# An Optimized Sliding Window Approach to Pedestrian Detection

Victor Hugo Cunha de Melo*, Samir Leão*, David Menotti†, William Robson Schwartz*
*Computer Science Department, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil
†Computing Department, Universidade Federal de Ouro Preto, Ouro Preto, Brazil

***Abstract*—While a large number of surveillance cameras available nowadays provide a safe environment, the huge amount of data generated by them prevents a manual processing, requiring the application of automated methods to understand the scene. However, the majority of the currently available methods are still unable to process this amount of data in real time, mainly those focusing on pedestrian detection. To optimize pedestrian detection methods, this work proposes a novel approach that performs a random filtering supported by the Maximum Search Problem theorem to select a very small number from all possible detection windows. Although the random filtering is able to select regions that capture every person on an image, some windows can cover only parts of a person, diminishing the accuracy. To solve that, a regression is applied to adjust the windows to the person's location. The computational cost reduction comes from the fact that the proposed approach does not need to perform any processing while selecting windows, differently from cascades of rejection that must evaluate at least simple features for every window. The experiments performed using a pedestrian detection based on Partial Least Squares show that the approach is effective in both accuracy and computational cost reduction.**

***Keywords*—*visual surveillance; pedestrian detection optimization; random filtering; location regression; Partial Least Squares***

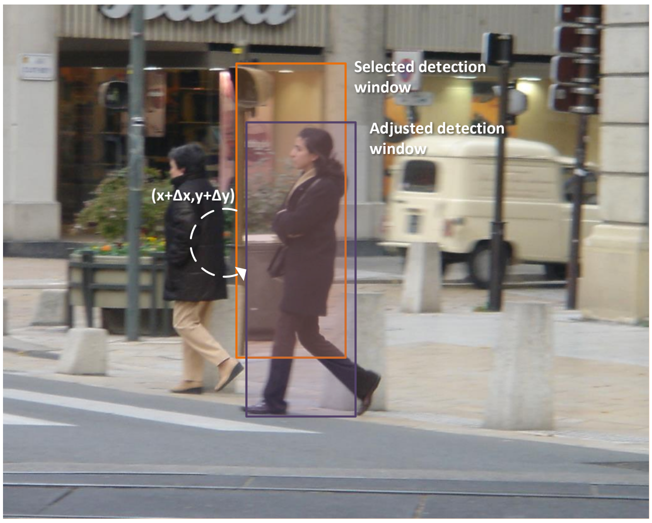

Fig. 1. Example of the adjustment of the detection window location after performing the location regression (best viewed in colors).

## I. Introduction

Pedestrian detection presents a fundamental importance in applications such as video surveillance and monitoring, since the location of the persons in the scene provides useful information to activity recognition and understanding, the main goals of such applications. Therefore, to enable the processing of the vast amounts of data provided by multiple surveillance cameras, the volume of data must first be reduced through pedestrian detection methods so that one can focus only on regions of interest to solve problems such as person tracking, face recognition, person re-identification, action and activity recognition, aiming at the scene analysis and understanding.

Several challenges are faced by the pedestrian detection [1]. Among them are changes in illumination and person's appearance, pose variations, low quality of the acquired data, and the small size of the pedestrian in the image, making the detection process harder. However, besides all these challenges, the majority of applications require high performance and reliable detection results, which increases the need for efficient and accurate pedestrian detection approaches.

Even though many pedestrian detection methods have been proposed in the past years, the methods currently available still do not present high performance [2] to be employed on domains in which many images need to be processed in real time. Therefore, it is desirable the development of methods to significantly reduce the computational cost. One way of achieving that is to focus on optimization approaches. The most common are those based on the computation of efficient feature descriptors [3], cascade of rejection that incrementally increases the feature complexity [4], region of interest filtering using saliency detection to reduce the number of detection windows to be evaluated [5], GPU-based approaches to parallelize the processing [6]. However, such approaches still are not enough to achieve real time processing in most of cases.

This work proposes a novel approach to optimize pedestrian detection methods based on sliding windows. The idea is to perform a random filtering in the image to select a very small number of detection windows and discard the remaining ones. Differently from the approaches described in the previous paragraph, the proposed approach does not perform any type of processing in the discarded windows, providing a significant speed-up. To the best of our knowledge this is the first time such approach is employed to optimize pedestrian detection.

Due to the random nature of the choice, the selected windows might be slightly dislocated of the person's body, which need to be fixed before presenting them to a classifier[1]. Therefore, a regression, referred to as *location regression*, is

[1] In this work we use the detectorPLS [7] to perform the classification, but any other sliding window based detector could be considered.

executed to each detection window to adjust its location in the image, as illustrated in Figure 1. A similar approach has been employed in an earlier work [8], however, with a different goal, detection of redundant windows.

The random filtering approach has been inspired by the Maximum Search Problem theorem [9, pp. 180] which states that it is very likely to find a good approximation for the maximum of a set by sampling a small subset. In the pedestrian detection context, this means that it is possible to find a detection window with high response (where a person is located) by looking into a small subset of randomly sampled windows. Previously, Zhu *et al.* [10] adopted this theorem with another focus. They were interested in selecting blocks to be used during the training phase of a cascade of rejection.

The experimental evaluation, performed using the publicly available INRIA Pedestrian Dataset [11] (widely employed to pedestrian detection evaluation), demonstrates that it is possible to discard a large number of detection windows and still achieve accurate results. In addition, the employment of the location regression has shown to be very effective on correcting the detection window locations. As a consequence, a significant reduction on computational cost can be achieved.

## II. Related Work

Several efforts have been made to address the optimization of pedestrian detection methods based on sliding windows. Among them, we highlight those approaches which use efficient subwindow search, saliency detectors, GPU, and cascade of classifiers. In addition, some heuristics such as subsampling windows for certain strides in order to evaluate windows in fixed sizes have been proposed [11], [12].

Lampert *et al.* [13], focusing on searching only promising regions of the image, proposed a branch-and-bound approach to perform object localization that finds the global optimum of a quality function over every possible sub-image. That approach typically runs in linear time or faster and requires fewer classifier evaluations than there are candidate regions in the image.

By simulating the behavior of the human visual system, saliency detectors are able to detect regions of interest [14], [5]. Initially, these methods quickly detect the possible positions of proto-objects in the image. An obtained saliency map suggests the position of the proto-objects. A filtering method that finds the salience of each window and segments the image into regions based on their similarities was proposed by Feng *et al.* [15]. It employs the difference among regions given their LAB histograms and spatial distances in order to find the most likely window to contain a salient object.

The multicore architecture of GPUs can also be very useful to accelerate computation and improve object detectors. For such aim, the algorithms must be designed to explore those hardware features. As feature extraction is a stage of high computational cost, pedestrian detection on GPUs can be highly accelerated. To extract HOG features Wojek *et al.* [16], Zhang *et al.* [17], and Masaki *et al.* [18] demonstrated efficient parallel techniques using GPU.

A common approach used to optimize object detection is based on a cascade of rejection, composed by weak classifiers learned during the training [19], [4]. These approaches use simple classifiers to discard detection windows that are easy to classify as counterexample (background samples). More complex classifiers are used as remaining windows of a stage advance through the cascade.

Using the cascade of rejection technique by successively combining classifiers with increasing complexity, Viola and Jones [20] were able to reject a large amount of windows in early stages. On a real time processing framework, Zhu *et al.* [10] applied a similar approach performing the extraction of HOG descriptors and achieved a detection rate comparable to that achieved by Dalal and Triggs [11]. Recently, Marin *et al.* [4] proposed the use of Random Forests to combine multiple local experts. To reduce computational cost, the multiple local experts share the extracted features. Then, the ensemble is integrated in a cascaded fashion.

Even though cascades of rejection are able to discard a large amount of windows, they still need to extract and evaluate features for all windows, which increases the computational cost [21], [4]. On the other hand, our proposed approach, the random filtering, does not need to execute any processing for discarded windows. Only after selecting a small subset of windows, we will need to extract and evaluate features for this small subset.

## III. Proposed Approach

To reduce the amount of data processed by the pedestrian detector, we propose a method based on a random filtering followed by adjustments on the detection window locations. To accomplish that, our method first generates a set of windows based on a sliding window approach. As these windows are generated in a wide range of scales and strides, we have a set of overlapping windows presenting high redundancy, which highlights the need of the next step of our approach. Here, we randomly select a fraction of windows that will be presented to a classifier (details in Section III-A). However, the selected windows might be slightly displaced from the pedestrian's location, which results in low responses. Therefore, before presenting them to the classifier, a regression is employed to adjust the window location to increase the responses (details in Section III-B). An overview of the steps in our approach is depicted in Figure 2.

### A. Random Filtering

The first step of any detector based on sliding window consists of generating a set of coordinates for several detection windows in multiple scales. After that, the detection is handled as a classification problem. If a large number of scales and small strides are used to move the detection window, we might have a large amount of windows to be classified, demanding high computational power. On the other hand, if few scales with large strides are considered, pedestrians might be missed. Therefore, the former option is more suitable to achieving high accuracy.

After generating an initial set of detection windows, instead of presenting all windows to a classifier, our approach performs a random selection of the windows, *i.e.*, a percentage of the total number of detection windows is selected. To ensure that every pedestrian is still detected, we use the Maximum Search

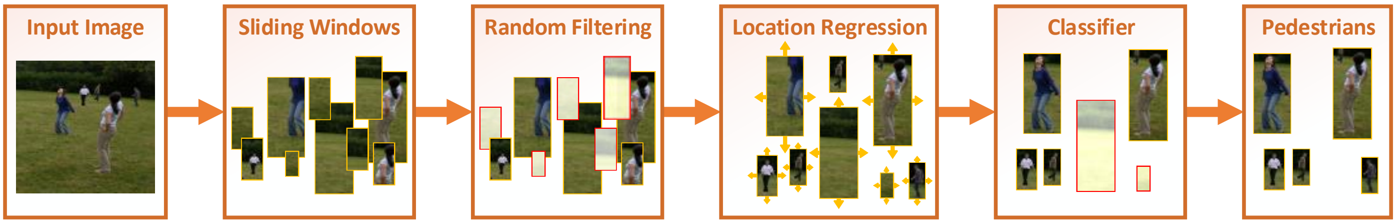

Fig. 2. Overall layout of the proposed approach based on random filtering and location regression (best viewed in colors).

Problem theorem [9, pp. 180]. The problem of classifying windows as containing pedestrians or not may be seen as the task of finding a subset of windows containing humans from a finite set of windows. As most maximum search problems, the exact solution is computationally expensive (every sample has to be evaluated). Instead, it is possible to find almost optimal approximate solutions by using probabilistic methods as the one described as follows.

The problem at hand might be formulated as follows. Given a set of $m$ windows, where $M := \{f_1, \ldots, f_m\}$ and $\mathcal{Q}[f]$ is a criterion to evaluate whether a detection window is covering an image region with a pedestrian, *i.e.*, the classifier response. Then, the problem can be stated as finding a window $\hat{f}_i$ that maximizes $\mathcal{Q}[f]$. In the pedestrian detection context, one is interested in finding not only the window $\hat{f}_i$ that maximizes $\mathcal{Q}[f]$, but also a subset of windows with large $\mathcal{Q}[\hat{f}_i]$, since more than one pedestrian might be in the image.

To solve the aforementioned problem, all terms $\mathcal{Q}[f_i]$ must be computed, which demands $m$ detection window evaluations. Due to the multiple scales and strides considered to locate all pedestrians in an image, the number of extracted windows is large for a given image, rendering this operation too expensive. For instance, for an image with $640 \times 480$ pixels, there are approximately 60,000 detection windows that need to be evaluated to detect pedestrian in multiple scales. Therefore, it is imperative to find a cheaper approximate solution.

Schölkopf *et al.* [9] demonstrated that by selecting a random subset $\tilde{M} \subset M$ sufficiently large, one can take the maximum over $\tilde{M}$ as an approximation of the maximum over $M$. If a small fraction of $\mathcal{Q}[f_i]$ $(i = 1, 2, \ldots, m)$, whose values are significantly smaller or larger than the average do not exist, one can obtain a solution that is close to the optimum with high probability.

To compute the required size, $\tilde{m} = |\tilde{M}|$ $(\tilde{M} \subset M)$, of a random subset to achieve a desired degree of approximation, Schölkopf *et al.* [9] showed that one can use the following equation

$$\tilde{m} = \frac{\log(1-\eta)}{\ln(n/m)} \tag{1}$$

where $\eta$ is the desired confidence and $n$ denotes the number of elements in $M$ having $\mathcal{Q}[f_i]$ smaller than the maximum of $\mathcal{Q}[f_i]$ among the elements in $\tilde{M}$.

The Equation 1 states that at least one element $f_i \in \tilde{M}$ will have $\mathcal{Q}[f_i]$ higher than the $\mathcal{Q}[f_i]$ of $m$ elements in the original set $M$ with a confidence of $\eta$. Therefore, this result would not be helpful when the element $f_i \in M$ with only the maximum $\mathcal{Q}[f_i]$ is required ($\tilde{m}$ would be very close to $m$). However, in the pedestrian detection problem based on sliding windows, we can take advantage of the fact that one human is covered by more than one detection window leading to a correct detection. This behavior is due to the redundancy resulting from the small strides in $x$ and $y$ and multiple scales (Figure 3 illustrates that, even when very few windows are randomly selected, there a number of windows covering the pedestrian location). Therefore, $n$ in Equation 1 can be fairly large, which reduces $\tilde{m}$ significantly.

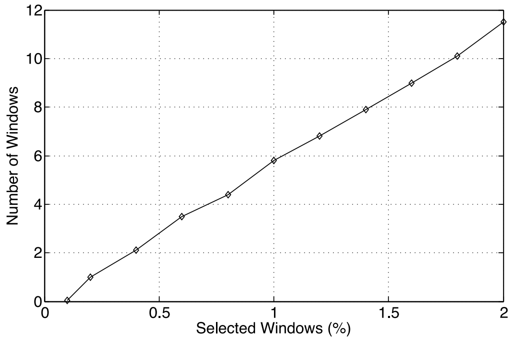

Fig. 3. Mean number of detection windows covering a pedestrian in the INRIA dataset as a function of the percentage of randomly selected windows. As standard approach to determine whether a pedestrian is covered by a detection window [2], we consider that a window $A$ covers a ground-truth window $B$ when their intersection divided by their union is greater than 0.5.

To illustrate the usage of the above result, we performed the following experiment. Given an image with $m = 60{,}000$ detection windows uniformly sampled from an $640 \times 480$ image pixels at multiple scales, we found that 583 windows contain the correct location of a pedestrian (windows that should lead to high values of $\mathcal{Q}[f_i]$, depending on the classifier being considered). Therefore, according to Equation 1, a random sampling with $\tilde{m} = 133$ (0.22% of the total windows) should contain at least a pedestrian with a probability of 95%, which is compatible with the plot shown in Figure 3.

Although the random filtering can provide a small subset of detection windows in such that almost every person in the image is covered, these windows might not provide the exactly location of a pedestrian. Hence, this pedestrian might be missed due to the low response achieved by the classifier. Therefore, we employ as extra step before presenting the window to the classifier to adjust the window location to the pedestrian, as will be described in the next section.

### B. Location Regression

Aiming at adjusting the bounding box delimited by a detection window, we learn a regression model (referred to as *location regression*) to correct it to the pedestrian's location. In this problem, we want to find displacements $\Delta x$ and $\Delta y$ such that, when added to the centroid $(x, y)$ of a given window, they move the detection window to the correct position of a pedestrian. Other variables may also be considered, such as the tuple $(\Delta w, \Delta h)$, in which $\Delta w$ and $\Delta h$ denotes width and height of a detection window, respectively. Nonetheless, in this work we consider only the $(\Delta x, \Delta y)$ tuple to demonstrate the detection improvement.

Previously, location regression has been applied on detection, but with a different purpose. In [8], the authors noticed that multiple redundant windows could be over a pedestrian, which should be presented to a generic classifier at the end of a detector stage. Hence, they applied location regression to reduce even further the number of detection windows that would be considered by the classifier by finding the corresponding pedestrian to each detection window. In this work, on the other hand, we only have few windows that were selected by random filtering and we want to predict their correct location. In other words, while the former wants to reduce redundant windows, the latter aims at finding the "best" location for the windows.

Unlike [8], our proposed method learns the regression model during an offline phase. First, we need to generate the training set to be presented to the learning algorithm. Given a training sample, we generate a set of displaced windows with the respective differences $(\Delta x, \Delta y)$ to their correct position. This set of displaced windows is generated in all directions, as long as the Jaccard coefficient between the ground-truth bounding box and the displaced window is greater than $50\%$ [2]. This ensures that we have a portion of the pedestrian within the window. The Jaccard coefficient $J(d_1, d_2)$ between two windows $d_1$ and $d_2$ is defined as

$$J(d_1, d_2) = \frac{|d_1 \cap d_2|}{|d_1 \cup d_2|}. \tag{2}$$

For instance, consider the sample shown in Figure 4. The blue bounding boxes represent the ground-truth annotation, while the green bounding boxes represent the generated samples. A bounding box that correctly matches a pedestrian, *i.e.*, it positioned exactly over a pedestrian, $(\Delta x, \Delta y)$ equal to $(0, 0)$. Another example is a bounding box displaced one pixel to the left, such as the first one, must add $(1, 0)$ to match the ground-truth centroid. Finally, a bounding box displaced one pixel above and to the right, such as the second one, must add $(-1, -1)$ to correct its location.

Once the training set is created, feature descriptors are extracted from the windows and associated to the displacements. Ideally, such descriptors should be simple to avoid extra computational cost. Then, a regression with two dependent variables is learned. Even though we have employed a regression based on Partial Least Squares due to its numerical stability and robustness to multicollinearity [22], other methods could have been applied.

Finally, when the detection is being executed, before presenting a window to the detector classifier, the regression is executed to correct its location in the image.

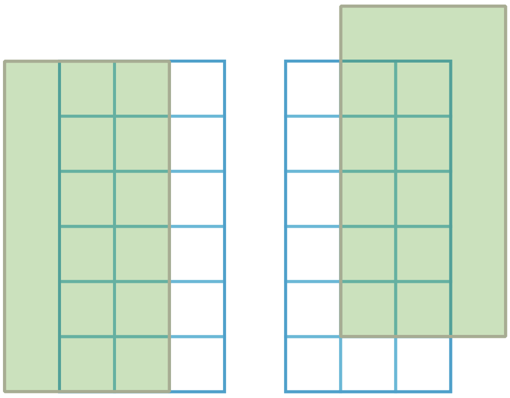

Fig. 4. Examples of sample generation used in the learning phase of the location regression.

## IV. Experimental Evaluation

This section evaluates the performance of the proposed method. First, the experimental evaluation focuses on two main aspects: the random filtering without applying a classifier (using only the ground truth) and the random filtering followed by the application of a classifier. We report and discuss the execution costs comparing to the employment of the original version of the detector (without discarding detection windows).

The first experiment examines the random filtering to determine whether it does not miss a pedestrian when used by itself. This can be shown by applying random filtering and evaluating the results obtained with respect to the ground-truth (*i.e.* a perfect classifier). In addition, we also evaluate whether the location regression is able to improve the detection results when applied after the random filtering.

Even though the method may obtain high recall when considering the ground-truth, the windows might be slightly shifted from the centroid of a pedestrian, which leads to low classification responses. Hence, we want to evaluate whether a detector is able to locate a large number of pedestrians in these particular windows. This analysis is conducted according to the trade-off between the amount of sub-windows randomly selected and the results of a pedestrian detector over these windows. Moreover, we also execute the location regression to investigate whether it is able to correct the locations of the detection windows and increase the recall achieved.

**Experimental setup.** The experiments were conducted on the INRIA Person Dataset [11]. From the training partition, we generated $500,000$ negative samples to be used in the learning phase of the detector. Note that, differently from [11], no bootstrapping is performed to generate hard negative samples for training the PLS detector, which may reduce the results achieved. Since in this work we are only interested in the relative comparisons, this does not affect the experiments. In addition, this work was developed using the Smart Surveillance Framework (SSF) [23].

We consider the following setup to execute all experiments. The detection window size is set up to $64 \times 128$ pixels. The images are resized by a range of scales (increased by $10\%$), so that pedestrians with sizes between $60$ and $700$ pixels can be detected. The detection window is shifted by a stride of $12\%$

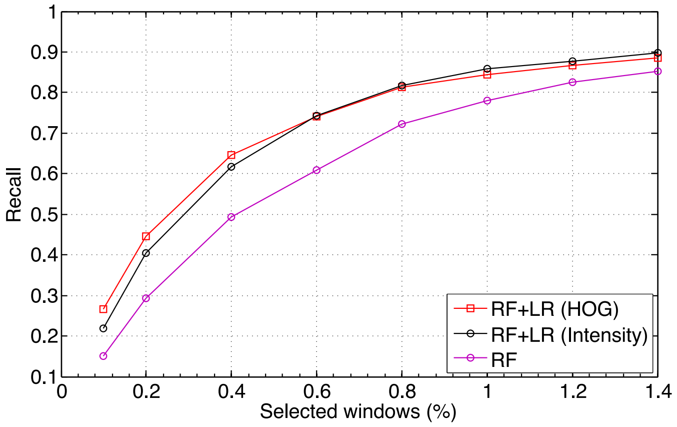

Fig. 5. Achievable recall as a function of the number of selected windows, evaluated on the INRIA dataset (RF: random filtering, LR: location regression).

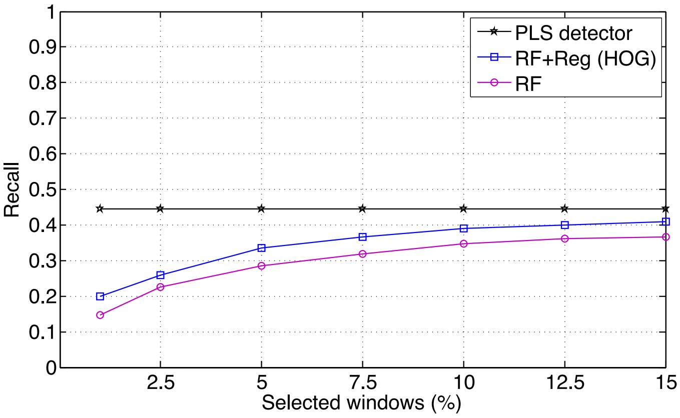

Fig. 6. Recall achieved at 1 FPPI when the selected detection windows are presented to the PLS detector. The PLS detector is shown as line because it is executed with 100% of the detection windows (without selection).

its width and 4% its height of the object size.

We use the PLS Detector [7] as our baseline (any other sliding window based detector could be used instead). The detector was trained using the same Histograms of Oriented Gradients (HOG) setup used by Dalal and Triggs [11], *i.e* a feature vector with $3,780$ dimensions. To execute the location regression, we consider two feature descriptors: pixel intensity and HOG. The former simply considers the pixel intensity within the detection windows, while the latter is the same one used by the classifier.

**Ground-truth comparison.** To verify the applicability of the theorem presented in Section III for pedestrian detection, this experiment determines the ratio of pedestrians that are being covered by at least one detection window as a function of the percentage of selected windows. This can be verified according to their correct position given by the ground-truth.

Figure 5 shows that a random selection of 1.4% of detection windows is enough to detect 83% of the pedestrians on the INRIA dataset (if the classifier provided perfect results). Note that according to the theorem, approximately 0.2% would be enough to approximate the maximum (find at least one pedestrian). However, since, on average, two people are present in each image, this value increases. In addition, the recall of 1 would not be achieved in this experiment because we are not padding the images, this way, people near to the edge of the images cannot be fit within a detection window.

Figure 5 also report results achieved applying the location regression using pixel intensity and HOG as feature descriptors. As we can see, the regression is able to correct the position of the detection windows and, consequently, increase the recall achieved by the random filtering to 90% at 1.4%. Note that these results show the maximum achievable recall if the detector provides perfect results. The next experiment evaluates the actual recall achieved by the PLS detector [7][2].

**Location Regression.** Now, we conducted experiments to determine which feature descriptor yields the best results for location regression. Since it must be fast, we trained the regression models with two simple feature descriptors. The first model uses pixel intensity as feature, which consists of concatenating all the pixels within the detection window into a feature vector. The second model is based on HOG, also with the Dalal and Triggs' setup. The results of both models are presented on Figure 5. As both present comparable results, we chose to use only HOG, because it has lower dimensionality and consequently is less subject to issues regarding the curse of dimensionality. This way, we discuss only results achieved with HOG in the following experiments.

**Pedestrian detector.** After performing the random filtering, it is possible to see in Figure 5 that the selected windows miss very few people in the dataset. However, these windows still need to be presented to a classifier, which may not obtain high accuracy due to some displacement of windows regarding to the person's location so that is does not result in high responses. This experiment evaluates how that may affect the accuracy of the detector/classifier.

The results in Figure 6 show the recall obtained at one false positive per image (FPPI). Even after executing the random filtering, the accuracy is still comparable to the original detector, which considers 100% of the detection windows (no windows are discarded). However, to achieve similar results, the number of selected detection windows had to be larger than the result achieved by the ground truth experiment. This indicates that, although the correct detection windows have been selected, the PLS detector does not provide high responses for all the correct windows. By using the location regression, we could improve the random filtering results, increasing the recall to 40.9% (close to the 45% achieved by the original detector).

The results achieved with this experiment demonstrate the high influence of the classifier in the accuracy. Even though a random selection of 1.4% detection windows covers more than 85% of the pedestrians in this dataset (Figure 5), when a classifier was employed to the same number of windows, the results decrease significantly. According to Figure 6, the maximum achievable by the detection PLS at 1 FPPI is 45%, but when 1.4% detection windows were selected, the recall achieved approximately 15%, which was increased to 20% when the location regression was considered. If it had followed the results in Figure 5, it should have obtained at least 40%

[2] For this work, the PLS detector was executed with a single stage. Therefore, it is not the same version employed in [7].

TABLE I. RELATIVE SPEEDUP ACHIEVED WITH THE PROPOSED METHOD WHEN COMPARED TO ORIGINAL DETECTOR ALONE (RF: RANDOM FILTERING, LR: LOCATION REGRESSION).

| SETUP | PERCENTAGE OF SELECTED WINDOWS | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1% | 2.5% | 5% | 7.5% | 10% | 12.5% | 15% | 100% |
| PLS Detector | – | – | – | – | – | – | – | 1.00× |
| RF | 67.91× | 37.64× | 21.81× | 15.58× | 12.10× | 9.62× | 8.45× | – |
| RF+LR | 64.83× | 35.13× | 20.40× | 14.59× | 11.39× | 9.09× | 7.97× | – |

of recall (90% of the maximum achievable). Therefore, these results emphasize the need for further studies to increase the robustness of classifiers to small displacements of the detection windows to allow the fully exploitation of techniques such as random filtering.

**Computational cost.** This last evaluation reports the speedup achieved when compared to the execution of the detector PLS alone. The results in Table I show the speedup for the results reported on Figure 6. The random filtering was able to achieve significant reduction in the computational cost, which also justifies the usage of the random filtering approach. In addition, we can see a low overhead when the location regression is employed after the random filtering.

## V. CONCLUSIONS AND FUTURE WORKS

This work proposed a detection optimization based on random filtering to discard a large number of detection windows, which is further improved by the application of a regression to correct the window location to fits the persons in the image. This approach can be applied as a early step of any sliding window based detector. For instance, it could be used as the first step of a cascade of rejection.

Compared to the application of a detector method alone, our experimental evaluation showed that accurate results at a reduced computation cost may be achieved by our method even when a large number of detection windows are discarded.

As future works, we intend to extend the study on the influence detection window displacements on the classifiers to further improve the location regression to achieve exactly the same result achieved by an sliding window based detector alone at a much smaller computational cost.

## ACKNOWLEDGMENT

The authors are grateful to FAPEMIG, CNPq, and FAPESP funding agencies.

## REFERENCES

[1] D. Geronimo, A. M. Lopez, A. D. Sappa, and T. Graf, "Survey of Pedestrian Detection for Advanced Driver Assistance Systems," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1239–1258, 2010.

[2] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian Detection: An Evaluation of the State of the Art," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[3] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast Feature Pyramids for Object Detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2014.

[4] J. Marin, D. Vazquez, J. Amores, A. Lopez, and B. Leibe, "Random forests of local experts for pedestrian detection," in *IEEE Intl. Conference on Computer Vision*, 2013.

[5] M.-M. Cheng, J. Warrell, W.-Y. Lin, S. Zheng, V. Vineet, and N. Crook, "Efficient salient region detection with soft image abstraction," in *IEEE Intl. Conference on Computer Vision*, 2013.

[6] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool, "Pedestrian Detection at 100 Frames per Second," in *IEEE Intl. Conference on Computer Vision and Pattern Recognition*, 2012.

[7] W. Schwartz, A. Kembhavi, D. Harwood, and L. Davis, "Human Detection Using Partial Least Squares Analysis," in *IEEE Intl. Conference on Computer Vision*, 2009.

[8] W. R. Schwartz, V. H. C. de Melo, H. Pedrini, and L. S. Davis, "A Data-Driven Detection Optimization Framework," *Neurocomputing*, vol. 104, pp. 35–49, 2013.

[9] B. Schölkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization and beyond*. MIT press, 2002.

[10] Q. Zhu, S. Avidan, M. chen Yeh, and K. ting Cheng, "Fast human detection using a cascade of histograms of oriented gradients," in *IEEE Intl. Conference on Computer Vision and Pattern Recognition*, 2006, pp. 1491–1498.

[11] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *IEEE Intl. Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886–893.

[12] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid, "Groups of adjacent contour segments for object detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 30, no. 1, pp. 36–51, 2008.

[13] C. H. Lampert, M. B. Blaschko, and T. Hofmann, "Beyond sliding windows: Object localization by efficient subwindow search," in *IEEE Intl. Conference on Computer Vision and Pattern Recognition*, 2008.

[14] J. G. d. Silva Filho, L. Schnitman, and L. R. d. Oliveira, "Multi-scale spectral residual analysis to speed up image object detection," in *Brazilian Symposium on Computer Graphics and Image Processing*, 2012.

[15] J. Feng, Y. Wei, L. Tao, C. Zhang, and J. Sun, "Salient object detection by composition," in *IEEE Intl. Conference on Computer Vision*, 2011, pp. 1028–1035.

[16] C. Wojek, G. Dorkó, A. Schulz, and B. Schiele, "Sliding-Windows for Rapid Object Class Localization: A Parallel Technique," in *Proceedings of the 30th DAGM symposium on Pattern Recognition*. Springer-Verlag, 2008, pp. 71–81.

[17] L. Zhang and R. Nevatia, "Efficient scan-window based object detection using GPGPU," in *IEEE Computer Vision and Pattern Recognition Workshops*, 2008.

[18] I. Masaki, B. K. Horn, B. Bilgiç *et al.*, "Fast human detection with cascaded ensembles," Ph.D. dissertation, Massachusetts Institute of Technology, 2010.

[19] X. Zeng, W. Ouyang, and X. Wang, "A cascaded deep learning architecture for pedestrian detection," in *IEEE Intl. Conference on Computer Vision*, 2013.

[20] Viola and Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Intl. Conference on Computer Vision and Pattern Recognition*, 2001.

[21] S. Paisitkriangkrai, C. Shen, and A. V. den Hengel, "Efficient pedestrian detection by directly optimizing the partial area under the ROC curve," in *IEEE Intl. Conference on Computer Vision*, 2013.

[22] R. Rosipal and N. Kramer, "Overview and Recent Advances in Partial Least Squares," *Lecture Notes in Computer Science*, vol. 3940, pp. 34–51, 2006.

[23] A. C. N. Jr., C. E. dos Santos, R. Ferreira, and W. R. Schwartz, "Smart Surveillance Framework: A Versatile Tool for Video Analysis," in *IEEE Winter Conference on Applications of Computer Vision*, 2014.