Semantic Channels for Fast Pedestrian Detection

Arthur Daniel Costea Sergiu Nedevschi
Image Processing and Pattern Recognition Research Center
Technical University of Cluj-Napoca, Romania
{arthur.costea, sergiu.nedevschi}@cs.utcluj.ro

Abstract

Pedestrian detection and semantic segmentation are high potential tasks for many real-time applications. However most of the top performing approaches provide state of art results at high computational costs. In this work we propose a fast solution for achieving state of art results for both pedestrian detection and semantic segmentation.

As baseline for pedestrian detection we use sliding windows over cost efficient multiresolution filtered LUV+HOG channels. We use the same channels for classifying pixels into eight semantic classes. Using short range and long range multiresolution channel features we achieve more robust segmentation results compared to traditional codebook based approaches at much lower computational costs. The resulting segmentations are used as additional semantic channels in order to achieve a more powerful pedestrian detector. To also achieve fast pedestrian detection we employ a multiscale detection scheme based on a single flexible pedestrian model and a single image scale. The proposed solution provides competitive results on both pedestrian detection and semantic segmentation benchmarks at 8 FPS on CPU and at 15 FPS on GPU, being the fastest top performing approach.

1. Introduction

A good perception and understanding of the surroundings is essential for an efficient and safe interaction with the environment. In this work we focus on traffic scenarios and in particular on the perception of pedestrians. They are the most important traffic participants and also the most vulnerable ones. Their perception can impose difficulties due to challenging weather, lighting conditions or difficult occlusion cases. In addition, their behavior can be sometimes very unpredictable.

Pedestrian detection is of high interest especially for the automotive industry, in order to design safe driver assistance systems or autonomous vehicles and represents one of the most challenging computer vision tasks.

Figure 1. Semantic context provided by the proposed solution

Computer vision based pedestrian detection is one of the most active research areas. The accuracy and precision of detectors increases every year, with a significant improvement over the last decade [6]. However, the computational cost of the top performing approaches still represents a bottleneck. Most of the approaches are impractical for real-time applications. Our main goal is to obtain a powerful detector that competes well with the top performing approaches at significantly lower computational costs.

In this work we propose a sliding window type detection solution based on multiresolution filtered LUV+HOG channels [53], focusing on computational cost reduction by optimizing the feature extraction, multiscale sliding window and classification schemes. We show that the same framework can be used to obtain semantic segmentations by classifying pixels into semantic classes such as sky, building, road, vehicle. As seen in figure 1, semantic segmentations provide a higher level representation. Background classes provide semantic context that can be used for search space reductions for different applications, while foreground classes can provide an alternative detection approach for obstacles.

We use the semantic segmentations as context informa-

tion for pedestrian detection and integrate them as semantic channels into the proposed solution. Eight semantic channels are used for six general semantic classes (sky, building, road, tree, vehicle, pedestrian) and two geometrical classes (horizontal and vertical structures). Experimental results showed improvements on detection rates using the additional semantic channels. The semantic channel for pedestrians provides an additional cue for their presence. This way, pedestrians are detected using two different recognition principles. The other 7 semantic channels provide the context.

Finally we achieve a solution that provides both state of art pedestrian detection and semantic segmentation at 8 FPS on CPU and 15 FPS on GPU using 640×480 pixel images. The main contributions in this work are:

- design of computationally efficient multiresolution filtered channels
- fast multiscale detection scheme based on a single classifier model, a single feature scale and adaptive classification feature sampling
- semantic segmentation by classifying pixels using short range and long range features over multiresolution filtered channels
- pedestrian detection using multiresolution channels and semantic channels

2. Related work

Extensive work has been carried out during the last decades regarding pedestrian detection. The state of art is rapidly improving and each year multiple approaches appear that outperform the previous state of art. Without a doubt, the availability of challenging detection benchmarks, such as Caltech-USA [18], KITTI [21], KAIST [26] has a significant impact on this increase.

There are several great surveys that can be considered for a detailed overview [6], [18] on the state of art. We mention here only some of the main approaches related to our work and focus on sliding window based detection from monocular images. Dalal *et al*. proposed the HOG descriptor in [13] which became one of the most used descriptors in pedestrian and object detection for more than 10 years. HOG descriptors are mostly used together with LUV color features in the form of image channels proposed by Dollar *et al*. in [17].

Over the years several multiscale detection schemes have been considered in order to achieve multiscale detection. The integral channel feature based approach [17] used a single classifier model for a fixed size sliding window and resized the image multiple times. The features were recomputed for each individual image scale. The Fastest Pedestrian Detector in the West [16] computed the image features only for half-octave scales and used approximations for the intermediate ones in order to reduce the computational costs. The VeryFast approach [4] achieved pedestrian detection at over 50 FPS on GPU using a single image feature scale and half-octave pedestrian models that relied on feature approximations for the intermediate scales. In a previous work we proposed a solution based on a single classifier model and a single feature scale using Word Channel features [11]. In [12] we proposed a solution based on 8 classifier models and channel features computed at 3 half-octave scales enabling competitive detection at over 100 FPS on CPU (over 20 FPS on mobile devices). Traditional integral channel features were further improved by proposing aggregate channel features (ACF) [14], informed haar features [52], locally decorrelated channel features [34] and checkerboard filtered channels [53], all being based on the same 10 LUV+HOG channels.

Other works focused on the introduction of additional features such as: LBP [35], [47] color from different color spaces [25], bag of words [11], covariance [45], [35]. Improved results have also been obtained by using additional information such as optical flow [36]. Part based approaches were considered in [19], [3]. Strong performances have been achieved recently using deep learning techniques [43], [24].

Another active research area is the one regarding semantic segmentation. One of the baseline approaches is the Texton-boost approach proposed by Shotton *et al*. [40]. Texton features, visual codebook based texture features, were used to generate texton channels. Individual pixels were classified using boosting over rectangular sums from different texton channels. The classification results were integrated as unary potential into a Conditional Random Field (CRF). Pairwise smoothness potentials were used to refine final segmentation. More complex CRFs have also been considered by using higher order P^n Potts models [27] and robust P^n Potts models [28] , hierarchical pixel and segment based CRF [38], global potentials based on co-occurrence statistics [30] or intra-class spatial relationships [23], and dense CRFs [29]. The computational cost for state of art CRF based approaches is dominated by the computation of unary potentials [37]. In this work we propose a fast solution for computing robust pixelwise unary potentials, that can be integrated into any CRF solution.

Non-parametric semantic segmentation approaches represent an alternative to the previously described parametric approaches [41], [44], [49], [39]. These approaches retrieve visually similar images from large databases, use label-transfer techniques for predicting class-labels and are more practical for dynamically changing large datasets with high number of semantic classes.

3. Multiresolution Channels

The 10 LUV+HOG image channels have served as baseline for several top performing approaches. Most of them used rectangular sums over these channels or variations of these channels in order to obtain classification features. Zhang *et al*. observed in [53] that these approaches can be generalized by adding a filtering layer to the feature generation process. Each approach was characterized by a different set of convolution kernels. This way the rectangular features become simple pixel lookups and there is no need for integral images. In [53] the best performance was achieved using 61 checkerboard kernels resulting in 610 image channels. Unfortunately, the convolution with a large set of kernels can be time consuming especially due to the large number of memory accesses.

Most of the filter sets that have been used in [53] consist of high pass and low pass filters at multiple scales. The low pass filters have the role to capture features at different scales, while the high pass filters capture different structures such as edges or corners. In order to have a reduced but still relevant set of filters we use a box filter for low pass filters and two edge filters for high pass filters and apply them at 5 different scales. We use a 2×2 pixel aggregation and $N - 1$ smoothings with 3×3 box filters to obtain N different channel scales, and apply simple vertical and horizontal difference kernels over each smoothing to obtain edges at different scales. We choose only two orientations for difference filters, because any edge direction can be described using them. We obtain a total of $N \times 3$ filterings using three different filter kernels as illustrated in figure 2. After computing the initial 10 image channels, we partition each channel into 2×2 pixel cells and compute the average. The convolutions are applied over these smaller resolution channels. For a 640×480 pixel image we obtain 320×240 pixel multiresolution channels.

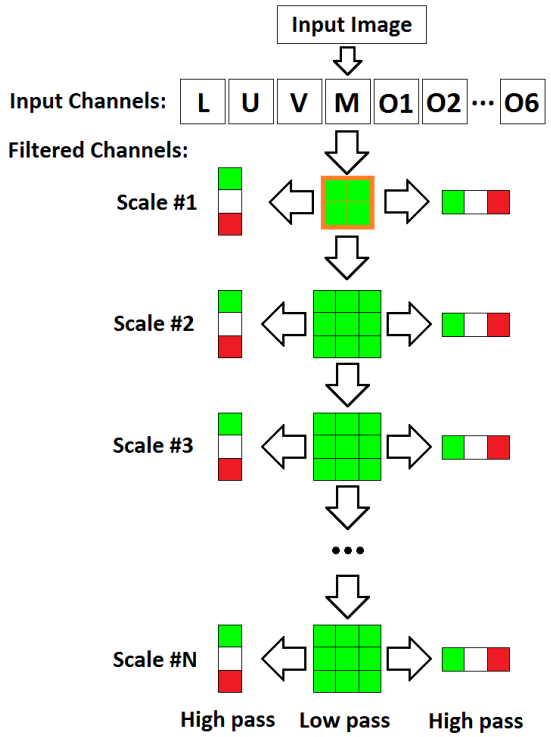

Figure 2. Multiresolution filtering scheme with N scales consisting of N low pass and $2 \times N$ high pass filters over the 10 LUV+HOG. Green and red colors indicate +1 and -1 coefficients. The first 2×2 kernel is an aggregation kernel.

4. Semantic context

Visual codebook based features were the baseline for several semantic segmentation approaches [40], [28], [30], [29]. Local descriptors were computed densely over the input image and were matched to a set of visual words from a codebook that was obtained using clustering over descriptor samples from a training database. Pixels or superpixels were classified based on the distribution of the surrounding visual words. Due to the usual large size of dictionaries and dense feature computation of more complex features, such as SIFT, unary potential estimation dominate computational costs [37] . We achieved full segmentation with 8 classes at 50 FPS in [11] using smaller codebooks, simpler features and a GPU based implementation. In this work we show that the multiresolution channels described in the previous section can be used for even more robust segmentation results at significantly lower computational costs.

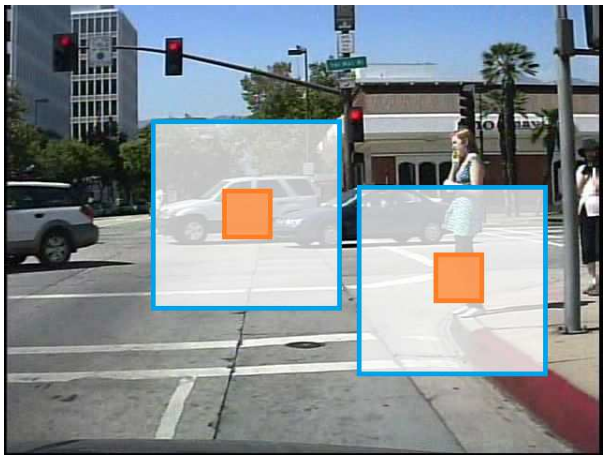

Figure 3. Field of interest for long (blue) and short (orange) features at different pixel locations. The short range features capture local structure, while the long range capture the context.

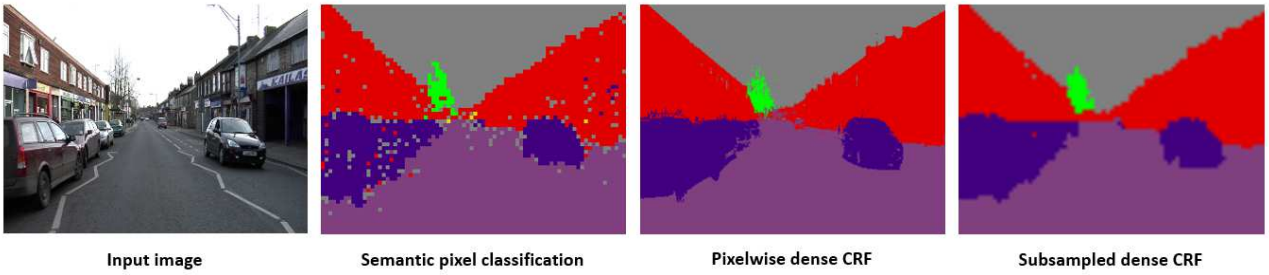

Figure 4. Semantic segmentation refinement using dense CRF

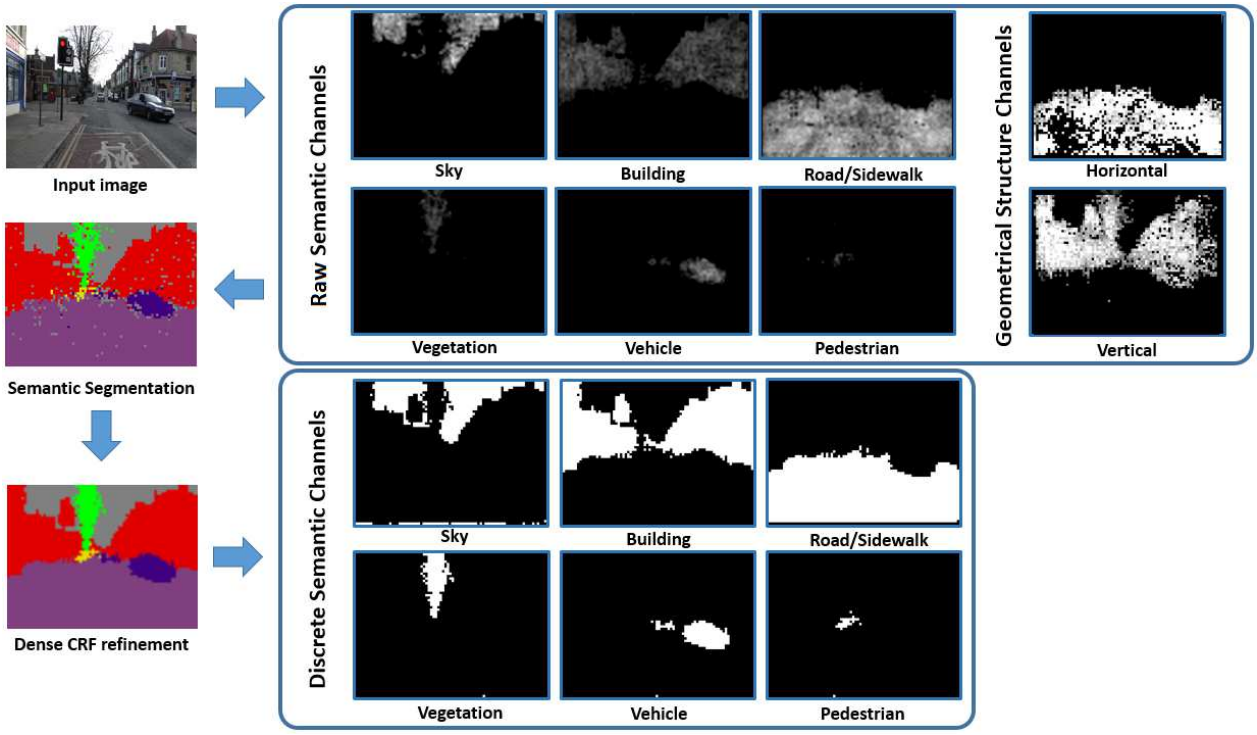

Figure 5. Semantic channels

4.1. Semantic Segmentation

In order to achieve multiclass segmentation we train an individual classifier for each semantic class to classify individual pixels. We use a sampling step rate of 4 pixels resulting in a 16 fold reduction of necessary classifications and show that the segmentation is still of good quality. Working with 320×240 pixel size multiresolution channels as input we need 4800 classifications to obtain full semantic segmentation for a single class.

As classification features for pixels we use the multiresolution channel features around them. We define two types of features sampled in a gridwise manner: short range and long range features (see figure 3). The short range features have the role to capture local structures and are sampled pixelwisely over a grid of 25×25 pixels around the superpixel center. The long range features have the role of capturing the context and are sampled again over a 25×25 grid but with a step rate of 4 pixels between grid points, as seen in figure 3. The two grids result in 625 long range and 625 short range features from each filtered channel that describe a 200×200 and 50×50 pixel region in the original image.

The used classification scheme for pixels is very similar to the one used for classifying sliding windows for pedestrians. For each class we train a binary boosting based classifier using 5 level decision trees. We use 6 bootstrapping rounds in order to train classifiers with 64, 128, 192, 256, 320 and a final one with 384 decision trees. Contrary to pedestrian detection datasets, semantic segmentation training datasets can result in a large number of possible positive

training samples, even of the order of millions. However, many training samples are almost identical and thus, redundant. To solve this issue, we use an initial training set with 5000 random positive samples and add 5000 hard positive samples after each boosting round. We do similarly with negative samples. For an accelerated classification we employ threshold based soft-cascading, used by most pedestrian detectors.

In order to provide semantic context for pedestrian detectors we train classifiers for the six most relevant semantic classes: sky, building, road/sidewalk, tree, vehicle and pedestrian. We train two classifiers also for horizontal and vertical structures, proposed in [22], that have the role to find foreground objects and their support regions. It is very important to have a consistent manually labeled training dataset that covers as many traffic scenarios as possible under different lightning and weather conditions at different times of the day. An ideal dataset would be the CityScapes database [9] that contains 5000 images with high quality pixelwise annotation for 25 semantic classes. Unfortunately the dataset is still under development and is not yet released, but will be available by the end of 2015. For our experiments we combine three different semantic segmentation datasets that cover urban traffic scenarios. We use 701 images from CamVid [7] , 552 from SiftFlow [32] (only highway and street images), and 107 from KITTI [48] datasets.

Example segmentations are illustrated in figure 4. Individual classification of pixels can result in noisy or inconsistent predictions. Several Conditional Random Fields (CRFs) have been defined for improving semantic segmentations. Outstanding results were achieved by Krahenbuhl and Koltun using dense CRFs [29]. Dense CRFs are defined over uniform 2D grids. Figure 4 shows the segmentation result after applying only 3 rounds of dense CRF iterations over each pixel and each 4th pixel. The best result is obtained using pixelwise CRF, however we prefer CRF defined over sparser grid-wise sampled pixels, considering the still relevant segmentation at 16 times lower computational costs (subsampling with a step rate of 4 pixels).

4.2. Semantic Channels

After training all classifiers, a classification cost can be determined for each pixel for each semantic class. Classification cost images are shown in figure 5 for each class. We intend to integrate the semantic context for pedestrian detection as semantic channels next to the multiresolution channels. We consider two alternatives:

- raw semantic channels: using classification cost values
- CRF semantic channels: using discrete predictions from dense CRF inference

An advantage of the pedestrian channel is, that it provides an additional pixel based detection scheme for pedestrians. Even if it can not be used as a full pedestrian detector, considering that it represents only a very small fraction of training samples, it can still recognize specific parts of pedestrians and indicate their potential presence. Another advantage is the full scale invariance of semantic channels which is useful for the multiscale detection scheme that we describe in the following section.

5. Multiscale Detection

Several multiscale schemes have been presented in section 2. Traditional approaches used a single classifier for a fixed size pedestrian model and relied on the recomputation or approximation of image features at multiple scales. We showed in [11], that it is possible to robustly detect pedestrians using a single classifier for variable pedestrian sizes and a single image scale, based on Word Channel features, achieving detection at 16 FPS on GPU. In this work we show that detection with a single classifier and single image scale can be also achieved with multiresolution LUV+HOG channels, which are computationally much simpler than codebook based Word Channels.

To detect pedestrians at multiple scales, we compute the filtered channels for the original scale and apply sliding windows at multiple scales using a scale factor of 1.07 (approximately 10 scales per half octave). We extract classification features by sampling from the filtered channels in a gridwise manner. The grid is adapted to the size of the detection window. This way, the same number of features is obtained for a pedestrian of any size, using different grid spacings. In the case of classifier training, the image features are computed only at the original scale and the pedestrian images are not resized. Feature sampling is illustrated in figure 6.

Based on the described classification features a single

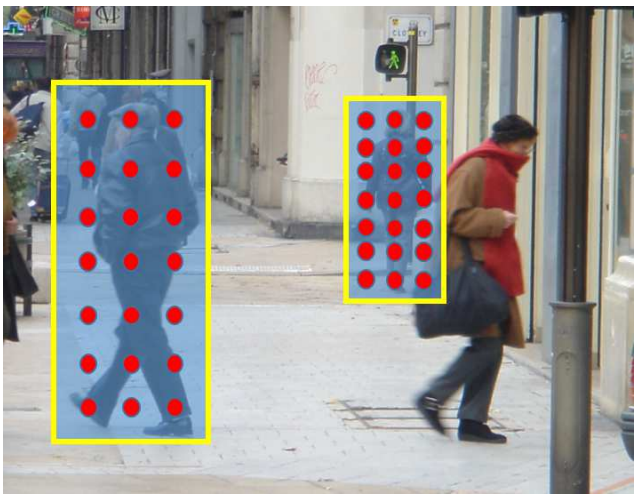

Figure 6. Multiscale detection. Grid of sampled features is adapted to the pedestrian window.

Table 1. CamVid segmentation benchmark results

	Execution time (seconds)	Global	Average	Building	Tree	Sky	Car	Sign-Symbol	Road	Pedestrian	Fence	Column Pole	Sidewalk	Bicyclist
Brostow et al. [8]	1	69.1	53.0	46.2	61.9	89.7	68.6	42.9	89.5	53.6	46.6	0.7	60.5	22.5
Costea et al. [10]	0.027	76.0	52.0	66.0	81.0	84.0	71.0	2.0	94.0	50.0	25.0	20.0	60.0	13.0
Sturgess et al. [42]	35	83.8	59.2	84.5	72.6	97.5	72.7	34.1	95.3	34.2	45.7	8.1	77.6	28.5
Zhang et al. [51]	-	82.1	55.4	85.3	57.3	95.4	69.2	46.5	98.5	23.8	44.3	22.0	38.1	28.7
Floros et al. [20]	22	83.2	59.6	80.4	76.1	96.1	86.7	20.4	95.1	47.1	47.3	8.3	79.1	19.5
Ladicky et al. [31]	-	83.8	62.5	81.5	76.6	96.2	78.7	40.2	93.9	43.0	47.6	14.3	81.5	33.9
Tighe et al. [44]	210	83.9	62.5	83.1	73.5	94.6	78.1	48.0	96.0	58.6	32.8	5.3	71.2	45.9
Ours – subsampled	0.05	82.4	55.9	84.6	78.6	94.5	71.3	3.7	97.8	58.1	38.5	14.8	41.9	31.8
Ours – pixelwise	0.32	83.0	55.5	86.4	79.8	96.1	73.0	3.2	97.8	56.7	39.4	10.4	39.2	28.8

real-boost classifier is learned using 5 level decision trees and 5 bootstrapping rounds. We use 32, 512, 1024, 2048 and 4096 5-level decision trees during these rounds. We start with 10000 random negative samples and add 10000 hard negative samples after each bootstrapping round. A similar setup was used also in [53]. For the sliding window we use the following adaptive step rates: 1/16 of the window width horizontally and 1/16 of the window height vertically. The search space is reduced by 35 % by conditioning the window centers to be between the rows 140 and 300 (valid for over 99 % percent of the pedestrians in the Caltech database [46]).

The filtered channel features are not scale invariant and pedestrians at different scales will have different representations. The pedestrians have significantly different representations also for different illumination, orientation or occlusions cases, and a single boosting classifier consisting of thousands of weak learners is still able to provide consistent results. Our intuition is that a powerful boosting classifier together with a large training dataset with pedestrians at various sizes is able to learn the relevant classification features independently for the different pedestrian representations.

6. Experimental results

In the following we evaluate the semantic segmentation and the pedestrian detection component of the proposed solution. In the case of pedestrian detection we also show the impact on detection performance when using semantic channels.

6.1. Semantic segmentation evaluation

Considering that we focus on traffic scenarios we evaluate the proposed semantic segmentation approach on the CamVid benchmark [7]. It is currently the largest traffic scene dataset with high quality pixelwise annotations for 32 semantic classes and consists of color video sequences captured by a camera mounted on a car. For evaluation we train an individual classifier for the classes Building, Tree, Sky, Car, Sign-Symbol, Road, Pedestrian, Fence, Column-Pole, Sidewalk and Bicyclist. We evaluate our semantic segmentation approach using pixelwise and sub-sampled dense CRF. For the sub-sampled dense CRF we use a step rate of 4 pixels for rows and columns.

Table 1 provides the classification accuracy for each individual class, average accuracy for all classes, global accuracy and execution time. We also provide a comparison with several state of art approaches [8] [10] [42] [51] [20] [31] [44] and show that our results are competitive at significantly lower computational costs. The low accuracy for *sign* class is due to the small number of training samples (only 0.07 % of the training data) and can be solved using class weight balancing. The most confused classes were *road* and *sidewalk*.

6.2. Pedestrian detection evaluation

We use the Caltech-USA pedestrian detection benchmark [18] for evaluating the performance of the proposed pedestrian detector. It is one of the mostly used pedestrian benchmarks and enables comparison between more then 50 state of art approaches. We use the extended training dataset for pedestrians by using each 3rd frame and the corresponding pedestrian annotations from the training videos sequences (the standard training set uses each 30th).

Training the 8 semantic pixel classifiers took around one hour using 24 Intel Xeon X5570 CPUs, part of the UTC-N GRID Center (POS CCE nr. 195) computing grid. The final pedestrian classifier was trained in less than an hour on the same grid.

As main detection performance metric we use the log-average miss rate for $[10^{-2}, 10^{0}]$ false positives per image (FPPI) precision range, which is the standard evaluation metric on the Caltech benchmark. As testing setup we use

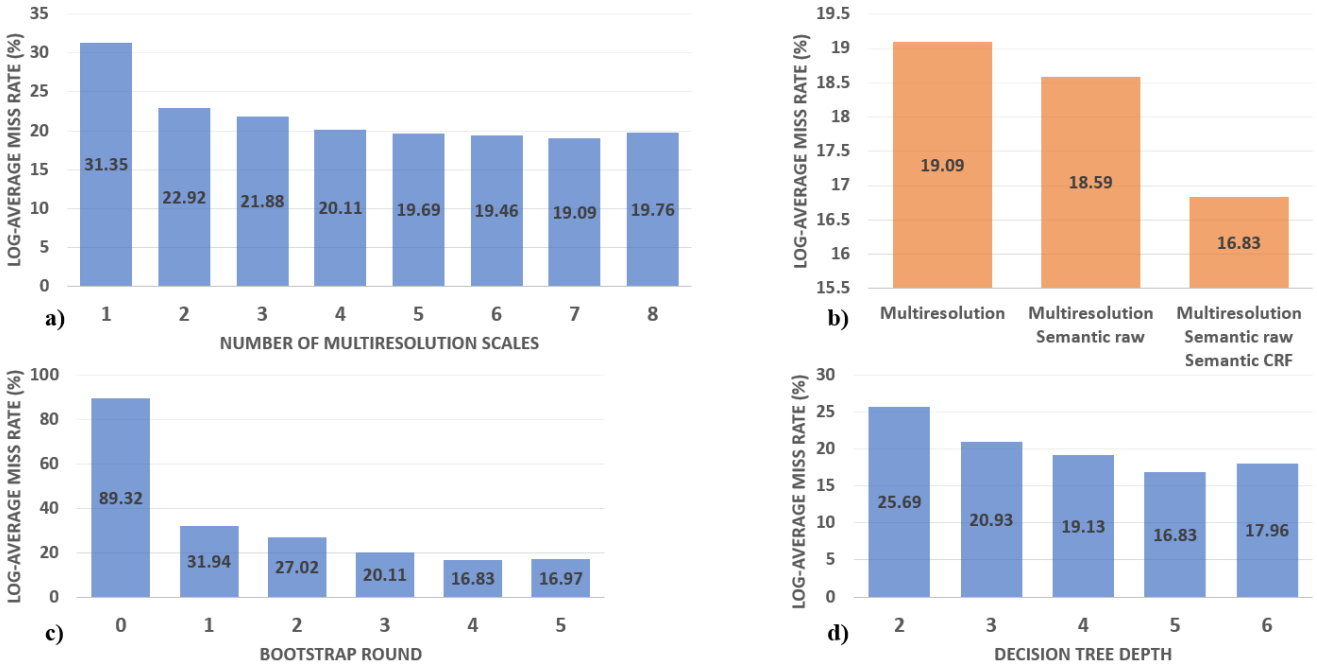

Figure 7. Detection performance on Caltech - reasonable test setup using different a) number of multiresolution scales, b) classification feature types, c) bootstrapping rounds and d) decision tree depths.

the *reasonable* setup. First we evaluate the detection performance using only the multiresolution filtered channels with different number of scales. As seen in figure 7a, the best performance is achieved using 7 scales (210 multiresolution filtered channels). In figure 7b we show the effect of adding raw semantic channels and semantic channels obtained from dense CRF inference. Figure 7c shows the performance of the boosting classifier after each bootrapping round. Using a 5th bootsrapping round (also with 4096 weak learners) provided worse results, most probably due to overfiting. A similar effect was observed also when using deeper decision trees (7d). In all our experiments we used a 20×10 grid for sampling classification features for detection windows, resulting in 200 features for each individual channel. Denser grids did not provide performance improvements. Figure 8 provides a comparison, based on ROC curves, of the proposed solution with the current top approaches [53] [43] [50] [35] [6] [24] [34] [2]. The approaches are ordered by log-average miss rates. The best performance is achieved using multiresolution channels together with semantic channels (raw and CRF).

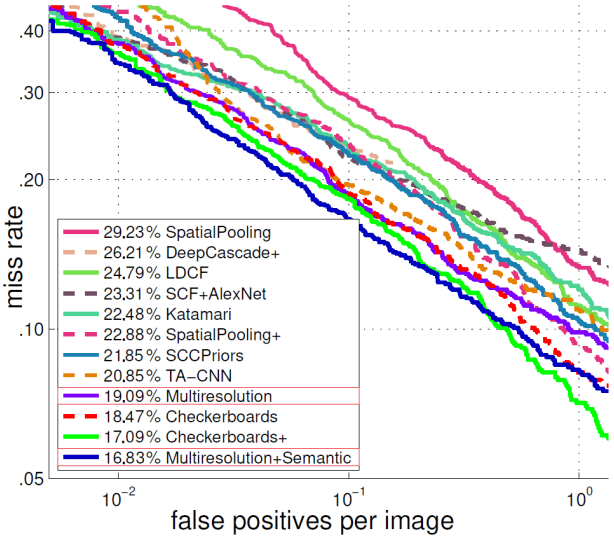

Figure 8. Benchmark results: Caltech - reasonable

6.3. Computational costs

Table 2 provides an overview of execution times and the achieved log-average miss rates on the Caltech - *reasonable* test setup for approaches that provided details regarding computational costs [16] [17] [15] [5] [14] [11] [33] [1] [52] [35] [2]. The proposed solution provides competitive results at significantly lower computational costs. In the following we provide the average execution times for different steps of the solution using GPU (Nvidia GTX 980 Ti) / CPU (Intel Core i7 3.0 GHz) implementation.

- 210 filtered channel computation: 2 ms / 21 ms
- 8 semmantic channel prediction: 22 ms / 45 ms
- dense CRF inference: - / 28 ms
- sliding window classifications: 14 ms / 29 ms

Table 2. Miss rate vs. frame rate

Approach	Miss rate	FPS
FPDW	57.40%	2.6
ChnFtrs	56.34%	0.2
CrossTalk	53.88%	14
Roerei	46.13%	1
ACF-Caltech	44.22%	30
WordChannels	42.30%	16 (GPU)
SDN	37.87%	0.67
FastCF	37.33%	105
LFOV	35.85%	3.6
SquaresChnFtrs	34.81%	1
InformedHaar	34.60%	0.63
SpatialPooling	29.23%	0.13
DeepCascade+	26.21%	15 (GPU)
Ours – Multiresolution	19.09%	20 60 (GPU)
Ours – Multiresolution & Semantic Channels	16.83%	8 15 (GPU)

The pedestrian detection for a 640 x 480 pixel image is achieved at an average rate of:

- 60 FPS on GPU and 20 FPS on CPU with 210 filtered channels
- 15 FPS on GPU and 8 FPS on CPU also with semantic channels

7. Conclusion

The main goal of this work was to provide a tool that can be used for visual perception in real-time applications and that can keep up with the robustness of current state of art approaches. We proposed a solution for pedestrian detection for validation purposes, however the approach can be also used for the detection of other object or obstacle types. The semantic segmentation is also an important visual cue and can help for a better higher-level understanding of the environment.

In this work we propose multiresolution channels for detection and semantic segmentation obtained from a computationally efficient filtering scheme. For fast detection we use a multiscale detection based on a single classifier model, a single features scale and adaptive classification features sampling. To obtain a more powerful detector, we integrate semantic segmentation as raw and CRF semantic channels next to the multiresolution channels. We focused also on keeping computational costs low and achieved a detection rate of 8 FPS on CPU and 15 FPS on GPU.

Acknowledgment This work has been partially supported by SmartCoDrive project (PNII-PCCA 18/2012) and MULTISENS project (PNII-ID-PCE-2011-3-1086), funded by the Romanian Ministry of Education and Research, UEFISCDI. We would like to thank Pusztai Kalman Communication Center the support in running our experiments.

References

[1] A. Angelova, A. Krizhevsky, and V. Vanhoucke. Pedestrian detection with a large-field-of-view deep network. In *ICRA*, 2015.

[2] A. Angelova, A. Krizhevsky, V. Vanhoucke, A. Ogale, and D. Ferguson. Real-time pedestrian detection with deep network cascades. In *BMVC*, 2015.

[3] A. Bar-Hillel, D. Levi, E. Krupka, and C. Goldberg. Part-based feature synthesis for human detection. In *ECCV*, 2010.

[4] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool. Pedestrian detection at 100 frames per second. In *CVPR*, 2012.

[5] R. Benenson, M. Mathias, T. Tuytelaars, and L. Gool. Seeking the strongest rigid detector. In *CVPR*, 2013.

[6] R. Benenson, M. Omran, J. Hosang, and B. Schiele. Ten years of pedestrian detection, what have we learned? In *ECCV*, 2014.

[7] G. J. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 2009.

[8] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV*, 2008.

[9] M. Cordts, M. Omran, S. Ramos, T. Scharwächter, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset. In *CVPR Workshop*, 2015.

[10] A. D. Costea and S. Nedevschi. Multi-class segmentation for traffic scenarios at over 50 fps. In *IVS*, 2014.

[11] A. D. Costea and S. Nedevschi. Word channel based multiscale pedestrian detection without image resizing and using only one classifier. In *CVPR*, 2014.

[12] A. D. Costea, A. V. Vesa, and S. Nedevschi. Fast pedestrian detection for mobile devices. In *ITSC*, 2015.

[13] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[14] P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *PAMI*, 2014.

[15] P. Dollár, R. Appel, W. Kienzle, and D. Ferguson. Crosstalk cascades for frame-rate pedestrian detection. In *ECCV*, 2012.

[16] P. Dollár, S. Belongie, and P. Perona. The fastest pedestrian detector in the west. In *BMVC*, 2010.

[17] P. Dollár, Z. Tu, P. Perona, and S. Belongie. Integral channel features. In *BMVC*, 2009.

[18] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *PAMI*, 2012.

[19] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 2010.

[20] G. Floros, K. Rematas, and B. Leibe. Multi-class image labeling with top-down segmentation and generalized robust p^n potentials. In *BMVC*, 2011.

[21] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *IJRS*, 2013.

[22] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *ICCV*, 2009.

[23] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller. Multi-class segmentation with relative location prior. *IJCV*, 2008.

[24] J. Hosang, M. Omran, R. Benenson, and B. Schiele. Taking a deeper look at pedestrians. *CVPR*, 2015.

[25] R. Khan, J. Van de Weijer, F. Shahbaz Khan, D. Muselet, C. Ducottet, and C. Barat. Discriminative color descriptors. In *CVPR*, 2013.

[26] S. H. J. P. N. Kim and Y. C. I. S. Kweon. Multispectral pedestrian detection: Benchmark dataset and baseline. In *CVPR*, 2015.

[27] P. Kohli, M. P. Kumar, and P. H. Torr. P3 & beyond: Solving energies with higher order cliques. In *CVPR*, 2007.

[28] P. Kohli, P. H. Torr, et al. Robust higher order potentials for enforcing label consistency. *IJCV*, 2009.

[29] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2012.

[30] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr. Graph cut based inference with co-occurrence statistics. In *ECCV*, 2010.

[31] L. Ladickỳ, P. Sturgess, K. Alahari, C. Russell, and P. H. Torr. What, where and how many? combining object detectors and crfs. In *ECCV*, 2010.

[32] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *PAMI*, 2011.

[33] P. Luo, Y. Tian, X. Wang, and X. Tang. Switchable deep network for pedestrian detection. In *CVPR*, 2014.

[34] W. Nam, P. Dollár, and J. H. Han. Local decorrelation for improved pedestrian detection. In *NIPS*, 2014.

[35] S. Paisitkriangkrai, C. Shen, and A. van den Hengel. Strengthening the effectiveness of pedestrian detection with spatially pooled features. In *ECCV*, 2014.

[36] D. Park, C. L. Zitnick, D. Ramanan, and P. Dollár. Exploring weak stabilization for motion feature extraction. In *CVPR*, 2013.

[37] G. Roig, X. Boix, R. De Nijs, S. Ramos, K. Kuhnlenz, and L. Van Gool. Active map inference in crfs for efficient semantic segmentation. In *ICCV*, 2013.

[38] C. Russell, P. Kohli, P. H. Torr, et al. Associative hierarchical crfs for object class image segmentation. In *CVPR*, 2009.

[39] A. Sharma, O. Tuzel, and D. W. Jacobs. Deep hierarchical parsing for semantic segmentation. In *CVPR*, 2015.

[40] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*, 2009.

[41] G. Singh and J. Kosecka. Nonparametric scene parsing with adaptive feature relevance and semantic context. In *CVPR*, 2013.

[42] P. Sturgess, K. Alahari, L. Ladicky, and P. H. Torr. Combining appearance and structure from motion features for road scene understanding. In *BMVC*, 2009.

[43] Y. Tian, P. Luo, X. Wang, and X. Tang. Pedestrian detection aided by deep learning semantic tasks. *CVPR*, 2015.

[44] J. Tighe and S. Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In *CVPR*, 2013.

[45] O. Tuzel, F. Porikli, and P. Meer. Pedestrian detection via classification on riemannian manifolds. *PAMI*, 2008.

[46] R. Varga, A. V. Vesa, P. Jeong, and S. Nedevschi. Real-time pedestrian detection in urban scenarios. In *ICCP*, 2014.

[47] X. Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for generic object detection. In *ICCV*, 2013.

[48] P. Xu, F. Davoine, and T. Denœux. Evidential combination of pedestrian detectors. In *BMVC*, 2014.

[49] J. Yang, B. Price, S. Cohen, and M.-H. Yang. Context driven scene parsing with attention to rare classes. In *CVPR*, 2014.

[50] Y. Yang, Z. Wang, and F. Wu. Exploring prior knowledge for pedestrian detection. In *BMVC*, 2015.

[51] C. Zhang, L. Wang, and R. Yang. Semantic segmentation of urban scenes using dense depth maps. In *ECCV*, 2010.

[52] S. Zhang, C. Bauckhage, and A. Cremers. Informed haar-like features improve pedestrian detection. In *CVPR*, 2014.

[53] S. Zhang, R. Benenson, and B. Schiele. Filtered channel features for pedestrian detection. In *CVPR*, 2015.