PRACTICE 1
Pattern Recognition

INTRODUCTION

This first practice aims to get used to the Matlab environment, the database of faces and expressions with which we will work in the practical sessions, and make an introduction to one of the basic methods in pattern recognition: template matching.

We will start with a database composed of images of real subjects doing different expressions and the labels corresponding to which expression is doing each subject (Neutral, Angry, Displeased, Fear, Happiness, Sadness or Surprise). The objective is the classification of new images of test using the Template Matching method. This method is based on the creation of a template (from the database) for each type of expression that we want to classify, and use a comparison metric to compare the new test images that we want to classify with these templates. This metric determines which expression is assigned to the new test images.

To do this, you must implement, with the help of the code provided, creation of templates (average of the intensity of the pixels, extraction of contours / Chamfer distance, coordinates of points of shape...) and comparison metrics (Euclidean distance, cross-correlation...) to compare those methods using cross-validation. For the moment, details as the alignment of the patterns or the scale factor is obviated, so you should focus only on what has been commented before.

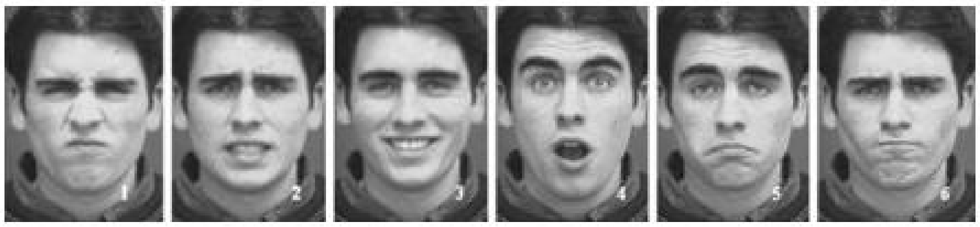

Illustration 1: 6 expressions that we want to classify.

THE CODE

Below we briefly detail the code provided and its functionalities. We recommend that you explore and familiarize yourself with all files:

- p1.m: skeleton on which the practice is constructed.
- createTemplate.m: function to "convert" samples from the database in useful patterns for later comparison. As an example, includes the case 'grayscaleMean', in which it will be considered as the average of the intensity of the pixels.
- classifyWithTemplateMatching.m: method to go through the images in the test set and compare them with the patterns. As an example of measure of similarity it

includes the calculation of the Euclidean distance between the values of the pattern and the original template.

- testMethod.m: support function that calls createTemplate and classifyWithTemplateMatching with methods selected according the passed parameters. Calculates the percentage of accuracy and the confusion matrix.
- extractData.m: support function to read the database and load the images and labels in the corresponding variables.
- drawShape.m: support function to draw the shape lines and the characteristic dots of the face images.
- ListFiles.m: support function to get all images from a directory.

DATABASE

The Face and Expression Database (CKDB) is structured in 8 directories (corresponding to 8 expressions) where we can find two types of files: images in TIFF format for visual patterns, and text files with coordinates the of control points corresponding to the face's shape. To simplify the practice, all images are aligned and trimmed to the same scale factor. To check your results you have 3 instances of the same database. CKDB is the less complex instance since the images contain very exaggerated expressions so it is easier to perform the classification. CKDBHard and CKDBVeryHard are more complex database instances.

WORK TO DO

We ask you to work on the files p1.m, createTemplate.m and classifyWithTemplateMatching.m and extend them in order to make a comparison between different methods to create the templates (createTemplate.m) together with different comparison metrics between templates (classifyWithTemplateMatching.m).

In the file p1.m you can see that one of the first steps is to obtain the indices of different test sets using the command crossvalind: this creates different sets within the number of samples to perform the comparisons using the method of Cross Validation. The objective is to iterate through the different feature extraction methods and metrics that you have defined in the previous files and perform different calls to the function testMethod, which will return the accuracy degree and the confusion matrix in each case to compare them later.

DELIVERY AND EVALUATION

Based on the methods you have defined and with which you have expanded the code provided, you must prepare a report explaining these methods in a clear and concise manner. In addition, you have to compare and comment the results obtained at the level of performance of each combination used for each of the 3 instances of the database. Try to analyze why the different combinations of template creation and comparison metrics work better than others, and to propose new methods that attempt to solve these problems. The memory should contain 4 basic sections: Introduction, Methods used, Qualitative results of the experiments, and Analysis of results and conclusions.

The evaluation of the practice will be given by different factors:

• Number of template creation methods used (2 points)

• Number of comparison metrics used (2 points)

• Quality of memory with justification of the methods used and analysis of results (4 points)

• Quantitative results obtained. The note will be given by the comparison of your results with those of the rest of the companions (2 points)