# Gradient-based Region of Interest Selection for Faster Pedestrian Detection

VARGA Robert, NEDEVSCHI Sergiu
Technical University of Cluj Napoca
Telephone: (800) 555–1212
Fax: (888) 555–1212

***Abstract*—This paper presents an approach to pedestrian detection that relies on a variable sized detection window. Its main aim is to facilitate a faster detection while maintaining a high detection rate. Speed-up is achieved by an efficient region of interest selection method and a clever detection system architecture. These two contributions can potentially enable real-time pedestrian detection on monocular images.**

***Keywords*—*Pedestrian detection; object recognition; region of interest selection; edge detection***

## I. Introduction

Increasing concern for pedestrian safety in the last year has resulted in the flourishing of pedestrian detection algorithms. These are essential in Advanced Driving Assistance Systems for preventing accidents involving pedestrians. Car companies are considering incorporating such systems into their models. For example, Volvo is planning to release cars that come with a pedestrian and cyclist detection module which will be able to stop the car automatically in case of an imminent collision.

Even though the problem was analyzed and tackled by many researchers it remains largely unsolved due to several difficulties: the various visual appearance and varied clothing of pedestrians, different possible postures and articulations, crowded scenes where partial occlusion prevents detection, the large range of scales. The problem is still open to research, with systems that meet real-time requirements being especially difficult to develop.

## II. Related work

For the purposes of this paper we will present only some related work in detail. These are papers which are strongly correlated with our approach and their description is needed for comparison. For a comprehensive overview of pedestrian detection algorithms the reader should consult the technical literature surveys [1], [2], [3], [4].

Despite the fact that there exist a multitude of approaches there is a tendency towards a general system architecture that is employed by most of them. We shall make use of this architecture to present and to emphasize different parts of the existing methods and our suggested approach. We mainly follow the description in [4] and state that the general pedestrian detection system has the following modules: preprocessing, feature extraction, region of interest selection (or foreground segmentation), object classification, postprocessing (verification and refinement), tracking.

To start off, we describe a traditional approach based on the method developed by Dalal [5] and extended by several other researchers [6], [7]. It is based on a fixed-size sliding window detection algorithm. To enable detection of pedestrians of different heights, the algorithm needs to resize the image and to recalculate the features for each scale. This is necessary because of two reasons: the detection window is of fixed size, and the features are not scale invariant. To obtain good results resizing must be done 4-8 times per octave and typically on 4-5 octaves. This leads to 16-40 feature recalculation steps. We consider this the weak point in similar approaches and our aim is to circumvent this situation.

Using two innovative ideas a recent publication by Benenson et al. [8] claim to achieve pedestrian detection at more than 100 frames per second. One of the ideas is to resize the features not images. The other is to use depth information for a successful region of interest selection. Our approach builds upon this work but it is different in many aspects. The most relevant would be that our method is for monocular images and the region of interest selection dos not require depth information.

The following subsections describe each module from the general architecture and provide additional details as well as references for each component relevant to our approach.

### A. Preprocessing

The module for preprocessing is responsible for operations aiming at reducing the noise from the images and also to improve image quality. Typical operations at this phase are low level image processing such as: filtering with low pass filter, histogram equalization, gamma correction, contrast enhancement, dynamic range etc. It is important to note that some descriptor types are sensible to these operations and some processing can lead to weaker detection performance.

### B. Feature types

One of the most useful feature types for pedestrian detection is the Histogram of Oriented Gradients (HOG) proposed by Dalal [5], [9]. Theses features are constructed from histograms where each bin corresponds to an orientation and each pixel contributes to the bin of the gradient angle with a value proportional to the gradient magnitude. The histograms from cells are grouped in blocks and normalized. This grouping in blocks preserves spatial distribution. Finally, all responses within the detection window are concatenated to form the full descriptor that will be fed to the classifier. Many of the best performing methods use this feature in conjunction with other information. This work has been extended to enable real-time computation of these features in [6] using integral images.

Haar wavelets were popularized by Viola and Jones [10] for fast detection. Theses are weighted sums of rectangular areas from within the detection window. Even though one can predefine such features based on simple observations of the structure of the object, it is recommended to generate these rectangular area randomly. By generating a large number of features one can apply an AdaBoost to select to most discriminant features automatically. This saves the developer the effort to find the best features and also ensures that non of the relevant feature configurations are missed if we let the feature to have a large dimensionality.

Integral Channel Features[11] generalize the concept of Haar wavelets. They are defined on a general image channel. This channel can be an intensity image; a color channel; gradient magnitude; channel corresponding to a histogram orientation bin etc. First order integral channel features are simply sums of rectangular areas from these channels. The optimization with integral images enables extremely fast calculation of theses features in constant time. (Integral images are cumulative sums along both the dimensions of the original image intensity). Despite their simplicity, these features can be used to achieve state-of-the-art results [2]. In [12] the authors present a fast detection method using these features and a scale correction method.

Other features used to complement the previous ones are presented next. Even though simple color is not helpful for classification relative color similarity between areas within the bounding box is a helpful feature. Color self-similarity [13] calculate histograms that encode second order statistics of colors. Motion cues are very helpful for detection when they are available. Works in this direction are: [14], [15], [16], [17].

### C. Region of interest selection

Good region of interest (RoI) selection methods can reduce the execution time of detection methods significantly because they eliminate unnecessary calls to the classifier. A survey by Geronimo [4] presents several approaches under the paragraph of Foreground segmentation. Most of the methods make use of stereo information to detect good candidate regions [8], [18], [19]. Monocular approaches are fewer in number and include: biologically inspired attentional algorithms [20], [21], vertical symmetry detection from infrared images [22], and segmentation algorithms. Simple and efficient region of interest selection methods using only monocular information are hard to find or inexistent.

### D. Classifier

The standard of-the-shelf classifier that is used in almost all classification tasks is the Support Vector Machine. Linear SVMs are fast enough to be applicable in this domain where hundreds of thousands of classifications must be made for each image. Radial basis function SVMs and other nonlinear kernels have better results but are much slower. Histogram intersection kernel SVMs have been proposed in [23] as an alternative to linear kernel variants for better results at the same speed. Boosted classifiers are more suitable for large dimensional feature vectors[24]. They successfully detect relevant features and have a good execution time. Cascading the weak classifiers can further speed up the process. Gavrila et al.[25] use hierarchical template matching to determine if a shape corresponds to a pedestrian or not. Another alternative involves neural networks.

### E. Non-maximum suppression

Typically pedestrian classifiers return true even for bounding boxes that partially overlap with the pedestrian. The result is that the detector will return a clutter of detections all centered around the true bounding box. Non-maximum suppression algorithms are employed in this stage to determine the best bounding box if there are overlapping ones. One of the more time-consuming approaches involves applying the mean-shift algorithm for this purpose. The other alternative is to retain the bounding boxes that have a higher confidence value in case of an overlap. We refer to this as the pairwise-max suppression algorithm.

## III. Proposed approach

We are aiming at a detector that does not need image resizing. As stated, this direction of research was already investigated in works such as [8] and [10], however these resolve the problem in a slightly different manor. Here we will train a classifier for each scale. With this approach the execution time can be reduced because a substantial time at detection is spent in the feature calculation phase for each scale.

Let us analyze the speed gain from this operation. Consider that feature calculation for an image of size $A$ is given by $\alpha A$. Then the cost for recalculating the features for 16 scales, corresponding to 4 scales per octave and 4 octaves will be:

$$C_{rescale} = \sum_{k=0}^{16} \alpha A s^{-2k} \approx \alpha A \frac{s^2}{s^2 - 1} = 3.41\alpha A \qquad (1)$$

In the last equation $s = 2^{0.25}$ is the scaling factor, which results from the 4 scales per octave requirement. We can see that this is 3-4 times larger than performing it only once on the large image $C_{noscale} = \alpha A$, not taking into account other necessary calculations.

To work in this framework, we must allow the detection window to have a variable height. The aspect ratio of width/height will be fixed to 0.5, this ratio can be easily changed to suit the prevailing mean ratio of the dataset. Using a variable size detection window requires a feature type that can be calculated on rectangular regions of arbitrary sizes. Integral channel features have this property and can be calculated very fast.

Even though some integral channel features are scale invariant the more discriminant ones are not. This depends on the channel type that was used. For example histogram bin channel yields an integral feature that is not scale invariant. This problem is solved in [12] by a scale correction and in [8] by correcting the responses of the classifier. Here, we will

**Algorithm 1** Detection method

**Require:** Input image.
**Ensure:** Pedestrians as an array of rectangles and confidence values.
1: Calculate channels for integral channel features.
2: Apply RoI selection using Algorithm 2.
3: Set *detections* = $\emptyset$
4: **for all** RoIs **do**
5: Calculate the features from within the RoI
6: Classify the features using the appropriate classifier
7: **if** confidence $> \theta$ **then**
8: Add the RoI to the *detections* list along with the confidence value
9: **end if**
10: **end for**
11: Apply pairwise-max on *detections*

consider different classifiers for each scale in order to eliminate the problem of scale variance.

Algorithm 1 formalizes the ideas presented above and describes the steps needed at detection time to obtain pedestrian bounding boxes. It is important to note, that feature calculation on the integral images is performed fewer times because of the reduced number of RoIs as opposed to calculating them for every region (step 5). This algorithm requires an already tuned region of interest selector and a trained classifier. Details regarding the first are presented in the next section, while the training procedure is described further in the experimental section.

## IV. RoI selection algorithm

Region of interest selection can be considered as a classification task that must be done quickly and must have a very low false rejection rate. In this sense, the classifier must be simple and fast. At the same time, it must reject as many regions as possible but must accept all possible future detection regions.

For this purpose we suggest a region of interest selection mechanism based on gradient information. The underlying simple idea is that object boundaries are found at positions where the gradient value is high. We search for the top and bottom of objects. We opt for vertical boundaries since pedestrian width has a lot a variance and there can be a lot of objects with vertical structure. The overview of the main steps of the algorithm is presented in Algorithm 2.

Step 1 helps to reduce noise, especially in images with a lot of texture (eg. dense foliage). In step 2, to obtain the top and bottom boundaries the y component (vertical) of the gradient is employed. We can use different methods for obtaining edge image such as: filtering with Sobel, Prewitt or Scharr edge filters, or applying the Canny edge detection algorithm. From hereafter we shall refer to the result of either of these operations as the *top* image.

We proceed by searching for locations where the gradient has a high value. For this, in step 3, we threshold to zero all pixels under a given value $t_1$. All non-zero locations will be considered as the middle point of the top of a potential bounding box. All that is left is to find the matching bottom and

**Algorithm 2** RoI selection

**Require:** Input image.
**Ensure:** Regions of interest as an array of rectangles.
1: Prefilter the image with a Gaussian filter
2: Obtain the edge image using a filter for the y direction (vertical).
Name the filtered image *top*.
3: Suppress small values using a fixed or dynamic $t_1$.
4: Filter the image *top* with a horizontal box filter of dimension $d$.
Name the filtered image *bottom*.
5: Set *RoIs* = $\emptyset$
6: **for all** possible rectangles with top center point $(x, y)$ and height $h$ **do**
7: **if** $top(x, y) > t_1$ and $bottom(x, y + h) > t_2$ **then**
8: Add the rectangle $(x - h/2, y) - (x + h/2, y + h)$ to *RoIs*.
9: **end if**
10: **end for**

the width is determined by the fixed aspect ratio. We observe that the bottom of a bounding box for a pedestrian will touch the feet, but it may touch it roughly at a single point (in the case of standing pedestrians when viewed from side) or in multiple points (for walking pedestrians). This suggests that it is not enough to search for the bottom of the bounding box under the first initial top point. We propose to sum up gradient values along the horizontal direction and to check these sums for possible bottom delimitators. To save time, the sums are precalculated using a horizontal 1-dimensional box filter. This corresponds to step 4.

The region of interest selector will then consider all possible rectangles and will decide it is a region of interest if the gradient at the *top* has a value larger than a threshold $t_1$ and also if the sum of gradients along the horizontal at the *bottom* is above a second threshold $t_2$ (steps 5-10).

The parameters for this classifier are: the type of edge detection (Sobel, Scharr, Prewitt, Canny), threshold value for the *top* image $t_1$, the dimension of the box-filter $d$, threshold value for the *bottom* image $t_2$, the heights of the admissible bounding boxes, the standard deviation of the Gaussian smoothing $\sigma$ applied before processing (0 for no presmoothing).

## V. Experimental results

We have performed tests on the INRIA pedestrian dataset. The training set contains 613 pictures with pedestrians, each picture can contain more than one pedestrian. The annotations are in the form of bounding boxes for each pedestrian. The negative set numbers 1218 images that do not contain pedestrians. It is one of the most widely used datasets for pedestrian detection evaluation.

All training procedures, including the parameter selection for the RoI selector, were done exclusively on the training set. For every scale we need to train a separate classifier. After studying the height distribution (see Figure 1) of the ground truth bounding boxes 4 scales were adopted: 64, 128, 256, 512 pixels. Note, this corresponds to canonical scales of: 0.5, 1, 2 and 4 for a 128x64 detection window. To obtain positive examples we resize the training images for each pedestrian bounding box from the ground-truth to match the fixed height of the classifier. The initial negative samples are obtained by sampling each of the negative example images randomly for 10 bounding boxes of the required height. Also, a random resizing is applied before cropping the negative image to match the resizing operations from the positive examples.

Once the initial training set is established the integral channel features are computed and saved along with the label of the sample. We follow the main guidelines from [12] and we use a feature vector of dimension 5000. As channel features we consider the channels of the Luv image, gradient magnitude channel, and 6 channels corresponding to the gradient orientation bins. Each of the 5000 features correspond to a randomly selected channel a and rectangular region defined on a 128x64 rectangle, having minimal area of 25. The size of the regions are adjusted for larger and smaller bounding boxes.

After all the descriptors are ready an initial boosted classifier is trained. We used Real AdaBoost with 1000 weak classifier consisting of 2-level decision trees from the OpenCV implementation. Using the predictions of this first classifier applied on the negative training set we obtain additional negative samples from all the mistakes performed by the classifier. We then retrain the classifier with these additional negative samples. This process is referred to as bootstrapping and is mainly useful in this case to reduce the number of false positives by establishing a relevant negative example set. We have observed that for smaller scales more negative examples were found during bootstrapping. This may prove that different scales pose different problems and training separate classifiers is a good way to solve these.

To evaluate the detection system on the test set the Pascal criteria is used to determine the correctness of our prediction. According to this, a predicted bounding box is correct if the ratio between the intersection and the union between the prediction and a ground truth bounding box is above a threshold set to 0.5. In order to eliminate bias to variable pedestrian width, ground truth bounding box widths are normalized to have width = height/2. This normalization is also adopted by Dollar to ensure correct evaluation in the pedestrian detection review [2].

We first analyze the effectiveness of the region of interest selection method. For this, we use the training set and apply the method on each training image. We then check what percentage of the ground truth bounding boxes is present in the returned regions. We notate this value as coverage and define it precisely as: the number of ground truth bounding boxes that are present in the selected regions divided by the number of ground truth bounding boxes. The boxes need not be exactly the same, but must overlap sufficiently. The same constant of 0.5 is used for this check. Another value of significance is the percentage of the bounding boxes retained. This will determine the speed-up that is achievable with the selection method. We define the speed-up as the mean value the speed-ups for each image. For a training image the speed-up is the ratio between the number of all possible bounding boxes divided by the number of accepted bounding boxes.

The minimal height is set to 24, the maximum height to 256, the dimension of the box filter is 32. For testing only the RoI selection we resize the input image to have a maximum dimension of 320 while retaining the original aspect ratio of the image. The data in Table I shows the importance of filtering and effect of other parameters. For Canny edge detection the parameter $t_1$ is the lower threshold and the higher is equal to $3t_1$. More smoothed images yield smaller coverage values because gradient magnitudes become smaller and this leads to rejection of more rectangles. This however simultaneously increases the gain in speed at the cost of false rejections. The speed-up is the theoretical speed gain obtained from the selector, the actual speed-up will differ from this value due to additional required calculations. Our aim is to pick a RoI selector that has a coverage very close to 1 and the highest speed-up ratio possible.

TABLE I: RoI parameter tests on the training set

| **Type** | $\sigma$ | $t_1$ | $t_2$ | $d$ | **speed-up** | **coverage** |
|---|---|---|---|---|---|---|
| Sobel | 0 | 100 | 2 | 32 | 46.52 | 0.98 |
| | 1 | 100 | 2 | 32 | 161.17 | 0.91 |
| | 2 | 100 | 2 | 32 | 430.21 | 0.83 |
| Scharr | 1 | 120 | 5 | 32 | 7.83 | 0.99 |
| | 2 | 120 | 5 | 32 | 10.05 | 0.98 |
| Prewitt | 0 | 100 | 2 | 32 | 148.41 | 0.94 |
| | 1 | 100 | 2 | 32 | 952.85 | 0.78 |
| Canny | 0 | 30 | 2 | 32 | 15.96 | 1.00 |
| | 1 | 30 | 2 | 32 | 23.90 | 1.00 |
| | 2 | 30 | 2 | 32 | 32.95 | 1.00 |
| | 2 | 30 | 5 | 32 | 144.21 | 0.99 |

Next, the ROC curves of three variants of the method are presented in Figure 2. First two methods use RoI selection method with Sobel and respectively Canny edge detection, while the third considers all possible detection windows. The parameters are chosen from Table I, row 1 from Sobel and row 3 from Canny. There is only a small deterioration in performance for using the RoI selector. In the critical zone from $10^{-2}$ to $10^{-1}$ the selector can actually help improve results. This demonstrates the effectiveness of the selection module. The detection system itself is good, the majority of the detection methods from the review [2] obtain a higher miss rate at $10^{-1}$ false positives per image. The best performing methods achieve around 0.25 miss rate at that mark. A better trained classifier with more rounds of bootstrapping and more scales would improve detection accuracy.

The speed gain from using the RoI selector can be obtained from comparing the running time of the detection algorithm with and without it on test images. Execution time measurements are given in Table II. Speed-up is dependent on the edge detection method used and on the input image itself. A noisy image will result in more regions accepted and a lower speed up. The implementation for feature extraction and classification is by no means fully optimized and even so, good execution times can be achieved. One of the smallest images in the test set has the size of 370x480, detection on this image is possibly in 200ms with RoI selection. Without the module it takes 10 times more. On larger images the speed gain can be even larger ( 33 less time needed using Sobel RoI selector ). Detection results for these sample images are shown in Figure 3. The time needed to perform testing on the whole test set shows an average speed-up of a factor of 10.

TABLE II: Comparison of execution times

| Test unit | Sobel | Canny | No RoI |
|---|---|---|---|
| Test set | 5 minutes | 4 minutes | 55 minutes |
| 370x480 img | 0.192 seconds | 0.182 seconds | 1.91 seconds |
| 960x1280 img | 1.032 seconds | 2.895 seconds | 33.02 seconds |

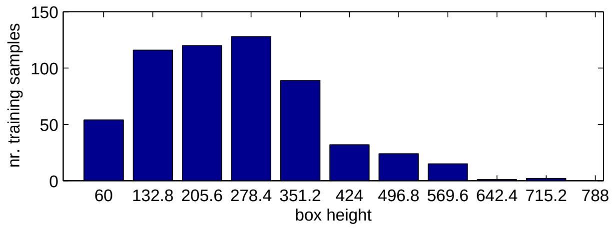


Fig. 1: Pedestrian height distribution in the INRIA training set

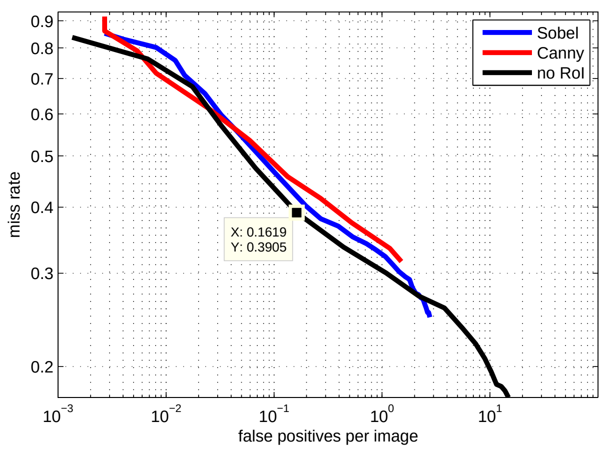


Fig. 2: Results on the INRIA test set - a sample value is emphasized near the $10^{-1}$ false positive mark

(a) 960x1280 image

(b) 370x480 image

Fig. 3: Sample detections with Canny RoI selection - The effects of using classifiers defined on fixed scales is visible but acceptable

## VI. Conclusion

This work presented a method for pedestrian detection that relies on a variable-sized sliding window approach and efficient use of integral channel features. The aim was to demonstrate that a simple and efficient region of interest selection can speed-up the execution time of the pedestrian detector while maintaining detection accuracy.

The first contribution of this paper is the original architecture for pedestrian detection that employs multiple classifiers, one for each scale, and uses a variable-sized sliding window for detection. The second contribution consists of the region of interest selection method that reduces the execution time and maintains detection accuracy.

There are many reasons why the proposed detection method is fast. Firstly, it is because integral features are inherently fast to calculate. Secondly, no image resizing is needed and features are only calculated once per image. Thirdly, we use an original region of interest selection method to reject most of the regions. Fourthly, boosted classifiers using 2-level decision trees are suitable and efficient classifiers for integral channel features.

In the future we plan to develop a training method for the RoI selector to automatically determine the thresholds from the training set. A better trained classifier with more bootstrapping rounds and more scales could help increase the detection rate. Another improvement would be using a cascaded classifier to lower execution for the system as a whole.

## Acknowledgment


This research was funded by the SmartCoDrive project, code PN II PCCA 2011 3.2-0742 from 03.07.2012 (2012-2015).


## References


[1] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *CVPR*, 2009, pp. 304–311.

[2] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 34, no. 4, pp. 743–761, 2012.

[3] M. Enzweiler and D. M. Gavrila, "Monocular pedestrian detection: Survey and experiments," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2179–2195, Dec. 2009.

[4] D. Gerónimo, A. M. López, A. D. Sappa, and T. Graf, "Survey of pedestrian detection for advanced driver assistance systems," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 32, no. 7, pp. 1239–1258, 2010.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005, pp. I: 886–893.

[6] Q. A. Zhu, M. C. Yeh, K. T. Cheng, and S. Avidan, "Fast human detection using a cascade of histograms of oriented gradients," in *CVPR*, 2006, pp. II: 1491–1498.

[7] T. Watanabe, S. Ito, and K. Yokoi, "Co-occurrence histograms of oriented gradients for pedestrian detection," in *PSIVT*, 2009, pp. 37–47.

[8] R. Benenson, M. Mathias, R. Timofte, and L. J. V. Gool, "Pedestrian detection at 100 frames per second," in *CVPR*. IEEE, 2012, pp. 2903–2910.

[9] N. DALAL, "Finding people in images and videos," Ph.D. dissertation, LINSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE, july 2006.

[10] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proc. CVPR*, vol. 1, pp. 511–518, 2001.

[11] P. Dollar, Z. W. Tu, P. Perona, and S. Belongie, "Integral channel features," in *BMVC*, 2009, pp. xx–yy.

[12] P. Dollár, S. Belongie, and P. Perona, "The fastest pedestrian detector in the west," in *BMVC*, F. Labrosse, R. Zwiggelaar, Y. Liu, and B. Tiddeman, Eds. British Machine Vision Association, 2010, pp. 1–11.

[13] S. Walk, N. Majer, K. Schindler, and B. Schiele, "New features and insights for pedestrian detection," in *CVPR*. IEEE, 2010, pp. 1030–1037.

[14] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *International Journal of Computer Vision*, vol. 63, no. 2, pp. 153–161, Jul. 2005.

[15] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *ECCV*, 2006, pp. II: 428–441.

[16] C. Wojek, S. Walk, and B. Schiele, "Multi-cue onboard pedestrian detection," in *CVPR*, 2009, pp. 794–801.

[17] S. Walk, N. Majer, K. Schindler, and B. Schiele, "New features and insights for pedestrian detection," in *CVPR*. IEEE, 2010, pp. 1030–1037.

[18] R. Labayrade, D. Aubert, and J. P. Tarel, "Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation," 2002, pp. 646–651.

[19] S. Nedevschi, S. Bota, and C. Tomiuc, "Stereo-based pedestrian detection for collision-avoidance applications," *IEEE Trans. Intelligent Transportation Systems*, vol. 10, no. 3, pp. 380–391, Sep. 2009.

[20] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 20, no. 11, pp. 1254–1259, 1998.

[21] T. Serre and T. Poggio, "A neuromorphic approach to computer vision," *Commun. ACM*, vol. 53, no. 10, pp. 54–61, 2010.

[22] A. Broggi, P. Cerri, and S. Ghidoni, "A correlation-based approach to recognition and localization of the preceding vehicle in highway environments," in *CIAP*, 2005, pp. 1166–1173.

[23] S. Maji, A. C. Berg, and J. Malik, "Classification using intersection kernel support vector machines is efficient," in *CVPR*, 2008, pp. 1–8.

[24] J. Friedman, T. Hastie, and R. Tibshirani, "Special invited paper. additive logistic regression: A statistical view of boosting," *The Annals of Statistics*, vol. 28, no. 2, pp. 337–374, 2000.

[25] D. M. Gavrila and S. Munder, "Multi-cue pedestrian detection and tracking from a moving vehicle," *International Journal of Computer Vision*, vol. 73, no. 1, pp. 41–59, Jun. 2007.