Lab 3 - Image Processing EEE412

Eigenfaces

Report is due **21 days** from the date of running this lab

Objectives:

1- To master the use of image transforms techniques

Introduction:

Eigenfaces is the name given to a set of eigenvectors when they are used in the computer vision problem of human face recognition. The eigenfaces form a **basis set** and recognition of faces can be achieved by comparing how faces are represented by this basis.

Download:

Download the files of Material.part1.rar, Material.part2.rar and Material.part3.rar from the ICE (actually these are three parts for one large rar file.), unzip the file into a folder *Lab3*. Now load into your Matlab workspace the file `data_for_labC.mat`; you will see then in your workspace the following variables:

1- `eigenfaces_blk` which contains 100 eigenfaces (Do not use the last eigenface in this lab report, i.e., you do not need to use the 101 eigenface in the mat)
2- `employees_DB` is an array containing the employees' ID and eigenface weighting parameters.

Background of the tasks:

Suppose you have been asked by a company to develop a software for face recognition. This software will be used by the company to check the identity of their employees, and prevent unauthorized persons from getting in.

This software needs to interface with a database containing the employee records (`employees_DB`). Each record contains the following information:

1) ID
2) Parameters describing the face (the weights of the Eigenface representation)

Tasks:

1. Orthonormal basis. (**15 marks**)

   The variable `eigenfaces_blk` contains 100 eigenfaces, verify that these eigenfaces are orthogonal, and if they are not orthonormal, please normalized them to orthonormal.

2. Evaluating the Eigenfaces weights of a face. (**20 marks**)

   When an image of a face is presented to the system for classification/ recognition, its own weights are found by projecting the image onto the collection of eigenfaces. This provides a set of weights describing this particular face.

   Write a Matlab function which evaluates the weights of a face. The function should have the following declaration:

   ```
   function [weights_of_face] = get_face_weights(im, eigenfaces_blk);
   ```

   Use the function `get_face_weights` to find the weighting parameters for the image *`find_id.jpg`* . Plot these weighting parameters, and comment on whether this plot carries any information about the employee's face or not.

3. Face generation from its "weights". (**20 marks**)

   Write a Matlab function which generates a face from its weights. The function should have the following declaration:

   ```
   function [im] = generate_face_from_weights(weights_of_face,
   eigenfaces_blk)
   ```

   Use the obtained `weights_of_face` in the previous task (i.e., task 2) to synthesise the image of the face using the function `generate_face_from_weights`.

   Comment on how 100 parameters are enough to describe an image of a face which has 450X300 pixels.

4. Recognizing an employee from his/her image. (**25 marks**)

   To recognize someone from his (/her) face's photo the weights of this face need to be compared against all weights in the database to find the closest match.

The Euclidean Distance between two vectors is a simple approach for finding two nearest neighbors; this metric could be used in this task to find the closest face in the database.

Write a Matlab function which finds the employee's ID starting from his/her photo. The function should have the following declaration:

```
function [ID] = get_employees_ID_from_DB (im, employees_DB,
eigenfaces_blk);
```

The first thing you need to do is to use the function `get_face_weights` to generate the weighting parameters for the given face.

Then you need to compare the face weights with all the employees' record in the `employees_DB` to find the closest to the given one. Output the ID of the closest face.

Find the employee's ID of the image `find_id.jpg`, and write it down in your report.

5. Discuss the robustness of the eigenface-based image recognition algorithm under various working conditions, like with salt & pepper noise. How can we make the algorithm more robust? You can thinking about this question following these issues (but not limited), like how to reject the image out of the data base or what should we do if the query image is with noise? (**20 marks**)

Lab Report

Write a **short** report which should contain a **concise description** of your results and observations. Include listings of the Matlab scripts that you have written. Describe each of the images that you were asked to display.

Submit the report electronically into ICE, and a hardcopy version into the collecting box beside the office EB310 (Hand written reports are not accepted) before the deadline.

This page last modified on 2019-10-15 8:13 PM