People Detection
with Histogram of Oriented Gradients
Project report

Chiara Bizzotto, Maurizio Ingrassia and Roberta Papa

July 23, 2015

Abstract

In this report we present a possible implementation in Python of Histogram of Oriented Gradient for people detection. For training and classification we used a Support Vector Machine using a linear kernel. A sliding window checks through a binary mask if there is movement in that region of interest avoiding in this way useless detections. Regions with movement are thus processed from the SVM which decides if it contains a person or not. We proceed then with calculating the performances both in the preliminary single-scale case and later in the multiscale one, added to get over some of the constraints.

1 Introduction

Detecting and tracking people is an important and widespread argument of research that in the last years has *roused* a lot of interest thanks to its wide range of application. Our project aims to apply one of the techniques used in literature, based on Histogram of Oriented Gradients (HOG), to detect people in a scene.

The minimum target of our work is that each person must be found at least once in a set of frames: in this way our algorithm could be easily used, after some improvements, in a video-surveillance system.

Our project has some constraints, due to the choices taken during the setup of the work, that make it suitable for specific contexts only: the detection is limited to moving pedestrians, i.e. our algorithm can find only people walking and not people standing still. Furthermore, to make the choice of parameters easier, we have decided to limit our range of action to fixed-camera videos and to use only one scale for the detection.

Section 2 describes the Data-sets chosen for the experiment both for the Training and for the Validation/Test of the model. Section 3 explains in details the algorithm we have implemented: both with respect to the theoretical side (such as HOG and SVM) and to the practical one, related to our additions to the standard algorithm. Section 4 presents the numerical results obtained applying our algorithm to the data-set and finally in Section 5 we discuss the results obtained and we propose some improvements to the method.

2 The Data-set

The data-set used for the training is composed, for the positive example, of the *ViPeR Data-set* and for the negative examples of a custom-made one (see Figures 1 a-e), where one can find background, objects and “pieces” of people. The size of the images is 48x128 pixels which we treat as the base dimension for the window (Section 3). In total we used more than 6000 images for the training.

For the validation and the training we took a set of videos from *3dPes* (for example Figure 1 f) of about 200 frames each on average where from 1 to 10 persons walk in the scene. One of the major obstacles of this data-set is the difficult light condition.

Data-set annotations have been done manually for the whole set, which leads sometimes to a little misalignment with the region detected: this affects the evaluation of the performance both in positive and in negative, thus we think that the total effect can be neglected.

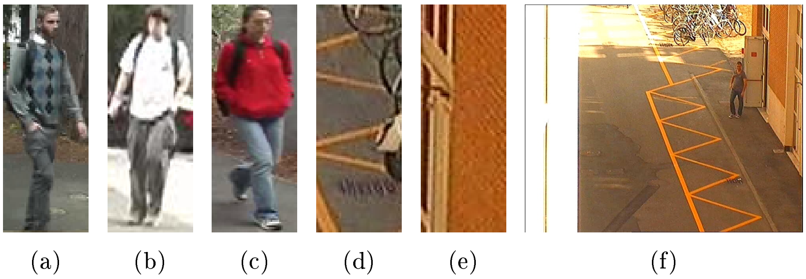


Figure 1: (a)-(c): Positive examples for training, (d)-(e): Negative examples for training, (f): Frame example for validation and test.

3 The Algorithm

The algorithm we implemented is briefly described in the block scheme in Figure 2. For each frame of the video we extract the background using an iterative method that update the background B as follows:

$$B(t) = \alpha B(t-1) + (1-\alpha)I(t)$$

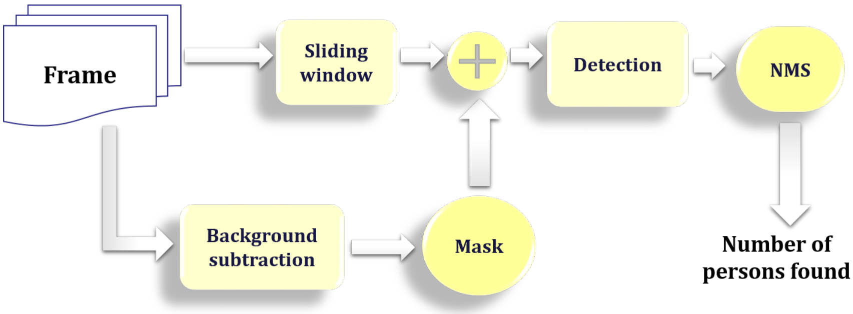
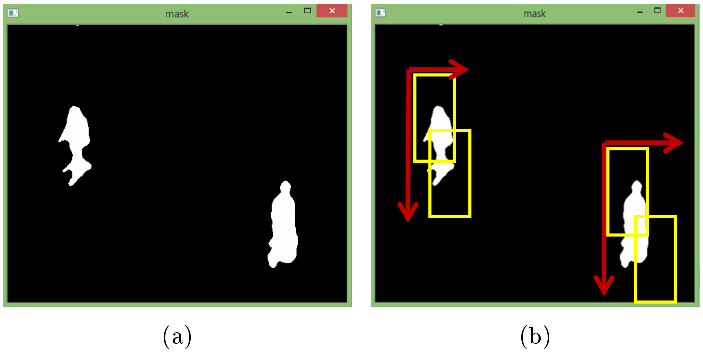


Figure 2: Block diagram of the algorithm

where $I(t)$ is the intensity (gray-scale) of the current frame and α is a weight that represents the trade-off between the relevance of the actual frame and the previous ones. This allows us to build a mask that highlights only the regions of the frame in which there is movement. We tried two types of masks: one based on contrast (using gray-scale frames) and the other based on saturation (using HSV frames). For our application we combined the two masks by summing them, to obtain more robust regions (see Figure 3a).



(a)

(b)

Figure 3: Example of mask with movement regions (a) and sliding window (b).

Once the mask has been calculated we perform the “sliding window” on the frame in exam (Figure 3b): a fixed-scale (with respect to the base one) window scans the frame and try for detection only if the number of white pixels, in the same ROI of the mask, is over a given threshold. For these windows we extract the feature vectors (HOG descriptor[1]) and we send them to the SVM classifier to get a prediction of the class they belong to: Person or Non Person. We use a probabilistic approach for the matching, which also allows us to discard multiple detections around the same ROI: we choose only those predictions that have the probability higher of a fixed threshold. For

[1] HOG parameters: 8 orientations bins, 16x16 pixels per cell, 1x1 cells per block

what concerns the classifier, we choose to use the linear kernel instead of the intersection one because of the less computational cost. At this point it is still possible to have multiple detections for a single person: for this reason we perform *Non Maxima Suppression* by keeping, in the neighborhood of a detected person, only the bounding box with the higher probability.

Afterwards one improvement has been made to the algorithm and one of its most important limits has been overcome: we introduced the multiscale detection by cycling on the dimension of the sliding window.

The results obtained from our detector and the variations due to the multiscale will be presented in Section 4.

4 Experimental results

The performance of the classification were evaluated by cross-comparison between ground-truth and people found by our system. To check the correctness of the prediction we used two different methods, the first based on a threshold on the distance between the corners of the two bounding boxes, and the second on the overlapping of the two areas. Information on the goodness of our predictions were obtained computing Accuracy rate, Recall, Precision and F1-score derived from the confusion matrix. In order to obtain the best performances these evaluation measures have been calculated changing the parameters of the model, in term of step of the sliding window, threshold of the mask and the C-value of the linear SVM (see Figures 4 and 5), for each video and then averaged them on the entire validation-set.

The first graph (Figure 4) shows how the method performs worse by increasing the step of the sliding window, because it increases the chance of missing a person. We choose not to decrease the step under 11 because the computational costs were too high compared with the gain in performance. The matching threshold represents the probability we set as a threshold during the prediction in a specific ROI.

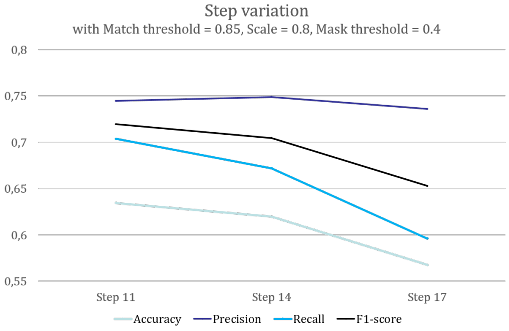
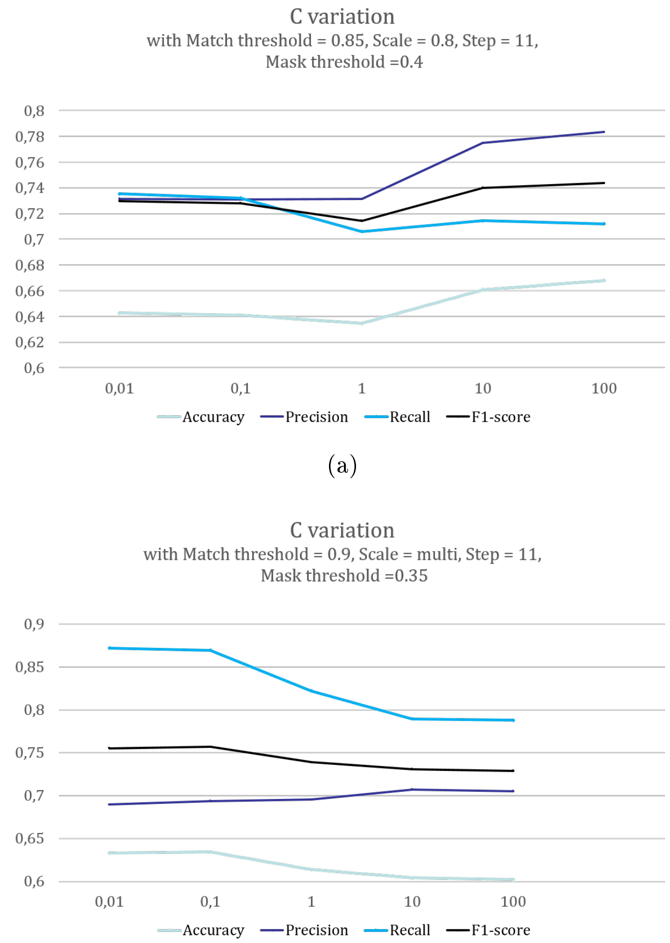


Figure 4: Performances as the step of the sliding window change (single scale case).



(a)

(b)

Figure 5: Performances as the C-value changes in the single-scale case (a) and in the multiscale case (b).

	Best fixed scale (=0.8)	**Multiscale**
Accuracy		
Recall		
Precision		
F1-score		

Table 1: Final results

5 Discussion

alle hog piacciono le gambe

real-time