CMPT 412 Assignment 4

Kai Zhang
301087916

Overview

The fourier face detection procedure is broken down into several stages. First, we convert the images into fourier frequencies, some preprocessing will be done into improve recognition results. After obtaining the fourier representation, we need selecting the most relevant frequencies to distinguish from. For that, lower quadrant method was used. Once frequencies were chosen, euclidean distance was calculated between target face and the rest of face database. Smaller distances tend to yield better results.

Getting Fourier Frequencies

Although fourier frequencies can be easily retrieved by calling single function in MATLAB, preprocessing needs to be experimented. Image sharpening appears to improve result in frequency definition. FFT shift can also center the frequencies for frequency selection in the next part.

Selecting Frequencies

The paper offered two approaches to frequency selection. The variance based method calculates the variance to find frequencies with greatest variance. But also, the paper suggested a lower quadrant method which simply take the lower quadrant of the frequency output.

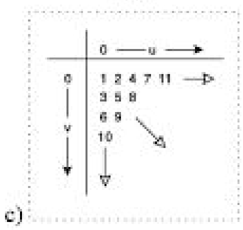

Figure. Lower Quadrant selection from Spies. et al

To achieve this, over ¾ of frequencies were thrown out and the inverse frequencies were calculated to show the remnants of the image.

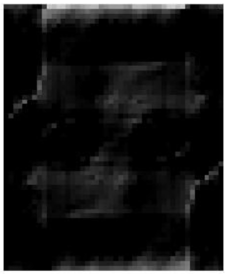

Figure. Lower Quadrant inverse transformed back to image

It appears that only facial features remains and most spatial information is lost after the lower quadrant method.

Euclidean Distance

Once the lower quadrant is calculated for every face in database, a feature vector is built from frequencies selected and compared to other feature vectors built in the same way. The paper did a poor job explaining how this step so it took many trials and error to accomplish this step. To build a single feature from face, consider the illustration below:

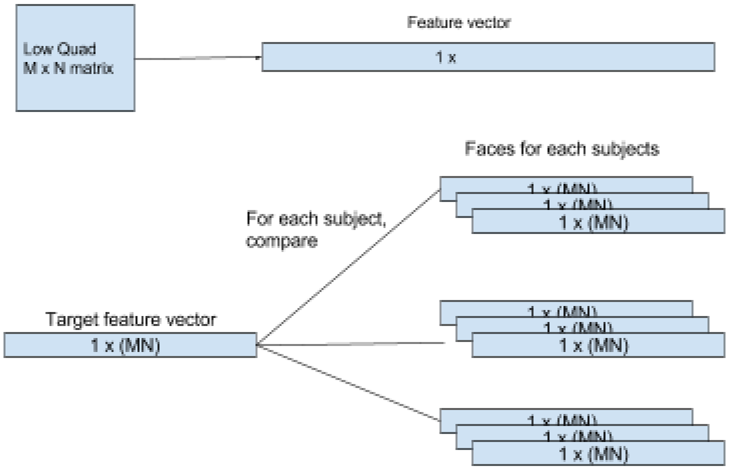

Figure. Illustration for how the comparison is done between target and face database

In MATLAB, a convenience function Pdist2(x,y) can be called to measure the euclidean distance between x and y matrices or vectors. This function is called for each comparison and smallest distance number suggested which subject the target belongs to.

Results

The paper suggested split into 5 train images and 5 test images, matching each testing image to all the available training images with a runtime of $O(N^2)$. In this database, we have 40 subjects with 10 images for each subject. Overall evaluation took 5.6 seconds with Accuracy score of 95%. Experiments with triangulation yields no improvements over the final result.