Histogram Of Oriented Gradients for Pedestrian Detection

Sourabh Kumar

04/12/2018

1 Abstract

This is a Project report on implementing one of the application of Histogram of Oriented Gradients Features. The entire code and steps to run can be found at
https://github.com/sourabhkumar0308/DigitalImageProcessing
The platform used is Linux(Ubuntu 17.04). The author has discussed about the extraction of a feature vector which can be used for pedestrian detection. I have used skimage's implementation of the hog descriptor to implement the detection of people across multiple scales of an image. The author has suggested Mean-shift to remove the bounding boxes, but I have used non-maxima suppression to eliminate multiple bounding boxes of detections having maximum overlap.

2 Feature Extraction and Training

Since the dataset was corrupted, I had to lookup for some alternate dataset and merge that with the original recovered one.

1. 1994 Initial Negative Set (997 from cornell[1] dataset + 997 mirror images)
2. 6000 Initial Positive Set (4000 from cornell[2] dataset(including mirror images) + 2000 from INRIA DATASET(including mirror images))
3. With the above dataset an initial linear SVM classifier was trained.
4. Using the initial trained model the 1218 negative training images from INRIA dataset are searched exhaustively using a sliding window for false positives(hard examples).
5. The final detector is then retrained using the augmented set (1994 + 6000 + hard examples)

3 Testing the Classifier

For testing the classifier 958 Positive Images from INRIA test dataset(including mirror images) and 4553 Negative Images extracted from INRIA test dataset.

Then the set of images are fed to the SVM classifier to predict whether the image is of a pedestrian or not.

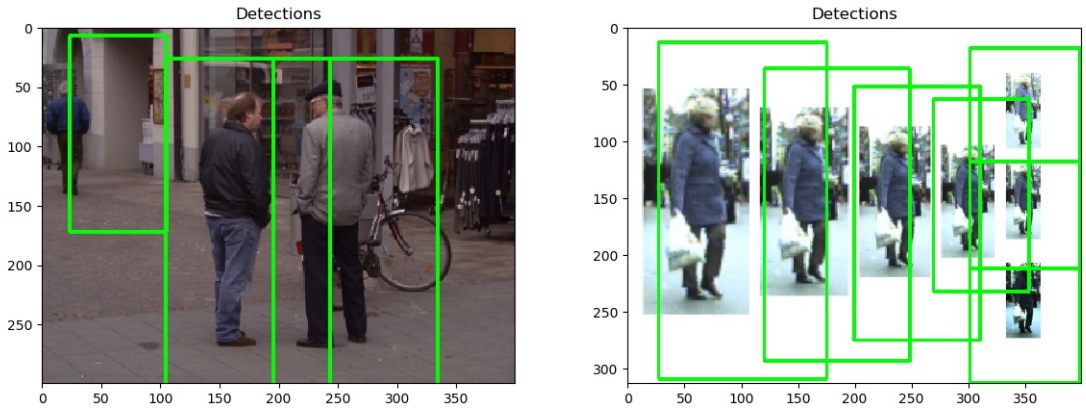
4 Detections

The detections are performed by creating a gaussian pyramid so that detections at different scales can be performed. At each scale, a sliding window approach is used to perform detections. The detections are controlled by parameters winStride, downscale. For most of the images, a winStride of (8,8) and downscale factor of 1.20 worked fine.

5 Observations

If the people are very near to each other or person is at the edge of image then the detector fails to isolate people as it needs a sufficient padding around the person. Also it is not able to detect partially or fully occluded people. The author have compared their implementation for pedestrian detections with the already available methods based on a Detection Error Tradeoff(DET) curve. But he has also suggested that ROC curve can also be used to measure performance of the detector. So I have used that.

6 Results



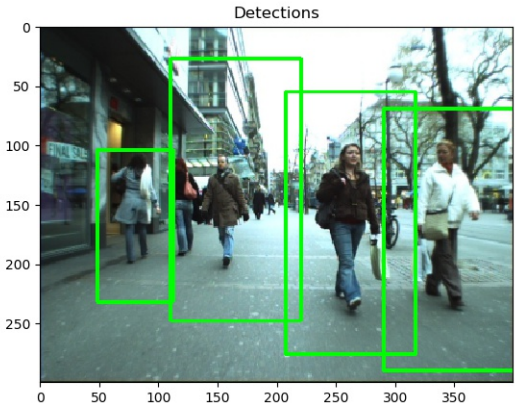
(a) (b)

Figure 1: Detection Results

(a) (b)

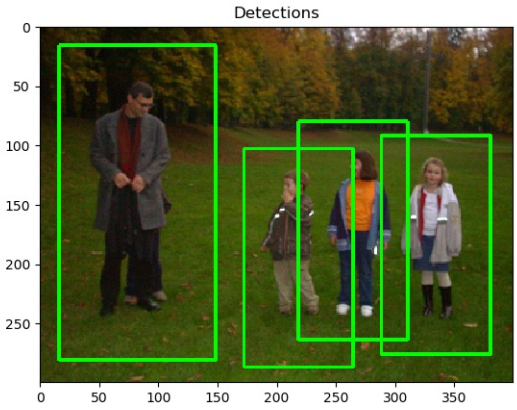
(c) (d)

Figure 2: Detection Results

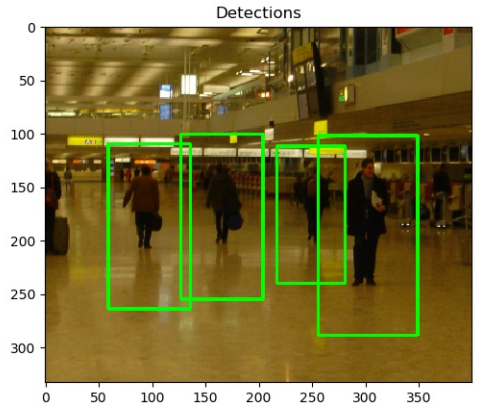


(a)

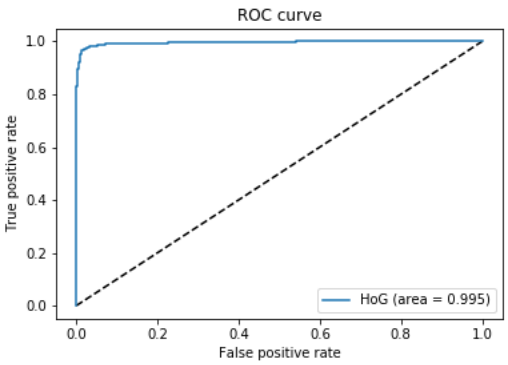
(b)



(c)



(d)



(e)

Average precision-recall score: 0.99

Classification Report

	precision	recall	f1-score	support
0	1.00	0.97	0.98	4553
1	0.86	0.98	0.91	958
micro avg	0.97	0.97	0.97	5511
macro avg	0.93	0.97	0.95	5511
weighted avg	0.97	0.97	0.97	5511

Accuracy:0.9682453275267646

Confusion Matrix

prediction / label	0	1
0	4396	157
1	18	940

(f)

Figure 3: Detection Results

7 References

1. https://www.pyimagesearch.com
2. http://www.cs.cornell.edu/courses/cs4670/2013fa/projects/p5/index.html[1][2]
3. https://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf