Semantic Road Segmentation via Multi-scale Ensembles of Learned Features

Jose M. Alvarez[1,3,⋆], Yann LeCun[1], Theo Gevers[2,3], and Antonio M. Lopez[3]

[1] Courant Institute of Mathematical Sciences, New York University, New York, NY
[2] Faculty of Science, University of Amsterdam, Amsterdam, The Netherlands
[3] Computer Vision Center, Univ. Autònoma de Barcelona, Barcelona, Spain

Abstract. Semantic segmentation refers to the process of assigning an object label (e.g., building, road, sidewalk, car, pedestrian) to every pixel in an image. Common approaches formulate the task as a random field labeling problem modeling the interactions between labels by combining local and contextual features such as color, depth, edges, SIFT or HoG. These models are trained to maximize the likelihood of the correct classification given a training set. However, these approaches rely on hand–designed features (e.g., texture, SIFT or HoG) and a higher computational time required in the inference process.

Therefore, in this paper, we focus on estimating the unary potentials of a conditional random field via ensembles of learned features. We propose an algorithm based on convolutional neural networks to learn local features from training data at different scales and resolutions. Then, diversification between these features is exploited using a weighted linear combination. Experiments on a publicly available database show the effectiveness of the proposed method to perform semantic road scene segmentation in still images. The algorithm outperforms appearance based methods and its performance is similar compared to state–of–the–art methods using other sources of information such as depth, motion or stereo.

1 Introduction

Road scene understanding from a mobile platform is a central task for vehicle environment perception. This process is the key to success in autonomous driving and driver assistance systems such as vehicle and pedestrian detection. Understanding road scenes involves comprehending the scene structure (e.g., sidewalks, buildings, trees, roads), scene status (i.e., traffic situations) or understanding the motion patterns of other objects present in the scene. A core component of road scene understanding systems is its semantic segmentation [1,2]. Semantic segmentation is the process of partitioning an image into disjoint regions and the interpretation of each region for semantic meanings (Fig. 1). Semantic segmentation provides important information to support higher level scene interpretation tasks. Therefore, in this paper, we focus on the semantic segmentation of road images.

⋆ This work was partially supported by Spanish Government under Research Program Consolider Ingenio 2010: MIPRCV (CSD200700018) and MINECO Projects TRA2011-29454-C03-01, TIN2011-25606 and TIN2011-29494-C03-02.

A. Fusiello et al. (Eds.): ECCV 2012 Ws/Demos, Part II, LNCS 7584, pp. 586–595, 2012.

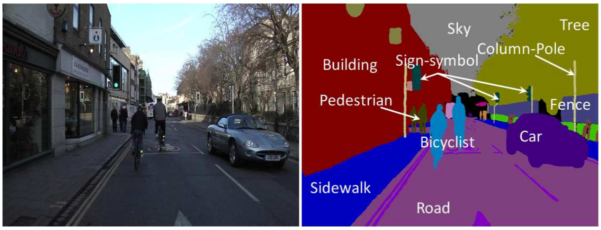

Fig. 1. Semantic scene segmentation aims at assigning every pixel in an image by one of the predefined semantic labels available (i.e., pedestrian, car, building, road, sidewalk, tree, sky). Image taken from [3].

Common semantic scene segmentation approaches formulate the problem as a random field labeling problem and model the dependencies of labels of pairs of variables by combining different types of features such as color, texture, depth, edges among others [4,5]. Then, a model describing these interactions is built and trained to maximize the likelihood of the correct classification. For instance, Gupta *et al.* [6] include the 3D geometry of the scene to improve the segmentation task by discarding physically implausible relations between segments. Floros *et al.* [5] include top–down segmentations from a densely sampled part–based detector. Other approaches include other sources of information such as structure from motion [2] or stereo disparity [4]. However, these conditional random field models have two main limitations. First, their dependency on hand–designed features that may not be appropriated for the specific task. Second, these approaches tend to be costly in terms of inference since inference requires searching over different label configurations.

In this paper, we focus on multi–scale learning features for road detection. Feature learning has received a lot of attention recently. For instance, a multi–scale end to end learning algorithm is proposed in [7]. In that approach, the authors train a Convolutional Neural Network in two steps. First, features are extracted at different scales and their output is concatenated to generate a feature vector. Then, in a second stage these features are trained to learn predictions of different classes. The approach shows promising results in different databases. However, the training stage is complex and also involve large (intermediate) feature vectors. Therefore, we propose a different approach to obtaining pixel potentials to represent the unary potentials of a conditional random field. The core of the algorithm is a Convolutional Neural Network trained to extract local features exploiting the 2D structures present in an image. In addition, the algorithm includes contextual information by extracting features at different visual scales (i.e., the larger scale, the smaller area of the image is occupied by the object being analyzed). Finally, robustness to scale variations is achieved by extracting these features at multiple resolutions. Then, all these features are considered as weak features and combined into a CRF as unary potentials. The ensemble of features is learned using global optimization to exploit inter–feature diversification. Different experiments conducted on a publicly available database show the effectiveness of the proposed method to perform semantic road scene segmentation.

The rest of this paper is organized as follows. First, in Sect. 2, we introduce conditional random fields for image segmentation. Convolutional neural networks for feature extraction are introduced in Sect. 3 and the algorithm to compute ensembles of multi–scale features is detailed in Sect. 3.1. Then, in Sect. 4, experiments are presented and the results are discussed. Finally, conclusions are drawn in Sect. 5.

2 Conditional Random Fields for Image Segmentation

Image segmentation consists of partitioning an image into several disjoint regions that show homogeneity to certain features such as color, texture, edges. This process is usually formulated as a random field labeling problem to aggregate local cues such as color, texture along with contextual cues describing the possible spatial interactions between labels [4,5]. To this end, an image is represented with a graph structure $\mathscr{G} = <\mathscr{V}, \mathscr{E}>$ where $\mathscr{V}$ is a set of N random variables $\mathbf{Y} = Y_1, \ldots, Y_N$ representing the nodes of the graph (i.e., $|\mathscr{V}| = N$) and corresponding to the pixels in the image. Each of these variables is allowed to take values from a discrete domain of labels $\mathscr{L} = l_1, , \ldots, l_K$. Furthermore, $\mathscr{E}$ is the set of edges modeling the relationships between neighboring pixels. Then, image segmentation is done by assigning every pixel in the image $x_i \in \mathscr{V}$ a meaningful label $l_i \in \mathscr{L}$. Finally, let $y = \{Y_i \in \mathscr{V}\}$ a label assignment with values in $\mathscr{L}$. Then, we consider a Conditional Random Field (CRF) to model the Gibbs energy as follows:

$$E(y) = \sum_{i \in \mathscr{V}} \psi_i(l_i, x_i) + \sum_{(i,j) \in \mathscr{E}} \psi_{ij}(l_i, l_j), \tag{1}$$

where $\psi_i(l_i, x_i)$ is the unary potential modeling the likelihood of a pixel taking a certain label, and $\psi_{ij}(l_i, l_j)$ is the pairwise potential modeling the coherence of neighboring pixels taking the same label. Then, the most probable label assignment $\hat{y}$ is obtained by minimizing the Gibbs energy on the graph structure: $\hat{y} = \arg\min_y E(y)$.

Conventionally, the unary potential is computed using features in an image such as color, texture, shape or hand–designed features such as SIFT or HoG. Then, the model is build and trained to maximize the likelihood of the correct classification. In the next section, we introduce the use of convolutional neural networks to extract specific features representing each possible label.

3 Feature Learning via Convolutional Neural Networks

In this paper we focus on learning/extracting features from training images using convolutional neural networks (CNN). CNNs are hierarchical architectures widely used for object detection and recognition [8] that alternate different type of layers (e.g., convolution, sub–sampling) to extract and combine visual patterns presents in the input data [9]. An example of this type of architectures is shown in Fig. 2. This architecture can be interpreted as a set of filter banks divided in three different layers and a set of connections to fuse them. The kernels of these filters and the connection weights are learned off–line using training data [9].

Based on this learning architecture we extract features at two different visual scales: fine and coarse. These two scales (levels from now on) consider different amount of

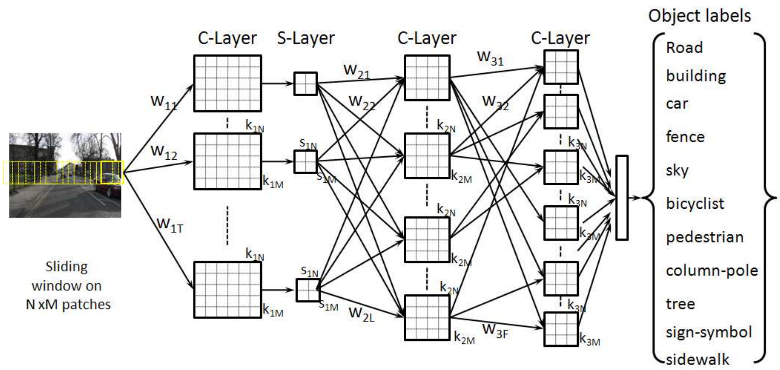

Fig. 2. Convolutional neural networks alternate layers of convolution (C–layers) and sub–sampling layers (S–layers) to learn high–order local features directly from training data. Connection weights are given by w_{ij} (i–th layer, j–th kernel), the size of convolutional kernels is $k_{iN} \times k_{iM}$ and the subsampling size for the S–layer is $s_{iN} \times s_{iM}$.

contextual information by varying the size of the input patch. In particular, we consider patch sizes of 32×32 and 64×64 for the fine and coarse levels respectively. Moreover, robustness to scale variations is improved by extracting features using different kernel sizes. In practice, this is done by resizing input images to 4 different scales: 1, $\frac{1}{2}$, $\frac{1}{4}$ and $\frac{1}{5}$. Hence, this stage takes a *RGB* image of size $X \times Y$ as input and extracts fine and coarse features at multiple resolutions by applying a sliding window on patches of size 32×32 and 64×64 respectively. The output is a $K \times X \times Y$ confidence map relating a set of K (i.e., number of labels available) floating point numbers, ranging from 0 to 1, to each pixel in the image to indicate their per–class potential. The higher the potential is, the more likely the pixel belongs to that class.

3.1 Multi–scale Feature Ensembles as Unary Potentials

Unary potentials (e.g., ψ_i in Eq. (1)) model the likelihood of a pixel taking a certain label. These potentials are usually estimated using common features extracted from incoming data (e.g., color, texture, depth, edges). If more than one feature is available, the unary potentials are estimated as a combination of them either using predefined rules (e.g., sum, product, maximum, minimum [10]) or learned weighted combinations. Using fixed rules does not exploit the fact that different objects (classes) have different needs in terms of context and scale information. Therefore, if training sets are available, a more powerful combining approach consists of a weighted combination of features. In this case, the unary potentials can be estimated as follows:

$$\psi_i(l_i, x|\Theta_i) = \sum_{r=1}^{R} w_r \psi_r(l_i, x|\Theta_r), \tag{2}$$

where $\Theta_1, \ldots, \Theta_R$ is the set of R features and w_r is the weight modeling the relative relevance of that feature for the given label. In this section, unary potentials are computed as a weighted combination of multi–scale learned features. To this end, we consider the output of each CNN (different scales and different levels) as a different feature and fuse them using a weighted linear combination. More precisely, we focus on class–dependent weighted linear combination were each feature (scale and resolution) for each class receives a different weight. These class–dependent weights are learned globally (all weights and all classes at the same time) off–line by minimizing the sum of squared errors between the output of the CNN and the target label.

4 Experiments

Experiments validating the proposed algorithm are conducted on the Cambridge-driving Labeled Video Database (CamVid) [3]. CamVid is a publicly available database of high-quality images acquired using a camera mounted on the windshield of a vehicle driving in an urban scenario at different daytime. Thus, these sequences include challenging situations as crowded scenes, different lighting conditions and different road type. Ground–truth is provided as manual annotations at 1fps. These annotations include 32 different classes [3]. However, for fair comparison with other approaches, we use 11 object categories as in [2,11]. We follow the experimental setup in [2,11] by dividing into 367 training and 233 testing images and providing evaluations by down–scaling the images by a factor of 3. Quantitative evaluations are provided using pixel–wise confusion matrices including global and average accuracy. The former is the number of pixels correctly classified over the number of pixels in the testing set. The latter is the number of pixel correctly classified per class divided by the total number of pixels in that class.

For testing purposes we devise a simple road scene segmentation algorithm based on CNN and CRF. The core of the algorithm is a CNN which takes an $N \times M$ image as input and outputs a $K \times N \times M$ confidence map relating a set of K (i.e., number of labels available) floating point numbers indicating the per–class potential of each pixel in the image. The higher the potential is, the more likely the pixel belongs to that class. Then, a single unary potential per class is computed using Eq. (2). Finally, the most probable label assignment is estimated minimizing the energy function in Eq. (1).

The parameters of the algorithm are empirically fixed as follows. First, the algorithm uses two levels (fine and coarse) and four different scales ($R = 8$): 1x, $\frac{1}{2}$x, $\frac{1}{4}$x and $\frac{1}{5}$x resolution. Further, the input layer (RGB data) at each level is sparsely connected to the first convolutional layer: each color plane is connected to two different kernels and then all three color planes are connected to two more kernels. The first connections enforce learning independent preprocessing kernels for each color while the last ones combine them.

4.1 Training and Data Preparation

The training set consists of 367 high–quality images manually labeled. This results in millions of highly correlated training samples that difficult the training process. Hence, to obtain a reasonable overhead we reduce the number of training samples by considering only a subset of training patches for each image. This subset per image is generated using over–segmentation and selecting a representative number of pixels within each region. In this paper, the over–segmented image is obtained using the turbo–pixel approach in [12]. Then, training samples are selected as the centroid of each superpixel. This sampling technique has two main advantages. First, it improves diversity in the training set by reducing the number of samples (i.e., iterative learning algorithms such as back–propagation can explore more samples in less time). Second, the selection of the centroid improves the intra–class variance since the centroid is the best representation of the superpixel area and maximize the distance between samples from two consecutive superpixels. Based on this sampling technique, two non–overlapping training sets are generated using different dilation masks around the centroid of each superpixel (Fig. 3). The first subset is used to train the CNN and, the second one is used to learn the weights. Thus, the weights of the ensemble are learned globally using the approach in Sect. 3.1 using unseen samples. Both training subsets are resampled to improve the balance between classes.

Fig. 3. The number of training samples is sub–sampled to reduce the computational training overhead. The incoming image is over–segmented using superpixels. Then, patches centered at the centroid of each superpixel and several pixels in its surrounding area are selected as training samples.

The layers of the CNN are trained in supervised mode using the pixel labels in the first training subset. To this end, image patches centered in the training set are extracted. Robustness to scale and noisy acquisition conditions is reinforced using jitter in each patch. More precisely, we consider a random scale per sample between $[0.6,\ldots,1.4]$, random Gaussian noise $\sigma = [0.3,\ldots,1.2]$ and random rotations in the range $[-17^\circ,\ldots,17^\circ]$. Given this resampled training set, CNN at each level is trained (weight learning) independently using classical back–propagation. The parameters of the CNN are corrected due to standard stochastic gradient descent by minimizing the sum of square differences between the output of the CNN and the target label. Training is stopped when the error in consecutive epochs does not decrease more than 0.001. Finally, the CRF is trained using the Conditional Random Field toolbox in [13].

The set of weights obtained is listed in Table 1. The larger the weight, the more important the feature. As shown, different weights are obtained for each class. For instance, high weights are given to the 4th–scale of the coarse level for the bicyclist and sign–symbol classes (i.e., more contextual information is needed to predict these classes) while the road ensemble mainly consists of fine scales.

Table 1. Set of weights obtained for the experiments. Weights are computed globally without bounding their possible values.

	Fine Level				Coarse Level			
Scales	1st	2nd	3rd	4th	1st	2nd	3rd	4th
Byciclist	0.023	0.360	0.120	−0.009	0.008	0.608	0.062	−1.0
Building	0.185	0.658	0.244	0.106	0.188	0.388	0.280	−1.0
Car	0.092	0.296	0.220	0.192	−0.007	0.337	0.189	−1.0
Column-pole	0.022	0.132	0.078	−0.020	0.101	0.251	0.021	−0.536
Fence	−0.145	0.344	0.182	0.027	0.058	0.487	0.181	−1.0
Pedestrian	0.020	0.197	0.100	0.032	0.079	0.320	0.092	−0.746
Road	0.050	0.272	0.074	0.109	0.089	0.478	0.3414	−1.0
Sidewalk	−0.040	0.309	0.174	0.051	−0.023	0.546	0.184	−1.0
Sign–symbol	0.060	0.247	0.191	0.048	0.076	0.314	0.040	−0.823
Sky	0.383	0.375	0.361	−0.008	0.026	0.096	0.062	−0.867
Tree	0.063	0.351	0.380	0.087	0.208	0.301	0.047	−1.0

4.2 Results

Representative qualitative results are shown in Fig. 4 and pixel–wise confusion matrices are shown in Fig. 5. These matrices provide per–class classification rates given by the total number of correctly classified pixels divided by the total number of pixels in the training set. For comparison, we provide confusion matrices for three different configurations. First, using a linear combination only at the finner level. Second, the confusion matrix using both levels and learned weights and finally, the confusion matrix using the CRF framework and CNN to estimate the unary potentials. As shown, the best average per–class accuracy is provided by the fusion of weights without the pairwise potentials Fig. 5b. However, using the pairwise potentials reinforce relation between neighboring pixels and improves the global accuracy of the algorithm. In this case, the accuracy of large classes (e.g., road, sidewalk) is improved at the expense of lowering the accuracy of classes with small presence in the dataset (e.g., sign–symbol). These contingency table suggests that miss–classifications are mainly located in small objects such as column-pole and sign-symbol corresponding with those classes with less examples in the training set. Further, the algorithm exhibits lower performance in classifying bicycles, buildings, fences and trees. The former are usually classified as pedestrian due to their similarity (Fig. 1).

The performance of the proposed CNN–CRF algorithm is also compared to several methods in the state–of–the–art. These approaches include different types of features such as appearance [2], motion cues (SfM) [2], their combination [2] , depth [11], semantic texton and superpixels [11] and their combination [11,4] to improve their performance. We also include approaches including object detectors [5,4] since they provide the highest accuracy within the state–of–the–art. Moreover, for comprehensive evaluation, we include five different instances of our algorithm. First, the complete CNN–CRF using all the scales and features. Then, two different multi–scale instances excluding the CRF: using both levels (MultiScale CNN–no CRF) and using only the fine level (CNN–MR Fine). Finally, an instance where histogram of superpixel labels is used to reinforce the spatial consistency of object labels (CNN–superpixels) is also included. We also include a CNN approach based on a single scale at the finer level (Fine Level). This configuration outputs the maximum response over each class,

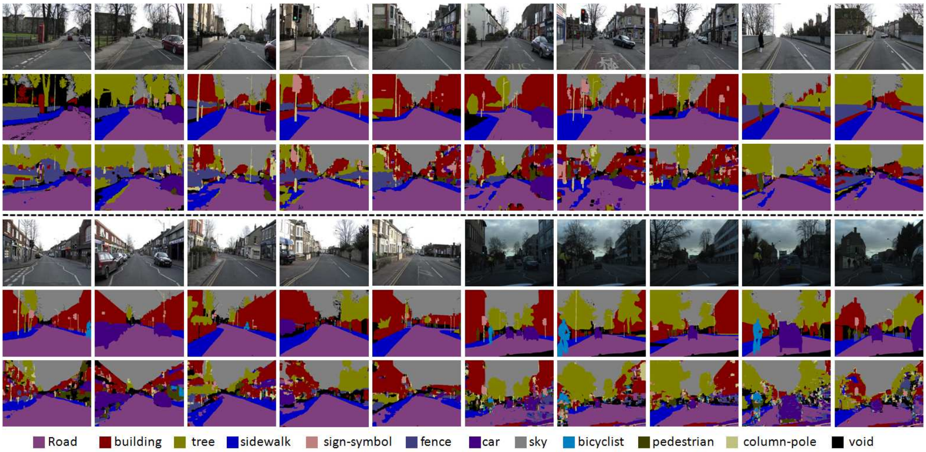

Fig. 4. Qualitative semantic road segmentation results. First and fourth rows: input image. Second and fifth rows: manually annotated labels. Third and sixth rows: results of our algorithm.

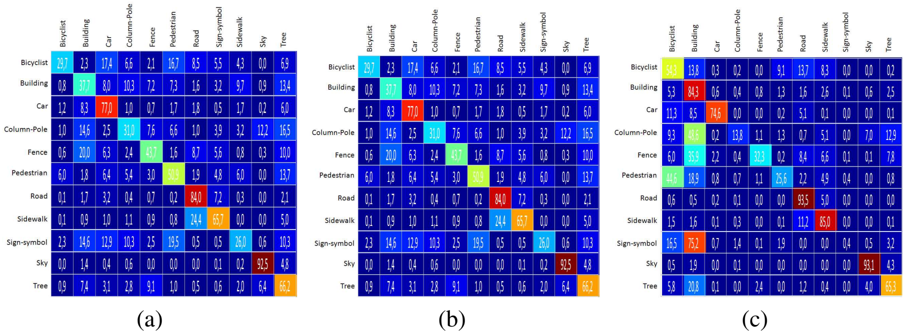

Fig. 5. Confusion matrix over the CamVid test set. a) Using directly the ensemble of learned features (average recognition rate per class is 54.95% ± 23.3). b) Using directly the ensemble of learned features (average recognition rate per class is 58.6% ± 21.7%). b) Using the ensemble of learned features and the CRF (average rate per class is 55.57% ± 33.41).

computes the superpixel histogram of object labels (using the approach in [14]) and finally, assigns the predominant class label to each pixel in the superpixel. The baseline is the appearance based algorithm in [2] since it is based on appearance features extracted in a single image. A summary of per–class accuracy is listed in Table 2. As shown, our approach significantly improves the performance of the maximum fusion method, the baseline and the combination of motion and appearance. Furthermore, the proposed approach provides similar per class average accuracy compared to the rest of methods. However, the proposed algorithm provides a lower global performance. This is mainly due to the lack of accuracy in the road class (89.0% compared to 95%) since this is the class with more pixels in the database. As shown, using depth information provides the highest accuracy for the road class and high global accuracy but lower per class average performance. Hence, we expect a significant improvement in the overall performance by including temporal information to our approach. From these results, we can conclude that using learning ensembles of trained multi–scale features provides promising semantic road image segmentation in a single image.

As shown, compared to state–of–the–art approaches, the proposed algorithm provides the higher accuracy for car, column–pole and bicyclist and it outperforms algorithms combining appearance and structure from motion features. In addition, it provides promising results compared to algorithms using CRF frameworks to combine multiple features from diversified sources of information. For instance, the top performing algorithm combines depth, semantic textons, object detection and superpixel features to achieve the higher class average accuracy. Nevertheless, the proposed algorithm outputs a per–class confidence map indicating the pixel potential per class that could be combined with the rest of features and integrated into a CRF framework improving the global and class average accuracy.

Table 2. Quantitative comparison of our method with state–of–the–art road scene recognition approaches on the CamVid database. These approaches include different cues such as motion, appearance, depth or stereo information. Bold names indicate an instance of the proposed method. Bold values indicate the best performing method.

	Building	Tree	Sky	Car	Sign-Symbol	Road	Pedestrian	Fence	Column–Pole	Sidewalk	Bicyclist	Average	Global
Our approach (CNN–CRF)	84.3	65.3	93.1	74.6	0.4	93.5	25.6	32.3	13.8	**85.0**	**54.3**	56.6	83.6
MultiScale CNN - noCRF	47.6	68.7	95.6	73.9	32.9	88.9	**59.1**	49.0	**38.9**	65.7	22.5	58.6	72.9
CNN–MR Fine	37.7	66.2	92.5	77.0	26.0	84.0	50.9	43.7	31.0	65.7	29.7	54.9	68.3
CNN–superpixels	3.2	59.7	93.5	6.6	18.1	86.5	1.9	0.8	4.0	66.0	0.0	30.9	54.8
Fine Level	33.2	53.9	87.8	67.1	23.2	83.9	42.7	44.1	31.3	63.0	26.1	50.6	63.5
[11] Unary	61.9	67.3	91.1	71.1	**58.5**	92.9	49.5	37.6	25.8	77.8	24.7	59.8	76.4
Baseline (App. [2])	38.7	60.7	90.1	71.1	51.4	88.6	54.6	40.1	1.1	55.5	23.6	52.3	66.5
[2] (SfM)	43.9	46.2	79.5	44.6	19.5	82.5	24.4	**58.8**	0.1	61.8	18.0	43.6	61.8
[2] (SfM combined)	46.2	61.9	89.7	68.6	42.9	89.5	53.6	46.6	0.7	60.5	22.5	53.0	69.1
[11] Unary & pairwise	70.7	70.8	94.7	74.4	55.9	94.1	45.7	37.2	13.0	79.3	23.1	59.9	79.8
[11] higher order	84.5	72.6	**97.5**	72.7	33.0	95.3	34.2	45.7	8.1	77.6	28.5	59.2	83.8
[4] (no det.)	79.3	76.0	96.2	74.6	43.2	94.0	40.4	47.0	14.6	81.2	31.1	61.6	83.1
[4] (det.)	81.5	**76.6**	96.2	78.7	40.2	93.9	43.0	47.6	14.3	81.5	33.9	**62.5**	**83.8**
[5] (top down)	80.4	76.1	96.1	**86.7**	20.4	95.1	47.1	47.3	8.3	79.1	19.5	59.6	83.2
[1] depth	**85.3**	57.3	95.4	69.2	46.5	**98.5**	23.8	44.3	22.0	38.1	28.7	55.4	82.1

Finally, the computational cost required to process a single image is analyzed. Currently we have a sub–optimal implementation based on Lua code and Matlab. Our implementation takes approximately 5 seconds to output the features of a 320×240 image and approximately 5 seconds to estimate the optimum labeling. Further, our approach is highly parallelizable and specially suitable for FPGA–based processors [15]. From these results, we can conclude that the proposed CNN–based algorithm provide promising semantic road scene segmentation in a single image.

5 Conclusions

In this paper, we proposed a semantic road image segmentation algorithm based on the fusion of multiple features. The algorithm first extracts learned features at multiple scales and multiple resolutions and then, fuses them at pixel level using a weighed linear combiner. Features and weights are learned off–line directly from training data.

Experiments conducted on a publicly available database demonstrate that a weighted combination outperforms other fusion methods based on fixed rules or single scale methods. Moreover, the algorithm outperforms state–of–the–art appearance based methods and it performs similar in terms of class average performance compared to algorithms using other types of cues such as motion, depth or stereo.

References

1. Zhang, C., Wang, L., Yang, R.: Semantic Segmentation of Urban Scenes Using Dense Depth Maps. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 708–721. Springer, Heidelberg (2010)
2. Brostow, G.J., Shotton, J., Fauqueur, J., Cipolla, R.: Segmentation and Recognition Using Structure from Motion Point Clouds. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 44–57. Springer, Heidelberg (2008)
3. Brostow, G.J., Fauqueur, J., Cipolla, R.: Semantic object classes in video: A high-definition ground truth database. Pattern Recognition Letters (2008)
4. Ladický, Ľ., Sturgess, P., Alahari, K., Russell, C., Torr, P.H.S.: What, Where and How Many? Combining Object Detectors and CRFs. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 424–437. Springer, Heidelberg (2010)
5. Floros, G., Rematas, K., Leibe, B.: Multi-class image labeling with top-down segmentation and generalized robust p^n potentials. In: BMVC 2011 (2011)
6. Gupta, A., Efros, A.A., Hebert, M.: Blocks World Revisited: Image Understanding Using Qualitative Geometry and Mechanics. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 482–496. Springer, Heidelberg (2010)
7. Farabet, C., Couprie, C., Najman, L., LeCun, Y.: Scene parsing with multiscale feature learning, purity trees, and optimal covers. In: ICML 2012 (2012)
8. Cecotti, H., Graser, A.: Convolutional neural networks for p300 detection with application to brain-computer interfaces. PAMI 33, 433–445 (2011)
9. LeCun, Y., Bengio, Y.: Convolutional networks for images, speech, and time-series. In: Arbib, M.A. (ed.) The Handbook of Brain Theory and Neural Networks. MIT Press (1995)
10. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience (2004)
11. Sturgess, P., Alahari, K., Ladicky, L., Torr, P.H.S.: Combining appearance and structure from motion features for road scene understanding. In: BMVC 2009 (2009)
12. Levinshtein, A., Stere, A., Kutulakos, K., Fleet, D., Dickinson, S., Siddiqi, K.: Turbopixels: Fast superpixels using geometric flows. PAMI 31 (2009)
13. Domke, J.: Graphical models toolbox, `http://phd.gccis.rit.edu/justindomke/JGMT/` (accessed July 31, 2012)
14. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. IJCV 59, 167–181 (2004)
15. Farabet, C., LeCun, Y., Kavukcuoglu, K., Culurciello, E., Martini, B., Akselrod, P., Talay, S.: Large-scale FPGA-based convolutional networks. In: Scaling up Machine Learning: Parallel and Distributed Approaches. Cambridge University Press (2011)