COL780
Assignment 3

Pratik Prawar(2018CS50415)
Aditya Mohan Mishra(2018ME10582)

Part 1:- Pretrained HOG:

OpenCV has a default people detector, which is based on HOGs of the images.(SVM trained on the HOG features).

Implementation

A simple script is written to read the images, then use the HOGDescriptor() with the pretrained svm HOGDescriptor_getDefaultPeopleDetector(). Then the rectangles are converted from x,y,w,h form to x,y,x+w,y+h form in order to use imutils' Non-maximal suppression function, to reduce overlapping images.

Finally we normalize the weights by Laplace smoothening(dividing by their sum along with a small term added to both numerator and denominator in order to avoid divide by zero errors,)

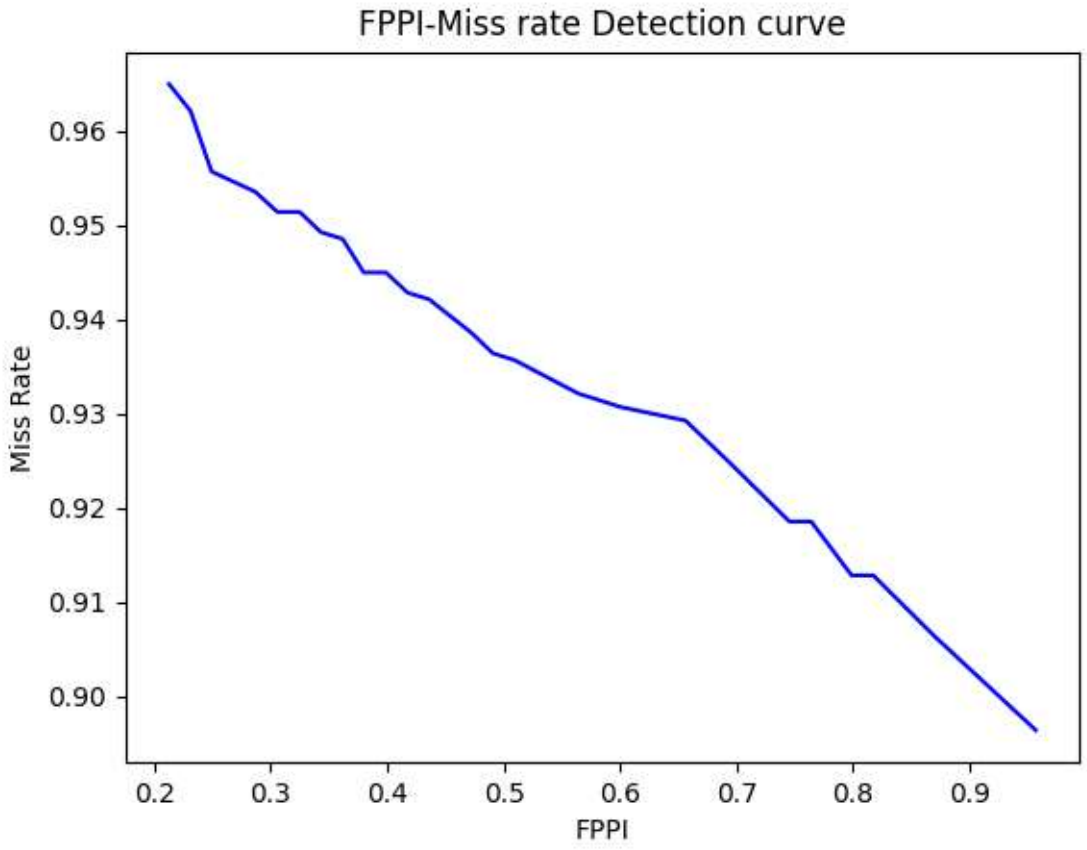
It was found by observation that the bounding boxes were a little too large for large detections, hence we scaled down the box with the same ratio on both dimensions while centered at the same place. The factor for larger boxes were of 0.8x the original on both the dimensions and for smaller we kept them the same.

Results:

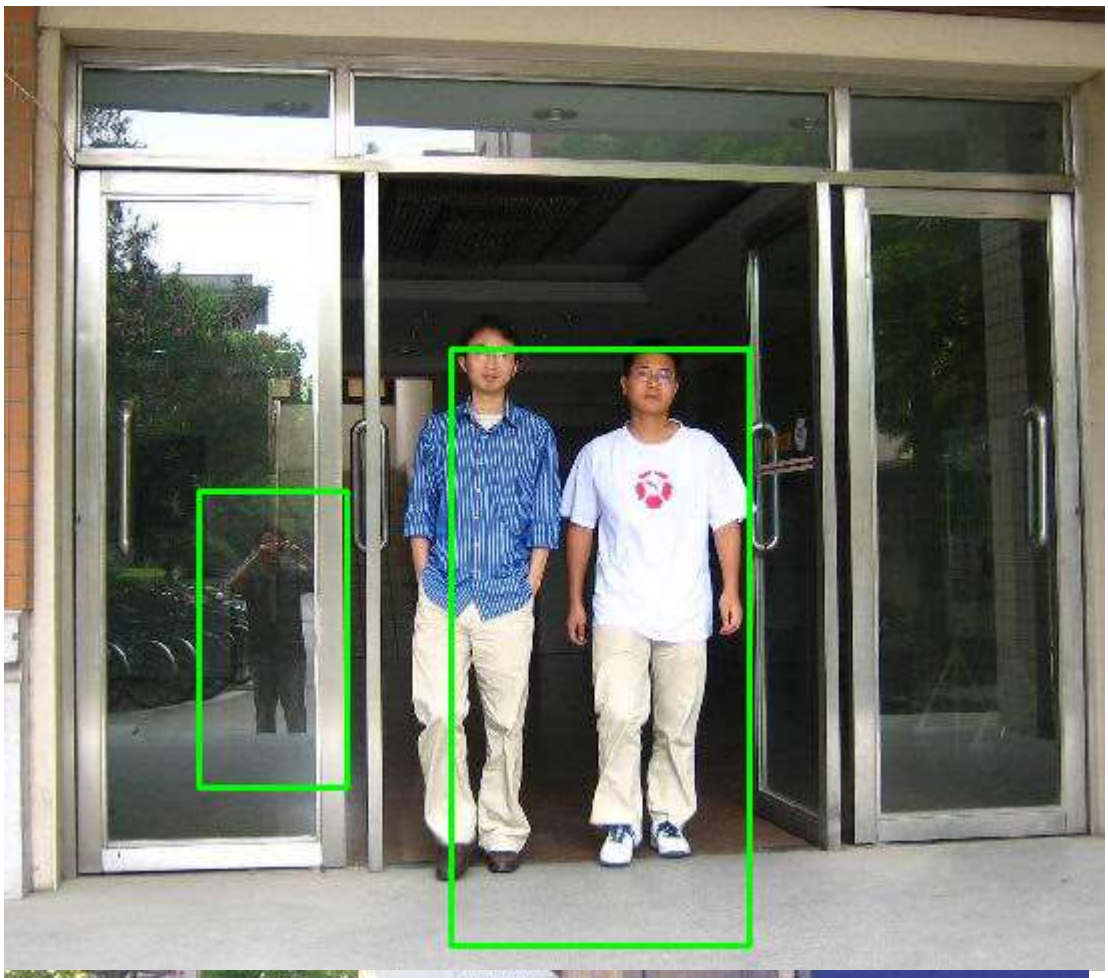
AP and AR

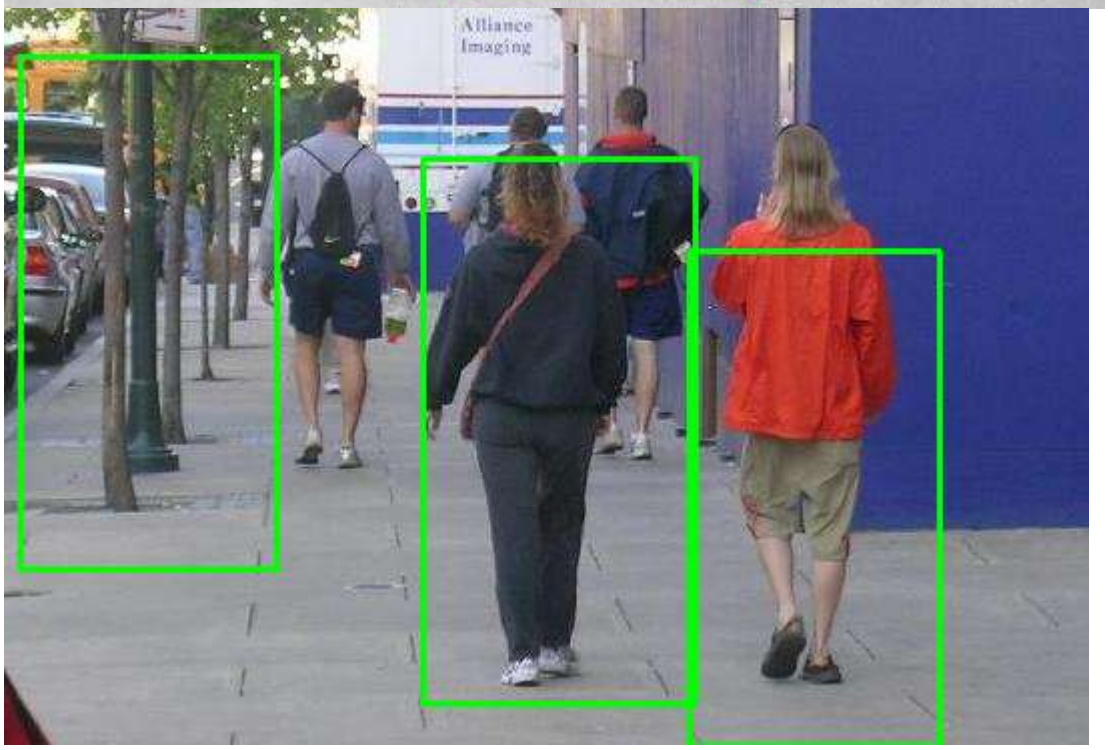
```
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.130
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.424
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.033
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.000
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.003
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.148
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.106
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.200
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.200
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.007
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.227

Average Precision =  0.12993365075567653
Average Recall @ 1 detection per image =  0.10571428571428572
Average Recall @ 10 detections per image =  0.2
```

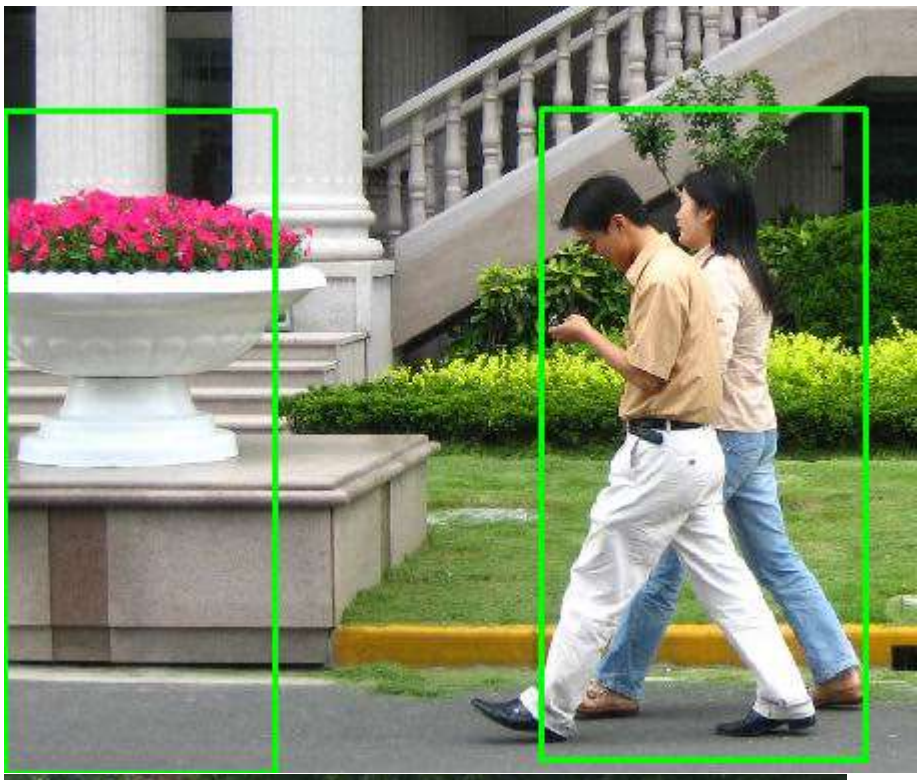
MR vs FPPI

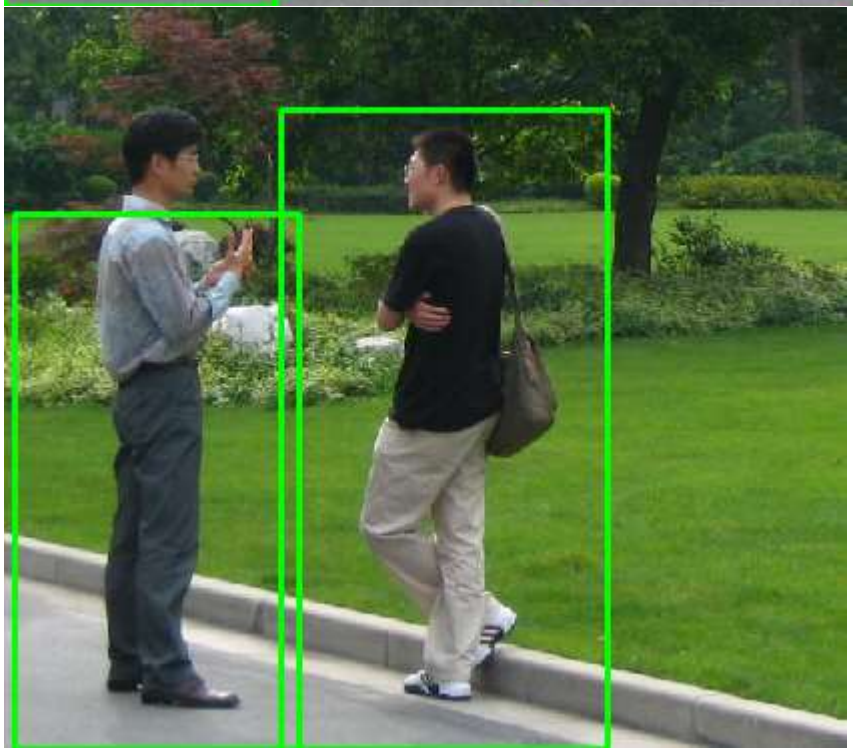


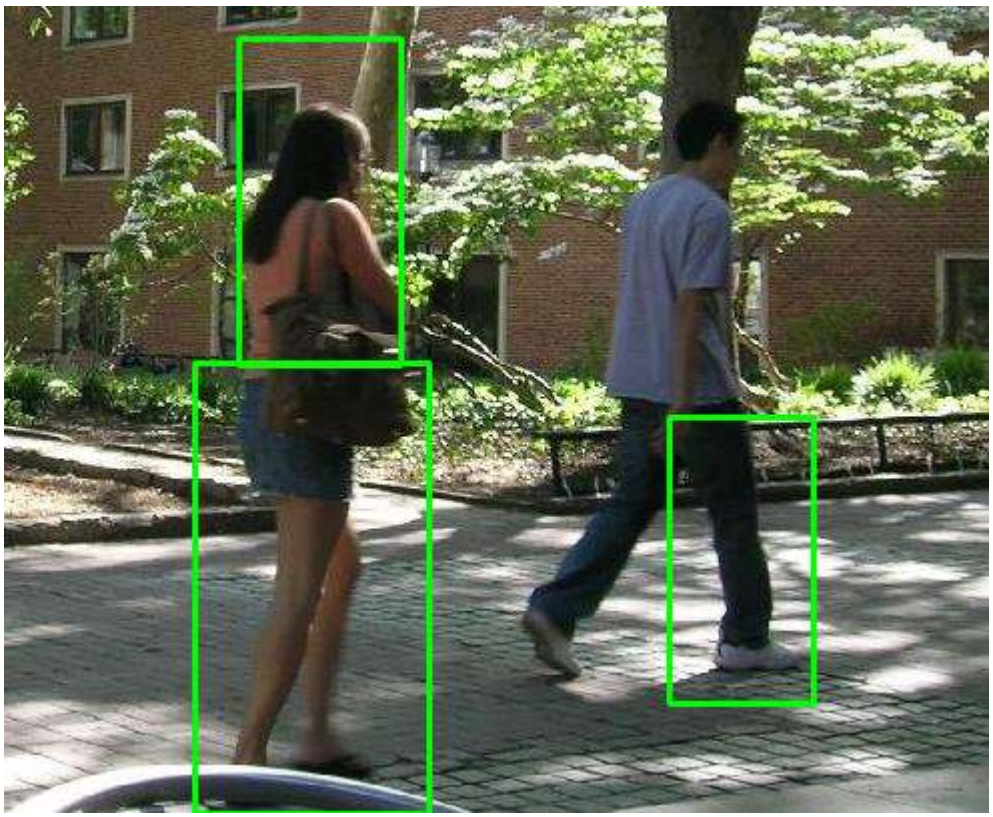
Some Samples:




Alliance
Imaging







Inferences:

The pre-trained model misclassifies many objects with similar aspect ratios as humans.
Many smaller and occluded humans also found but we were supposed to ignore them according to the dataset.

2)Custom Trained SVM Using HOG features

Implementation

So to train the svm we generated negative samples via manual annotation, and then converted it to a similar dataset as given to us. We annotated over 800 boxes for the whole dataset. The scripts that

we created to make the process human friendly are generate_negative_samples.py and process.py.

The whole idea is to take the hog features of positive and negative samples and then train a SVM classifier to differentiate between them.

This had one problem. The SVM classifier needs the input feature to be of constant dimension but it wasn't possible if we directly converted the incoming image to a feature vector. So we explored two solutions. First was to resize the image to any same standard size for all image and then take the HOG features. This solved the input feature size problem but this process destroyed aspect ratios of the image and was not translational invariant. Plus the dimentionality was really high.

Hence we first chose a window size and took a sliding window on every image and saved all the resulting vectors. We trained a KMeans clustering model on these vectors for k-centroids. Now for every sample in the training corpus, we took all the sliding windows possible on those images and recorded which centroid group a feature belongs in. Then for the image we create a frequency vector where every dimension is the frequency of a certain centroid. Since no of centroids are k, every image will now have a k-dimensional vector. We use this with the labels to train our SVM.

During testing, we take all the sliding windows possible on the images, take the boxes over a certain threshold score. Then apply non-maximal suppression over it to generate the final result.

Results:

AP and AR

MR vs FPPI

Some Samples:

Inferences:
The model detects less small sized and occluded humans, as was intended in the dataset.

The model still makes misclassifications on poles and other objects which slightly look like humans.

Model Link: https://drive.google.com/drive/folders/1sxsOZ-tHByAWuFMwa1zPiVQnxorIcqXd?usp=sharing

3) Faster RCNN

Implementation:

Faster RCNN model with a resnet 50 backbone, pre-train on COCO classification data set is used. It is from the torchvision library. We have used a DataLoader class to manage the dataset and then selected the bounding boxes with label 1.

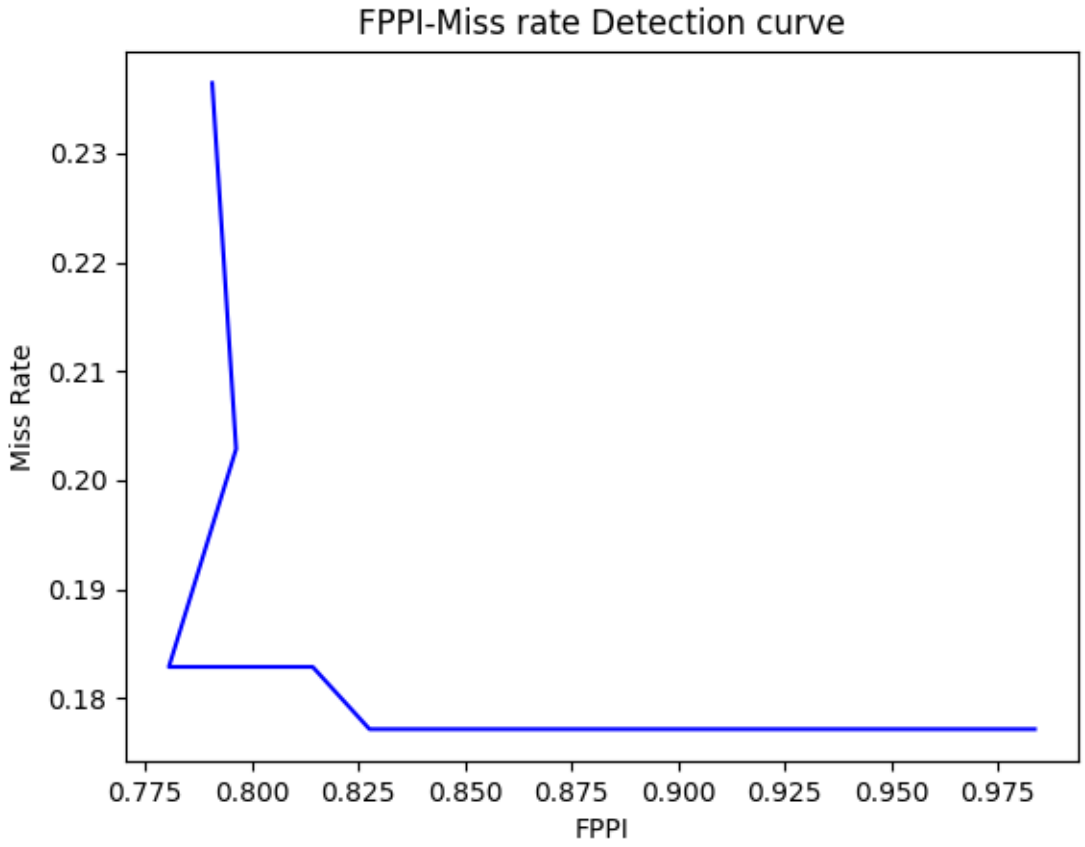
Because the original dataset did not have small images with the val

AP and AR

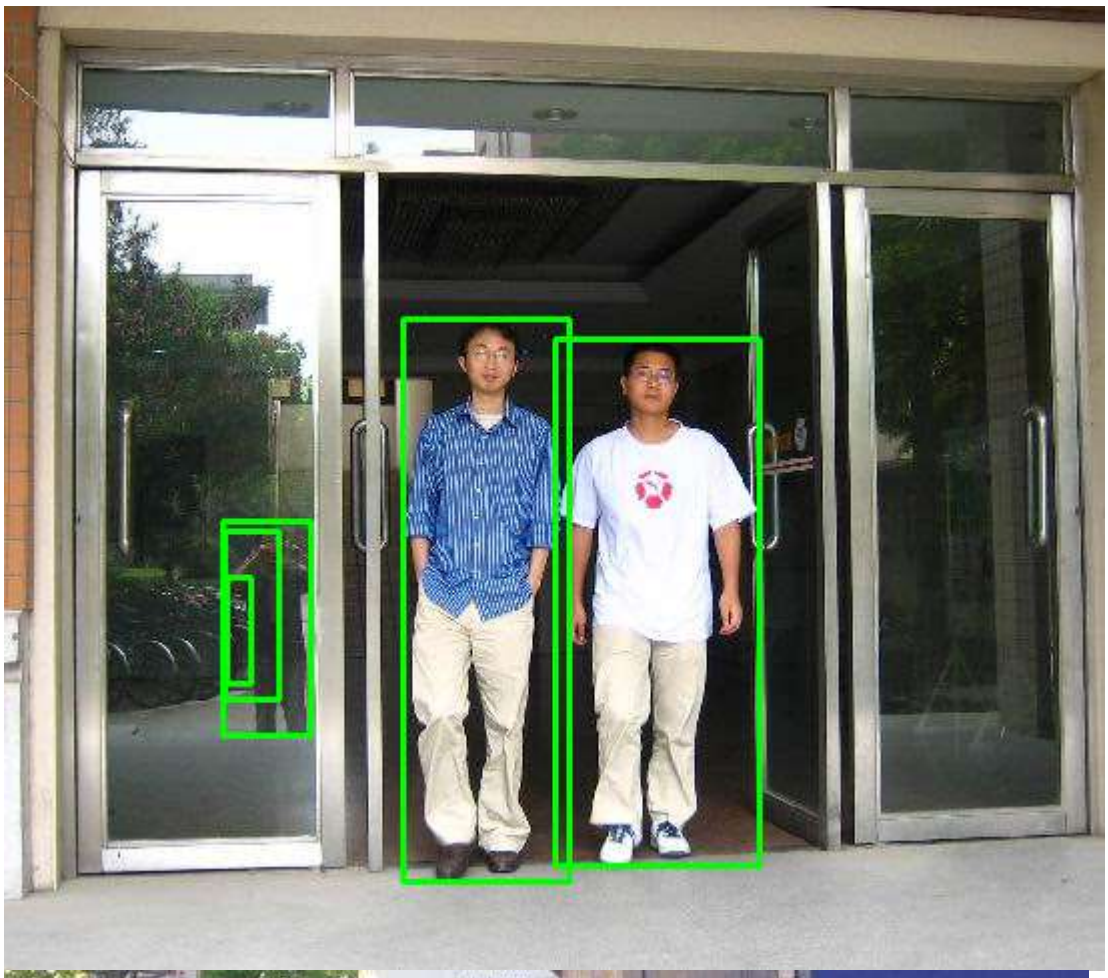
```
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.764
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.962
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.879
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.164
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.586
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.786
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.296
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.824
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.824
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.650
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.772
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.833
```

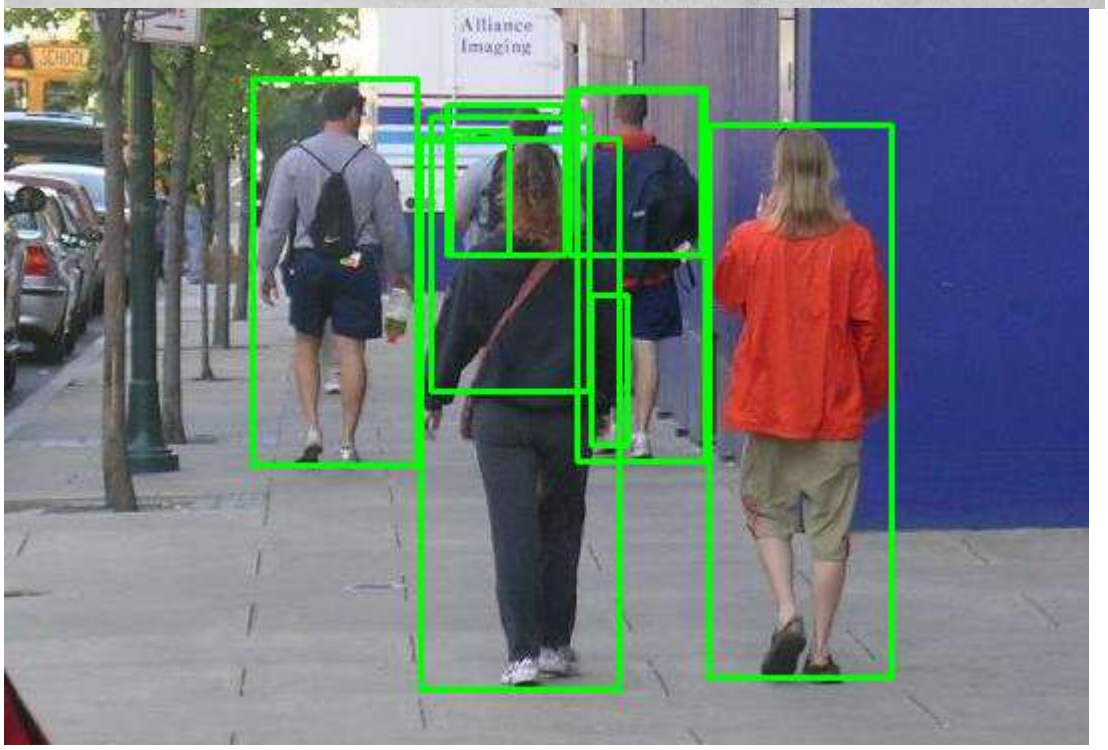
```
Average Precision =  0.7637724775988447
Average Recall @ 1 detection per image =  0.29642857142857143
Average Recall @ 10 detections per image =  0.8235714285714286
```

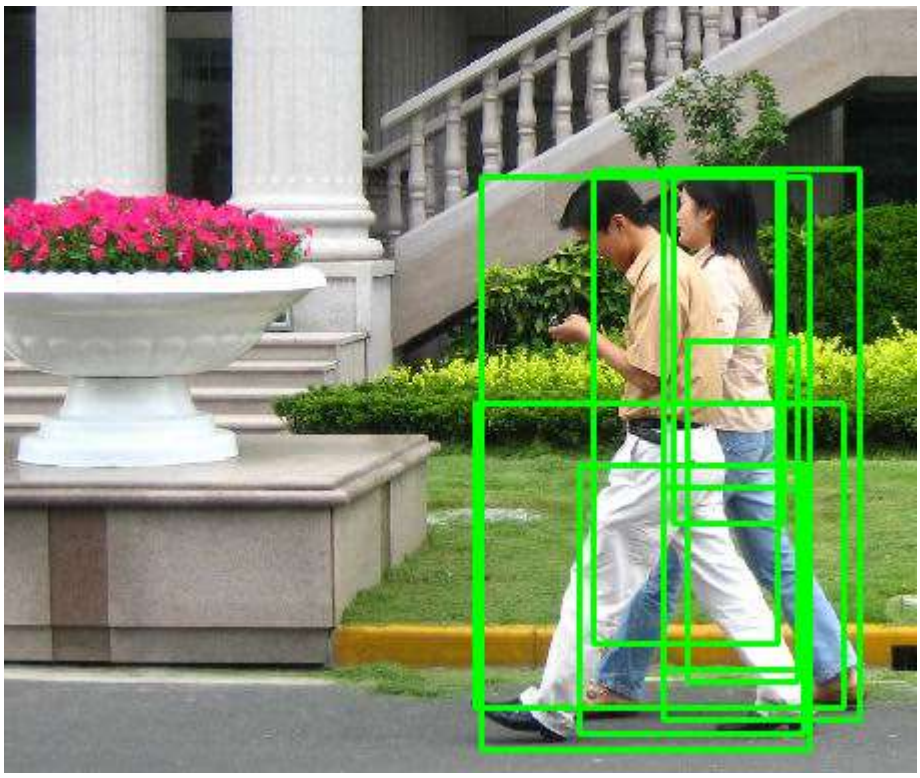
FPPI vs MR

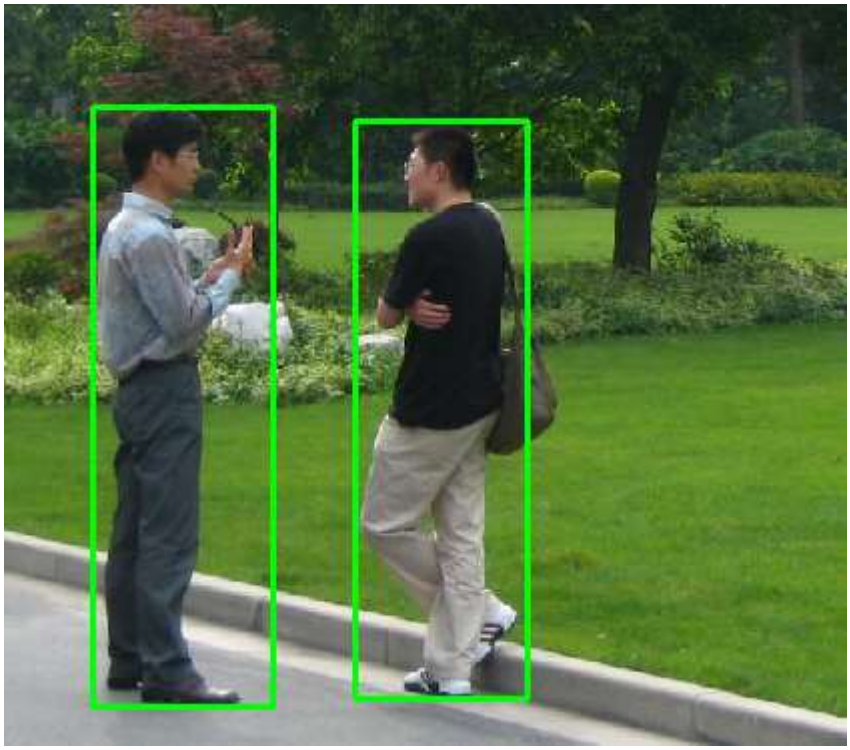


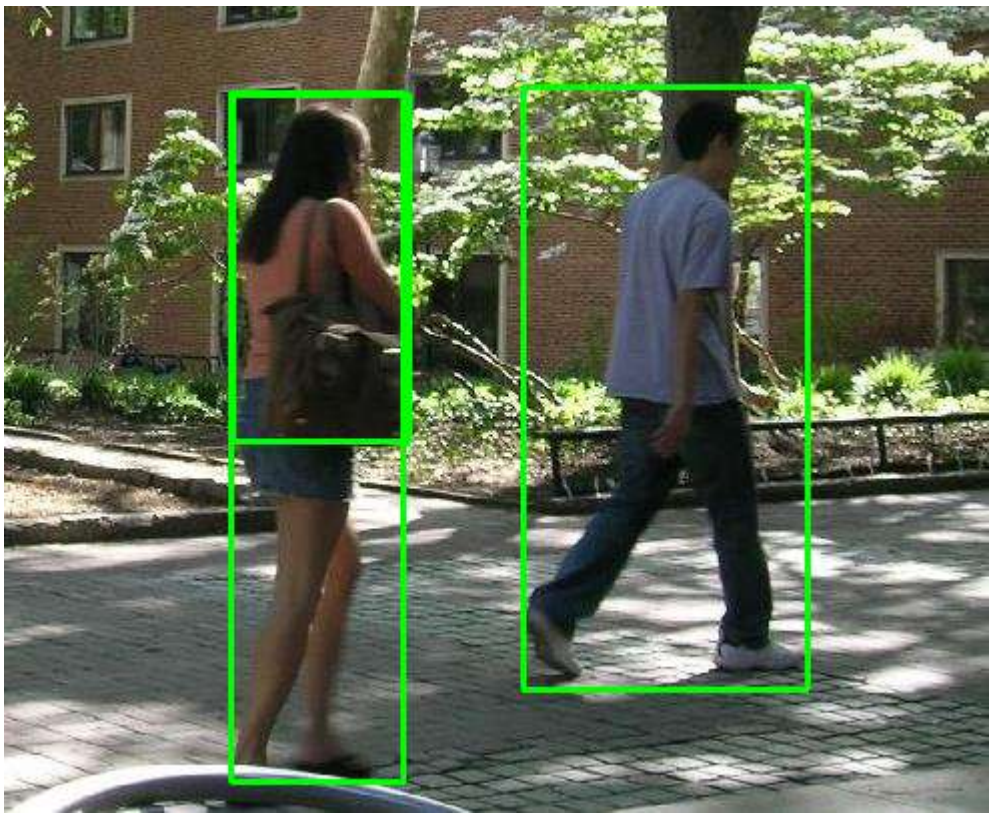
Samples:




Alliance
Imaging







Conclusion:
In the sample images
1) In part1, when two people are standing close, they are covered with a single bounding box.In part3,even when two people are standing close, they are treated as different objects of interest with separate bounding boxes.
2) In part1, even other objects like trees are getting detected, ignoring some background humans.Whereas part 3, background humans are detected while it does not detect trees and similar objects with similar aspect ratio
3) Again, in part 1, objects other than humans are getting detected. As can be seen in the figures, only humans are getting detected in 3rd part.
4) In part 1, only half a human is getting detected. In part3, unlike part1, full bounding box is getting detected.
5)The man on the right is correctly detected instead of just his leg in rcnn vs pretrained-HOG. An interesting observation is the woman on the left is detected

For the AP-AR values:
We can see the large amount of change in both accuracy and recall.
Faster RCNN performs much better than PretrainedHOG with svm

For MR vs FPPI graph:

The area under the curve of MR-FPPI graph is more for the first part than the third. (Directly proportional to average MR). This also shows that the fasterRCNN model is better than pretrained HOG