LABORATORY PROJECT ON FACE IDENTIFICATION SYSTEM

Objectives

This goals of this project is to (1) accomplish a software implementation of face identification in MATLAB using the Discrete Cosine Transform (DCT) with a k-nearest neighbor (kNN) classifier and (2) develop an understanding of how the kNN classifier works.

Files supplied

att_faces.zip, findfeatures.m, face_recog_knn_train.m

AT&T Database

The database used for the experiments is the AT&T Laboratories face database. This is also known as the Olivetti Research Laboratory (ORL) face database. The database will be emailed to you as the file 'att_faces.zip'. In this database, there are 40 subjects corresponding to the subdirectories s1 to s40. In each subdirectory, there are 10 images labeled as 1.pgm, 2.pgm, … , 10.pgm. Therefore, there are 10 images per subject and the entire database consists of 400 images. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement). The files are in PGM format, and can conveniently be viewed on UNIX (TM) systems using the 'xv' program. They can also be easily read into MATLAB (the programming language used in the project). The size of each image is 112 by 92 pixels, with 256 grey levels per pixel.

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

Acquaint yourself with the AT&T database. Pick any image from the database. The file for this image will be referred to as face.pgm but you can call it as you deem convenient. Do the following:

1. Read and plot the image.
2. Find the 2-D DCT of the image.
3. Plot the 2-D DCT
4. Find the inverse 2-D DCT to recover the original image and plot it

Note that when you read the image, you get a 112 by 92 matrix of greyscale values from 0 to 256. In your report, explain the concept of greyscale values. The MATLAB commands given below will be of use.

```
[img,map]=imread('face.pgm');
imshow(img,map);
img2dct=dct2(img);
imshow(img2dct,map);
```

```
imgrecover=idct2(img2dct);
imshow(imgrecover,map);
```

The 2-D DCT plot does not clearly reveal that the DCT coefficients with large magnitudes are concentrated in the upper left-hand corner of the matrix (that is at the low frequencies). Implement the following lines of code after the code given above so that this concept can be realized using the log magnitude of the DCT.

```
% Compute and plot log magnitude of 2-D DCT
t1=0.01.*abs(imgdct);
t2=0.01*max(max(abs(imgdct)));
c_hat=255*(log10(1+t1)/log10(1+t2));
imshow(c_hat,map);
title('Log Magnitude of 2-D DCT');
```

Comment on what you learned in this part.

Part 2 – Feature Extraction

Program supplied: findfeatures.m

Develop an understanding of the program that converts an image into a DCT feature vector of a specified dimension. Note that the scanning of the 2-D DCT of an image is done in a 'zigzag' fashion (see Figure 1) to convert it into a one-dimensional (1-D) feature vector. Add comments to the code to give clear explanations. The left part of Figure 1 illustrates the concept of converting a 2-D array into a 1-D array. The right part of Figure 1 shows how the 2-D array is scanned ('zigzag' fashion) to get the 1-D array.

For any image in the database not picked in Part 1, the 1-D feature vector can be truncated to any dimension L by retaining only the first L components. Plot the 1-D feature vectors of dimension 9, 35 and 100 for any image in the database and comment on your results. Do the same for an image (not picked in Part 1) from another subject. Do you observe any differences between the feature vectors of the two subjects?

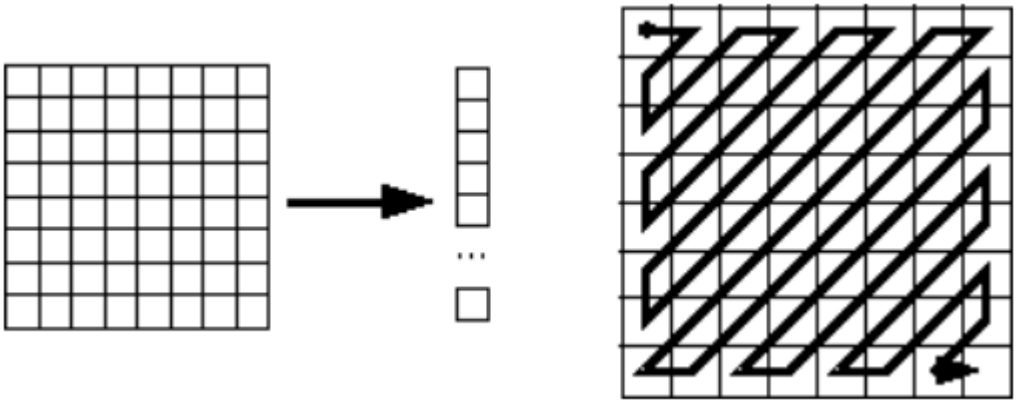

Figure 1. A 'zigzag' scanning of a 2-D DCT array to get a 1-D feature vector

Part 3 – Training the Face Identification System

Program supplied: face_recog_knn_train.m

For each subject, use the first five files (1.pgm to 5.pgm) for training. The program face_recog_knn.train configures the kNN classifier. Run the code as follows: [trdata_raw,trclass]=face_recog_knn_train(subject_range,dct_coef) such that subject_range=[1 40] and dct_coef=70. Explain the code to demonstrate your understanding. Examine the file raw_data.mat and explain what each variable in this file means.

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

Use the files 6.pgm to 10.pgm for the performance evaluation. The identification success rate is the number of face images identified correctly divided by the total number of face images tested (expressed as a percent). If 40 subjects are used, the total number of images tested is 200 (40 subjects multiplied by 5 test images for each subject).

Write a MATLAB function to implement the kNN classifier. It is convenient for your function to load raw_data.mat. The value of k should be an input argument (there may be more inputs depending on your function). Your function should at least output the true identity of each test image, the identity determined by the kNN classifier and the identification success rate. Use the L2 distance measure (consider using the 'norm' function in MATLAB for this).

Investigate the identification success rate as a function of:

- Dimension of the 1-D feature vector (from 25 to 100 in steps of 15). Note that you have to retrain the system for each dimension of the 1-D feature vector.
- Values of k = 1,3,5,7.

Show a 3-D plot of (k, dimension of feature vector, identification success rate as a percent). Comment on your results. What are the best values of k and the dimension of the 1-D feature vector?

Format for Project Write-up Due December 11

Submit the project as a pdf file on canvas.

Outline of Report

1. Title Page
2. Table of Contents
3. Introduction
   Discuss the following:
   - Give a brief synopsis on face recognition. Why is this problem important? Give references.
   - In your own words, list the objectives of the project.

4. Protocol, results and discussion
    - 4.1 Part 1 – Include discussion, results, plots and figures.
    - 4.2 Part 2 – Include discussion, results, plots and figures. Explain findfeatures.m. Answer all questions given in this part.
    - 4.3 Part 3 – Explain face_recog_knn_train.m and answer all questions given in this part.
    - 4.4 Part 4 – Give discussion of results and answer all questions given in this part. Include all plots, results and figures.

    Do not put any listing of code. You do not have to put any listing of MATLAB code.
5. Summary and Conclusions
6. References: Have at least 2. Referencing the lab protocol does not count.
7. Appendices – This is optional and to be used if you deem necessary. Each Appendix should start on a new page and be labeled A.1, A.2, … . No MATLAB code. No plots, tables, experimental results and discussion.