# People Detection with Histogram of Oriented Gradients

Project report

Chiara Bizzotto, Maurizio Ingrassia and Roberta Papa

July 26, 2015

**Abstract**

In this report we present a possible method for human detection in videos. We implemented an algorithm in Python that uses Histograms of Oriented Gradient descriptors, associated with Support Vector Machine (SVM) classifier with linear kernel, for detecting people in scene. A sliding window checks through a binary mask if there is movement in that region of interest avoiding in this way unnecessary checks and wrong detections. The features extracted from those rectangular regions with movement are thus processed from the SVM which decides if it contains a person or not. Finally we evaluated the efficiency of our method both in the preliminary single-scale case and in the multiscale one, added to get over some of the constraints.

## 1 Introduction

Detecting human beings in a given scene represents one of the most important and challenging tasks in computer vision and is becoming a key technology for many areas. 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 set-up 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.

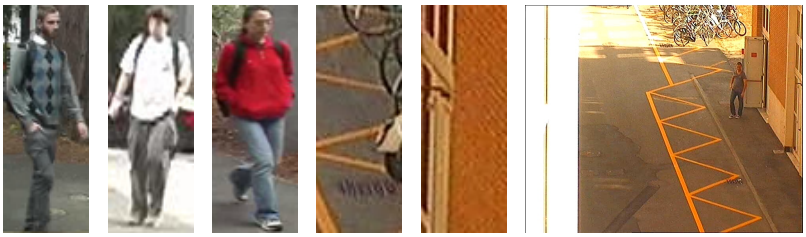
Section 2 describes the Datasets chosen for the experiment both for the Training and for the Validation/Test of the model. Section 3 explains in details the additions we have implemented to the standard algorithm (HOG descriptor and SVM classifier). Section 4 presents the numerical results obtained applying our algorithm to the dataset and finally in Section 5 we discuss the results obtained and we propose some improvements to the method.

## 2 The Dataset

The dataset used for the training is composed, for the positive example, of the *ViPeR Dataset* 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 dataset is the difficult light condition.

Dataset 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.



(a) (b) (c) (d) (e) (f)

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

## 3 The Algorithm

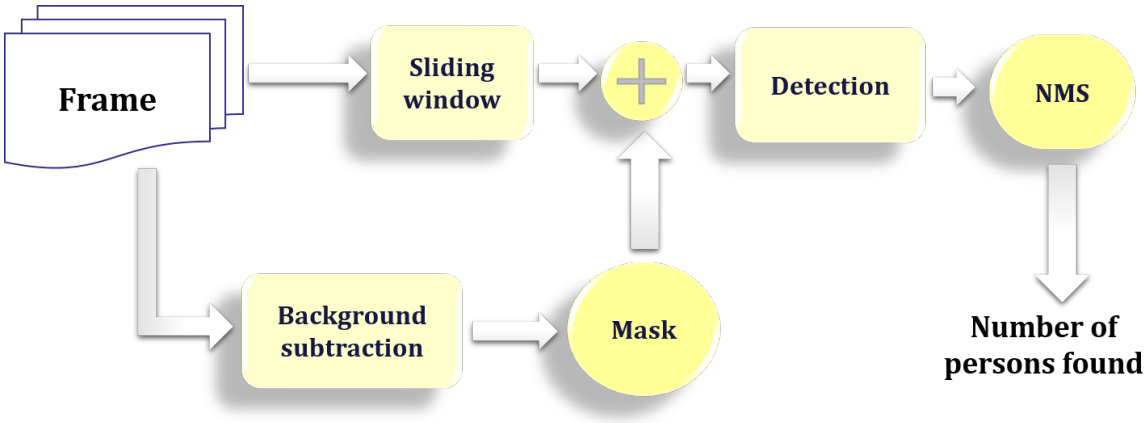



Figure 2: Block diagram of the algorithm

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:

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

where $I(t)$ is the intensity (gray-scale) of the current frame and $\alpha$ 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 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 neighbourhood 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. Figure 4 shows an example of how our algorithm works.

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


(a)

Figure 4: Example of detection in a scene with two people.

## 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 as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - Score = \frac{2TP}{2TP + FP + FN}$$

- False positives (FP): examples predicted as positive, which are from the negative class.
- False negatives (FN): examples predicted as negative, whose true class is positive.

- True positives (TP): examples correctly predicted as per- taining to the positive class.
- True negatives (TN): examples correctly predicted as be- longing to the negative class

The evaluation measure most used in practice is the accuracy rate. It evaluates the effectiveness of the classifier by its percentage of correct predictions. The recall is the proportion of examples belonging to the positive class which were correctly predicted as positive. Precision is a measure which estimates the probability that a positive prediction is correct and the F1-score is the harmonic mean of precision and recall.

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 5 and 6), for each video and then averaged them on the entire validation-set.

The first graph (Figure 5) 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 window stride under 11 because the computational costs were too high compared with the gain in detection accuracy. The matching threshold represents the probability we set as a threshold during the prediction in a specific ROI.

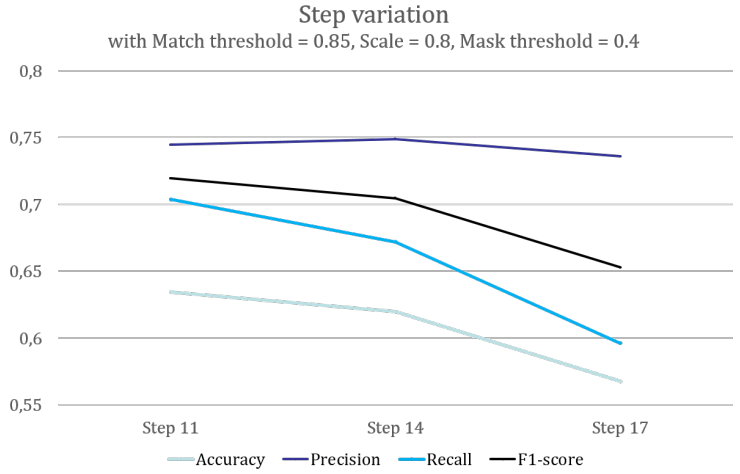
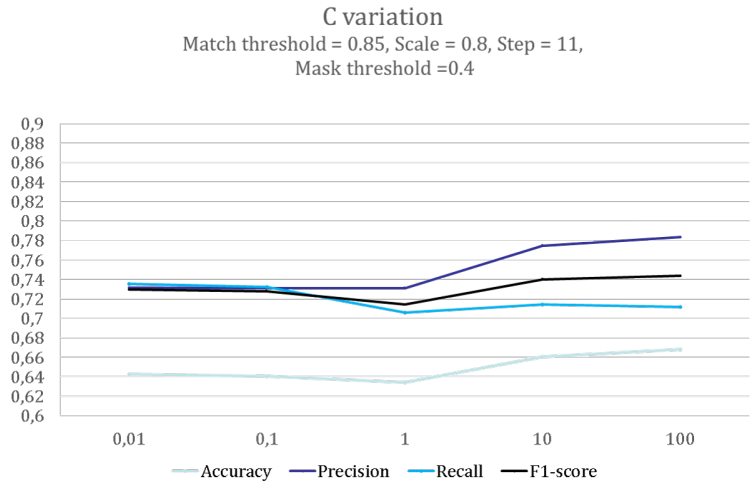
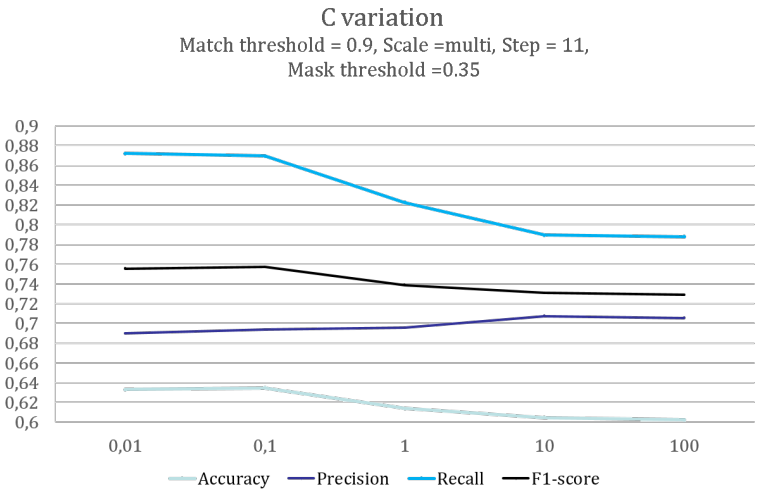



Figure 5: Performances as the step of the sliding window change (single scale case).

The second chart (Figure 6) show the consequences of the variation of the C-value with single-scale of the sliding window (a) and multiscale (b) related to the best parameters choosen by tuning them on the entire validation set. It is clear how the general performance of the algorithm, given by the F1-Score, does not change significantly from the first to the second case; on the contrary there is a slightly larger difference in terms of precision and recall (see Table 1). In fact with multiscale the recall increases because, by looking for people of "different sizes", the algorithm has more possibilities of detecting a person; however on the other side, because of the low robustness of HOG descriptors, the precision decreases: in our case this is not due to the increasing of the number of false positives but to an incorrect positioning of the detection window.




(a)




(b)

Figure 6: Performances as the C-value changes in the single-scale case (a) and in the multiscale case (b) with best parameters for each algorithm. Accuracy, Precision, Recall and F1-score represent the Mean Average value of each measure on the Validation Set.

| | **Fixed scale** | **Multiscale** |
|---|---|---|
| **Accuracy** | 0.660 | 0.633 |
| **Recall** | 0.714 | 0.872 |
| **Precision** | 0.775 | 0.690 |
| **F1-score** | 0.740 | 0.755 |

Table 1: Final results with the best parameters of each algorithm: fixed scale = 0.8, step = 11 (for both algorithms), match threshold = 0.85 for single-scale and 0.9 for multiscale, mask threshold = 0.4 for single-scale and 0.35 for multiscale. Accuracy, Precision, Recall and F1-score represent the Mean Average value of each measure on the Validation Set.

# 5 Discussion

In this report we implemented a people detection algorithm: in particular we proposed two variations, one based on fixed scale detection and the other on multiscale, that is an improvement to the first solution. Despite the better results, the greater computational cost of the multiscale does not let the algorithm run real-time like the single-scale does. Interesting results are obtained using the multiscale in those videos where there is a grater number of people: with multiscale we can better manage the "crowd" achieving improvements in classification performances (see Table **??**).

Concerning future modifications, we can think of enhancing the dataset, in particular by adding more negatives examples. Moreover, we can introduce some kind of parallel computing to have a better response and work in real-time even with multiscale.

Other possible future developments of our algorithm could be done applying Principal Component Analysis (PCA) to reduce the dimensionality of the feature vectors and testing different classifiers. Performing PCA could be also a good solution to decrease the computational cost of multiscale making it faster and letting the algorithm run real-time. Furthermore other kinds of descriptors could be tried because the HOG is not robust in case of occlusions or object with high variations with respect to the background.

| | **Fixed scale** | **Multiscale** |
|---|---|---|
| **Accuracy** | 0.504 | 0.610 |
| **Recall** | 0.665 | 0.776 |
| **Precision** | 0.629 | 0.740 |
| **F1-score** | 0.647 | 0.758 |

Table 2: Performance comparison between Fixed-scale = 0.8 and multiscale in a video with more people.