Project Report

ECE 09.341 Section #3: Final Project
15 December 2017

Face identification system using MATLAB

Stephen Glass
Electrical & Computer Engineering, Rowan University

Table of Contents

Introduction

A facial recognition system is a capable of identifying or verifying a person from an image or video. For the case of this project, facial features are extracted from an image and compared against a trained feature database. Facial features can be extracted from an image using processing techniques such as the discrete-cosine-transform (DCT) and matched to a database using a simple k-nearest-neighbor (kNN) classifier.

Facial recognition systems are popular as biometric security applications. Recently, it has become popular in commercial products such as mobile phones and home security. The newly released Apple iPhone X uses facial recognition as a method for users to unlock their devices.[1] Additionally, facial recognition is important for other applications which include criminal identification and people tracking. In a study by Georgetown University, it was found that half of all American adults are in a police face recognition database for criminal detection.[2] However, even commercially, it was found that department stores, casinos, and cruise ships used facial recognition to detect and track people.[3]

The objectives of this project are to:

- Use MATLAB to extract facial features with discrete-cosine transform (DCT).
- Use MATLAB to implement a kNN classifier for facial identification.
- Investigate the success rate of facial identification as a function of the feature vector dimension and value of K.

Protocol, results and discussion

Part 1 - Two dimensional (2-D) DCT and inverse 2-D DCT

The facial database used in this project is the AT&T Laboratories face database, also known as the Olivetti Research Laboratory (ORL) face database. The database contains images of 40 subjects and there are 10 images per subject. For some subjects, the images were taken at different times, varying lighting, facial expressions, and facial details. All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position. The size of each image is 112 by 92 pixels, with 256 grey levels per pixel.

This part will show the following:

- Reading and plotting a database image
- Finding the 2-D DCT of the image
- Plotting the 2-D DCT
- Finding the 2-D DCT to recover the original image and plot it

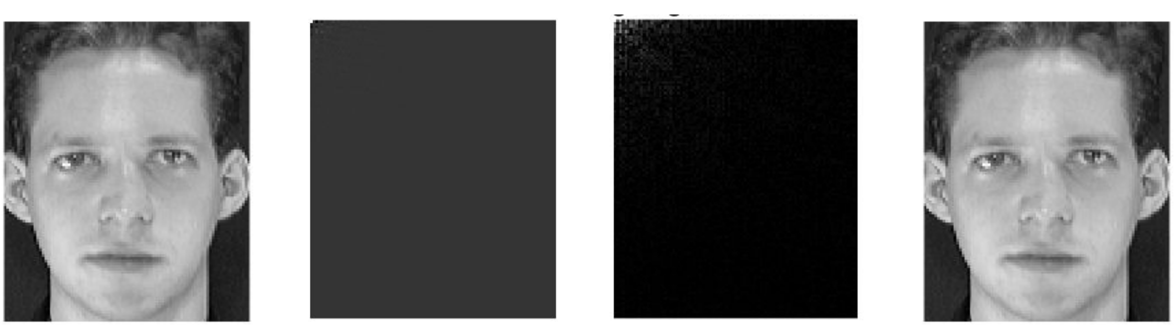

Fig. 1 - 4 (from left to right): original image, 2-D DCT of image, log magnitude of 2-D DCT, inverse 2-D DCT image

MATLAB reads the original image as a 112 by 92 matrix of grayscale values from 0 to 256. Grayscale values represent the brightness of the pixel (byte image) where the number is stored as an 8-bit integer giving a possible value range from 0 to 255. Typically zero is taken to be black, and 255 is taken to be white. The values in between make up different shades of gray.[4] Figure 1 shown above is the original image read by MATLAB from the facial database.

Figure 2 shows the 2-D discrete cosine transform (DCT) of the original image. However, the 2-D DCT plot does not clearly show the DCT coefficients with large magnitudes are concentrated in the top-left corner of the matrix. Figure 3 shows the computation of the log magnitude of the 2-D DCT. The DCT coefficients are now more easily visible in the top-left corner of the figure.

Figure 4 shows the inverse 2-D DCT of the transformed image. It is seen the inverse 2-D DCT recovers the original image (Figure 1 and Figure 4 are identical). *Part 1 - Two dimensional (2-D) DCT and inverse 2-D DCT* explained and visualized the concept of discrete cosine transform (DCT) and acquainted myself with the AT&T Laboratories facial database. Additionally, this part explained the importance of grayscale values in image processing.

Part 2 - Feature Extraction

This part will develop an understanding of the program which converts an image into a DCT feature vector of a specified dimension. The scanning of the 2-D DCT of an image is performed using a 'zigzag' method for conversion to a one-dimensional (1-D) feature vector as shown in Figure 5 below.

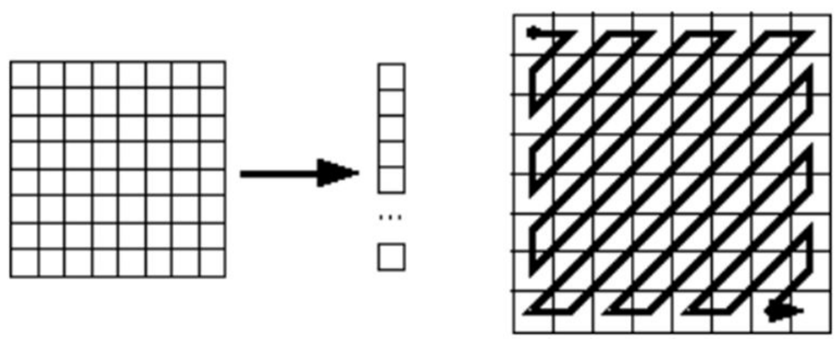

Fig. 5: Zigzag conversion of a 2-D DCT array to a 1-D feature vector

The provided code *findfeatures.m* can be used to truncate the 1-D feature vector to any dimension *L* by retaining only the first *L* components. Figures 6 - 8 below show the feature vectors of dimension 9, 35, and 100 for *Subject 8, Image 1* in the facial database.

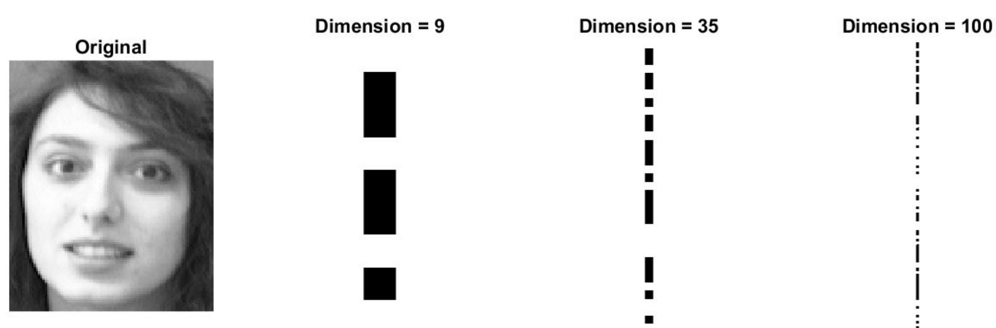

Fig. 6 - 8 (left to right): Feature vectors of dimension 9, 35, and 100 for subject 8, image 1

The feature vectors contain a larger set of data as the dimension is increased. When the dimension is lower, the feature vector becomes truncated and contains less feature data. Figures 9 - 11 below show the feature vectors of dimension 9, 35, and 100 for *Subject 40, Image 1* in the facial database.

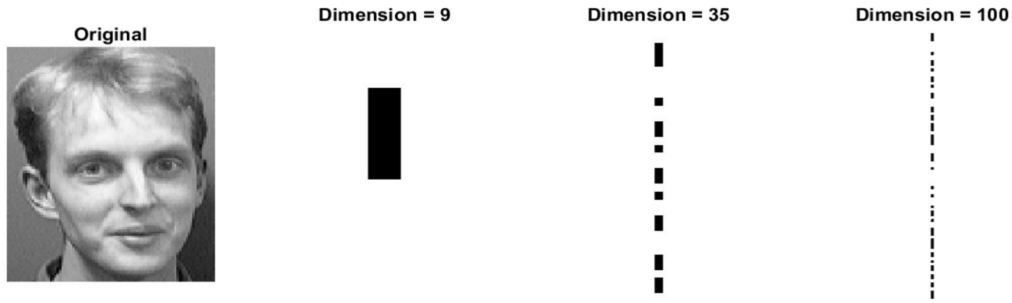

Fig. 9 - 11 (left to right): Feature vectors of dimension 9, 35, and 100 for subject 40, image 1

There are clear differences between feature vectors of the two subjects. Visually inspecting the two original images, there are differences between the average pixel brightness, where subject 40 seems to be brighter. These inspections translate to the results shown as seen by the different patterns in the feature vectors.

Part 3 - Training the Face Identification System

This part will explain the process for the training the facial database for the k-nearest-neighbor (kNN) classifier. The provided code *face_recog_knn_train.m* is able to train the the classifier using the first five images of each subject (1.pgm to 5.pgm). The code will train the classifier using all 40 subjects (200 images total) with a 2-D DCT feature dimension of 70. Figure 12 shown below explains the process of constructing the kNN classifier database.

Fig. 12: Flowchart describing the process of constructing the kNN classifier database

The feature vector of each image (1.pgm - 5.pgm) are gathered and placed into a database that will be used to compare against for the image recognition system. The following variables are output to *raw_data.mat*:

- DCT_Coef: the length of the 2-D DCT feature vector (70)
- F_range: the range of subjects to train (1 x 40)
- nSubjects: the amount of subjects trained (40)
- Trdata_raw: the database of feature vectors of each analyzed image
- Trclass: database assigning a label of the subject number to each feature vector

The contents of *raw_data.mat* can be used in the facial identification system as a database to compare feature vectors against and determine the true identity of subject.

Part 4 - Performance Evaluation of the Face Identification System - Clean Data

This part will implement the kNN classifier to evaluate the performance of the trained face identification system. The last five images of each subject (6.pgm to 10.pgm) will be used to compare against the classifier and determine the true identity of the subject. Figure 13 below describes the process of the performance evaluation procedure.

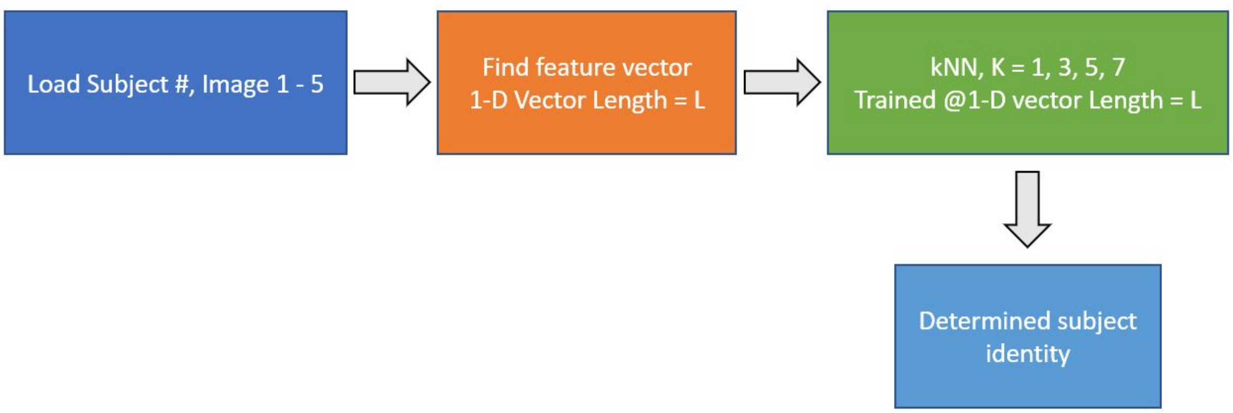

Fig. 13: Performance evaluation procedure using kNN with variable 1-D feature length and K

As shown in Figure 13, the function loads the remaining five images for each subject and finds the feature vectors of each image. The feature vector is input into the kNN search to determine the subject identity. An example output from the function is shown below in Figure 14.

```
True Subject: 39, Calculated Subject: 39
True Subject: 39, Calculated Subject: 39
True Subject: 40, Calculated Subject: 5
True Subject: 40, Calculated Subject: 40
True Subject: 40, Calculated Subject: 40
True Subject: 40, Calculated Subject: 5
True Subject: 40, Calculated Subject: 40


 Amount Correct: 183
 Percent Success: 91.500000%>> 1
```

Fig. 14: Sample output of face recognition program when DCT_Coef = 70 and K = 1

The identification success rate is determined by the amount of subjects **correctly** determined divided by the total number of face images tested expressed as a percent. The identification success rate is a function of:

- Dimension of the 1-D feature vector (from 25 to 100 in steps of 15).
- Values of K = 1, 3, 5, 7.

The system is retained for each new dimension of the 1-D feature vector. A 3D-surface plot of the identification success rate is shown below in Figure 15.

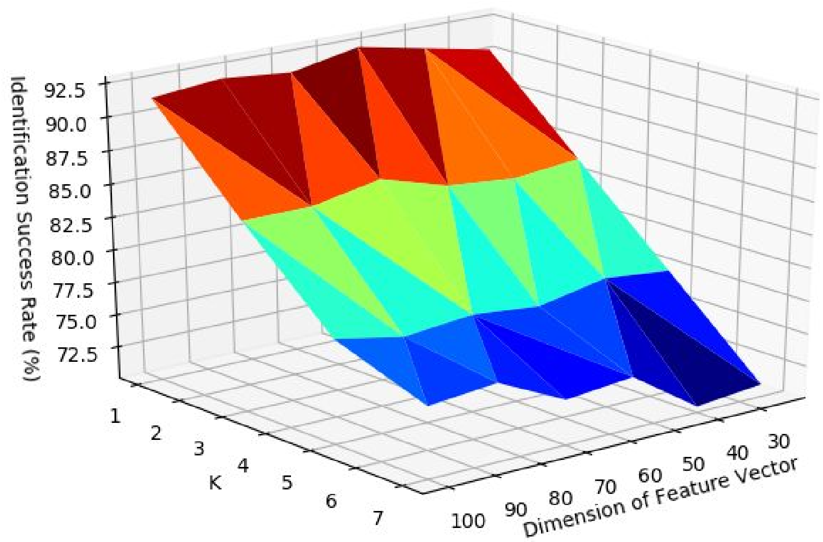

Fig. 15: 3D-Surface plot for identification rate as function of feature vector dimension and K

The identification success rate can be seen at the highest percent yield when K = 1. Feature vector length had minimal impact on overall performance. The maximum identification success rate of 92.5% occurred when the dimension of feature vector is 55 and K = 1. The minimum identification success rate of 70.5% occurred when the dimension of feature is 40 and K = 7. The results of identification results are shown in Figure 16 below.

Identification Success Rate (%)

Dimension of Feature Vector \ K	1	3	5	7
25	90.5	84.0	77.5	71.0
40	91.5	83.5	78.0	70.5
55	92.5	84.0	77.0	74.0
70	91.5	85.5	77.5	73.5
85	92.0	84.5	77.0	76.0
100	91.5	84.5	78.0	75.5

Fig. 16: Chart of the results for identification rate as function of feature vector dimension and K

Summary and Conclusions

The objectives set forth in this project were successfully completed, analyzed, and discussed. Using the AT&T Laboratories face database we were able to extract facial features using discrete-cosine transform (DCT) and utilize a kNN classifier to implement facial identification.

The kNN classifier was trained at varying DCT feature vector lengths and evaluated with values of K = 1, 3, 5, 7. The performance of the kNN classifier was evaluated as a function of the DCT feature vector length and the value of K.

It was found that the identification success rate increased as K approached 1. The dimension of the feature length was found to have minimal impact on the identification success rate. The best values for dimension of feature length and K is 55 and 1 which yields the identification success rate of 92.5%. The results of the project show how performance can be impacted due to mismatched training and testing conditions. To increase the success rate you would require a larger training data set and/or use another method for classifying and comparing data.

The project explained the basic concept of image processing and the usage of signals and systems for a real-world applications such as facial recognition.

References

[1] “About Face ID advanced technology,” Apple Support, 05-Nov-2017. [Online]. Available: https://support.apple.com/en-us/HT208108. [Accessed: 10-Dec-2017].

[2] “Half of All American Adults are in a Police Face Recognition Database, New Report Finds,” Half of All American Adults are in a Police Face Recognition Database, New Report Finds, 18-Oct-2016. [Online]. Available: https://www.law.georgetown.edu/news/press-releases/half-of-all-american-adults-are-in-a-police-face-recognition-database-new-report-finds.cfm. [Accessed: 10-Dec-2017].

[3] “Facial Recognition: Who's Tracking You in Public?,” Facial Recognition: Who's Tracking You in Public?, 30-Dec-2015. [Online]. Available: https://www.consumerreports.org/privacy/facial-recognition-who-is-tracking-you-in-public1/. [Accessed: 10-Dec-2017].

[4] R. Fisher, S. Perkins, A. Walker, and E. Wolfart, “Pixel Values,” Pixel Values, 2003. [Online]. Available: https://homepages.inf.ed.ac.uk/rbf/HIPR2/value.htm. [Accessed: 10-Dec-2017].