Road Scene Segmentation from a Single Image

Jose M. Alvarez[1,3], Theo Gevers[2,3], Yann LeCun[1], 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. Road scene segmentation is important in computer vision for different applications such as autonomous driving and pedestrian detection. Recovering the 3D structure of road scenes provides relevant contextual information to improve their understanding.

In this paper, we use a convolutional neural network based algorithm to learn features from noisy labels to recover the 3D scene layout of a road image. The novelty of the algorithm relies on generating training labels by applying an algorithm trained on a general image dataset to classify on–board images. Further, we propose a novel texture descriptor based on a learned color plane fusion to obtain maximal uniformity in road areas. Finally, acquired (off–line) and current (on–line) information are combined to detect road areas in single images.

From quantitative and qualitative experiments, conducted on publicly available datasets, it is concluded that convolutional neural networks are suitable for learning 3D scene layout from noisy labels and provides a relative improvement of 7% compared to the baseline. Furthermore, combining color planes provides a statistical description of road areas that exhibits maximal uniformity and provides a relative improvement of 8% compared to the baseline. Finally, the improvement is even bigger when acquired and current information from a single image are combined.

1 Introduction

Segmenting road scenes is an important problem in computer vision [1] for applications such as autonomous driving [2] and pedestrian detection [3]. Common road scene segmentation approaches use information from dense stereo maps [4] or structure from motion [5] to obtain reasonable results at the expense of higher computational cost. An important pre–processing step, to speed–up these algorithms, is road segmentation. Road segmentation is used to discard large image areas and to impose geometrical constraints on objects in the scene. Road scenes mainly consist of vertical surfaces (i.e., buildings, vehicles, pedestrians) positioned on a horizontal ground (i.e., road or sidewalk) with possible parts of the sky. Hence, recovering the 3D structure of these scenes plays an important role in their understanding. Current algorithms to recover the 3D scene layout are mainly based on an initial color segmentation step followed by a hand–designed feature classifier [6,7] which is trained on a general database. Hence, these algorithms assume similar test images and often fail for a different database such

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part VII, LNCS 7578, pp. 376–389, 2012.

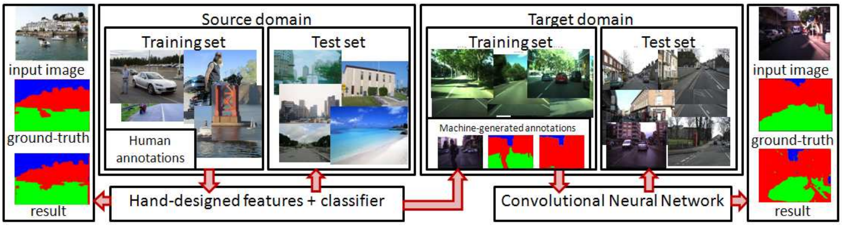


Fig. 1. Current algorithms to infer the 3D scene layout of an image (e.g., Hoiem *et al.* [6]) are trained on general databases (i.e., LabelMe dataset) and their performance may degenerate when applied to unseen images on a different domain dataset (e.g., on–board road images). Our approach first generate training samples on the new dataset using the output of these classifiers and then learn specific features based on a Convolutional Neural Network. Color codes are blue-sky, green-horizontal, red-vertical).

as on–board road images (Fig. 1). This lack of robustness is mainly due to two reasons. First, the use of hand–designed features may be too restrictive to model complex patterns in the training database. Second, the dissimilarities between the training and target datasets. There are two common approaches to improve the performance of these classifiers. The former consists of retraining the classifier with label instances from each new dataset. The latter consists of adapting the classifier kernels to the new domain exploiting domain adaptation methods [8,9,10]. However, these methods usually require manually–labeled instances in the new domain which is time consuming. Therefore, in this paper, we focus on learning features from machine generated labels to infer the 3D scene structure of a single image.

Our approach consists of using a convolutional neural network to learn high–order features from noisy labels for road scene segmentation. These networks are usually trained in a supervised mode on manually labeled data. However, as novelty, we propose training the network using labels generated as predictions of a classifier trained on a general image dataset (Fig. 1). Further, we focus on the on–line learning of patterns in stochastic random textures (i.e., road texture). More precisely, texture is described using statistical moments (e.g., uniformity) of a grey–level histogram of areas in an image obtained by exploiting variant and invariant properties of different color planes in a weighted linear combination that is learned on–line. Finally, acquired (off–line) and current (on–line) information are combined to detect road areas in single images. The former computes road areas based on general information acquired from other road scenes. The latter computes road areas based on information extracted from a small area in the image being analyzed. Hence, the combination exploits the generalization capabilities of the first method and the high adaptability of the latter. As a result, we obtain a robust road scene segmentation algorithm for still images.

The main contributions of this paper are three: (1) we propose an efficient method to learn from machine–generated labels to label road scene images. (2) We propose a novel texture descriptor based on a learned color plane fusion to obtain maximal uniformity in road areas. (3) We combine acquired and on–line information to obtain a diversified ensemble for road scene segmentation in a single image.

The rest of this paper is organized as follows. First, in Section 2, related work is reviewed. Then, in Section 3, the learning algorithm to compute the 3D scene layout of an image is described. The algorithm to detect road areas based on the fusion of color planes is detailed in Section 4. The framework for combining off–line and on–line information is outlined in Section 5. Next, in Section 6, experiments are presented and results are discussed. Finally, conclusions are drawn in Section 7.

2 Related Work

Vision-based road segmentation aims at the detection of the (free) road surface ahead the ego–vehicle and is an important research topic in different areas of computer vision such as autonomous driving [2] or pedestrian crossing detection [3]. Detecting the road in images taken from a mobile camera in uncontrolled, cluttered environments is a challenging problem in computer vision. The appearance of the road varies depending on the daytime, shape, road type, illumination and acquisition conditions. Common approaches model the appearance of the road using cues at pixel–level (such as color [11] or texture [12]) to group pixels in two different groups: drivable road or background. *Color information* has been exploited to minimize the influence of lighting variations and shadows. For instance, Alvarez *et al.* [11] convert the image into an illuminant–invariant feature space and then apply a model–based classifier to label pixels. However, color based approaches may fail for severe lighting variations (strong shadows and highlights) and may depend on structured roads. *Road–texture information* is used to estimate the vanishing point of road scenes [12]. These approaches are based on dominant texture orientations and are sensitive to consistent road marks (i.e., off–road navigation). Using texture for road detection has two main disadvantages. First, the strong perspective effect of road scenes. Second, roads usually present random aperiodic textures that are not easily characterizable.

Therefore, in this paper, we propose a novel texture descriptor based on color plane fusion to obtain maximal uniformity in road areas. The proposed descriptor does not depend on the shape of the pattern and can be learned on–line using a few pixel samples.

Contextual information is used for vanishing point estimation [13] or road scene segmentation as road scenes mainly consist of ground planes, vertical walls and possibly sky [14,15]. Current algorithms to recover the 3D scene layout are mainly based on an initial color segmentation step followed by a classifier [6,7]. However, there is a significant decrease in performance when these algorithms are applied to a different domain (e.g., on–board images, see Fig. 1). A common method to improve their performance is retraining the classifier with label instances from each new domain. However, the collection of labeled instances is time consuming. Another approach consists of adapting the classifier kernels to the new domain exploiting domain adaptation methods [8,9,10]. However, these methods usually require manually–labeled instances in the new domain to improve generalization capabilities of the classifiers. Further, all these systems use hand–designed features that may not be suitable to model complex patterns in the training data. Therefore, in the next section, we propose an algorithm to learn high–order features to obtain the 3D scene layout of the scene based on machine–generated annotations.

3 Learning–features for Recovering Surface Layout from a Single Image

In this section, we propose an effective method based on Convolutional Neural Networks [16] to recover the layout of road images (i.e., images acquired using a camera mounted on a mobile platform). Learning is based on predictions of a classifier trained for recovering the scene layout on a general image dataset (i.e., images from the LabelMe dataset). Convolutional neural networks (CNNs) are multi–layer feed–forward architectures widely used in visual applications such as detection [17], recognition and segmentation tasks [18]. Common CNN architectures are composed of one to three stages (levels) as shown in Fig. 2. Each stage consists of a convolutional filter bank layer (C layers), a non–linear transform layer and a spatial feature layer or down–sampling (S layers). The combination of these three architectural concepts ensures a degree of shift, scale and distortion invariance and improves robustness against variations in the position of distinctive features in the input data. CNNs are trained off–line to extract local visual features (structures) exploiting the fact that images have strong 2-D local structures. During the training process (learning), local features are extracted from the input image (RGB pixel values) and combined in subsequent layers in order to obtain higher order architectures. Hence, the training process consists of learning the kernels and connection weights for each layer (Fig. 2) to infer the label instances in the training set. The output of this process is a set of filters banks (kernels) combined and tuned for the task at hand.

The core of the proposed algorithm is a CNN which takes an $N \times M$ image patch as input and outputs a set of 3 floating point numbers to indicate the potential of the patch to belong to the sky, horizontal or vertical areas. These potentials range from 0 to 1. The higher the potential is, the more likely the pixel belongs to that class. For larger areas (e.g., image) a sliding window is applied to provide a potential set for each pixel in the image based on its $N \times M$ neighborhood. CNNs are usually trained in a supervised mode based on large amounts of manually annotated images. However, collecting and labelling target instances is time consuming and not always possible. Therefore, in the next section, we propose using predictions of a classifier (machine generated labels) to train a CNN to infer the 3D scene layout.

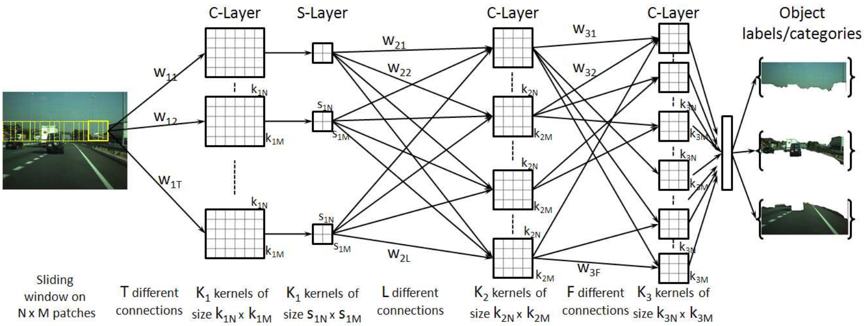


Fig. 2. Convolutional neural networks exploit strong 2-D local structures present in an image to learn high order local features using multi–layer architectures. The learning capacity of these networks depends on the number and size of the kernels in each layer and the number of kernel combinations between layers.

3.1 Training Data Preparation

Training data is generated using machine–generated target labeling. In this paper, the layout of each image (sky, vertical and ground pixels) is obtained using the approach of Hoiem *et al.* [6]. This method is trained on a general domain dataset (i.e., LabelMe) to provide, for each pixel, a confidence map for each of these classes. Hence, target labels are generated considering the maximum support obtained for each class. To reduce the number of training samples (i.e., reduce the overhead in the training process) only a subset of training patches is considered for each image. This subset per image is generated as follows. First, superpixels are obtained using the approach of Levinshtein *et al.* [19]. Then, patches centered at the centroid of each superpixel are selected for training. To improve spatial invariance, 4 more patches around the centroid are also selected.

CNN is trained (weight learning) 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. Once the model is trained, it is fixed during evaluation. Training is stopped when the error in consecutive epochs does not decrease more than 0.001. Bootstrapping can not be applied as we aim to achieve maximum generalization on unseen samples and target labels are noisy. Bootstrapping training samples would lead to models specialized in certain data that may be wrongly labeled (i.e., errors in the label generation algorithm).

4 Learning Road–Texture Patterns by Color Plane Fusion

Road texture has a stochastic and random nature. Moreover, perspective effects are present in road images. Common texture-based approaches describe patterns presents in a specific region by using a gray–scale image area as input [20]. Some improvement is achieved by extending these descriptors to color images by concatenating the output of the descriptor for each color plane [21]. For instance, Local Binary Patterns (LBP) are rotation invariant texture descriptors designed to characterize granular image areas [20]. However, LBP are based on spatial relations that are not suitable to describe random textures. Moreover, these descriptors are scale dependent and fail due to the perspective effect.

Therefore, in this section, we propose a novel approach to describe textures by minimizing the variance of small areas. This variance is estimated using statistical moments of the histogram of a road area. Histograms are independent of the relative position of pixels with respect to each other. As consequence, the proposed descriptor is scale and rotation invariant and suitable for characterizing random stochastic patterns. In particular, we consider a measure of histogram uniformity U defined as follows [22],

$$U = \sum_{j=1}^{L} p^2(j), \tag{1}$$

where $p(j)$ is the j-th bin of the histogram of intensity levels in a region and L is the number of possible intensity levels of the histogram. U ranges from [0..1], and is maximum when all pixel levels are equal (maximally uniform).

Based on this measure, instead of using a given color plane (i.e., grey–level image), we aim to find a transformation that maximizes the uniformity of a road area. That is, reducing the deviation of pixel values around the mean of the patch. To this end, we use a linear combination of different color planes to obtain new road intensity values. Hence, the intensity value of the i-th pixel in a road patch, $y(i)$, can be obtained as follows:

$$y(i) = \sum_{j=1}^{N} w_j x_j(i), \tag{2}$$

where N is the number of color planes (i.e., R, G, B, nr, ng, L, a, b, among others), w_j is the support of each color plane to the final combination and $x_j(i)$ is the value of the i-th pixel using the j-th color plane.

Our goal is obtaining a linear combination that minimizes the variance of intensity values. Hence, a minimum variance estimate of y is obtained by taking the weighted average (Eq. (2)) and restricting the coefficients w_j to sum up to one, that is, $\sum_{j=1}^{N} w_j = 1$. An optimal solution to estimate the minimum variance weights is given by:

$$\mathbf{w} = \Sigma^{-1} I (I^T \Sigma^{-1} I)^{-1}, \tag{3}$$

where $\mathbf{w} = [w_1, \ldots, w_N]^T$ is the vector of weights to be estimated, Σ is the data (pixel representation) covariance and I is an N-element vector with ones.

Finally, once the set of weights is estimated, new pixel intensities and U values are computed for image areas. Pixels exhibiting high U are more likely to belong to the road surface.

The algorithm to characterize road areas is summarized as follows:

– Given an input image, select a training area containing K pixels (i.e., the central–bottom part of the image).
– Obtain $\mathbf{X}$, a $K \times N$ matrix containing N different color representations for each pixel in the training area.
– Compute $\mathbf{w} = [w_1, \ldots, w_N]^T$, using Eq. (3).
– Obtain $\mathbf{y} = [y_1, \ldots, y_M]$ using Eq. (2), where M is the number of pixel in the input image.
– Compute U using Eq. (1) of image regions obtained either using superpixels [19] or using fixed square regions.
– Threshold the output U to obtain road areas. The higher value the more likely to be road areas.

5 Combining Off–line and On–line Learning for Road Detection

In this section, acquired (off–line) and current (on–line) information is combined to detect road areas in single images. The former computes road areas based on general information learned off–line from other road scenes. The latter computes road areas based on learning the current appearance of the road from a small training area. Thus, the combination exploits the generalization capabilities of the former method and the high adaptability of the latter.

The combination algorithm consists of a Naive Bayes framework. The algorithm considers the output of each learning method (Section 3 and Section 4) as road likelihoods: $\mathcal{L}_L(x_i = R)$ is the i–th pixel road likelihood according to the scene layout algorithm and $\mathcal{L}_T(x_i = R)$ is the i–th pixel road likelihood according to the color fusion texture descriptor. Then, the probability of each pixel x_i in an image being a road surface pixel given two observations $p(x_i = R|R_o), o \in [L, T]$ is formulated as follows:

$$p(x_i = R|R_o) \propto p(x_i = R)\mathcal{L}_L(x_i = R)\mathcal{L}_T(x_i = R), \quad (4)$$

where $p(x_i = R)$ is the road prior for that pixel.

Given $p(x_i = R|R_o)$, road or background pixel labels are assigned based on a fixed threshold λ. Hence, a road label is assigned if $p(x_i = R|R_o) > \lambda$. Otherwise, a background label is assigned. In this way, only considerations about the road are taken into account. This is a major advantage due to the diversity of background samples (different scenarios, vehicles, buildings, pedestrian, sky and so on).

6 Experiments

In this section, two different experiments are conducted to validate the proposed method. The goal of the first experiment is evaluating the ability of the proposed learning scheme to infer the 3D scene structure from a single image based on predictions from a classifier trained on a general database. The goal of the second experiment is evaluating the fusion of color planes for describing road textures and the proposed road detection algorithm. Experiments are conducted on two different datasets of images acquired using a camera mounted on a mobile platform. The first dataset consists of 2000 on–board road images taken at different days, different daytime and in different scenarios. Thus, images exhibit different backgrounds, different lighting conditions and shadows and the presence of other vehicles due to different traffic situations. The second dataset is the Cambridge-driving Labeled Video Database (CamVid) [23]. CamVid is a publicly available collection of videos captured from the perspective of a driving automobile with ground truth labels that associate each pixel with one of 32 semantic classes. This dataset is divided in two subsets of images (training and testing). In our experiments we use the training set as validation set.

6.1 3D Scene Layout

The first experiment consists of evaluating the 3D scene layout learning system. To this end, a non–overlapping randomly selected subset of 500 images from the first dataset is processed using the approach in [6] to obtain target labels using the training procedure described in Section 3.1. As a result, we obtain a training set consisting of 4.8M patches with noisy labels (i.e., classifier trained on the LabelMe dataset is applied to a completely different domain and results are not human–supervised). Robustness to scale and noisy acquisition conditions is reinforced using jitter in each patch. In particular, we consider a random scale between $[0.6, \ldots, 1.4]$, random Gaussian noise $\sigma = [0.3, \ldots, 1.2]$ and random rotations in the range $[-17°, \ldots, 17°]$. Quantitative evaluations are provided using the accuracy of the algorithm tested on the CamVid dataset. Accuracy is computed by comparing ground–truth pixels to the output of our algorithm as in [1]. Pixel–wise confusion matrices are provided including the global average accuracy (normalized diagonal of the pixel–wise confusion matrix). Ground–truth is generated considering road and sidewalk semantic classes as horizontal surfaces, sky as sky and the remaining 29 semantic classes as vertical surfaces.

For a comprehensive evaluation, we compare the performance and computational cost of 15 different network configurations (Fig. 2) on the validation set (training set of the CamVid database) using three different RGB input patch sizes ($N \times M \in \{32 \times 32, 48 \times 48, 64 \times 64\}$). Network configurations using gray level patches are also considered. However, due to their low accuracy, their results are not included. For each input size, we vary the number of kernels in each layer ($k_1 \in \{4, 6\}, k_2 \in \{10, 16, 60, 50, 70, 100, 130\}, k_3 \in \{3\}$) and the kernel sizes ($k_{1N} \times k_{1M} \in \{5 \times 5, 7 \times 7, 9 \times 9, 13 \times 13\}, k_{2N} \times k_{2M} \in \{7 \times 7, 9 \times 9, 13 \times 13\}, k_{3N} \times k_{3M} \in \{5 \times 5, 7 \times 7, 9 \times 9\}$). Varying these parameters differentiates the learning capacity and the learning cost of the network: the higher number of kernels, the higher learning capacity at the expense of a higher learning cost. The larger kernel, the more contextual information is used to infer the pixel label. Figure Fig. 3 show the summary of evaluations for each CNN

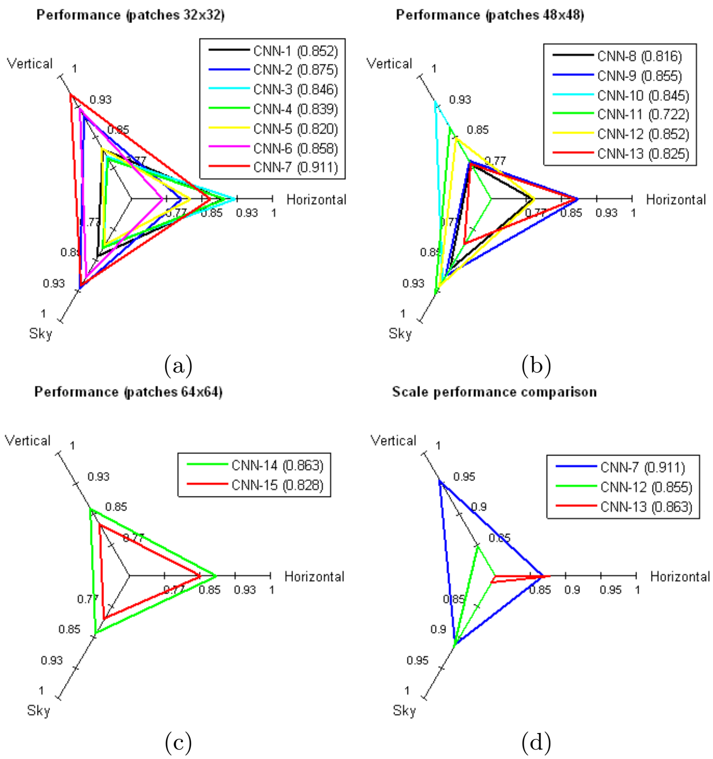
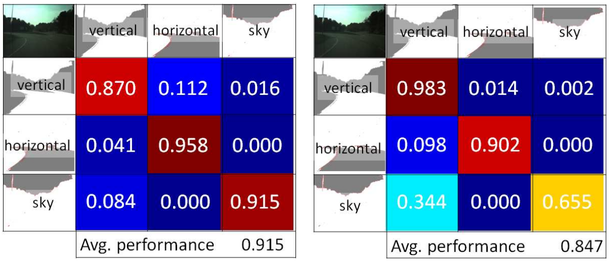


Fig. 3. Performance evaluation of different CNNs tested on the training set of the CamVid [23] dataset (used as validation set). This dataset has not been used for generating training samples. Input patch size varies from: a) $N \times M = 32 \times 32$, b) $N \times M = 48 \times 48$, c) $N \times M = 64 \times 64$. d) Comparison between best performance CNN in each scale. Better viewed in color.

configuration at each scale and the performance comparison between the best configuration per scale. As shown, the performance depends on the topology and the input size. Nevertheless, differences in performance are not significant. That is, in this application, the amount of contextual information does not influence the performance of the algorithm. A summary of current processing time of representing CNN topologies and the baseline approach are listed in Fig. 4a. CNNs are tested on non–optimized Lua code and the baseline is tested using the code publicly available. As shown, CNNs significantly reduce the time required for processing each image. In addition, CNN processing time is highly dependent on the network architecture and the number of filters per layer. The higher number of layers and kernels, the higher learning capacity and higher computational cost (i.e., higher number of convolutions per patch). Nevertheless, CNN–7, the selected configuration for the rest of the experiments is $30\times$ faster than the baseline and reaches real–time at $1/3$ resolution (320×240).

	Time per image	
	1/1 res.	1/3 res.
Hoiem *et al.* [6]	47.6s.	6.49s.
CNN–2	4.83s.	0.48s.
CNN–7	1.96s.	0.21s.
CNN–9	3.56s.	0.36s.
CNN–11	0.58s.	0.06s.
CNN–13	22.2s.	2.08s.
CNN–14	1.68s.	0.17s.

(a)



(b) (c)

Fig. 4. a) Average time per image at full (960×720) and 1/3rd resolution. Significant reduction in time can be achieved at the expense of a lower accuracy, see Fig. 3. b) Confusion matrix of CNN–7 ($C_1 : k_1 = 4, k_{1N} \times k_{1M} = 7 \times 7$. $S_1 : s_{1N} \times s_{1M} = 2 \times 2$. $C_2 : k_2 = 16, k_{2N} \times k_{2M} = 7 \times 7$. $C_3 : k_3 = 3, k_{3N} \times k_{3M} = 7 \times 7$). c) Confusion matrix of the baseline approach in [6], both matrices on the CamVid [23] test dataset.

Finally, confusion matrices comparing CNN–7 and the baseline (3D scene layout approach in [6]) on the test set (CamVid testing sub–set) are shown in Fig. 4b and Fig. 4c. Representative qualitative results from both datasets are shown in Fig. 5. It can be derived that the baseline algorithm fails and degenerates on complex color situations since it is based on a RGB image segmentation and on color information. However, the proposed approach to learn features based on noisy labels is able to properly recover the layout of different unseen images without degeneration. Moreover, the proposed approach significantly improves (relative improvement of 7%) the average accuracy of the baseline approach and, thus, exhibits better generalization capabilities. The proposed approach use machine–generated labels to tune and combine a set of filter banks. From these results we can conclude that learning features is a recommended approach to transfer labels from a general image domain to a specific one. The bottleneck of the proposed algorithm is the selection of its topology. However, for this application, the difference is not representative (Fig. 3).

6.2 Combining On–line and Off–line Learning for Road Detection

The second experiment consists of evaluating the fusion of color planes to detect road areas and the proposed algorithm for combining off–line and on–line learning for road detection. Quantitative road evaluations are provided using ROC curves on the pixel–wise comparison between the segmentation results and ground–truth. ROC curves are graphical representations of the trade–off between true positive rate ($TPR = \frac{TP}{TP+FN}$) and false positive rate ($FPR = \frac{FP}{FP+TN}$) for different cut–off points of a parameter. In this paper, the parameter is λ (Section 5).

The setup of the algorithm is done as follows: the input image is converted into the following color planes, R, G, B, nr, ng, $opp1$, $opp2$, S, and V from common color spaces such as RGB, normalized RG, opponent color space and HSV. We refer the reader to [22] for their derivation. Besides, training pixels

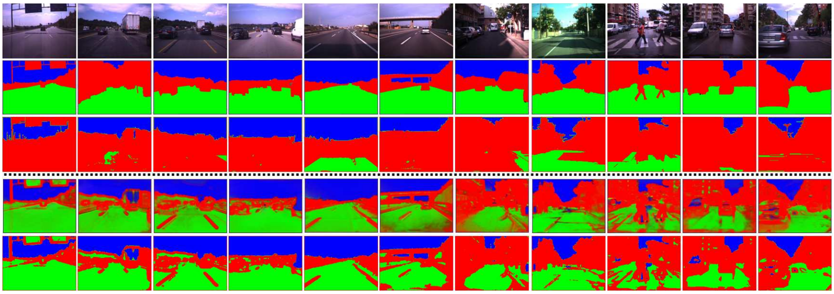
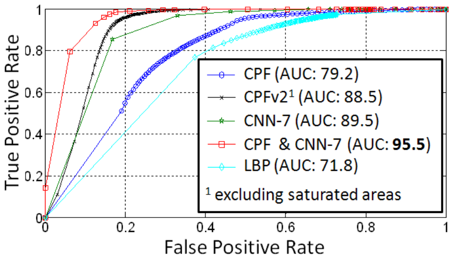


Fig. 5. Example 3D scene layout results. First row show the input image. Second row shows manually annotated ground truth. Third row show the results (binarized using the maximum output of the classifier) using the approach in [6]. Fourth row show the output of the convolutional neural network (color codes are blue-sky, green-horizontal, red-vertical). Bottom row shows the binarized masks based on the maximum output confidence of the classifier.

are selected using a fixed area in each image. In particular, we use the common assumption that the bottom part (30×80 pixels) of the image belongs to the road surface [11]. Non–overlapping image regions are estimated using the superpixel algorithm in [19]. The layout of the scene is estimated using CNN–7. For testing purposes, the road prior, $p(x_i = R)$ in Eq. (4) is considered uniform for the image. Finally, isolated road areas are discarded using connected components [11]. For comparison, two different texture approaches are considered. The former estimates U directly on the gray–level image. The latter, the baseline, estimates a road likelihood based on LBP texture descriptors [20]. In this way, rotational invariant LBP is computed at each pixel in the image and then, for each small image area, histograms of LBP codes are computed and compared to seed histograms (located on the bottom central part of the image), see [20] for algorithm details. Qualitative examples of texture results for different scenarios are shown in Fig. 7. Moreover, ROC curves and their corresponding area under the curve AUC (as a global measure of accuracy) for the proposed layout algorithm, two instances of the color plane fusion: CPF and CPFv2 (CPF adding a pre–processing step to exclude saturated areas) and their combination are shown in Fig. 6a. Finally, qualitative road detection results are shown in Fig. 8.

As shown in Fig. 7, the proposed descriptor exhibits high discriminative power and high robustness against road intra–class variations in images containing shadows, vehicles, sidewalks and concrete roadside areas. Furthermore, it is highly adaptive since the optimum combination of color planes is estimated independently at each frame. Moreover, the proposed texture approach significantly improves the baseline performance (8% bigger). The improvement is even bigger (24%) when the texture descriptor is combined with the layout of the scene. From these results, we conclude that color plane fusion for obtaining maximal uniformity in road areas is a robust and suitable descriptor for



(a)

State–of–the–art methods	Acc.
Motion [1]	82.5
App. [1]	88.6
Mot. & *App.* [1]	89.5
App. & *SFM.* (v1) [5]	92.9
App. & *SFM.* (v2) [5]	94.1
App. & *SFM.* (v3) [5]	95.3
Our approach (CPF & CNN–7)	**95.5**

(b)

Fig. 6. a) ROC curves and area under ROC on the CamVid testing dataset. CPF: color plane fusion. CPFv2: color plane fusion including a preprocessing step to exclude saturated areas. CNN–7: 3D scene layout using convolutional neural networks and their combination. The baseline is a LBP–based texture algorithm. b) Road accuracy of different state–of–the–art road scene segmentation algorithms using additional information such as stereo, or temporal information (*App.* is appearance, *Mot.* is motion, *SFM* is Structure From Motion).

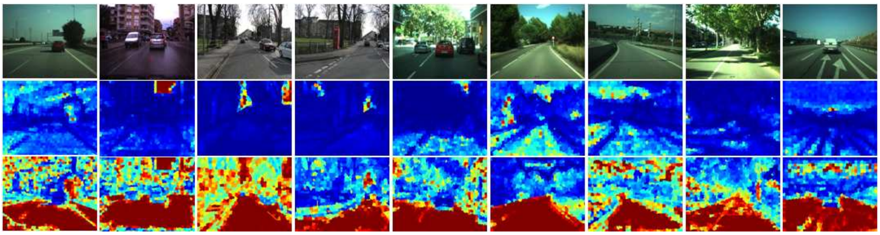


Fig. 7. Example road detection results using the proposed combination of color planes. Top row: original images. Middle row: result of estimating U directly on a gray–level image. Bottom row: road likelihood using color plane fusion as road–texture descriptors.

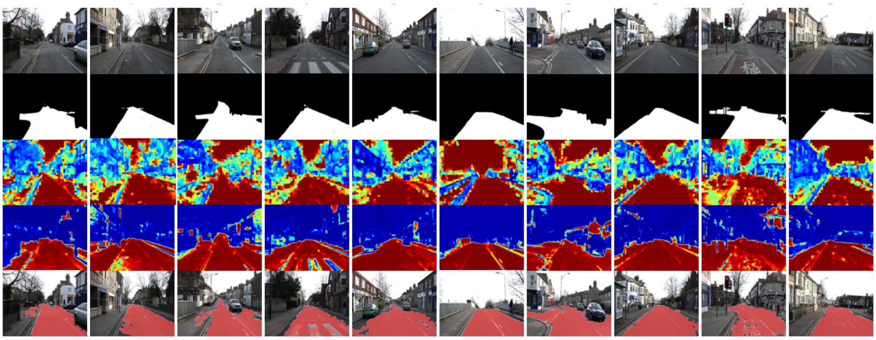


Fig. 8. Example road detection results combining texture and scene layout. First row: input image. Second row: ground truth. Third row: road likelihood using color plane fusion. Fourth row: road likelihood using CNN for road scene layout. Bottom row: overlaid results obtaining using the proposed algorithm.

detecting drivable road areas at the expense of using a small number of training pixels. Analysis reveals limitations to distinguish completely saturated images areas (i.e., sky areas) as shown in Fig. 8. This is mainly due to the lack of color information. However, this could be solved at acquisition time using high dynamic range cameras or polarizer filters. Moreover, combining off–line and on–line approaches exhibits higher performance due to lower false positive ratios. This is mainly due to the capabilities of the texture descriptor to discard sidewalk areas present in horizontal surfaces. Further, the performance of the proposed algorithm working on still images is comparable to the performance of state–of–the–art algorithms (Fig. 6b) using additional information such as dense depth maps or structure from motion. These algorithms rely on human–labeled data from the same database and require a large computational time per image (30 – 40 sec. per image [5]). In contrast, the proposed approach uses images from the CamVid dataset exclusively for testing. From these results, we can conclude that combining acquired information with road descriptors learned from the current image results in a robust method for road scene understanding from a single image. The proposed algorithm does not require specific training and relies only in the assumption that the bottom part of the image belongs to the road surface. The classification step is based on a single threshold at pixel–level. Hence, we expect significant improvement by including additional spatial reasoning [4].

7 Conclusions

We proposed an efficient method to learn from machine–generated labels to label road scene images based on convolutional neural networks. Further, we propose a novel texture descriptor based on learning a combination of color plane to obtain maximal uniformity in road areas. Finally, a combination strategy is proposed which resulted in a robust road scene segmentation algorithm.

From quantitative and qualitative experiments conducted on publicly available datasets, it can be concluded that using convolutional neural networks to learn from noisy data provides a 7% of relative improvement compared to the baseline. Further, combining color planes provides a statistical description results in a significant improvement. The improvement is even bigger when both methods are combined to perform robust road scene segmentation.

Acknowledgements. 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.

References

1. 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)

2. Lookingbill, A., Rogers, J., Lieb, D., Curry, J., Thrun, S.: Reverse optical flow for self-supervised adaptive autonomous robot navigation. IJCV 74, 287–302 (2007)
3. Geronimo, D., Lopez, A.M., Sappa, A.D., Graf, T.: Survey of pedestrian detection for advanced driver assistance systems. PAMI 32, 1239–1258 (2010)
4. Ladicky, L., Sturgess, P., Russell, C., Sengupta, S., Bastanlar, Y., Clocksin, W., Torr, P.: Joint optimization for object class segmentation and dense stereo reconstruction. IJCV, 1–12 (2011)
5. 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)
6. Hoiem, D., Efros, A.A., Hebert, M.: Recovering surface layout from an image. IJCV 75, 151–172 (2007)
7. Saxena, A., Min, S., Ng, A.Y.: Make3d: Learning 3d scene structure from a single still image. PAMI 31(5), 824–840 (2009)
8. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting Visual Category Models to New Domains. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 213–226. Springer, Heidelberg (2010)
9. Duan, L., Tsang, I.W., Xu, D.: Domain transfer multiple kernel learning. PAMI 34, 465–479 (2012)
10. Kulis, B., Saenko, K., Darrell, T.: What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In: CVPR 2011, pp. 1785–1792 (2011)
11. Alvarez, J.M., Lopez, A.M.: Road detection based on illuminant invariance. IEEE Trans. on ITS 12(1), 184–193 (2011)
12. Rasmussen, C.: Grouping dominant orientations for ill-structured road following. In: CVPR 2004 (2004)
13. Kong, H., Audibert, J., Ponce, J.: Vanishing point detection for road detection. In: CVPR 2009, pp. 96–103 (2009)
14. Ess, A., Mueller, T., Grabner, H., Gool, L.J.V.: Segmentation-based urban traffic scene understanding. In: BMVC 2009 (2009)
15. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: ICCV 2009 (2009)
16. LeCun, Y., Bengio, Y.: Convolutional networks for images, speech, and time-series. In: The Handbook of Brain Theory and Neural Networks. MIT Press (1995)
17. Cecotti, H., Graser, A.: Convolutional neural networks for p300 detection with application to brain-computer interfaces. PAMI 33, 433–445 (2011)
18. Turaga, S.C., Murray, J.F., Jain, V., Roth, F., Helmstaedter, M., Briggman, K., Denk, W., Seung, H.S.: Convolutional networks can learn to generate affinity graphs for image segmentation. Neural Comp. 22, 511–538 (2010)
19. Levinshtein, A., Stere, A., Kutulakos, K.N., Fleet, D.J., Dickinson, S.J., Siddiqi, K.: Turbopixels: Fast superpixels using geometric flows. PAMI 31 (2009)
20. Petrou, M.: Image Processing: Dealing with Texture. Wiley (2006)
21. van de Sande, K.E.A., Gevers, T., Snoek, C.G.M.: Evaluation of color descriptors for object and scene recognition. In: CVPR 2008, pp. 453–464 (2008)
22. Gonzalez, R., Woods, R.: Section 10.4. In: Digital Image Processing, 2nd edn. Prentice Hall (2002)
23. Brostow, G.J., Fauqueur, J., Cipolla, R.: Semantic object classes in video: A high-definition ground truth database. Pattern Recognition Letters (2008)