Embedding Structured Contour and Location Prior in Siamesed Fully Convolutional Networks for Road Detection

Qi Wang, *Senior Member, IEEE*, Junyu Gao, and Yuan Yuan, *Senior Member, IEEE*

***Abstract*—Road detection from the perspective of moving vehicles is a challenging issue in autonomous driving. Recently, many deep learning methods spring up for this task because they can extract high-level local features to find road regions from raw RGB data, such as Convolutional Neural Networks (CNN) and Fully Convolutional Networks (FCN). However, how to detect the boundary of road accurately is still an intractable problem. In this paper, we propose a siamesed fully convolutional networks (named as "s-FCN-loc"), which is able to consider RGB-channel images, semantic contours and location priors simultaneously to segment road region elaborately. To be specific, the s-FCN-loc has two streams to process the original RGB images and contour maps respectively. At the same time, the location prior is directly appended to the siamesed FCN to promote the final detection performance. Our contributions are threefold: (1) An s-FCN-loc is proposed that learns more discriminative features of road boundaries than the original FCN to detect more accurate road regions; (2) Location prior is viewed as a type of feature map and directly appended to the final feature map in s-FCN-loc to promote the detection performance effectively, which is easier than other traditional methods, namely different priors for different inputs (image patches); (3) The convergent speed of training s-FCN-loc model is 30% faster than the original FCN, because of the guidance of highly structured contours. The proposed approach is evaluated on KITTI Road Detection Benchmark and One-Class Road Detection Dataset, and achieves a competitive result with state of the arts.**

I. INTRODUCTION

Recently, autonomous driving has drawn great attention with the popularity of intelligent vehicles. It is a core component for the intelligent transportation systems (ITS) and aims at avoiding accidents during the driving period. Since most traffic accidents happen on road, it is important to precisely detect the road region. An accurate road detection can not only make the vehicle navigate in the correct way but also prompt the driving system to focus on the specific tasks in the street scene, such as lane detection [1], vehicle detection [2], pedestrian detection [3] and anomaly detection [4]. If the road region can be ensured, other detection tasks will also benefit from it.

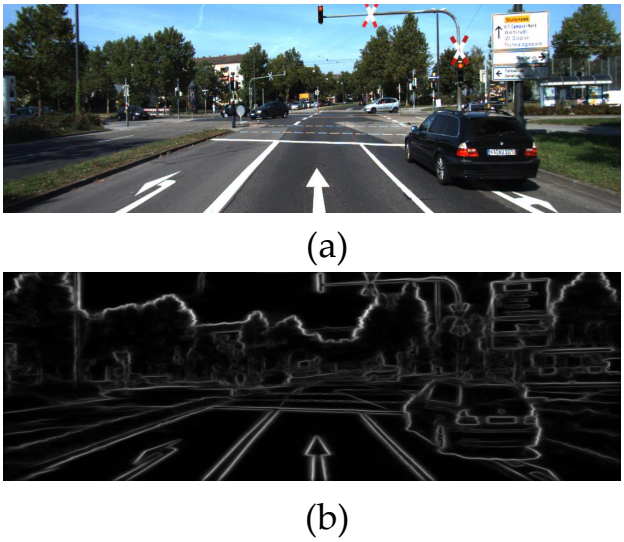


Fig. 1. The exemplar display of an original street scene image (a) and the corresponding contour map (b).

Although this problem is named as "road detection", it is actually a per-pixel classification task which is a type of semantic segmentation or labeling. In the street scene, road can be viewed as a background object, which is usually occluded by foreground objects (such as vehicles, pedestrians or other obstacles) so that the road surface has no definite shape. Thus, road detection is completely different from other object detection tasks which only need to locate the objects using bounding boxes and it is more challenging than the latter.

For detecting road region accurately, some traditional methods ([5], [6], [7] and [8]) exploit 3D point clouds or location information by extra sensors such as laser scanner and GPS. In the real world, nevertheless, a human can drive a vehicle safely under the complex traffic environment without the above extra information. Thus, how to dig out deeper vision information is still an important issue, which is our focus in this paper.

With the rise of deep learning, the Convolutional Neural Networks (CNN) improves the image comprehension by learning more discriminative features ([9], [10], [11] and [12]). Farabet *et al.* [9] propose a multi-scale CNN to extract dense feature vectors that encode regions of multiple scales centered on each pixel. Fully Convolutional Networks (FCN) [10] is a

This work was supported by the National Natural Science Foundation of China under Grant 61379094, Fundamental Research Funds for the Central Universities under Grant 3102017AX010, the Open Research Fund of Key Laboratory of Spectral Imaging Technology, Chinese Academy of Sciences.

Qi Wang is with the School of Computer Science, with the Unmanned System Research Institute, and with the Center for OPTical IMagery Analysis and Learning, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: crabwq@gmail.com).

Junyu Gao and Yuan Yuan are with the School of Computer Science and Center for OPTical IMagery Analysis and Learning, Northwestern Polytechnical University, Xi'an 710072, Shaanxi, China (e-mail: gjy3035@gmail.com; y.yuan1.ieee@gmail.com).

variant of traditional CNN, which leads to great improvement in many applications, especially in object detection and image semantic segmentation. Seg-net [11] proposes an encoder-decoder architecture for image segmentation, in which the encoder is fully convolutional networks and the decoder is deconvolutional networks. The above architectures focus on what an object is but ignore the essential spatial structure and location information in images. In view of this, we introduce spatial structures and location priors into the traditional network:

*1) **Spatial Structures**:* As we all know, the contours in an image represent the essential edges of objects. Different from classic approaches such as Sobel and Canny edge detectors, current methods (e.g. [13], [14] and [15]) focus on detecting semantic edges, which represent the contours of a whole image. Given a contour image of a street scene, human beings are capable of recognizing important objects and their boundaries. Fig. 1 illustrates an example of the original image and its corresponding contour map. Based on the above observation, we would like to train a new CNN model to recognize objects from contour information.

*2) **Location Priors**:* In the street scenes, the objects' spatial distributions are regular. For example, road region is usually located at the bottom of an image. So how to utilize the location prior is critical to remove the false detection. In the previous methods [16], location prior is generated according to the position of each patch in the image, which is cumbersome in the preprocessing stage. Considering these facts, a 2-channel location map is designed to describe the location priors of the whole image and is incorporated in the designed deep model.

In summary, the overview of our method is described below. Given an input image, the semantic contour map is firstly generated by Structured Forests (SF) [15]. Then, the RGB image and the contour map are fed into the proposed siamesed fully convolutional networks, exploring the location feature map simultaneously. Finally, the road region is output by the networks. The concrete flowchart is shown in Fig. 2.

The main contributions of this paper are:

1) Propose a siamesed FCN (s-FCN) for road detection, which learns more discriminative features of road boundaries than the original FCN to detect more accurate road regions. The proposed s-FCN consists of two siamesed convoultional streams. It tackles RGB and contour information by sharing the parameters of convolution layers, which prompts the FCN focuses on extracting features of road boundaries. Meanwhile, higher features than raw data also significantly improve the generalization capacity the model.
2) Append the location prior to s-FCN to reduce the mistaken detection. Specifically, the location prior is viewed as a 2-channel feature map to directly concatenate the existing feature map of the network. To our knowledge, the strategy of considering the location prior (s-FCN-loc) is the first time, which is easier than other traditional methods, namely different priors incorporation for different inputs (image patches or superpixels).
3) Accelerate the training process of the original FCN. The convergent speed of training s-FCN-loc is 30% faster than FCN. The contour maps are regarded as higher-level features than the raw RGB images. The neural network can easily learn more effective semantic representation from the highly structured contour maps, which guides the model to converge to a good solution more quickly.

This work is an extension of our earlier conference paper [12]. The more detailed method description and the further experimental analysis, results are shown in this version.

The rest of this paper is organized as follows. Section II reviews the related work briefly. Section III describes the proposed approach in detail. Section IV shows the experimental settings and results on the two challenging datasets and reports the further discussions and analysis about important strategies in our proposed method. Finally, we summarize the work in Section V.

II. RELATED WORK

In recent years, many approaches for road detection have been proposed. There are more than 50 methods on KITTI road detection benchmark since 2013. According to their pipelines, the algorithms usually consist of several important modules: feature extraction, object classification, contextual inference, and priors combination. In this paper, we only briefly review the important works about the two most related modules: feature extraction and priors combination.

Before the popularity of deep learning, many approaches about road detection are usually comprised of hand-craft feature extraction, per-pixel (superpixel, or block) classification and contextual refinement. Álvarez *et al.* [17] propose illumination invariant features to improve the performance in shadowed street scenes. Mendes *et al.* [18] present a block scheme that classifies small images patches using self-designed features (RGB, grays-cale, entropy, LBP and Leung-Malik filters responses) to efficiently incorporate contextual cues. Since road is background object, which is more cluttered and heterogeneous, Lu [19] proposes a self-supervised method only using hand-crafted color features without priori knowledge of the road structure. Wang *et al.* [20] design a novel context-aware descriptor for superpixels by using depth map and transfer labels in a nearest neighbor search set. Yuan *et al.* [21] propose an on-line structural learning method for exacting drivable road region from video sequences, which uses the fusion of Dense SIFT, HOG and LBP features for their robustness to intensity change and shadow.

Because of the powerful feature learning ability of CNN, many methods exploit it in recent years. Álvarez *et al.* [22] train a CNN model from noisy labels to recover the 3D layout of a street image. Then they design a texture descriptor based on a learned color plane fusion to get maximal uniformity in road regions. After Long *et al.* [10] proposed FCN, some approaches (such as [23], [24] and [25]) also adopt this type of architecture to segment road region. Laddha *et al.* [23] propose a self-supervised approach which does not require any manual road annotations. Then, they fintune a FCN based on VGG-net [26] using these noisy labels for road detection. Mendes *et al.* [24] train a FCN model based on Network-in-Network

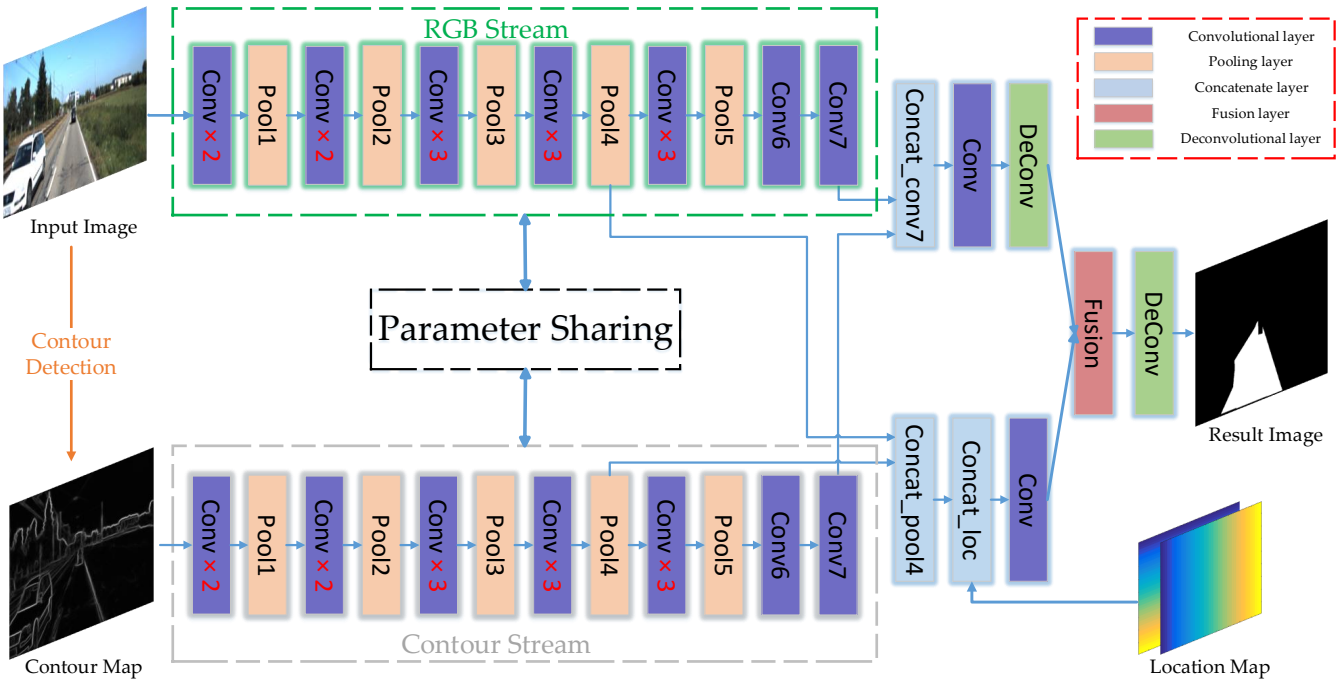


Fig. 2. The flowchart of our proposed siamesed FCN with location prior (s-FCN-loc). First, given an input image, the semantic contour map is generated by a fast contour detection. Then, the RGB-channel image and the contour map are fed into s-FCN-loc, which makes s-FCN-loc focus on learning discriminative features of road boundaries and spatial structure in street scene images. At the same time, the location prior is appended to concat_pool4 layer for alleviating false detection. Finally, the feature map is mapped to each pixel by deconvolution operation to predict the per-pixel road regions.

(NiN) architecture, which utilizes large amounts of contextual information. Mohan [25] proposes a deep deconvolutional network in combination with traditional CNNs for feature learning to road detection. Similar to seg-net architecture [11], Oliveira *et al.* [27] propose a smaller network based on an encoder-decoder symmetric network to achieve a near real-time road detection.

Structured priors (such as shape, edge/contour and location) is important to detection results. Many methods also focus on it. He *et al.* [28] model a boundary estimation to improve the detection performance. Yu *et al.* [29] proposed a new binary local representation for action recognition from RGB-D video sequences, which adopts an orthogonal projection matrix to preserve the pairwise structure with shape constraints. Álvarez *et al.* [30] present an algorithm to estimate road priors by using geographical information systems (GISs), which can provide relevant initial information of the road. Song *et al.* [31] present an algorithm that obtains road boundary information and can be applied to other similar unstructured road environment. Nam *et al.* [32] propose a vision-based road detection algorithm, which adopts robust color-based region merging and edge-based filtering mechanisms. Zitnick and Dollar [33] present a method to locate object proposals based on edge information (the number of edges that are wholly enclosed by a bounding box) in images. Liu *et al.* [34] combine CNNs' output and simple edge map via Conditional Random Field for semantic face segmentation. Brust *et al.* [16] propose convolutional patch networks and incorporate location information into the learning process.

III. APPROACH

In this Section, we first explain the core components of the original FCN [10] in brief. Then the details of the adopted semantic contour map is described. Next, we show the architecture of the proposed s-FCN. Finally, the strategy of incorporating location priors in s-FCN is explained.

A. Fully Convolutional Network(FCN)

For traditional CNN, the convolutional ("conv" for short) layers focus on extracting local features in an image, and on the top of multiple conv layers, the fully connected ("fc" for short) layers integrate those high-level local feature maps into a n-D vector by the inner product operation to predict the image's label. Nevertheless, the architecture of this network does not predict the label for each pixel. Until 2015, Long *et al.* [10] propose the Fully Convolutional Networks (FCN) to tackle the dense prediction problem, which replaces all fc layers with conv layers to produce arbitrary-size output. However, since the deep layer's output loses a lot of location and edge clues, the authors of FCN combine deep and shallow layers' feature maps to obtain finer results, which is called as "FCN-xs". Here x denotes that the fused feature maps need to be x times upsampled to predict the input-image per-pixel label.

In this paper, we adopt the FCN-16s architecture of VGG16-net [26], which fuses the pool4 layer and conv7 layer (convolutionalized fc7 of the original network) by a summing operation. It should be noted that pool4's output is cropped and the conv7's is 2 times upsampled before the fusion for consistent dimensions. VGG16-net can recognize more

than $1,000$ categories objects from images, which consists of 13 conv and 3 fc layers. In 2014, it wins the second prize in ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2014, which achieves 7.32% Top-5 error in image classification task.

B. Semantic Contour Map

Although FCN fuses the deep and shallow layers' information for alleviating imprecise boundary segmentation to some extent, it is difficult to learn the spatial structure and skeleton clues of an image. Fortunately, semantic contour maps represent them more effectively than traditional edges such as Sobel, Canny, Roberts and so on. In addition, contour map is gray-scale instead of binary image, so that the intensity of contour is quantified. To be specific, Fig. 3(b) is an exemplar of contour map, and the pixels with larger value mean that they are more principal contour in the original image.

In order to validate the point that contour is important for semantic labeling, we design a simple trial that let the examined subjects segment each objects from a semantic contour map of a street scene. And they do not go through any special training to recognize objects from contour images. Fig. 3 illustrates the results of this trial. From manual segmentation results, we find human vision are capable of understanding scenes just using the semantic contour map. Although there are some recognition errors contrast to the original image, it is undeniable that the boundary segmentation is elaborate.

The above trail confirms our assumption in a way. Furthermore, we think CNN model can also learn similar ability by supervised training. Therefore, the semantic contour map is generated by SF[1] [15] and a new stream is added to traditional neural network to process contour information. The concrete description is reported in the next section.

C. Siamesed FCN (s-FCN)

Our proposed siamesed network is based on FCN-16s [10], which is shown in Fig. 2. It consists of two streams that handle RGB image and semantic contour map simultaneously. For integrating the two streams' features, the outputs of pool4 and conv7 layer are concatenated together (the sizes are $\mathrm{n}\times 1024\times 44\times 44$ and $\mathrm{n}\times 8192\times 16\times 16$ respectively, where n denotes the size of each mini-batch). Considering the correspondence of the RGB image and the contour map per-pixel, two streams should interact with each other. Thus, at the training stage, the parameters (kernel weights and biases) of conv layer of two streams are shared with each other. However, since the contour map is a gray-scale image, its channel number is not equal to that of RGB image. For sharing parameters, the contour map is replicated on the three channels to be similar to a RGB image.

During the training process, we fine-tune the proposed s-FCN based on the original VGG16-net weights according to the thought of previous section, and minimize the sum of unnormalized soft-max loss for each pixel by SGD.

[1]The source code is provided by Piotr Dollár in https://github.com/pdollar/edges

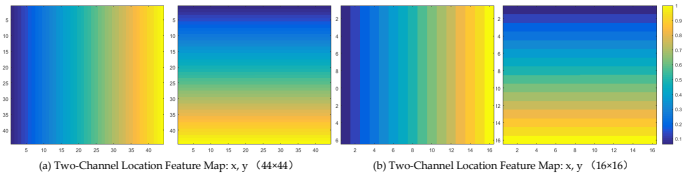


Fig. 4. The visualization of the two different sizes of location feature maps. the coordinate values of x and y axis are normalized in $[0\ ,1]$). (a) and (b) illustrate the 44×44 and 16×16 location feature maps respectively.

D. Incorporating Location Priors in s-FCN

In the street scene, the location prior is important: the objects' spatial distributions are regular. For example, road region is usually located at the bottom of images, and the buildings and trees are on both sides of the road. Thus, utilizing this location prior is essential to remove the false detection. However, the traditional FCN is only sensitive to local appearance features instead of location priors, which causes some unreasonable results. For example, building regions might be mistakenly recognized as road. For alleviating the above problem, [16] proposes a Convolutional Patch Networks (CPN) with location priors to classify the small patches in images as "road" or "not road". Specifically, different patches have different location priors, which means that the location prior needs to be generated independently and enter into CPN. However, the CPN's strategy is not flexible in practice. In order to avoid the disadvantage, in this paper, the location prior is viewed as a type of feature map in s-FCN and it can be appended to convolutional layer's output directly. This way, the location feature map is generated only once for all images.

To be specific, location prior is designed as a 2-channel feature map for the x and y axis in the image, which is appended to the last feature map in s-FCN. The value of a position in x- or y- channel feature map is defined as the coordinate values (normalized in $[0\ ,1]$) of x or y axis in the input image. Since the height and width of the feature map are smaller than the input's, location maps should be resized to the size of the last feature map for concatenating them. It's important to note that there are two final feature maps to be fused in s-FCN: the outputs of concat_pool4 layer and concat_conv7 layer (the $\mathrm{height}\times\mathrm{width}$ of the outputs are 44×44 and 16×16). Fig. 4 shows the visualization of the two different sizes of location feature maps. Different colors represent different values: from blue to yellow correspond to $[0\ ,1]$. From Fig. 4, the 44×44 feature map has more accurate location priors than the 16×16 feature map. Thus, we choose to append the 44×44 location feature map to concat_pool4 layer's output. Moreover, we also compare the different effects of the above two strategies by the further experiments in Section IV-H.

IV. EXPERIMENT

In this section, we respectively report the three comparative results: the original FCN-16s, the proposed s-FCN and the s-FCN-loc on the two challenging road detection datasets. Section IV-A shows the evaluation criteria in road detection.

(a) (b) (c) (d)

Fig. 3. The manually annotated results of the trial in Section III-B. (a) is the original RGB image; (b) is the contour map, which is given to the subjects; (c) and (d) are the annotated results of two randomly selected subjects.

Section IV-B briefly describes the two selected dataset. Section IV-C lists some important implementation details and parameter setup in the experiments. Section IV-D and IV-E shows our road detection results in KITTI Dataset and Section IV-F displays our road detection results in OC Dataset. Then, we analyze the convergent speed about different networks architectures in Section IV-G. In Section IV-H, we discuss the differences of the different location feature maps. Furthermore, we analyze the generalization capacities of FCN, s-FCN and s-FCN-loc in Section IV-I. Finally, the comparison of Contour Map v.s. Depth Map in s-FCN is in Sectioin IV-J.

A. Metrics

For evaluating the algorithm performance, similar to [35], we adopt the following criteria:

$$\text{Precision} = \frac{TP}{TP+FP}, \tag{1}$$

$$\text{Recall} = \frac{TP}{TP+FN}, \tag{2}$$

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}, \tag{3}$$

$$F\text{-}measure = \left(1+\gamma^2\right)\frac{\text{Precision}\cdot\text{Recall}}{\gamma^2\text{Precision}+\text{Recall}}, \tag{4}$$

where TP, FP, TN and FN denote the number of true positive, false positive, true negative and false negative samples under a classification threshold τ, respectively. F-measure is a trade off between precision and recall. In this paper, we set $\beta = 1$ in Eq. 4 (called "F1-measure"), which is the harmonic mean of precision and recall. The KITTI benchmark ranks all methods according to max F-measure, which is defined as below:

$$\max F = \arg\max_{\tau} \ F\text{-}measure, \tag{5}$$

where τ is the classification threshold to maximize the F-measure.

B. Dataset

In order to evaluate the proposed approach we select the road detection dataset in KITTI Vision Benchmark Suite [35] [2] ("KITTI dataset" for short) and One-Class Road Detection Dataset [3] [36] ("OC Dataset" for short).

[2]http://www.cvlibs.net/datasets/kitti/eval_road.php
[3]http://scrd.josemalvarez.net/

1) KITTI Dataset: KITTI Dataset consists 579 images (289 training images and 290 testing images respectively) with a resolution of 375×1242 pixel. The entire data set is divided into three categories, the concrete descriptions of which are shown in Table I. The evaluation server of the benchmark ranks all submitted methods according to their max F-measure on the Birds Eye View ("BEV" for short) by assuming a flat real world for the transformation from the perspective image to the BEV space. The benchmark features color stereo images, GPS information and Velodyne laser scans data for each scene. As for this dataset, we only exploit monocular color data to detect road region in the experiment. For showing the effect of each component, the training set is randomly divided into two classes (272 images for training and 17 images for validation).

TABLE I
THE DETAILS OF KITTI DATASET, INCLUDING THE SCENE CATEGORY, THE NUMBERS OF IMAGES IN TRAINING AND TESTING SETS.

Scene category	Training	Testing
UU (urban unmarked)	98	100
UM (urban marked two-way)	95	96
UMM (urban marked multi-lane)	96	94
URBAN(All)	289	290

2) OC Dataset: The dataset consists of 755 street scene images 640×480 pixels. These images include a variety of scenes (e.g. daybreak, morning, noon, afternoon, sunny, cloudy, rainy), which are selected to cover the major challenges in real world. The above challenges contains strong shadows, wet surfaces, sidewalks similar to the road, direct reflections, crowded scenes, lack of lane markings and so on. Because of our supervised machine learning method, the original dataset is randomly divided into two parts (605 images for training and 150 images for testing) to evaluate the proposed algorithms.

C. Implementation Details and Experimental Setting

In the entire experiment, original images are resized to 500×500 to enter into the neural networks. Contour maps are generated by default parameters (the number of decision trees is 1) of SE-SS in SF [15]. We use two fixed learning rates of 10^{-10} for weights and 2×10^{-10} for biases, a mini-batch size of 4 images, momentum of 0.99 and decay of 0.0005. We also set dropout ratio of 0.5 in conv6 and conv7 layers. Besides, the size of location map is 44×44 for correspondence

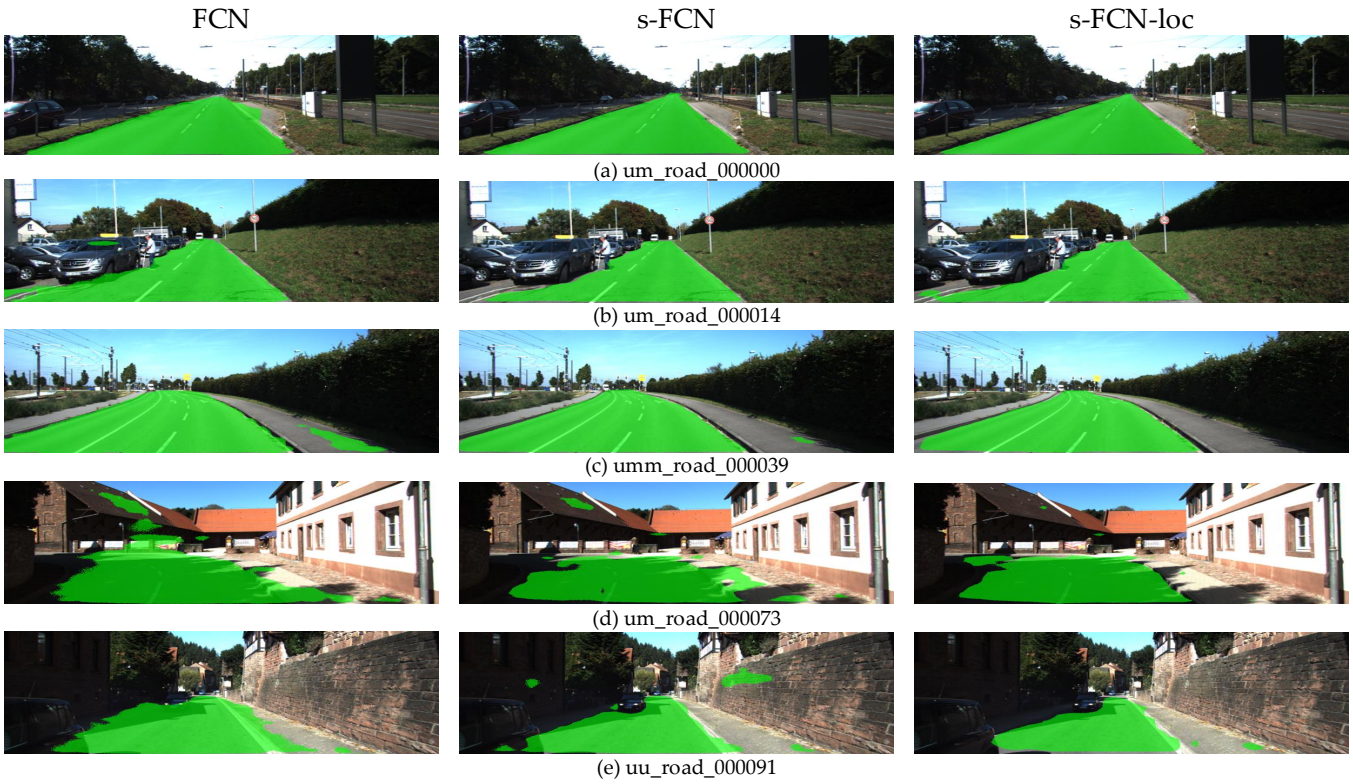


Fig. 5. Exemplar results of the different models (from left to right are: the original FCN, s-FCN and s-FCN-loc) on our randomly selected validation set. The green region is the predicted road region.

of concat_pool4's output. For the classification threshold τ, we set it as a default value of 0.5.

For evaluation, the above metrics are computed in the BEV space in KITTI dataset and in the perspective images in OC dataset.

The experimental environment is equipped with Intel(R) CPU Xeon(R) E5-2697 v2 @ 2.70GHz, 128GB RAM, and four NVIDIA Tesla K80 GPUs. As for the software environment, we modify the standard Caffe[4] by merging the #2016[5] pull request (PR) of Caffe for saving memory during training process.

D. KITTI Dataset: Performance on Validation Set

Since the KITTI website only allows the test data to be used strictly for reporting the final results, the stepwise models are evaluated on validation set for showing their effectiveness, and all of the stage results are evaluated on the validation set. Moreover, we also list the result of our full version on the benchmark server to compare with other popular methods in the next subsection.

Table II presents the four metrics (F1-measure, accuracy, precision and recall) of different models on the validation set. Through quantitative results, our proposed full version ("s-FCN-loc") achieves the best result on the four criteria. In addition, we find each criterion has been improved to some extent except the recall rate, which demonstrates the effectiveness of our proposed siamesed FCN and location prior incorporation.

[4]http://caffe.berkeleyvision.org/
[5]https://github.com/BVLC/caffe/pull/2016

TABLE II
COMPARISON OF DIFFERENT STEPWISE MODEL (THE ORIGINAL FCN-16S, S-FCN AND S-FCN-LOC) ON OUR SELECTED VALIDATION SET (IN %).

Methods	F1-measure	Acc.	Pre.	Rec.
Baseline(FCN)	92.53	97.58	89.40	95.90
s-FCN	94.60	98.31	93.64	95.60
s-FCN-loc	**95.38**	**98.56**	**94.29**	**96.48**

In order to analyze the detection performance further and intuitively, Fig. 5 displays the visualization results of road detection from UM, UMM and UU category. From the first three rows, FCN's results are unclean, especially at the road boundary. For example, the distant sidewalk is mistakenly recognized as road in the third row. By comparison, however, s-FCN and s-FCN-loc are sensitive to the contours of objects, which can segment road region accurately. In the last two sets of exemplars, some building regions are mistaken for road by FCN and s-FCN. The positions of those regions that appear in images are rarely where the road locates. As we can see from the results of the third column, s-FCN-loc incorporating location priors alleviates this problem. These results give a hint that the proposed s-FCN and s-FCN-loc are more effective than the original FCN.

Fig. 6. Exemplar results on the KITTI server. The green, blue and red regions denote respectively true positives, false positives and false negatives.

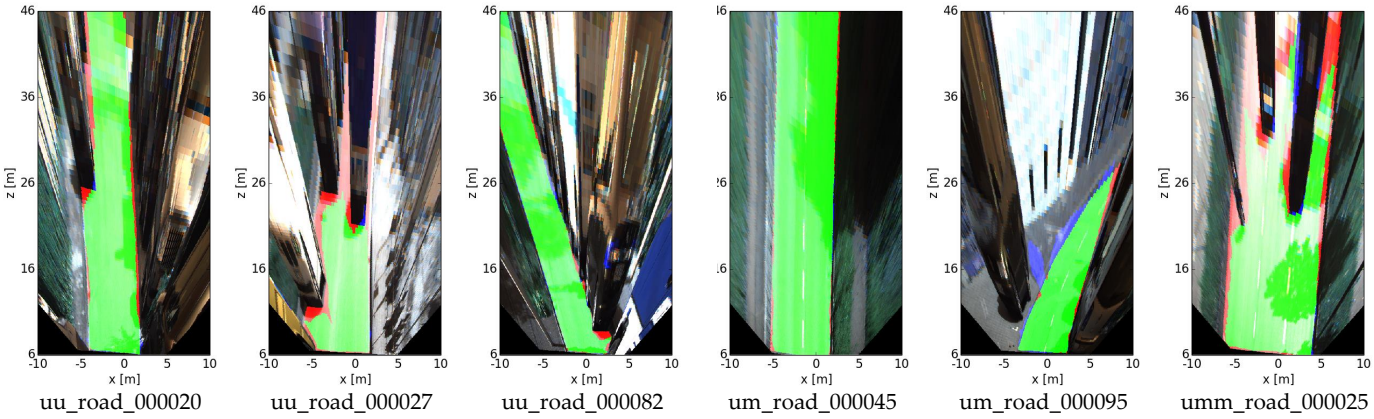


Fig. 7. Exemplar BEV space results on the KITTI server. Similarly, the green, blue and red regions denote respectively true positives, false positives and false negatives. A BEV representation covering $[-10m, +10m]$ in lateral(x) direction and $[+6m, +46m]$ in longitudinal (z) direction is used for evaluation.

E. KITTI Dataset: Performance on the Benchmark

For comparing our proposed s-FCN-loc with other popular methods, we submitted the final results to the KITTI server. Note that the server evaluates the algorithm performance in the BEV space. Table III shows the results of the first ten real-name submissions [6] and ours in the Urban Road category. Our method achieves a competitive (the second place) result of Max F-measure 93.26%, which does not differ much from the best 93.43% of DDN [25]. In the listed methods, DDN [25], FTP [23], FCN_LC [24], StixelNet [37] and MAP [23] are deep learning methods and only take advantage of RGB information; NNP [5], FusedCRF [6] and ProbBoost [7] exploit 3D information such as stereo vision and LIDAR data; HIM [38] and CB [18] make use of hand-crafted features to detect road region. In addition to the max F-measure, the results of other four criteria are in the top three. As for runtime, the proposed method is the 4-th place in all 11 algorithms. Compared with the faster methods, the proposed method is superior to them according to the max F-measure. In general, our method is more competitive than other mainstream algorithms in terms of the detection accuracy and the time performance.

[6]The leaderboard on the KITTI server includes some anonymous submissions. As for these anonymous submissions, because without their detail information, we do not list them in this paper. It is noted that the proposed model obtains the 8-th prize in all 52 submissions.

Fig. 6 shows our final results on the KITTI benchmark server. The green, blue and red regions denote respectively true positives, false positives and false negatives. As we can see from the displayed exemplars, our proposed "s-FCN-loc" model has strong generalization ability from the training set to the testing set. From the "uu_road_000027" and "uu_road_000082", however, we find our model still cannot segment accurately in the small corner of road region. The main : deep neural networks output small-size feature map, the receptive field of which is too large to describe the independent features for the small regions.

Fig. 7 shows our BEV space results on the KITTI benchmark server. The 400×800px BEV image represents the $20m \times 46m$ (meters) real world. From the result of the original "umm_road_000025" image (in Fig. 6), the performance of our method is good, and only the distant road region can not be segmented accurately. However, in the BEV space, the drawback is magnified. The same phenomenon also exists in other images, such as the "uu_road_000027" and "um_road_000095".

Like the proposed s-FCN-loc, FTP [23] and FCN_LC [24] also belong to Fully Convolutional Networks. But they process small image patches. FTP adopts the traditional FCN provided by DeepLab. FCN_LC designs a small FCN model based on Network-in-Network (NiN) architecture and takes advantage

TABLE III
LEADERBOARD OF THE TOP-10 REAL-NAME ALGORITHMS ON THE URBAN ROAD CATEGORY ON THE KITTI VISION BENCHMARK SUITE SERVER (IN %). THE INPUT SOURCE THAT CORRESPONDS TO THE ✓ IS EXPLOITED BY THE ALGORITHMS. THE RED, BLUE AND GREEN FONTS RESPECTIVELY REPRESENT THE FIRST, SECOND AND THIRD PLACE IN THE CORRESPONDING COLUMN.

Methods	Input Sources			Metrics					Runtime
	RGB	Stereo	Laser	Max F-measure	Precision	Recall	FPR	FNR	
DDN [25]	✓			93.43	95.09	91.82	2.61	8.18	2s
Ours: s-FCN-loc	✓			93.26	94.16	92.39	3.16	7.61	0.4s
FTP [23]	✓			91.61	91.04	92.20	5.00	7.80	0.28s
FCN_LC [24]	✓			90.64	90.87	90.72	5.02	9.28	0.03s
HIM [38]	✓			90.07	91.62	89.68	4.52	10.32	7s
NNP [5]	✓	✓		89.68	89.67	89.68	5.69	10.32	5s
StixelNet [37]	✓			89.12	85.80	92.71	8.45	7.29	1s
CB [18]	✓			88.97	89.50	88.44	5.71	11.56	2s
FusedCRF [6]	✓		✓	88.25	83.62	93.44	10.08	6.56	2s
MAP [23]	✓			87.80	86.01	89.66	8.04	10.34	0.28s
ProbBoost [7]	✓	✓		87.78	86.59	89.01	7.60	10.99	150s

of large amounts of contextual information. Compared with them, the proposed s-FCN-loc obtains the best on all four criteria according to Table III. As for visualization results, s-FCN-loc can more accurately segment the road boundary than FTP and FCN_LC. In addition, the mistaken detection are reduced in s-FCN-LC, which is caused by without location priors in the results of FTP and FCN_LC.

F. OC Dataset: Performance

TABLE IV
COMPARISON OF DIFFERENT APPROACHES ON OC TESTING SET (IN %). THE BOLD FONTS REPRESENT THE BEST IN THE CORRESPONDING COLUMN.

Methods	F1-measure	Acc.	Pre.	Rec.
FCN-32s [10]	94.95	96.07	94.68	95.22
FCN-16s [10]	96.97	97.65	96.66	97.30
FCN-8s [10]	97.31	97.89	96.56	98.06
Seg-net [11]	96.56	97.35	96.86	96.27
ENet [39]	96.21	97.02	94.85	97.61
Our Methods:				
s-FCN	97.46	98.02	97.02	**97.91**
s-FCN-loc	**97.56**	**98.10**	**97.24**	97.88

The results of FCN-32/16/8s [10], Seg-net [11], ENet [39] and our models are listed in Table IV. FCN-32/16/8s [10] are Fully Convolutional Networks, and the last two models combine deep and shallow layers' feature maps to obtain finer results; Seg-net [11] consists of encoder (a Fully Convolutional Networks) and decoder (a DeConvolutional Networks) architecture to predict pixel-wise label. Like Seg-net [11], ENet [39] is also a symmetric encoder-decoder network, which has a smaller architecture than Seg-net [11]. Among these algorithms, FCN-32/16/8s [10] and Seg-net [11] adopt the VGG-16 net [26] as a pre-trained model. From the table, we can see our full version ("s-FCN-loc") achieves the best result on the first three criteria. For another criterion - recall, the best performance belongs to "s-FCN". It can be seen, after incorporating location priors in s-FCN, the false detection is reduced but the missing detection is increased. In addition, from the last rows, the improvement of incorporating location priors is not significant on OC dataset than KITTI dataset. The main reason is: the distribution area of road in OC dataset is so variable that some infrequent road regions are wrongly removed.

Four typical visual results are shown in Fig. 8 to intuitively explain the effects of the siamesed FCN and incorporating location priors. From the "input_103" and "input_433", the original FCN cannot segment the road boundaries accurately, but the s-FCN and s-FCN-loc alleviate this problems effectively. In the results of "input_002" and"input_287", the original FCN and s-FCN take some inconceivable regions for road (the positions of these regions are where the road is rarely located). After incorporating location priors in s-FCN, the phenomenon is greatly alleviated.

G. Analysis of Convergent Speed

Fig. 9 illustrates the trends of convergence for three different models on the two datasets. We find the convergent speeds of s-FCN and s-FCN-loc are faster than the original FCN, and the curve lines of s-FCN and s-FCN-loc are very close during the training process. As for the KITTI dataset, the original FCN converges after $240,000$ iterations, but the proposed s-FCN and s-FCN-loc only need $80,000$ iterations to converge. In terms of iteration number, the convergent speeds of the latter two are about 70% faster than that of the original FCN. As a matter of fact, it is unfair to measure the convergent speed of each model by iteration number, because the computation time of each iteration is not equable for different models. Since the original FCN has only one stream, the time of its one iteration is only half of that of s-FCN and s-FCN-loc. Even so, the convergent speeds of s-FCN and s-FCN-loc are still 30% faster than the original FCN according to the overall training time. It's worth mentioning that the difference of convergent speeds is more larger in the initial stage (the first 1,000 iterations) of training. Similarly, the consistent phenomena also appear on the OC dataset.

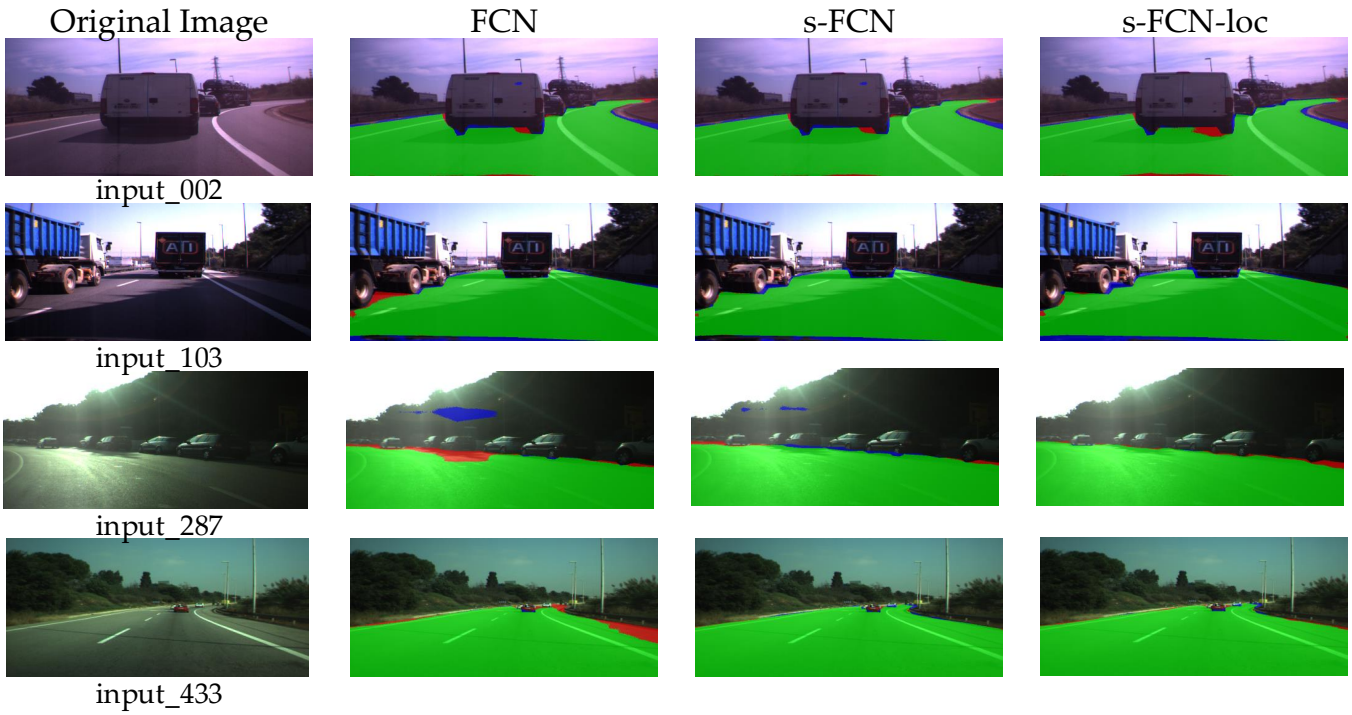
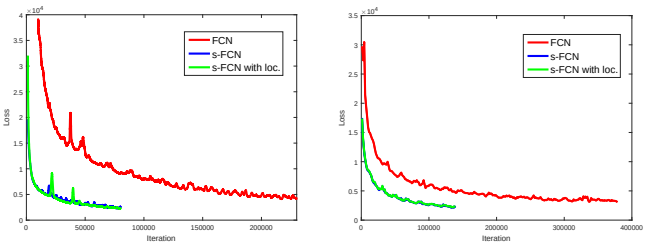


Fig. 8. Exemplar results on OC testing set. The green, blue and red regions denote respectively true positives, false positives and false negatives.



(a) KITTI dataset. (b) OC dataset.

Fig. 9. The convergent trends of FCN (red), our proposed s-FCN (blue) and s-FCN-loc (green) during the training process. (a): KITTI dataset; (b): OC dataset.

The above phenomenon demonstrates that effective spatial structures and contour information speed up the training models. In essence, the semantic contour maps are regarded as higher-level feature than raw RGB images. The neural network can easily learn more effective semantic representation from highly structured contour maps, which guides the model to convergence more quickly. So it saves more training time than traditional single-stream network. Nevertheless, the neural network can not extract features from only semantic contour maps because it loses a lot of detailed and colorful information. Thus, it needs to have two streams to handle RGB images and semantic contour maps, respectively.

H. Comparison of the Different Location Feature Maps

When incorporating location priors in s-FCN, two sizes of feature maps are candidates, namely 16×16 and 44×44 location maps. In Fig. 4, we can find that the 44×44 feature map has more accurate location priors than the 16×16 feature map. For an input image (500×500), the more large-size location map has the smaller respective filed so that the location map can describe more diverse and irregular shape. Thus, the location of the distant road and the small corner road region can be fine represented in the former than the latter. And we believe that the former can achieve more accurate results than the latter. In order to confirm our conjecture, further contrast experiments (s-FCN with 16×16 and 44×44 location maps) are conducted on the two dataset. Table V shows the quantitative results of the two sizes of feature maps in the s-FCN-loc. From the results, the s-FCN with 44×44 feature map can achieve a higher F1-measure than the s-FCN with 16×16 feature map. Therefore, s-FCN-loc incorporates the 44×44 location feature map for the higher performance.

TABLE V
QUANTITATIVE COMPARISON OF S-FCN-LOC WITH DIFFERENT LOCATION PRIORS (16×16 AND 44×44 FEATURE MAPS) ON THE TWO DATASETS. THE BOLD FONTS REPRESENT THE BEST PERFORMANCE IN THE CORRESPONDING COLUMN.

Methods	F1-measure	Acc.	Pre.	Rec.
KITTI dataset				
s-FCN-loc(16×16)	94.76	98.42	**96.54**	93.04
s-FCN-loc(44×44)	**95.38**	**98.56**	94.29	**96.48**
OC dataset				
s-FCN-loc(16×16)	97.50	98.05	96.80	**98.22**
s-FCN-loc(44×44)	**97.56**	**98.10**	**97.24**	97.88

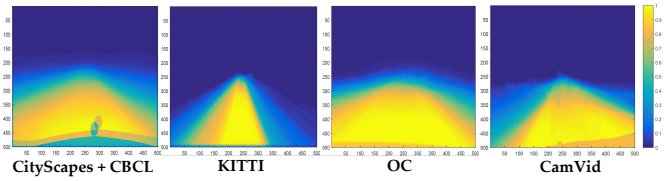


Fig. 11. The frequencies of road distribution on the four datasets: CityScapes + CBCL, KITTI, OC and CamVid.

I. Generalization Analysis

Since the above datasets are too small (579, 755 images in KITTI and OC dataset, respectively) for deep learning methods, the experiments are conducted in order to demonstrate the generalization of the proposed s-FCN-loc. To be specific, the training dataset is a combination of CityScapes's 2,975 [40] and CBCL StreetScenes's 3,547 images [41], totaling 6,522 training samples. The testing is conducted on the KITTI, SC, Cityscapes and CamVid [42] datasets to show the generalization of the proposed methods.

Fig. 10 illustrates the generalization of the three models (FCN, s-FCN and s-FCN-loc). Overall, the generalization of s-FCN and s-FCN-loc is better than that of FCN. As for the original FCN, the performance on different testing data is very poor. The essential reasons are: 1) the road has not constant shapes and structures like vehicles or pedestrians, so the network tend to learn the local appearance features (such as texture, color information); 2) the different cities adopt the different materials to build roads, and the different grades of the road need different materials, which cause the appearance features are distinct. For the proposed s-FCN and s-FCN-loc, both of them consider the higher structured information of the global scene, which is scarcely affected by changes of scenes. And the network can learn the robust structured features to represent the objects, which is an important complement. Thus, the generalization abilities of them is stronger than the original FCN.

Furthermore, we find that the generalization capacity of s-FCN-loc is weaker than s-FCN on KITTI and CamVid datasets. The substantial reason is that the location priors are different because of the different camera's properties. Fig. 11 shows the frequencies of road distribution on the four datasets. The location priors of KITTI and CamVid are quite distinct from that of the combination of CityScapes and CBCL. Thus, the four metrics of s-FCN-loc are clearly inferior to that of s-FCN. On the contrary, the location prior of OC is similar to that of the combination dataset. Hence, the performance of s-FCN-loc is superior to s-FCN, which is consistent with Section IV-D and IV-F. In summary, given plenty of diversified training data from the camera with the same parameters, s-FCN-loc will work better than s-FCN.

J. The Effects of Contour Maps vs. Depth Maps

Generally, providing extra data sources may prompt the accuracy and convergent speed for the same model on some tasks. However, we think the contour map has instinctive advantages for road detection than other data sources: 1) the contour map is a processed, highly structured feature, which can represent the edge intensity of the global scene; 2) it can significantly improve the segmentation of the road boundaries; 3) the generation of it is easy and super real-time.

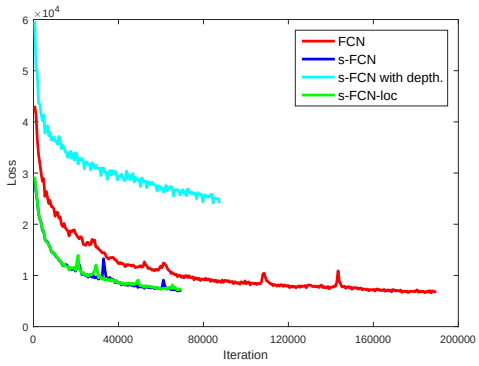


Fig. 12. The convergent trends of FCN (red), s-FCN (blue) s-FCN with depth (cyan) and s-FCN-loc (green) during the training process on CityScapes Dataset.

TABLE VI
COMPARISON OF DIFFERENT STEPWISE MODEL (THE ORIGINAL FCN-16S, S-FCN, S-FCN WITH DEPTH AND S-FCN-LOC) ON CITYSCAPES VALIDATION SET (IN %).

Methods	F1-measure	Acc.	Pre.	Rec.
Baseline(FCN)	94.68	96.46	93.69	95.70
s-FCN	95.15	96.74	93.37	**97.01**
s-FCN with depth	93.47	95.57	90.90	96.19
s-FCN-loc	**95.36**	**96.92**	**94.63**	96.11

Furthermore, we compare the effects of contour map and depth information on CityScapes Dataset. CityScapes is a unique large dataset that contains 2,975 training samples, and provides the RGB data as well as the depth information. The s-FCN's contour input is replaced with depth input, which is called as "s-FCN with depth". Quantitative results are listed in Table VI. In terms of the four metrics, s-FCN thoroughly defeats the s-FCN with depth. In addition to accuracies, the training time of s-FCN is less than that of s-FCN with depth from Fig. 12. To be specific, the learning rate is 10^{-12} during training s-FCN with depth and 10^{-10} in other models. When given learning rate of 10^{-10}, the s-FCN with depth model cannot converge. Actually, since the depth maps are raw data, the model is hard to quickly learn the effect representation from it. On the contrary, contour maps have three above-mentioned advantages to improve the original FCN and speed training up.

V. CONCLUSIONS

This paper presents an s-FCN-loc model based on VGG-net for road detection, which is able to learn discriminative features of road boundaries and location priors. Specifically, the RGB-channel image, the semantic contour and the location prior are simultaneously integrated into a neural network without any postprocessing. Stepwise experimental results verify the effectiveness of each component in the proposed method.

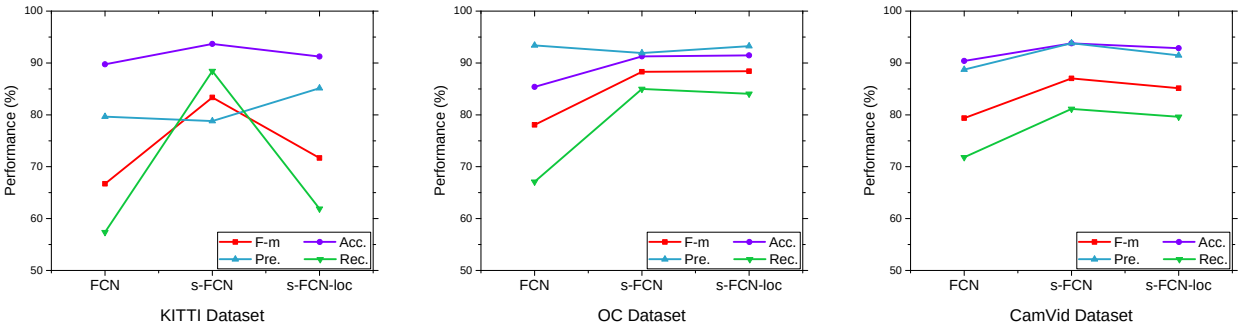


Fig. 10. Generalization on the different testing datasets: KITTI, OC and CamVid. All models are trained on the combination of CityScapes and CBCL StreetScenes datasets.

We also find that the proposed s-FCN-loc converges faster than the original FCN during the training stage, which saves more training time.

In the proposed s-FCN-loc, the contour stream can also be added to other networks (for instance, CNN and DeConv NN) to promote the capacity. Thus, we will transform the thought to other dense prediction tasks, such as saliency detection and semantic image segmentation in the future.

REFERENCES

[1] M. Revilloud, D. Gruyer, and M.-C. Rahal, "A new multi-agent approach for lane detection and tracking," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 3147–3153.

[2] X. Wen, L. Shao, W. Fang, and Y. Xue, "Efficient feature selection and classification for vehicle detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 3, pp. 508–517, 2015.

[3] A. Angelova, A. Krizhevsky, and V. Vanhoucke, "Pedestrian detection with a large-field-of-view deep network," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 704–711.

[4] Y. Yuan, D. Wang, and Q. Wang, "Anomaly detection in traffic scenes via spatial-aware motion reconstruction," *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, no. 99, pp. 1–12, 2016.

[5] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun, "3d object proposals for accurate object class detection," in *Advances in Neural Information Processing Systems*, 2015, pp. 424–432.

[6] L. Xiao, B. Dai, D. Liu, T. Hu, and T. Wu, "Crf based road detection with multi-sensor fusion," in *2015 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2015, pp. 192–198.

[7] G. B. Vitor, A. C. Victorino, and J. V. Ferreira, "A probabilistic distribution approach for the classification of urban roads in complex environments," in *Workshop on IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[8] X. Wang, L. Xu, H. Sun, J. Xin, and N. Zheng, "On-road vehicle detection and tracking using mmw radar and monovision fusion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 2075–2084, 2016.

[9] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1915–1929, 2013.

[10] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[11] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *arXiv preprint arXiv:1511.00561*, 2015.

[12] J. Gao, Q. Wang, and Y. Yuan, "Embedding structured contour and location prior in siamesed fully convolutional networks for road detection," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 219–224.

[13] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 5, pp. 898–916, 2011.

[14] R. Xiaofeng and L. Bo, "Discriminatively trained sparse code gradients for contour detection," in *Advances in neural information processing systems*, 2012, pp. 584–592.

[15] P. Dollár and C. L. Zitnick, "Structured forests for fast edge detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1841–1848.

[16] C.-A. Brust, S. Sickert, M. Simon, E. Rodner, and J. Denzler, "Convolutional patch networks with spatial prior for road detection and urban scene understanding," *arXiv preprint arXiv:1502.06344*, 2015.

[17] J. M. Á. Alvarez and A. M. Lopez, "Road detection based on illuminant invariance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 1, pp. 184–193, 2011.

[18] C. C. T. Mendes, V. Frémont, and D. F. Wolf, "Vision-based road detection using contextual blocks," *arXiv preprint arXiv:1509.01122*, 2015.

[19] X. Lu, "Self-supervised road detection from a single image," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2989–2993.

[20] Q. Wang, J. Fang, and Y. Yuan, "Adaptive road detection via context-aware label transfer," *Neurocomputing*, vol. 158, pp. 174–183, 2015.

[21] Y. Yuan, Z. Jiang, and Q. Wang, "Video-based road detection via online structural learning," *Neurocomputing*, vol. 168, pp. 336–347, 2015.

[22] J. M. Alvarez, T. Gevers, Y. LeCun, and A. M. Lopez, "Road scene segmentation from a single image," in *European Conference on Computer Vision*. Springer, 2012, pp. 376–389.

[23] A. Laddha, M. K. Kocamaz, L. E. Navarro-Serment, and M. Hebert, "Map-supervised road detection," in *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2016, pp. 118–123.

[24] C. C. T. Mendes, V. Frémont, and D. F. Wolf, "Exploiting fully convolutional neural networks for fast road detection," in *2016 IEEE International Conference on Robotics and Automation, ICRA 2016, Stockholm, Sweden, May 16-21, 2016*, 2016, pp. 3174–3179.

[25] R. Mohan, "Deep deconvolutional networks for scene parsing," *arXiv preprint arXiv:1411.4101*, 2014.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[27] G. L. Oliveira, W. Burgard, and T. Brox, "Efficient deep models for monocular road segmentation," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 4885–4891.

[28] Y. He, H. Wang, and B. Zhang, "Color-based road detection in urban traffic scenes," *IEEE Transactions on intelligent transportation systems*, vol. 5, no. 4, pp. 309–318, 2004.

[29] M. Yu, L. Liu, and L. Shao, "Structure-preserving binary representations for rgb-d action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 8, pp. 1651–1664, 2016.

[30] J. M. Álvarez, A. M. López, T. Gevers, and F. Lumbreras, "Combining priors, appearance, and context for road detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 3, pp. 1168–1178, 2014.

[31] W. Song, L. Liu, X. Zhou, and C. Wang, "Road detection algorithm of integrating region and edge information," in *Proceedings of the International Conference on Artificial Intelligence and Robotics and the International Conference on Automation, Control and Robotics Engineering*. ACM, 2016, p. 14.

[32] J.-H. Nam, S.-H. Yang, W. Hu, and B.-G. Kim, "A robust real-time road detection algorithm using color and edge information," in *International Symposium on Visual Computing*. Springer, 2015, pp. 532–541.
[33] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *European Conference on Computer Vision*. Springer, 2014, pp. 391–405.
[34] S. Liu, J. Yang, C. Huang, and M.-H. Yang, "Multi-objective convolutional learning for face labeling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3451–3459.
[35] J. Fritsch, T. Kuehnl, and A. Geiger, "A new performance measure and evaluation benchmark for road detection algorithms," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE, 2013, pp. 1693–1700.
[36] J. M. Alvarez, T. Gevers, and A. M. López, "Road detection by one-class color classification: Dataset and experiments," *arXiv preprint arXiv:1412.3506*, 2014.
[37] D. Levi, N. Garnett, E. Fetaya, and I. Herzlyia, "Stixelnet: A deep convolutional network for obstacle detection and road segmentation." BMVC, 2015.
[38] D. Munoz, J. A. Bagnell, and M. Hebert, "Stacked hierarchical labeling," in *European Conference on Computer Vision*. Springer, 2010, pp. 57–70.
[39] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *arXiv preprint arXiv:1606.02147*, 2016.
[40] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.
[41] S. M. Bileschi, "Streetscenes: Towards scene understanding in still images," Ph.D. dissertation, Citeseer, 2006.
[42] G. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," *Computer Vision–ECCV 2008*, pp. 44–57, 2008.

Qi Wang (M'15-SM'15) received the B.E. degree in automation and Ph.D. degree in pattern recognition and intelligent system from the University of Science and Technology of China, Hefei, China, in 2005 and 2010 respectively. He is currently a Professor with the School of Computer Science, with the Unmanned System Research Institute, and with the Center for OPTical IMagery Analysis and Learning, Northwestern Polytechnical University, Xi'an, China. His research interests include computer vision and pattern recognition.

Junyu Gao received the B.E. degree in computer science and technology from the Northwestern Polytechnical University, Xi'an 710072, Shaanxi, P. R. China, in 2015. He is currently pursuing the Master degree from Center for Optical Imagery Analysis and Learning, Northwestern Polytechnical University, Xian, China. His research interests include computer vision and pattern recognition.

Yuan Yuan (M'05-SM'09) is currently a full professor with the School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an 710072, Shaanxi, P. R. China. She has authored or coauthored over 150 papers, including about 100 in reputable journals such as IEEE Transactions and Pattern Recognition, as well as conference papers in CVPR, BMVC, ICIP, and ICASSP. Her current research interests include visual information processing and image/video content analysis.