VIC Assignment 2

Florez de la Colina Inès

1. Introduction

In this assignment we had to implement a pedestrian detector using non deep learning methods. In this report, I will present the three methods I have implemented: HOG/SVM, background subtraction and finally a concatenation of both methods. The idea of implementing these methods came from [1] and [2].

2. HOG/SVM approach

I have decided to start with the following approach. I have used the INRIA person's dataset to train my model. I created a training set of 14596 samples with 2416 positive samples and 12180 negative samples. The HOG descriptors have been computed on images of size 128x64 and the model has been saved under the name svm_model_final.p. I have tested this model on a test set of size 375 (75 positive samples and 300 negative samples) and have obtained the following results:

$ConfusionMatrix = \begin{bmatrix} 297 & 3 \\ 8 & 67 \end{bmatrix}$, and $F1\_score = 0.924$. Thus, according to those results, the model performs very well. Now that we have trained our SVM, we can compute the bounding boxes. To do so, we compute, for a given image, a pyramid of gaussians by downscaling the image by a certain downscaling factor (I have chosen 1.6) until the image reaches the limit size of 128x64. For each image in the pyramid we slide a window of size 128x64 and compute the HOG features for all windows. We apply the SVM to the feature vector found and if it predicts 1 (i.e a pedestrian), we save this window as a bounding box. The size of the bounding box depends on the scale of the resized image considered. Finally, by applying this sliding window method, we can find bounding boxes that overlap. Hence, we apply non-maximum suppression to obtain the final boxes. The results obtained with this method are presented on Figure 1.

The drawbacks of applying this method are that it is a time-consuming method. It takes quite a lot of time to compute the bounding boxes as we go through all images from the gaussian pyramid, and for each image we compute the hog features of all windows. Moreover, it presents some difficulties in detecting the pedestrians when they are far away and when they are quite close. In addition to this, this method predicts some false boxes.

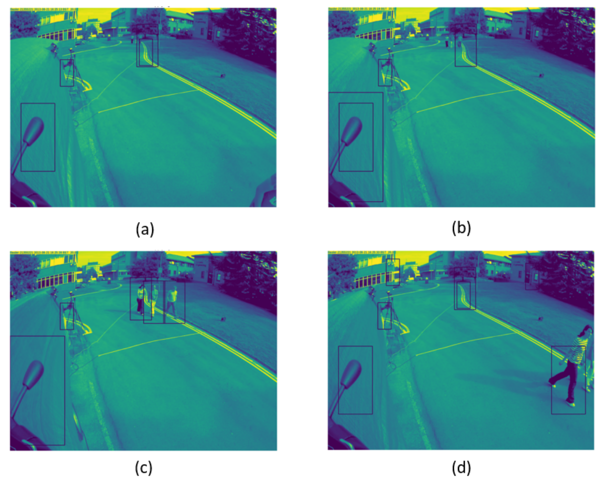

Figure 1. Bounding boxes computed with SVM/HOG: (a) Frame 1, (b) Frame 120, (c) Frame 300, (d) Frame 399

3. Background Subtraction

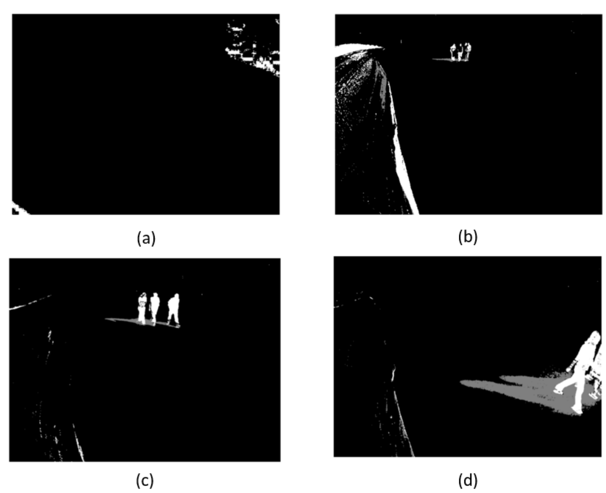

Figure 2. Background Suppression applied on the video sequence: (a) Frame 1, (b) Frame 120, (c) Frame 300, (d) Frame 399

In this assignment I have also implemented a background subtraction method. Subtracting the background of a video sequence allows for a better detection of the pedestrians as can be seen on figure 2. Nevertheless, even with background subtraction, we can't detect the pedestrians when they are very far away (cf Frame 1). To compute the bounding boxes, I have used an OpenCV implementation that highly speeds up the computation of the boxes.

Before using this implementation I tried applying the method above to the black and white frames and to the frames after having applied the Canny edge detector but I reached some unsatisfying results. To reach better results for this technique, I would need to broaden my training dataset by adding positive samples obtained by applying background subtraction to videos of pedestrians walking. Unfortunately, I didn't find the right material to do that on the internet. Hence, I used OpenCV's implementation. The results obatained with this method are presented on Figure 3.

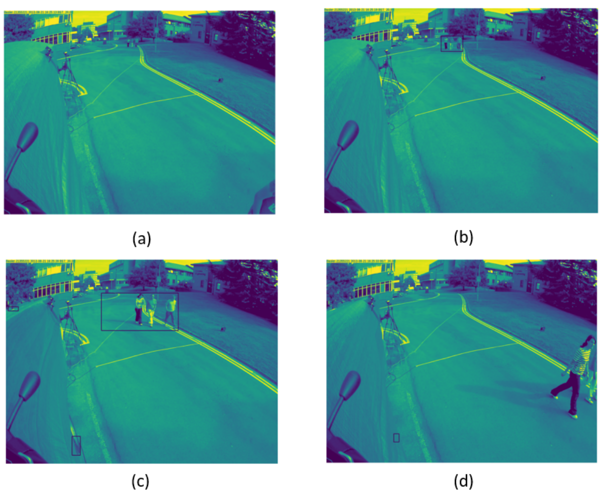

Figure 3. Bounding boxes computed using Background subtraction: (a) Frame 1, (b) Frame 120, (c) Frame 300, (d) Frame 399

We can see from these results that this method is able to detect pedestrians from further away than the SVM/HOG method. Nevertheless, it only detects one box containing all pedestrians. Furthermore, this method isn't able to detect the pedestrians when they are close. From the results obtained with both methods, I decided to concatenate both methods, as SVM/HOG performs better when pedestrians are close and background subtraction performs better when they are far.

4. Concatenation of both methods

The results obtained by concatenating both methods are presented on Figure 4. After having concatenated the bounding boxes from both methods, I applied again non-max suppression to get rid of overlapping boxes. In this non-max suppression, I favor the bounding boxes found with the HOG/SVM method. We can see that we still have some false boxes, as well as some nested boxes. Nevertheless, by concatenating both methods I managed to improve the results when the pedestrians are either far away or close.

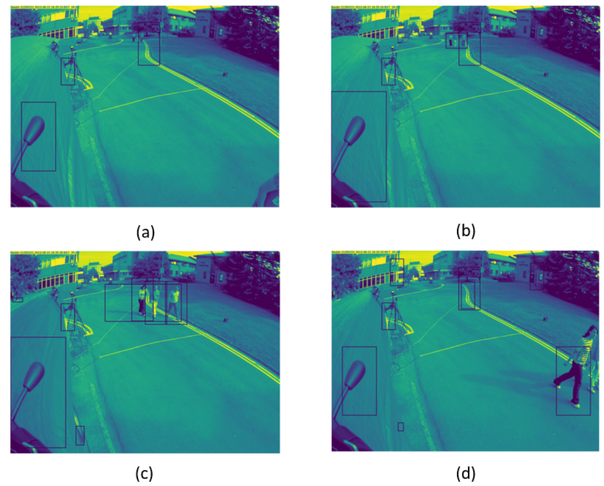

Figure 4. Bounding boxes computed concatenating methods: (a) Frame 1, (b) Frame 120, (c) Frame 300, (d) Frame 399

5. Choice of method

As I previously mentionned, the major drawback of my implementation of the HOG/SVM method from scratch is that it is very time consuming. Indeed it takes a little less than 10 minutes to predict the bounding boxes for one frame. Hence, it is impossible for me to compare the performances of the three methods as the computation is two long. Thus, I have decided to replace my implementation of the HOG/SVM with openCV's implementation that can compute the bounding boxes for 10 images in less than 2 minutes, just for the comparison of the methods. For my choice of model, I have compared the accuracy of the three models on the given image data. I have added one method here which consists of applying HOG to the back and white frames obtained after background subtraction. Here are the accuracies obtained by every method:

Table 1. Accuracy obtained for each method

HoG/SVM	Background Subtraction
0.0947	0.1177
Combination of HoG/SVM and Background subtraction	HoG applied to the black and white frames
0.0686	0.1153

According to those results, the best method seems to be the background subtraction method. Hence, I will use this method as my final method.

6. Conclusion

To conclude, this assignment allowed me to discover some computer vision methods widely used in pedestrian detection, and more generally in object detection. As my implementations from scratch were quite time consuming I wasn't able to test them on the whole datset but only on a few images. Nonetheless, they really helped me understand how the detection is carried out. The main advantage of the background subtraction method chosen as my final method is that it is an OpenCV implementation. Hence the computation of all bounding boxes only takes a minute overall.

References

[1] Azar Fazel, Viet Vo *Pedestrian Detection and Tracking in Images and Videos* 1

[2] Yuan Xu, Lihong Xu, Member, IEEE, Dawei Li, Yang Wu *Pedestrian detection using background subtraction assisted Support Vector Machine* November 2011 1