Comp 408

Assignment 3

Kağan Cenan

Oğuzhan Uz

# Comp408/508 - Assignment 3
# Face detection with a sliding window

The goal of this project is to make a face detection system with using sliding window based on Dalals and Triggers 2005. The following things in this project are:

1. Extract fixed-sized sliding window at each position and scale.
2. Compute HOG (histogram of gradient) features within each window.
3. Score the window with a linear SVM classifier (need to train the classifier first).
4. Perform non-maxima suppression to remove overlapping detections with lower scores.
5. Evaluate performance of the detector on test data.

## Implementation

We first use the get_positive_features.m. It gives us the features for the given set of images which are 36x36. We create a feature matrix which has the size of number of images x feature size. After that we iterate over all images and we get their histogram of gradients. We do this with using vl_hog from vl_feat library. After that, we store all this data into a matrix.

Second step is to get the histogram of gradients for negative features which are not faces. We do this in two steps. First we get an image from the image library. Then we get numerous random samples from each image which are 36x36. We determine the samples with getting a random point on the image which are not close to the boundary of the image. After getting this point, this point becomes the reference point for that sample. We get a 36x36 frame down and right of this point. After that , we get the histogram of gradients for each sample. After that we store all this data to a matrix.

After that we train the SVM with this data from positive features and negative features. We train with labelling the positive data with +1 and labelling the negative data with -1. After that we used the vl_svmtrain to train the SVM.

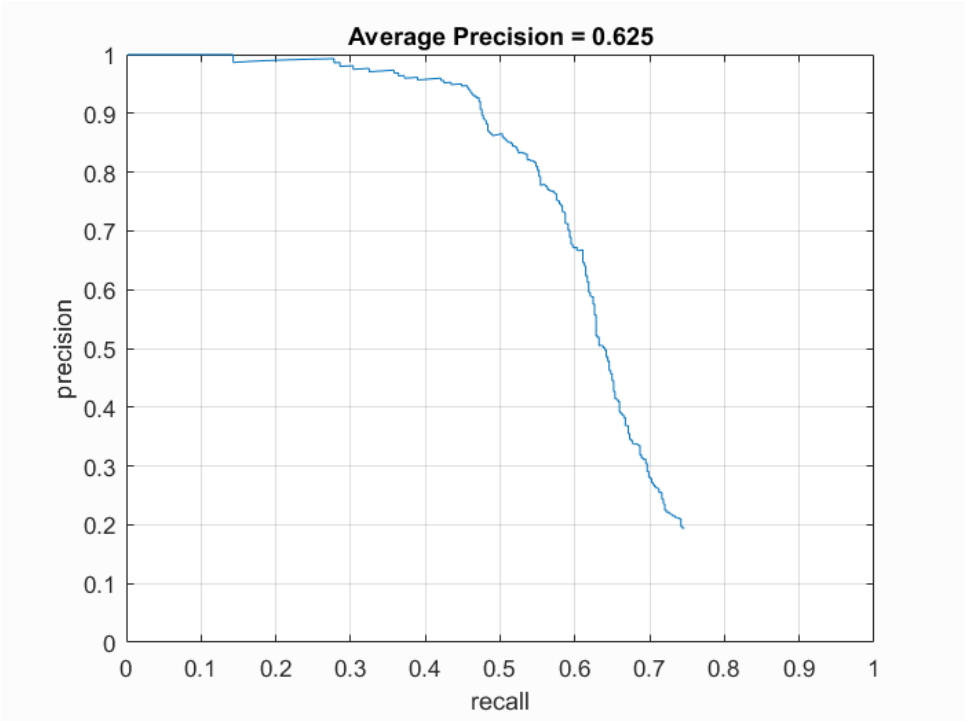
Finally we have written the run_detector.m. We get all images from test_scenes and iterate over them and detect the faces with the trained SVM. We use 8 different scaled windows to try detection of different size of faces. For each image, various sized windows slide over the image and try to find a face. We first take the hog of the image, then the sliding window slides over this hogged image. For each frame, we calculate a score. We found how to calculate this score from the vl library. This scoring was given in vl_trainsvm:

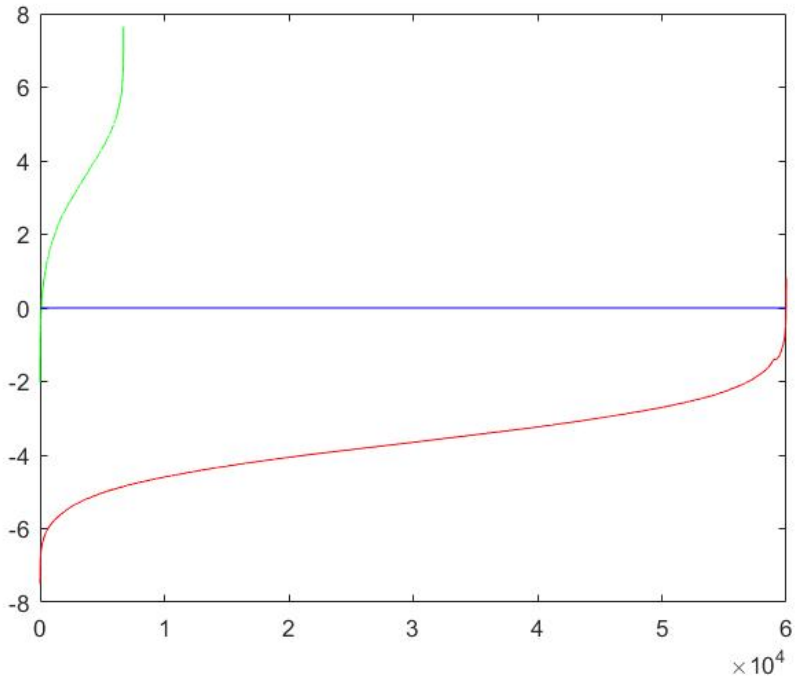
score W'*X(:,i)+B

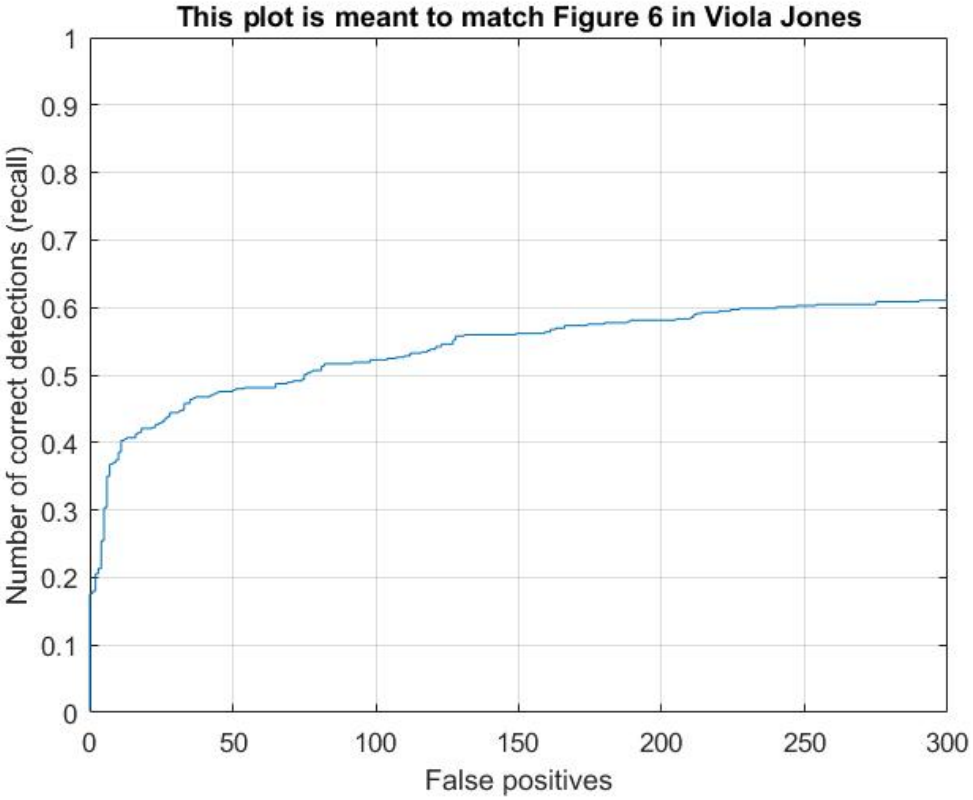
We get this score for each of the frames for each of the sizes for each of the images. W is the weight vector. This corresponds to the w which we got from training the SVM. B is the offset which we got from b with training the SVM. After this, we check whether the score of that frame is greater than the threshold or not. If its greater than that, we add the sized coordinates of the frame to the cur_boxes. And we also add the score to the confidences with using the vertical function of matlab. We hold a count whether any frame has a greater score than the threshold. If count is 1, it iterates to next image. If the count is not 1, we do non max suppression and thus we get the detected frame.

We got a bunch of design parameters in this assignment. First of all we got the lambda parameter for the SVM training. We tried bunch of values and we found out that 0.001 is the best for the results.

## Example Results and Graphs


Average Precision = 0.625
precision
recall
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


8
6
4
2
0
-2
-4
-6
-8
0
1
2
3
4
5
6
×10^4


This plot is meant to match Figure 6 in Viola Jones
Number of correct detections (recall)
False positives
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
50
100
150
200
250
300

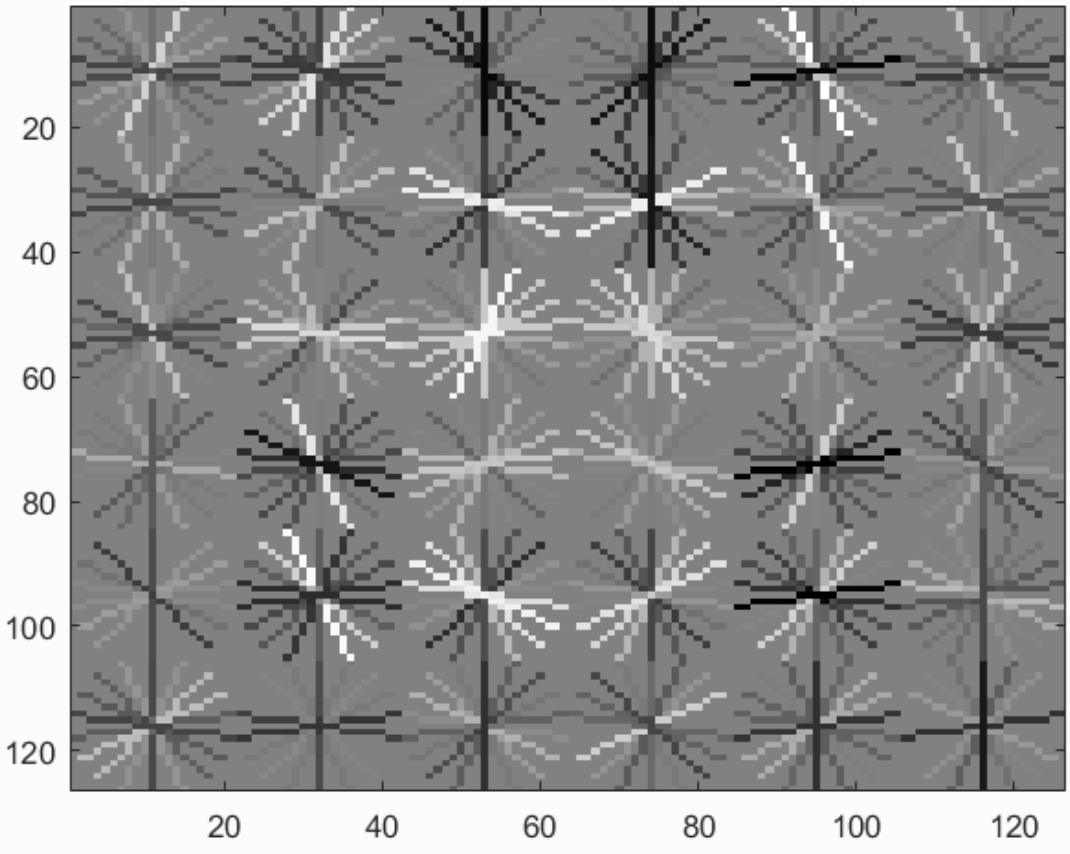

20
40
60
80
100
120
20
40
60
80
100
120

image: "kaari1.jpg" (green=true pos, red=false pos, yellow=ground truth), 1/1 found

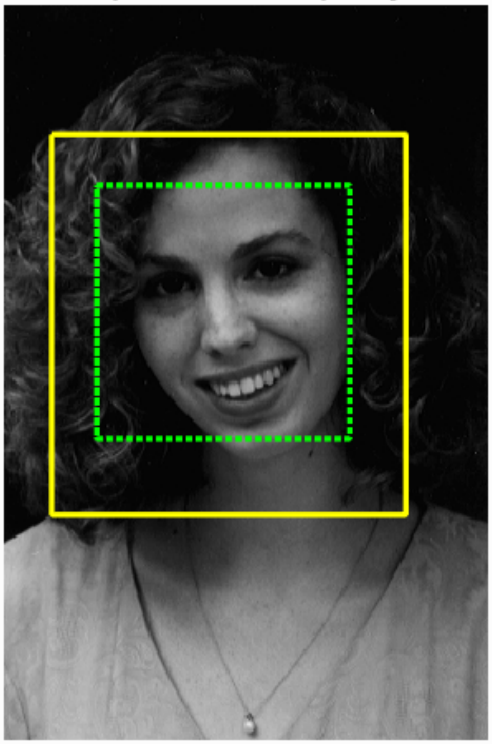
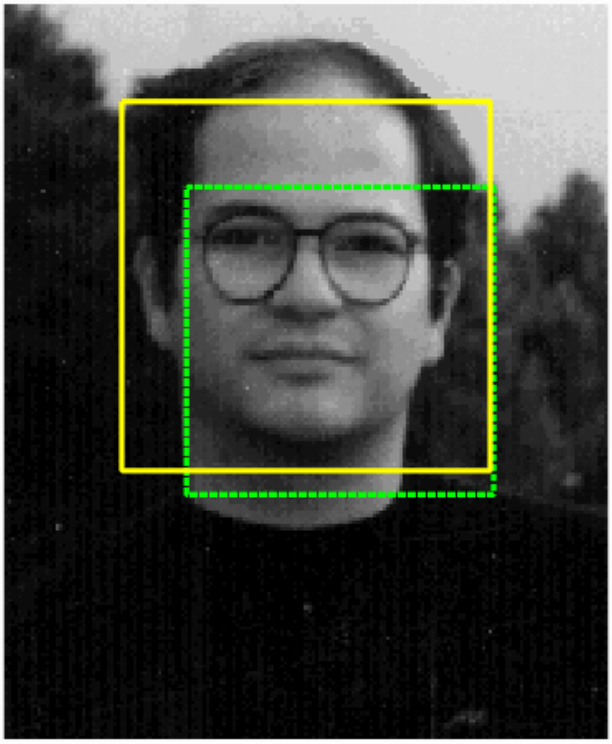


image: "cfb.jpg" (green=true pos, red=false pos, yellow=ground truth), 1/1 found



nage: "aerosmith-double.jpg" (green=true pos, red=false pos, yellow=ground truth), 5/5 four

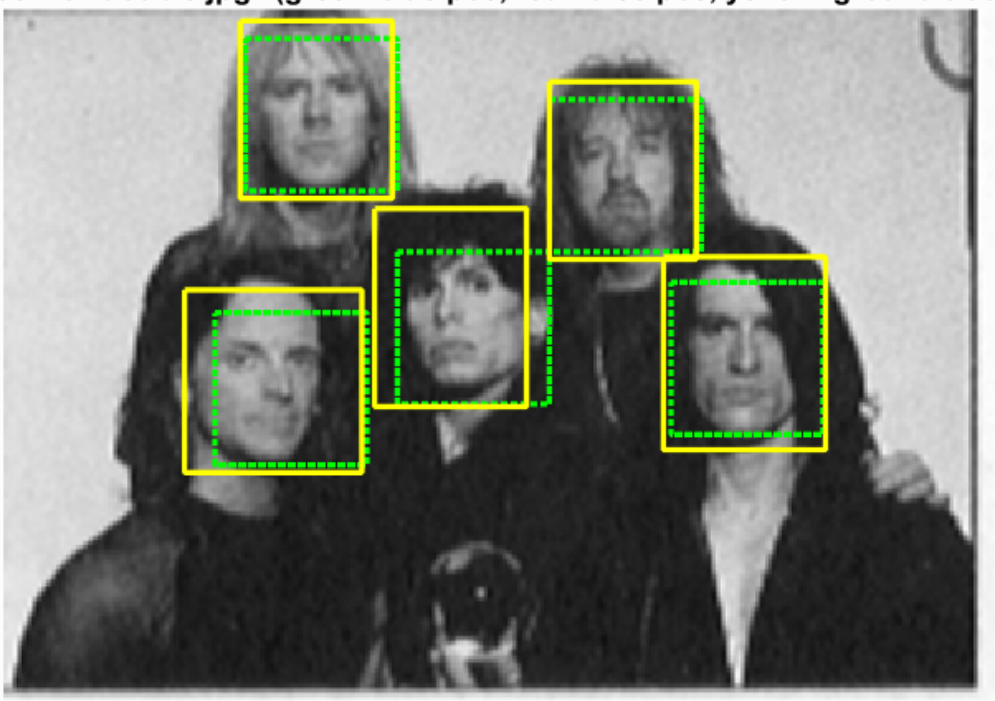


image: "Brazil.jpg" (green=true pos, red=false pos, yellow=ground truth), 8/11 found

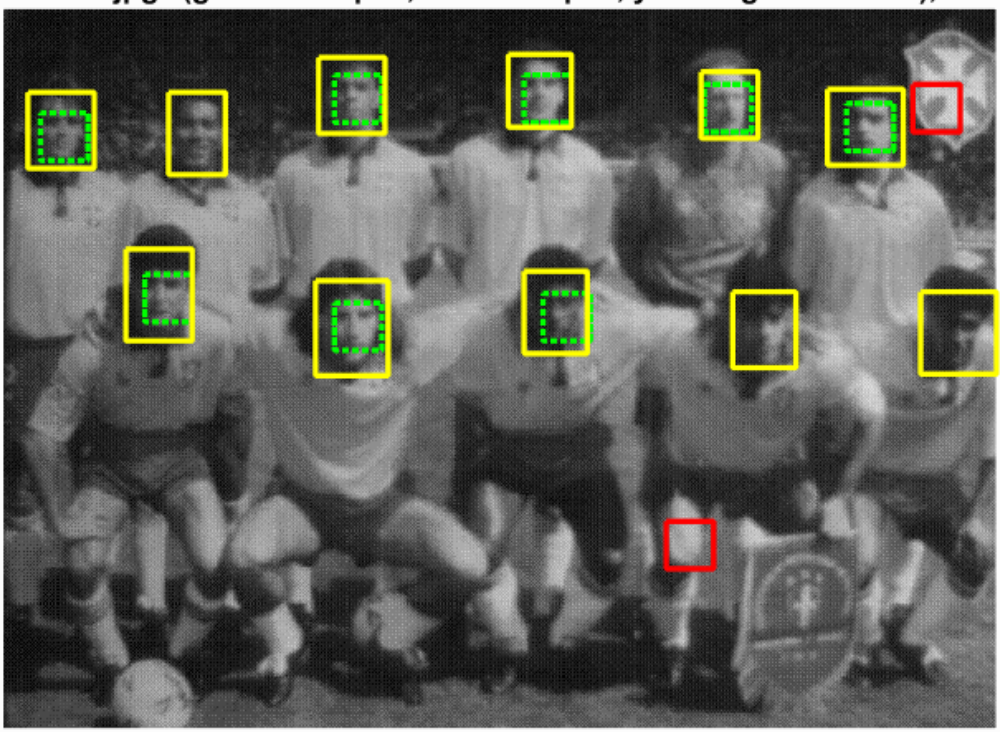


image: "Argentina.jpg" (green=true pos, red=false pos, yellow=ground truth), 9/11 found

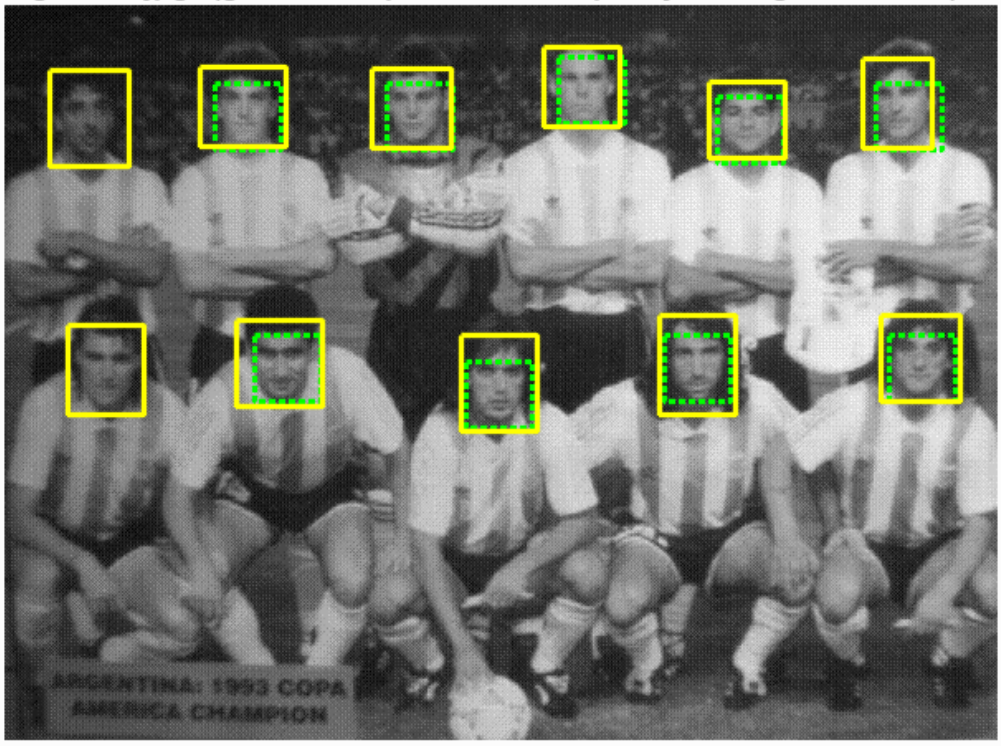



image: "knex37.jpg" (green=true pos, red=false pos, yellow=ground truth), 1/1 found