

**Department of Artificial Intelligence**

**Faculty of Computer Science and Information Technology**

**University of Malaya**

**WID3008: (REPORT)**

**TITLE:**

**FACE RECOGNITION USING MATLAB**

**NAME: MOSTAFA RAFIUR WASIB**

**MATRIC NO: WEA150702**

**LECTURER NAME: DR WOO CHAW SENG**

# **Table of Contents**

|  |  |
| --- | --- |
| **Item** | **Page** |
| **Abstract** | 3 |
| **1. Chapter 1 Introduction** | 4-5 |
| a) 1.1 Description of face recognition | 4 |
| b) 1.2 Problem Statements | 4 |
| c) 1.3 Objectives | 5 |
| d) 1.4 Project Schedule | 5 |
| **2. Chapter 2 Literature Review** | 6 |
| **3. Chapter 3 Research Methodology** | 6-7 |
| **4. Chapter 4 Requirements** | 7 |
| **5. Chapter 5 System Design and Analysis** | 8-13 |
| a) 5.1 [Preparing dataset](https://www.nzfaruqui.com/face-recognition-using-matlab/#prepare), | 8-10 |
| b) 5.2 [Loading the dataset](https://www.nzfaruqui.com/face-recognition-using-matlab/#dataset) | 10-11 |
| c) 5.3 [Recognizing the face](https://www.nzfaruqui.com/face-recognition-using-matlab/#recognize) | 12-13 |
| **6. Result** | 14 |
| **7. Conclusion** | 14 |
| **8. References** | 15-16 |

**Abstract:**

Face recognition has become an important issue in many applications such as security systems, credit card verification and criminal identification. Face recognition is more secure in security system because facial image had been used as the ID. It also helps to avoid any duplicated identification. Face recognition helps to recognize the facial image specially to identifying certain criminals. Identifying and comparing faces in images is a very complex task, this is probably why it has attracted so many researchers in the latest years. Common method used in face recognition like eigenface method will be discussed. The objectives of this project are to design and develop a face recognition using MATLAB software beside to comprehend eigenfaces method of recognizing faces images. The face space is defined by the "eigenface", which are the eigenvectors of the set of faces, they do not necessarily correspond to isolated features such as eyes, ears and noses. Eigenfaces approach seems to be an adequate method to be used in face recognition due to its simplicity, speed and learning capability. Experimental results are given to demonstrate the viability of the proposed face recognition method.

**Chapter 1: Introduction**

* 1. **Description of face recognition**

Face recognition is the process of identifying one or more people in images or videos by analyzing and comparing patterns. Algorithms for face recognition typically extract facial features and compare them to a database to find the best match.

**1.2 Problem statement:**

(1) In security system, many types of password are used to access the private and confidential data. Such password can be as insert characters (key in pin) and touch smart card using RFID technology. Passwords and PINs are hard to remember and can be stolen or guessed; cards, tokens, keys and the like can be misplaced, forgotten, purloined or duplicated; magnetic cards can become corrupted and unreadable. By developed face recognition it more secure because facial image had been used as the ID. It also helps to avoid any duplicated identification.

(2) Other problem is to identify certain criminals especially in identification technique used by the police. Face recognition helps to recognize the facial image in more efficient and accurate in order to match with the identity stored in the database.

**1.3 Objectives:**

(1) To comprehend eigenfaces method of recognizing faces images and tests its accuracy.

(2) To design and develop a face recognition using MATLAB

(3) To set up test platform for determining the accuracy of this technique.

**1.4 Project Schedule:**

|  |  |  |
| --- | --- | --- |
| **Activity** | **Duration** | **Start Date** |
| Introduction of MATLAB | 1 week | 23.10.2018 |
| Face Recognition introduction | 1 Week | 30.10.2018 |
| System requirement for the project | 1 Week | 06.11.2018 |
| Algorithm pseudocode for the project | 1 week | 13.11.2018 |
| System development | 1 Week | 20.11.2018 |
| Project Report Preparation | 1 Week | 27.11.2018 |

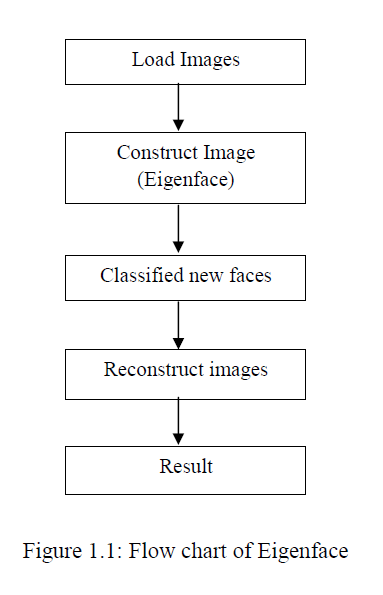
**Chapter 2: Literature Review**

Face recognition is typically used in security systems. Besides that, it is also used in human computer interaction. In order to develop this project eigenfaces method is used for training and testing faces. It has received significant attention up to the point that some face recognition conferences have emerged. A general statement of the problem can be formulated as follows, given still or video images of a scene, one or more persons in the scene can be identified using a stored database of faces. The solution of the problem involves face detection, feature extraction from the face regions and recognition. To develop this project, we used the eigenfaces method.

**Chapter 3: Methodology**

Development of Eigenface method have some process which the first stage is load images from database. This process is to load all training images and return their contents (intensity values). Then construct the image which is in this part there have some calculation to get mean image, normalized image, covariance matrix and determine the eigenvector. Third stage is classified the new image where we need to insert the new name image to continue the process. The functionality of this process is to give a test image, this function can determine whether it is a face image or not. If it is a face image, does it belong to any of the existing face classes? If so, which face class does it correspond to?

The result will appear as bar chart after done the process. To more clearly about the flow refer to Figure 1.1



**Chapter 4: Requirements**

* Need clear photo of the loaded image
* The loaded photo must have cleat edges which can be detected correctly.
* The photo must comply with MATLAB session.
* File format should be corrected and greyscale format.
* Loaded image should avoid any object eg, hat,wig which can alter results
* Image should be front faced and straight.

**Chapter 5: Analysis and Design**

To keep the face recognition system as simple as possible, I used eigenvector-based recognition system.

For recognizing faces using MATLAB we need to do three things to implement such a system.

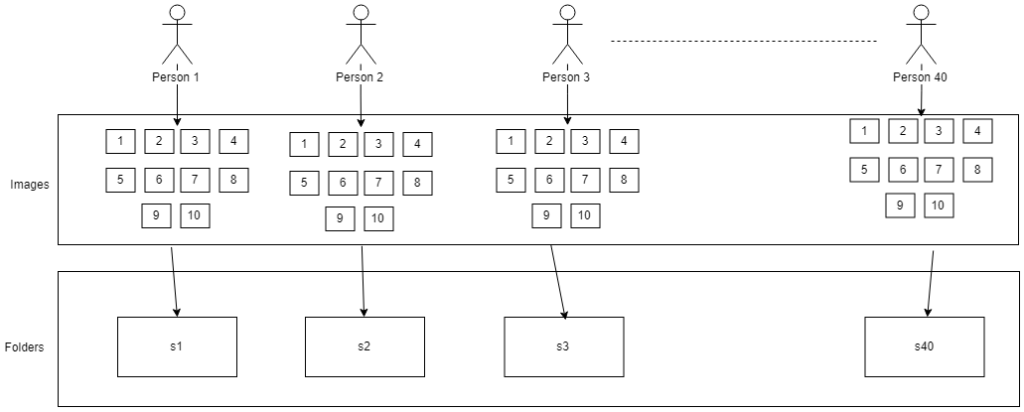
These are –

1. [Preparing dataset](https://www.nzfaruqui.com/face-recognition-using-matlab/#prepare),
2. [Loading the dataset](https://www.nzfaruqui.com/face-recognition-using-matlab/#dataset) and
3. [Recognizing the face](https://www.nzfaruqui.com/face-recognition-using-matlab/#recognize)

First, we will prepare the dataset. Although we prepare dataset automatically, I will do it manually here, so that you can understand what is happening. Then I will show you how to load the dataset in MATLAB. Finally, I will show how to recognize faces using these datasets. Let’s start with preparing the dataset.

5.1 Preparing the Dataset

A dataset of 40 people is created. Each of these people has 10 images with different poses. That means in total there are 40 x 10 = 400 images. For every individual, there is separate folders. Explaining it in this way will create confusion. That is why I have prepared figure Fig 1.1. I hope it will help you to understand how to prepare the dataset.



In figure 1.1, you can see in the top row, there are 40 people marked as 1, 2, 3 to 40. Each of them has 10 images. These images are in grayscale. All of these images must have to be of same dimension and resolution. Finally, I have kept images of every individuals into separate folder. In the figure 1.1, the s1, s2, s3 ….. s40 are representing the folders. I think you have understood the structure of the dataset.

Let’s summarize the steps of preparing the dataset –

1. 10 images for every person
2. 1 folder for each person (s1, s2, s3 and so on)
3. Images must be in grayscale
4. Images must have to be of same resolution and dimension. I have taken 92 x 112-pixel image.
5. The name of the image must have to be numeric such as 1, 2, 3.
6. And the images must have to have same extension such as bmp, pgm and so on. Do not mix up different extensions of image.

**5.2 Loading the Dataset**

After preparing the dataset, next task is loading the dataset. We need to implement a function in MATLAB to load the dataset. We can use this function to load another dataset as well.  Let’s name it ‘load\_database.m’.



In the first line, I have declared the name of the function ‘load\_database()’ It doesn’t take any input. That is why I have used empty parenthesis. But it returns numeric form of images. The images to be returned will be stored in a variable named ‘output\_value’.

After that I have taken two more variables named ‘loaded’ and ‘numeric\_Images’. These variables are persistent type.

*(N.B: “The persistent variables are local to the function in which they are declared; yet their values are retained in memory between calls to the function. Persistent variables are similar to global variables because MATLAB creates permanent storage for both. They differ from global variables in that persistent variables are known only to the function in which they are declared. This prevents persistent variables from being changed by other functions or from the MATLAB command line)*

After that, in the ‘if’ condition, I have checked if the ‘loaded’ variable is empty. If it is empty only then we will load the dataset. The persistent variables permanently store the data. And we need to load the dataset only once. That is why it is important to check if the variable is empty or not at the beginning.

We have 40 images. Each of them has 92 x 112 = 10304 pixels. That is why we have to take 10304 zeros for 40 times. Later the pixel values of the images will replace these ‘zeros’. Then inside the ‘for loop’ using [strcat function](https://www.youtube.com/watch?v=J_iilGCrF1M), I am concatenating the names of the folders (s1, s2, s3 and so on), names of the images (1, 2, 3, and so on) and the extension of the images (I used .pgm images).

After that using ‘imread’ function, I’m loading the images. It is necessary to reshape the images. After loading the images, I have used ‘reshape’ function to convert the images into single column matrix. I also used ‘size’ function which gives the size of the rows and columns of the images.

After that I have converted the images into 8 bit unsigned integer to reduce the memory usage. Using ‘uint8’ function, we can directly convert data into 8 bit unsigned integer in MATLAB. You might have noticed I used ‘loaded = 1’ before the last line. It is to prevent the function from further loading the dataset. There is an ‘if’ condition before the ‘for loop’. It will become false when the ‘loaded’ variable has a value. Finally, this function returns the images in 8-bit unsigned integer form.

5.3 Face Recognition using MATLAB

To recognize the faces, I loaded the dataset first. After that using random function, I generated a random index. Using the sequence of random index, I loaded the image which will be recognized later. Rest of the images are also loaded into a separate variable.

After that, I calculated the mean of all the images and subtracted the mean from them.  The eigenvectors were calculated on these images. Upon having the eigenvalues, I created matrix where each row contains the signature of individual images. That means now we have the eigenvalues and the signature of the image to identify them.

In the last section, I subtracted the mean value from the image which we want to recognize. Then multiplied it with the eigenvector. Finally based on the difference between current image signatures with the signature I have mentioned above, I have predicted the recognized face.

The accuracy of this method is around 94%. That means out of 100 trial, it may make four mistakes.



**Chapter 6: Result**

Here are four results of this program –



The result of face recognition using MATLAB

**Conclusion**

In this project I did not test images with color as well as different lighting environment. This is a very simple project thus it can be improved. By using machine learning and neural network the results can be more accurate and it can handle real life situations. I plan to work with face recognition algorithm with other programming language which are open source. Though MATLAB provided various tools I cannot use it more than one month since it is a trial version. This project can further be improved with more resources which is difficult to obtain for single student. I hope university will support us with the license of this software for better project.

**References:**

|  |
| --- |
| 1. Zelelew, H. M., Almuntashri, A., Agaian, S., & Papagiannakis, A. T. (2013). An improved image processing technique for asphalt concrete X-ray CT images. *Road Materials and Pavement Design*, *14*(2), 341-359. |
|  |
| 1. *Costarelli, D., & Vinti, G. (2013). Approximation by nonlinear multivariate sampling Kantorovich type operators and applications to image processing. Numerical Functional Analysis and Optimization, 34(8), 819-844.* 2. *Kawaji, H., Hatada, K., Yamasaki, T., & Aizawa, K. (2010, October). Image-based indoor positioning system: fast image matching using omnidirectional panoramic images. In Proceedings of the 1st ACM international workshop on Multimodal pervasive video analysis (pp. 1-4). ACM.* 3. *Plaza, A., Plaza, J., Paz, A., & Sanchez, S. (2011). Parallel hyperspectral image and signal processing [applications corner]. IEEE Signal Processing Magazine, 28(3), 119-126.* 4. *Arbelaez, P.,* *Maire, M.,* *Fowlkes, C., &* *Malik, J. (2011). Contour detection and hierarchical image segmentation. IEEE transactions on pattern analysis and machine intelligence, 33(5), 898-916.* 5. *Joyce, K. E., Belliss, S. E., Samsonov, S. V., McNeill, S. J., & Glassey, P. J. (2009). A review of the status of satellite remote sensing and image processing techniques for mapping natural hazards and disasters. Progress in Physical Geography, 33(2), 183-207.* 6. *Lu, D., & Weng, Q. (2007). A survey of image classification methods and techniques for improving classification performance. International journal of Remote sensing, 28(5), 823-870.* 7. *K* *Kuzniecky, R. I., & Jackson, G. D. (2005). Magnetic resonance in epilepsy: Neuroimaging techniques. Amsterdam: Elsevier. doi: https://doi.org/10.1016/B978-012431152-7/50012-4. 249–269* 8. *Sz25. eliski, R. (2010). Computer vision: algorithms and applications. Springer Science & Business Media.* 9. *Gokturk, S. B., Sumengen, B., Vu, D., Dalal, N., Yang, D., Lin, X., ... & Vanhoucke, V. (2010). U.S. Patent No. 7,657,100. Washington, DC: U.S. Patent and Trademark Office.* 10. *Joutou, T., & Yanai, K. (2009, November). A food image recognition system with multiple kernel learning. In Image Processing (ICIP), 2009 16th IEEE International Conference on (pp. 285-288). IEEE.* 11. *Nuruzzaman Faruqui (October 22, 2018) Face Recognition using MATLAB -* [*https://www.nzfaruqui.com/face-recognition-using-matlab/*](https://www.nzfaruqui.com/face-recognition-using-matlab/) 12. *The MathWorks, Inc. (1994-2005)- MATLAB Function Reference* [*http://matlab.izmiran.ru/help/techdoc/ref/persistent.html*](http://matlab.izmiran.ru/help/techdoc/ref/persistent.html) |

*((N.B Dataset reference:*

[*https://drive.google.com/file/d/14g\_fD74pDmX9AsgkAQG59cxQSh0fgRr2/view*](https://drive.google.com/file/d/14g_fD74pDmX9AsgkAQG59cxQSh0fgRr2/view) ))