbox interprets image data, depending on the class of the data array.

Image Types in the Toolbox

The Image Processing Toolbox supports four basic types of images:

- Indexed images
- Intensity images
- Binary images
- RGB images

Indexed Images

An indexed image consists of two arrays, an image matrix and a colormap. The colormap is an ordered set of values that represent the colors in the image. For each image pixel, the image matrix contains a value that is an index into the colormap.

The colormap is an m-by-3 matrix of class double. Each row of the colormap matrix specifies the red, green, and blue (RGB) values for a single color:

color = [R G B]

R, G, and B are real scalars that range from 0 (black) to 1.0 (full intensity).

The pixels in the image are represented by integers, which are pointers (indices) to color values stored in the colormap. The relationship between the values in the image matrix and the colormap depends on whether the image matrix is of class double or uint8. If the image matrix is of class double, the value 1 points to the first row in the colormap, the value 2 points to the second row, and so on. If the image matrix is of class uint8, there is an offset; the value 0 points to the first row in the colormap, the value 1 points to the second row, and so on. The uint8 convention is also used in graphics file formats, and enables 8-bit indexed images to support up to 256 colors. In the image above, the image matrix is of class double, so there is no offset. For example, the value 5 points to the fifth row of the colormap.

2. Literature Review

A **face recognition system** is a computer application for automatically identifying or verifying a person from a digital image or a video frame from a video source. One of the ways to do this is by comparing selected facial features from the image and a facial database.

It is typically used in security systems and can be compared to other biometrics such as fingerprint or eye iris recognition systems.

Some facial recognition algorithms identify facial features by extracting landmarks, or features, from an image of the subject's face. For example, an algorithm may analyze the relative position, size, and/or shape of the eyes, nose, cheekbones, and jaw. These features are then used to search for other images with matching features. Other algorithms normalize a gallery of face images and then compress the face data, only saving the data in the image that is useful for face recognition. A probe image is then compared with the face data. One of the earliest successful systems is based on template matching techniques applied to a set of salient facial features, providing a sort of compressed face representation.

Recognition algorithms can be divided into two main approaches, geometric, which looks at distinguishing features, or photometric, which is a statistical approach that distills an image into values and compares the values with templates to eliminate variances.

Popular recognition algorithms include Principal Component Analysis using ***eigen faces, Linear Discriminate Analysis, Elastic Bunch Graph Matching using the Fisherface algorithm, the Hidden Markov model, the Multilinear Subspace Learning using tensor representation, and the neuronal motivated dynamic link matching.***

A newly emerging trend, claimed to achieve improved accuracies, is three-dimensional face recognition. This technique uses 3D sensors to capture information about the shape of a face.

This information is then used to identify distinctive features on the surface of a face, such as the contour of the eye sockets, nose, and chin.

One advantage of 3D facial recognition is that it is not affected by changes in lighting like other techniques. It can also identify a face from a range of viewing angles, including a profile view. Three-dimensional data points from a face vastly improve the precision of facial recognition. 3D research is enhanced by the development of sophisticated sensors that do a better job of capturing 3D face imagery. The sensors work by projecting structured light onto the face. Up to a dozen or more of these image sensors can be placed on the same CMOS chip—each sensor captures a different part of the spectrum.

Even a perfect 3D matching technique could be sensitive to expressions. For that goal a group at the technique applied tools from metric geometry to treat expressions as isometries a company called Vision Access created a firm solution for 3D facial recognition. The company was later acquired by the biometric access company Bioscrypt Inc. which developed a version known as 3D FastPass.

Skin texture analysis

Another emerging trend uses the visual details of the skin, as captured in standard digital or scanned images. This technique, called skin texture analysis, turns the unique lines, patterns, and spots apparent in a person's skin into a mathematical space.

Tests have shown that with the addition of skin texture analysis, performance in recognizing faces can increase 20 to 25 percent.

As one of the most successful application of image analysis and understanding, face Recognition has recently received significant attention, especially during the past several years. This is evidenced by the emergence of face recognition conferences such as AFGR and AVBPA, and systematic empirical evaluations of face recognition techniques, including the FERE and XM2VTS Protocols. There are at least two reasons for this trend; the first is the wide range of commercial and law enforcement applications and the second is the availability of feasible technologies after 30 years of research.

3. SYSTEM REQUIREMENTS

Hardware Requirement:

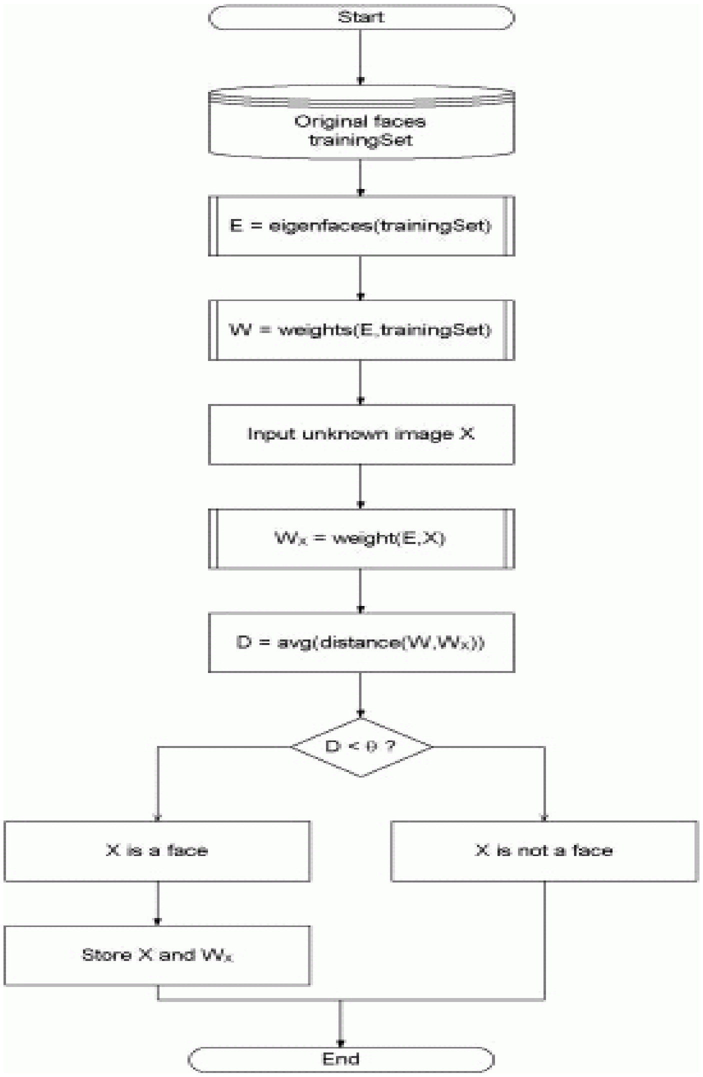
- PROCESSOR: Intel(R) Core(TM) i3-2310M CPU@2.27Ghz
- INSTALLED MEMORY (RAM): 2.00 GB
- SYSTEM TYPE: 32/64 bit operating system
- SYSTEM NAME: Windows 7 ultimate

Software Requirement:

- MATLAB
- PICASSA
- PhotoShop
- JRE.

4. PROPOSED DESIGN OF MODEL

FLOWCHART


Start
Original faces trainingSet
E = eigenfaces(trainingSet)
W = weights(E,trainingSet)
Input unknown image X
W_x = weight(E,X)
D = avg(distance(W,W_x))
D < θ ?
X is a face
X is not a face
Store X and W_x
End

5. IMPLEMENTATION

1. First obtain a set S of M face images. Each image is transformed into a vector of size n and placed into the set.

$$S = \{ \Gamma_1, \Gamma_2, \Gamma_3, \ \ldots\ldots\ldots \ , \Gamma_M \}$$

2. After we have obtained our set, we will obtain mean image Ψ

$$\Psi = \frac{1}{M} \sum_{n=1}^{M} \Gamma_n$$

3. Then we will find the difference Φ between the in-put image and mean image.

$$\Phi_i = \Gamma_i - \Psi$$

4. Now we seek a set of M orthonormal vector, un, (which best describe the distribution of data). Now the kth vector, uk, is chosen such that.

$$\lambda_k = \frac{1}{M} \sum_{n=1}^{M} \left(u_k^T \Phi_n \right)^2$$

5. We obtain the covariance matrix C in the following manner.

$$\begin{aligned} C &= \frac{1}{M} \sum_{n=1}^{M} \Phi_n \Phi_n^T \\ &= AA^T \end{aligned}$$

$$A = \{ \Phi_1, \Phi_2, \Phi_3, \ \ldots\ldots \ , \Phi_n \}$$

6. AT

7. $$L_{mn} = \Phi_m^T \Phi_n$$

8. $$u_l = \sum_{k=1}^{M} v_{lk} \Phi_k \qquad\qquad l = 1, \ldots\ldots , M$$

9. Once we have found the Eigen vectors we obtain the Eigen images of the set of images.

6. RECOGNITION PROCEDURE

A new face is transformed into its Eigen image components. Now we compare our input image with our mean image and multiply their difference with each Eigen vector of matrix. Each value would represent a weight and would be saved on a vector (say omega).Now we determine which face class provides the best description for the input image. It is done by minimizing the Euclidean distance.

The input face is considered to belong a class, if Euclidean distance is below than established threshold, and then face is considered to be a known face if difference is above than given threshold, the image can be considered as unknown face. If input image is below than two thresholds, the image is determined not to be a face.

7. Limitation

The limitation of this model is that this will response to those inputs which will match the mean value of the training set image and corresponds to the eigen faces.

8. Area of Application

This research can be used in setting the passwords in the personal computers where your face is taken as the password of your personal computer.

This can be used in the various offices to maintain the records of the employees with their faces as their account identity.

This help in the medical assistance of the patients in the medical field, setting their face identity as the proof of all reports belonging to him.

9. Scope and Conclusion

First we input a known image and observed the Euclidean distance. This distance tells us how close the input image is from the image on our training set. Based on maximum and minimum distances we can make a decision of whether the face is a known face, an unknown face or not a face at all.

This model can be designed using other various programming techniques and the languages such as C++, JAVA, etc. but matlab can do this with ease using its image processing applications and tool box.

REFERENCES

[1] MIT University media laboratory, USA.

[2] Western Carolina university online li-brary,Cullowhee,North Carolina(USA).

[3] History behind Eigen faces.

[4] D. Pissaarenko "Eigenface-based facial recognition.

[5] Delac, k.Grgic,M.Liatsis "Appearance based statistical methods for facial recognition"(Croatia).

[6] A. Marion *An Introduction to Image Processing*, Chapman and Hall, 1991

[7] Azeema Sultana, Dr. M. Meenakshi, "Design and Development of FPGA based Adaptive Thresholder for Image Processing Applications" ,on line access

[8] Gerhard X. Ritter; Joseph N. Wilson, " Handbook of Computer Vision Algorithms in Image Algebra" CRC Press, CRC Press LLC ISBN:0849326362 Pub Date: 05/01/96

[9] N. Nacereddine, L. Hamami, M. Tridi, and N. Oucief , "Non-Parametric Histogram-Based Thresholding Methods for Weld Defect Detection in Radiography " ,online access.

[10] Otsu,N., "A Threshold Selection Method from Gray-Level istograms,"IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1, 1979, pp. 62-66.

[11] Elham Ashari , Richard Hornsey, " FPGA Implementation of Real-Time Adaptive Image Thresholding" ,online access