Computer Vision
Assignment - 4
Face Detection by HOG features

Abdullah Atakan Guney
2018700069

December 22, 2018

Contents

1 Description

In this assignment, I have implemented a face detection algorithm which is based on Histogram of Oriented Gradients which is defined in (Dalal and Triggs 2005).

2 HOG calculation

2.1 Computing Gradients of the image

To compute gradients of image, I have used 1-D kernel $[-1, 0, 1]$ which is proposed (Dalal and Triggs 2005, here) after Gaussian smoothing the image.

$$G_x = [-1, 0, 1] \circledast I$$
$$G_y = [-1, 0, 1]^T \circledast I$$

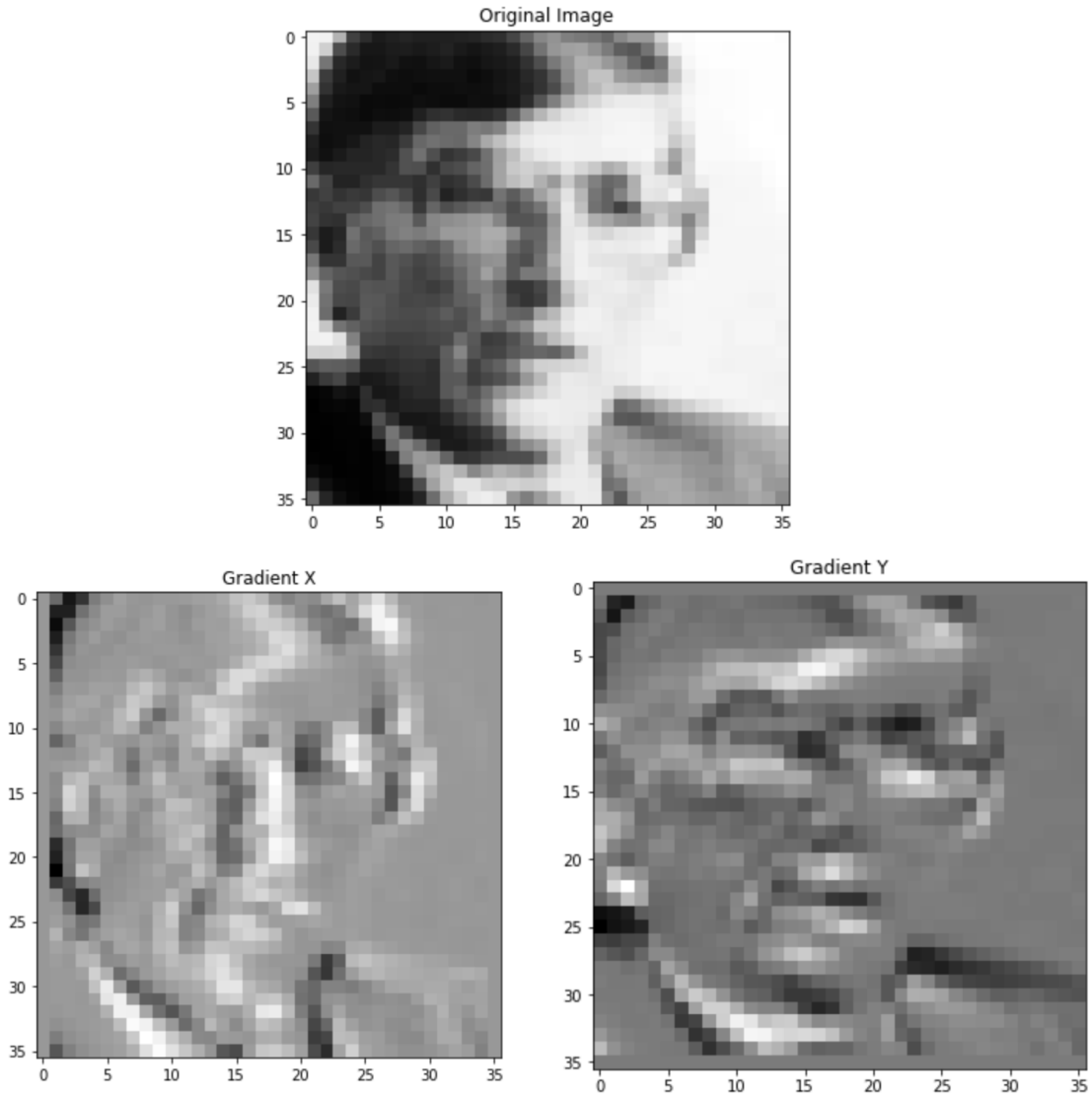

Figure 1: Gradients

2.2 Computing Magnitudes and Angles of Gradients

I have first calculated magnitudes by using gradients in x-direction and y-direction, and I have also calculated direction by calculating arctan $\frac{gy}{gx}$.

2.3 Calculation of Weighted Vote into Spatial and Orientation Cells

While calculating orientations of cells, I followed the steps described on the website(MALLICK n.d.). I basically calculated orientations around unsigned angles, which are between 0-180. I used 9 bins for 0, 20, 40, ..., 160. For each gradient, it contributes to bins according to its distance. For example, if the angle is 10 and magnitude of the gradient is 50, this gradient

contributes to bin-0 by 25, and bin-20 by 25.

2.4 Normalizing Bin Vectors Over Overlapping Block

After calculating orientations of gradients for each cell, I have normalized these orientation vectors by grouping as blocks. So, let say we have chosen cell shape as (6, 6) and block shape as (12, 12), for each cell we have a vector, which is 9 element-length, and each block will have 4 cells, by concatenating its cells vectors, each block is represented by 36-length vectors. In this part, I have normalized these vectors, it is again explained in detail in here(Dalal and Triggs 2005).

2.5 Extracting HOG from Image

By applying above procedures iteratively, I have calculated HOG features for an image by given block shape, block stride, and cell shape. I have assumed block shape and block stride are multiple of cell shape.

3 Linear SVM Classifier

By using hog script I have implemented, I have calculated hog features for given face images, and I have also cropped random window which is the same shape with face images, and created negative samples. By using face images as positive samples and random crops as negative samples, I have trained a linear SVM classifier. While training, I have also tested this classifier by using some portion of labeled data. So, there were:

- 18675 **Negative Samples**
- 6713 **Positive Samples**

I have used 20% of this data set as test data for my classifier, after tuning some parameter, I have combined 2 sets and trained the classifier again. Here are some results on test data.

- Precision: [0.99079838 0.97975416]
- Recall: [0.99240987 0.97552196]
- F-Score: [0.99160347 0.97763348]
- Support: [3689 1389]
- Accuracy: 0.98779046868846

For window shape (48, 48), window stride (48, 48) and block shape (16, 16) and block stride, cell shape (8, 8)

- Precision: [0.99184117 0.98329537]

- Recall: [0.99400382 0.97735849]
- F-Score: [0.99292132 0.98031794]
- Support: [3669 1325]
- Accuracy: 0.9895875050060072

For window shape (72, 72), window stride (48, 48) and block shape (24, 24) and block stride, cell shape (12, 12)

- Precision: [0.99259716 0.99392559]
- Recall: [0.99752015 0.9819955]
- F-Score: [0.99505257 0.98792453]
- Support: [3226 1333]
- Accuracy: 0.9929809168677342

4 Face Detection

4.1 Pyramids

In this part, I have resized image in different scales and used sliding window approach to detect faces. First, I have calculated pyramid scales for an image by given downscale factor, than by sliding window method I have tried to detect faces through windows.

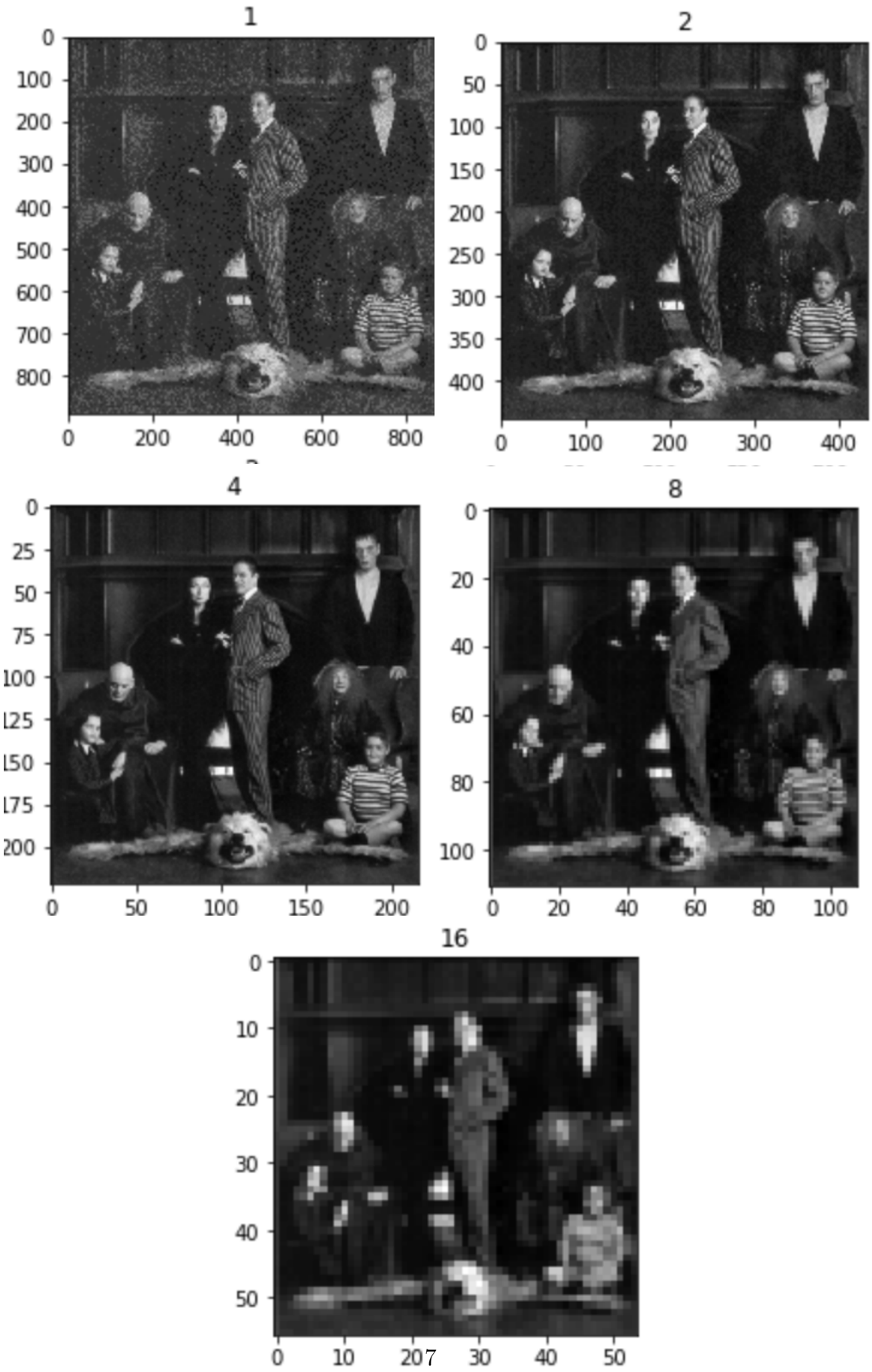

Figure 2: Pyramids

4.2 Sliding Window

By given window shape and window stride, by sliding window, I have calculated hog features for each window and by using classifier, I have decided to that the window is face or not. I have followed the idea from the post of (Rosebrock n.d.[a], see)

4.3 Non-Maximum Suppression

I have taken this part from (Rosebrock n.d.[b], see). But I have changed a little bit of this code. In the blog post, author sorts candidate bounding boxes according to their bottom-right points, but I have used scores that are calculated by classifier to sort bounding boxes which resulted better.

5 Evaluation

5.1 Mean Intersection over Union

Intersection over union is a metric that is used to evaluate object detector, which is based on comparing overlapping area of detectors bounding box and grand truth bounding box.

$$\text{Mean IuO} = \frac{\text{Overlapping Area}}{\text{Union of 2 Bounding Boxes}}$$

5.2 Average Precision

I have calculated precision as follows:

- Calculate IoU between any 2 bounding boxes from grand truth bounding boxes set, and detectors bounding boxes set.
- If this IoU is greater that the some threshold, I have considered this is a true positive.
- If the IoU is greater than 0 but, less than threshold, it is considered as false positive
- By calculating TP, and FP, average precision will be $\frac{TP}{TP+FP}$

To calculate mean Average Precision, calculate average precision for each image and take the mean. I followed the idea from this blog post(Hui n.d., see).

6 Visualization

To visualize results, I have plot grand truth bounding boxes as red and my detectors ones as lime.

7 Results

7.1 Results of My Pictures

Here is the results of my pictures with/without non maximum suppression. As I have observed, my camera took these images with higher width and height. To detect my face in reasonable time, I have set window shape for image 1 (600, 600), for image 2 (450, 450), and for image 3 (300, 300). I have set block shape, block stride and cell shape according to that because my classifier needs to have 900 feature vectors for this experiment. As you can observe for first and second images, this detector worked quite well but for the last, it could not detect my face easily. I can conclude that this detector is not robust to rotations.

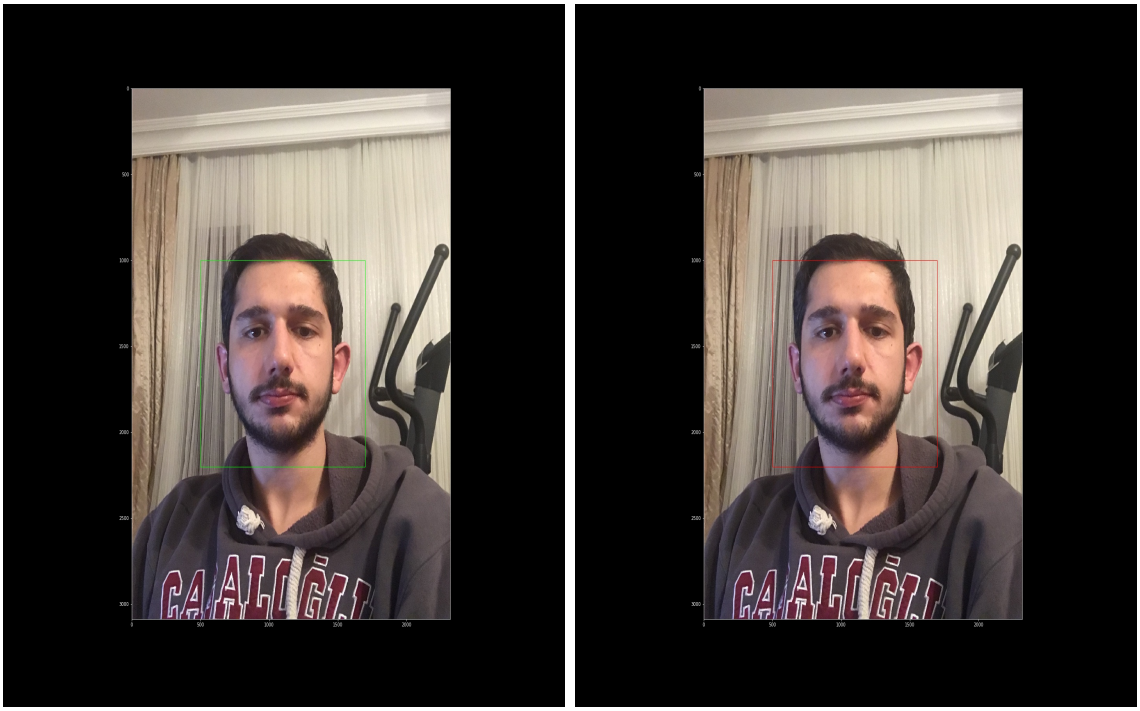

Figure 3: Atakan - 1

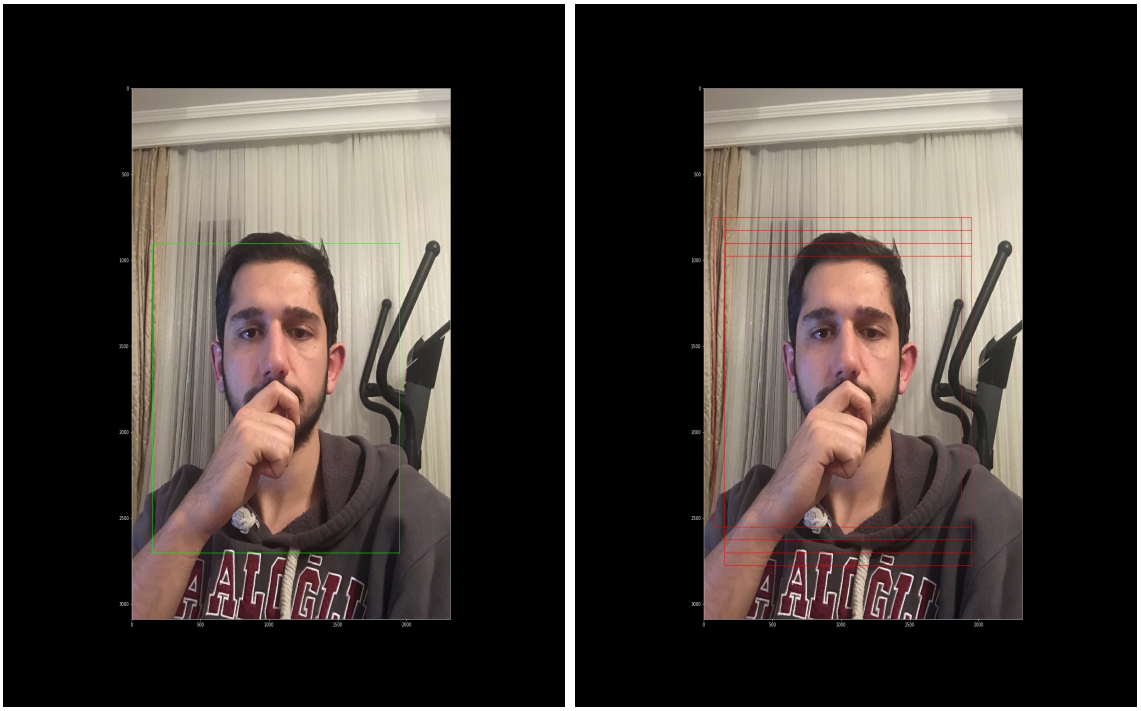

Figure 4: Atakan - 2

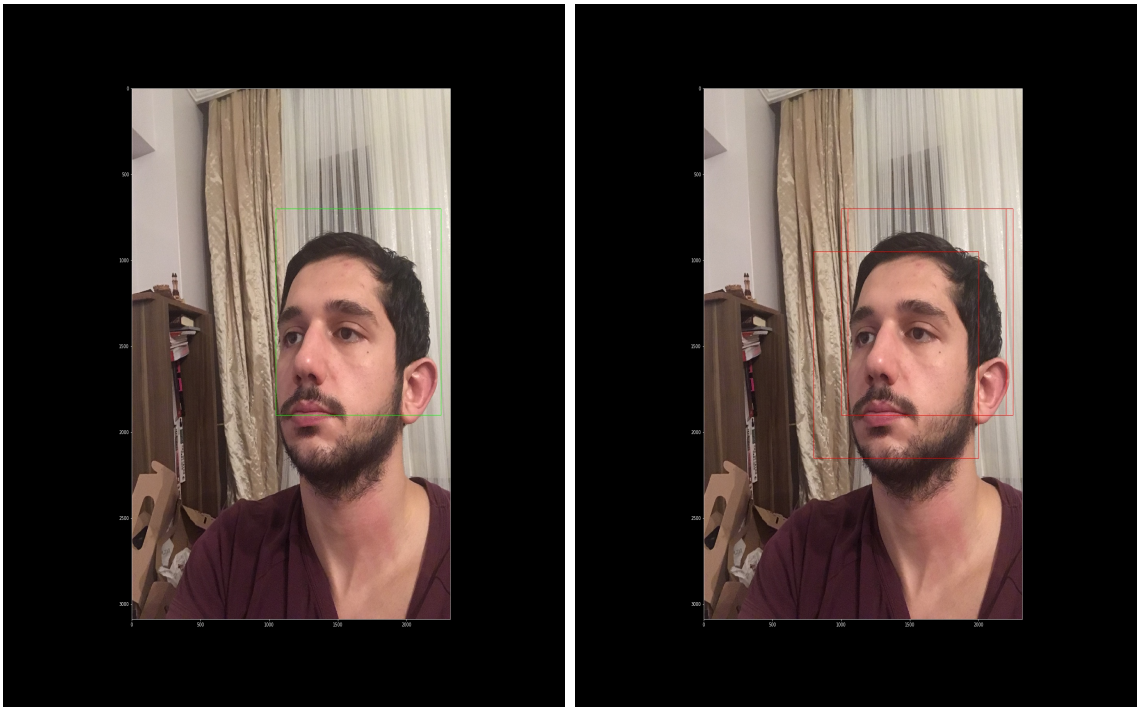

Figure 5: Atakan - 3

7.2 Results for Validation Set

7.3 Classifier - 1

First classifier trained with random crops, cropped with

- window shape (72, 72)
- window stride (48, 48)
- block shape (12, 12)
- block stride (6, 6)
- cell shape (6, 6)

Here are some results:

Figure 6: Classifier 1 - Results

Figure 7: Classifier 1 - Results

Average Precision and Mean Intersection over Union for this classifier as follows:

- Mean IoU: 0.1980598357735915
- Average Precision: 0.5814236216379877

7.4 Classifier - 2

In this classifier, in training step I have used following settings:

- window shape (36, 36)
- window stride (48, 48)
- block shape (12, 12)
- block stride (6, 6)
- cell shape (6, 6)

Here are some results obtained by this classifier on validation data set by using:

- window shape (36, 36)
- window stride (6, 6)
- block shape (12, 12)
- block stride (6, 6)
- cell shape (6, 6)

Figure 8: Classifier 2 - Results 1

Figure 9: Classifier 2 - Results 1

Average Precision and Mean Intersection over Union for this classifier as follows:

- Mean IoU: 0.23953190120474918
- Average Precision: 0.7069559970195768

I have also tried to detect faces by using this classifier with another parameters settings on validation set:

- window shape (72, 72)

- window stride (8, 8)
- block shape (24, 24)
- block stride (12, 12)
- cell shape (12, 12)

Figure 10: Classifier 2 - Results 2

Figure 11: Classifier 2 - Results 2

Average Precision and Mean Intersection over Union for this classifier as follows:

- Mean IoU: 0.16795356548306106
- Average Precision: 0.5420791287365101

References

Dalal and Triggs (2005). "Navneet Dalal, Bill Triggs. Histograms of Oriented Gradients for Human Detection. Cordelia Schmid and Stefano Soatto and Carlo Tomasi. International Conference on Computer Vision". In: *IEEE Computer Society*. DOI: 10.1109/CVPR.2005.177.

Hui, Jonathan. URL: https://medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173.

MALLICK, SATYA. *Histogram of Oriented Gradients*. URL: https://www.learnopencv.com/histogram-of-oriented-gradients/.

Rosebrock, Adrian. URL: https://www.pyimagesearch.com/2015/03/23/sliding-windows-for-object-detection-with-python-and-opencv/.

— URL: https://www.pyimagesearch.com/2015/02/16/faster-non-maximum-suppression-python/.