# Practicum Computational Vision: Face Detection

Tuesday, November 11 , 2014

**Laura Igual**

## Contents

# Abstract

In this session, we work with the concepts involved in the face detection method from Viola & Jones: Integral Images and Haar-like features. You can download from the Campus Virtual all the material you will need for this practicum.

# Integral Images and Haar-like features

## Exercise 1: Haar-like features and Classification

First folder "CV_Face_detection_1" contains a matlab function and two images. Open the file FD_ex1.m and follow the comments to complete the exercise.

### Part 1: Compute and visualize Haar-like features

The code is devoted to compute the two firsts Haar-like features in six windows of the image "barcelona.jpg". We have chosen three windows containing faces and three windows without faces. These Windows are defined by the coordinates (x, y) of the top-left corner, are squared and the length of the sides is "L" pixels. The two Haar-like features to compute correspond to the rectangles described in the Figure 1. The area of these rectangles can be computed by means of the integral image S, computed from the original image I. You have to compute features F1 and F2 over the six windows with and without faces. The values of the features are stored in two matrices, and finally, the features of the considered windows are visualized in a figure. The points are displayed in two different colors to remark the difference, in the feature space, between windows with and without faces. Finally, you can display the windows over the original image in two different colors for windows with and without faces.
Answer the following questions:

- Explain the obtained 2-dimensional plot on the feature space.
- Given this 2-dimensional plot, can we infer the defined Haar-like features are appropriate for face/non-face discrimination?

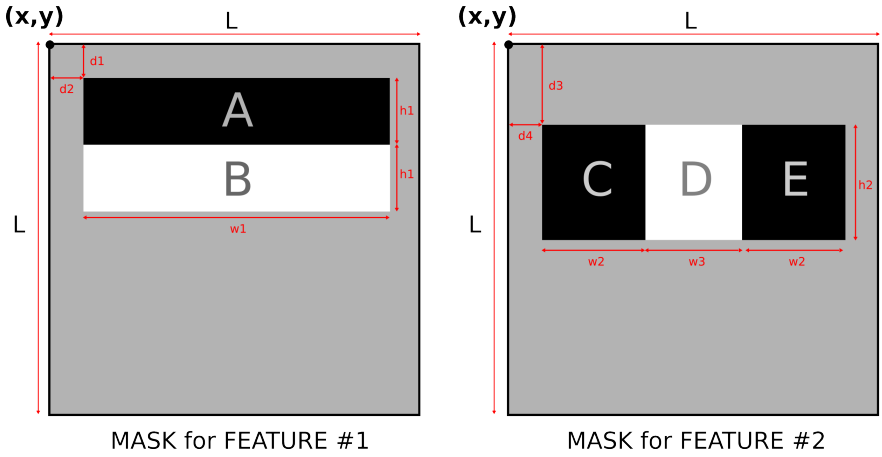


Figure 1: Two masks of Haar-like features.

**Part 2: Classification in the feature space**

The objective of this exercise is to train a k-Nearest Neighbor (NN) classifier using a set of features extracted from the image "barcelona.jpg" and evaluate the classifier in a new image "madrid.jpg".
Consider the 11 windows from image "barcelona" as training set. Define the test set by 22 windows manually selected in the image "madrid.jpg". You can use "ginput" matlab comand to manually capture the positions of these windows. Then concatenate all the faces coordinates in the same matrix and compute the rectangular features for all the windows. Consider the 2 Haar-like features corresponding to the rectangles described in the Figure 1.
In the feature space, 2-dimensional space, use k-NN classifier. For this purpose, the Matlab function "knnclassify" can be used. Display the results of the classification over the test image using two different colors of the rectangles for face / non-face.
Answer the following questions:

- Is the result good enough? Explain your response.

## Exercise 2: Haar-like features and Classification

Repeat the same classification problem as in Exercise 1, but considering the 6 Haar-like features corresponding to the rectangles described in the Figure 2. Create the file FD_ex2.m and write the code.
Answer the following questions:

- Is the result better? Explain your response.

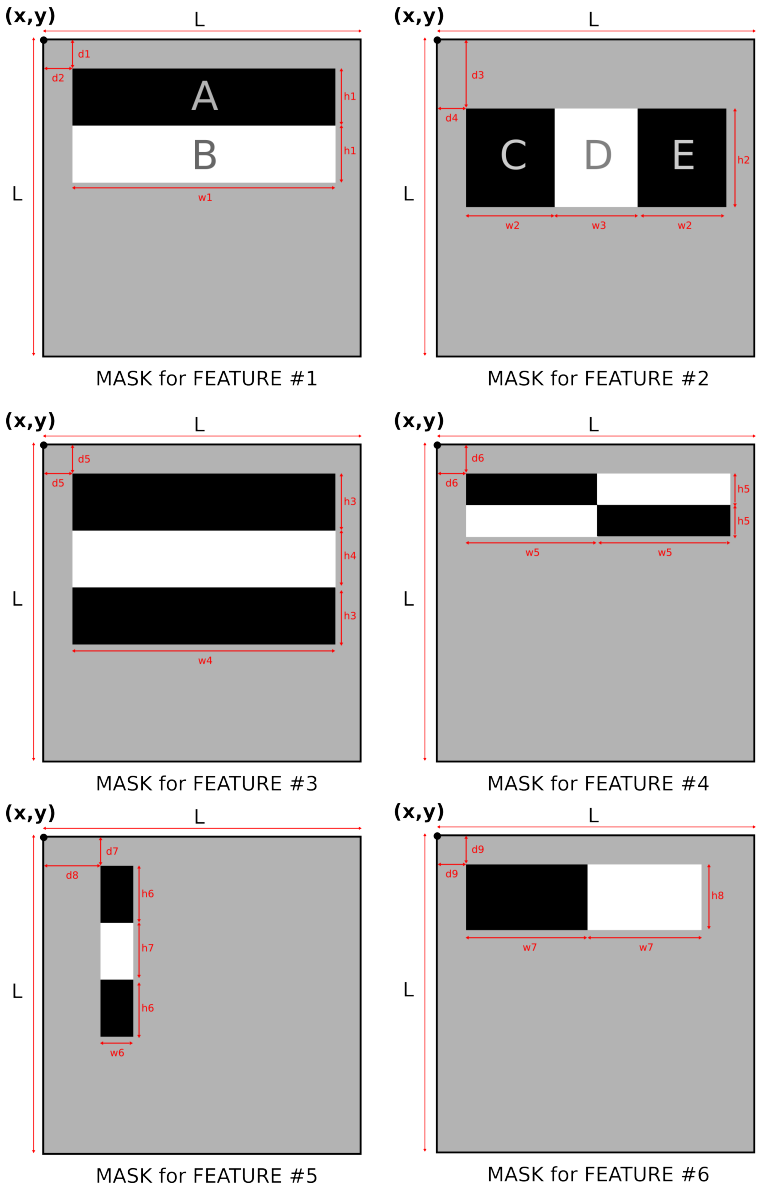


Figure 2: Six masks of Haar-like features.

# Optional: Viola & Jones Evaluation

## Exercise 3: Apply and evaluate Viola & Jones method on a video

Folder "CV_Face_detection_2" contains matlab code and files, and the video "MOVIE.mat". In this video there are three persons which appear and disappear. Open the file FD_ex3.m and follow the comments. The objective of this exercise is to apply Viola & Jones face detection method on the video "MOVIE.mat". From the result of the automatic detection, measure the MEAN DETECTION RATE over the whole sequence of images. The DETECTION RATE is defined by: DR = TP/(TP+FN), where: - TP = True Positives - FN = False Negatives

In order to decide if a detected region is a face or not we provide the "GROUND TRUTH", the set of coordinates from all the faces in the image. These coordinates are stored in the file GROUND_TRUTH.mat, where for every frame of the sequence (stored as cells of a cell-array): - Each row is a face - For every row, there are 4 columns, which define the following parameters: . x = position x of the top-left corner of the window (position 1) . y = position y of the top-left corner of the window (position 2) . w = number of columns of the window (width) (position 3) . h = number of the rows of the window (height) (position 4)

To know if a given detection produces a TP, we have to compare the region defined by the automatic detection with all the regions of the ground truth. And vice versa, to know if we have lost a face (FN), we have to compare the region of the ground truth with all the regions of the automatic detection. To this purpose we provide the function "checkFace", at the end of the file. This function has as input the coordinates of two rectangles, and returns a value "true" if the rectangles have a certain common area (face correctly detected) o "false" otherwise. Once the detection rate (DR) is computed for every frame, the mean can be computed using the Matlab function "mean()".

**Note**:

In folder "CV_Face_detection_2", we provide a file "AUTO_FACES.mat" containing the result of the function FaceDetect. In case you cannot run this function in your operative system you can load and use the matrix in this file.

# Practicum submission

Deadline: 25 of November, 23:55h by Campus Vitual.

The material to submit is a file "StudentName_CV_LabFaceDetection.zip" containing: " A report entitled "Face Detection" including the results of the problems properly commented. The report should include all necessary images to fully understand your discussion. The structure of Sections can be: Title, specification of the problem, discussion of experiments and conclusions. " The files FD_ex1.m, FD_ex2.m and FD_ex3.m of each exercise, where code should be completed.