Universitat Politècnica de Catalunya
Universitat de Barcelona
Universitat Rovira i Virgili

Master in Artificial Intelligence

Computational vision

Face detection

Authors:
Alejandro Suárez Hernández
Johannes Heidecke

November 2016

Contents

List of Figures

1 Exercise 1: Haar-like features and classification

1.1 Compute and visualize Haar-like features

Question 1:

- **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?**

Since we have just two features, each of the data points can be represented in a plane. The scatter plot can be found in figure 1. Moreover in figure 2 we show the square regions from which these features have been extracted. The first observation is that the positive and negative examples are not linearly separable, so this is not a trivial classification task. It turns out that the negative examples are more or less distributed along a vertical line while the positive examples, although somewhat more scattered, follow a horizontal trend. The neighbourhood around the intersection point between the two trends (approximately $(F_1, F_2) = (2.5 \cdot 10^3, 4 \cdot 10^4)$) seems like it is going to be troublesome when we get to classify new data.

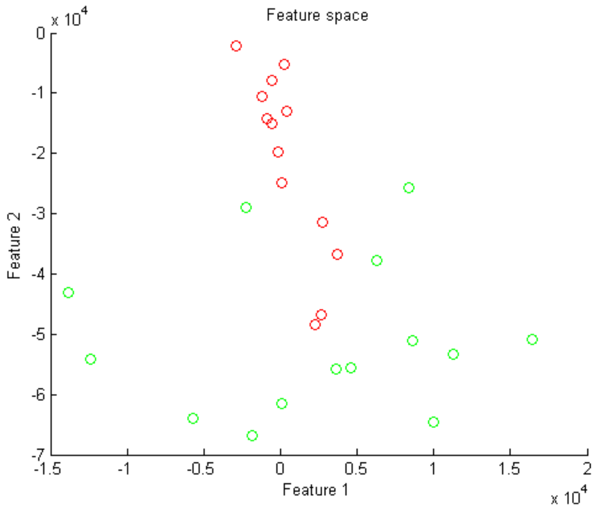

Figure 1: Scatter plot of the features for faces and non-faces

These two features can be useful to discriminate between clear faces and clear foreground. However they cannot be relied upon when it comes to serious face detection and that is the very reason why more sophisticated face detection techniques like the one proposed by Viola & Jones exist.

1.2 Classification in the feature space

Question 2:

- **Is the result good enough? Explain your response**

The highlighted rectangles can be seen in figure 3. Clearly there are some errors. We can see that there are 36 data rectangles and 23 of them have been correctly classified. Therefore, the predictive accuracy in this reduced test set is about 63.9%, which is not a very appealing result.

For the sake of giving a more detailed conclusion, we have analysed the numbers with a bit more of depth. Table 1 shows the number of true/false positives/negatives. From

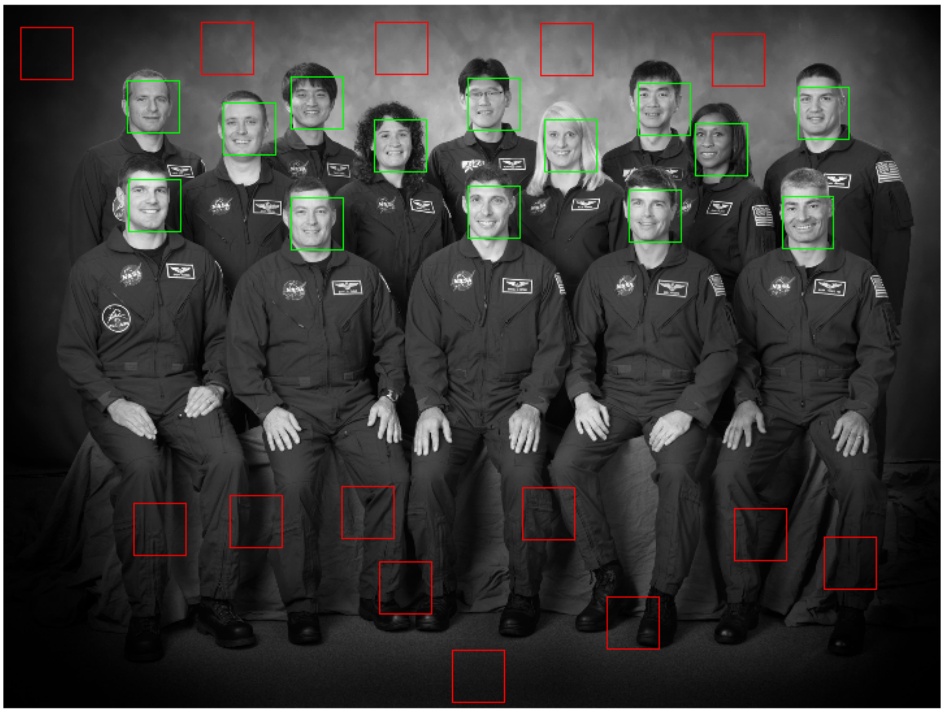

Figure 2: Highlighted faces & non-faces

Figure 3: Highlighted faces & non-faces (test set)

	Detected P	Detected N
Real P	12	11
Real N	1	12

Table 1: Ascertained label vs real one

these numbers we infer that the precision of the classifier in the test set $(TP/(TP+FP))$ is about 92.3%, while the recall $(TP/(TP+FN))$ is 52.2%. In more intuitive terms, this means that if the classifier is biased towards classifying an example as a negative one and that once it classifies a rectangle as a face, the probability that it is effectively a face is very high. We can see this noticing that many faces are misclassified as non-faces, while there is just a single non-face that is classified as a positive example.

In any case, the accuracy of the model is very low to consider it in a serious application.

Question 3:

- **What do you infer from the figure? Explain your response**

The scatter plot is shown in figure 4. Our previous observation is more or less coherent which this new plot (if only the negative examples are more of a round cluster now, instead of a line). The first thing we can observe is that the negative examples are more densely packed than the positive ones, which lie more scattered and are distributed with a greater variance. Thanks to this the classifier has been able to correctly detect as faces the points located far from the "center of mass". On the other hand, we can see that the points that are in the middle of the plot and that are classified as negative examples probably correspond to the badly classified faces. This is just as we have foreseen in the former section of this document.

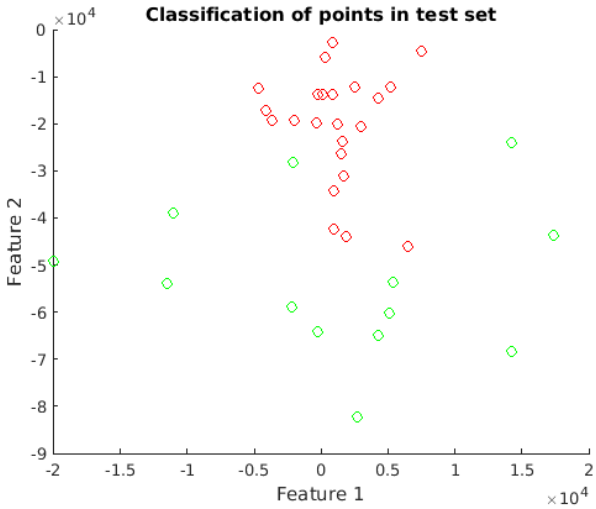

Figure 4: Highlighted faces & non-faces (test set)

2 Exercise 2: Apply and evaluate Viola & Jones method on a video

Question 4:

- **Is the Viola & Jones method detecting faces in the video frames?**

- **When is the Viola & Jones method not able to detect the faces? Explain your response.**

The Viola & Jones method is robust enough to correctly detect the subjects' faces provided they are looking to the front, even when changing from one subject to another or when they open their mouth or adopt contrived gestures. This is the case in figures 5, 6 and 9. However, it fails to detect the faces when the subjects' of the video look to the sides which is understandable given that the model is trained to recognize faces from a frontal perspective. When the method fails, it can identify a wrong section of the image as a face, much like in figure 7, or it can detect no face at all, like in figure 8.

Figure 5: In this case the man is looking to the front and the face is detected correctly.

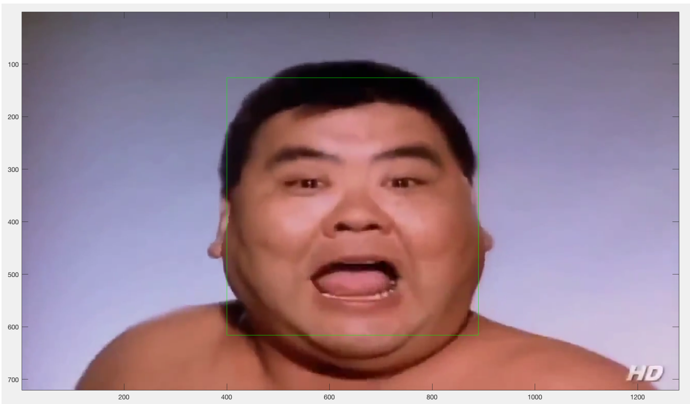

Figure 6: In this case the man is looking to the front with his mouth with open. Even so the Viola & Jones' features are robust enough to correctly distinguish the face.

Figure 7: Frame from the video in which a man appears with his head turned to the side. Notice how his ear is incorrectly detected as a face.

Figure 8: Frame from the video in which the same man appears with his head turned to the other side. In this case, no face is detected at all (which is preferable than an wrongly detected face).

Figure 9: Now the detector is correctly detecting the face of a smiling woman

Appendices

A Annex

Here we list all the delivered script and function files delivered for this practice.

We include relevant observations. For full insight, we refer the reader to the source code.

- `FD_ex1.m`: code for the first section (Haar-like features and classification) of the assignment. The code has been completed to perform the required functionality.
- `FD_ex2.m`: code for the second section (Apply and evaluate Viola & Jones method on a video). Notice that only the 100 first frames of the video are processed as per indication in the comments of the source code (although this can be easily changed). Notice also that we create the `CascadeObjectDetector` specifying a minimum size of [50, 50] (this can be increased even further), because the faces in the video have a rather big size and this contributes to perform the computation more efficiently.