

Artificial Intelligence: HW 2

Jeong Min Lee

October 22, 2023

1 Face Detection

In this section, I will discuss the `face_recognition()` method.

First, let's take a look at the result of `face_recognition()`:

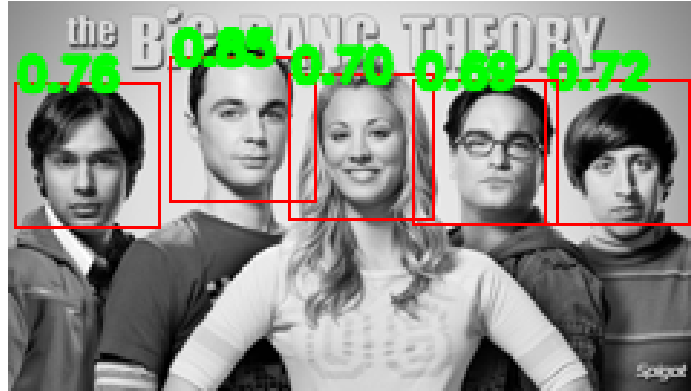


Figure 1: Result of `face_recognition()` applied to the given target and template.

As you can see, `face_recognition()` produces highly accurate face detection results.

Now, let me provide a brief explanation of how `face_recognition()` works. The process consists of two main steps: collecting bounding boxes and applying non-maximum suppression.

We begin by extracting HOG features from patches obtained from both the template and the target. These features are used to calculate the correlation or dot product after subtracting their averages. We save the coordinates of the top-left corner of bounding boxes along with their respective scores, provided their scores exceed 0.65.

Subsequently, we reduce the number of bounding boxes through non-maximum suppression. This is achieved by using the Intersection over Union (IoU) metric to determine whether bounding boxes are neighbors. The algorithm for non-maximum suppression is adapted from the article specified in the instructions. Calculating the IoU was also informed by a different article which proposed the specification. I had to adjust the IoU calculation code to accommodate the requirement that both input boxes have the same size. After applying non-maximum suppression, we obtain the final bounding boxes by removing the first element added for implementation purposes.

I would like to mention that I implemented `face_recognition()` by modularizing the `non_maximum_suppression()` and `getIOU()` functions. However, the TAs requested that I implement `face_recognition()` without introducing new functions. As a result, I embedded the code of these functions directly into `face_recognition()`. This may lead to the redundancy of variables, and some might find the implementation less streamlined.