Fast Road Scene Segmentation using Deep Learning and Scene-based Models

Conference Paper · December 2016

DOI: 10.1109/ICPR.2016.7900220

6 authors, including:

Vijay John
Toyota Technological Institute
39 PUBLICATIONS **199** CITATIONS
SEE PROFILE

Chunzhao Guo
Toyota Central R & D Labs., Inc.
44 PUBLICATIONS **264** CITATIONS
SEE PROFILE

Kiyosumi Kidono
Toyota Central R & D Labs., Inc.
31 PUBLICATIONS **256** CITATIONS
SEE PROFILE

Hossein Tehrani
Denso Corporation
39 PUBLICATIONS **283** CITATIONS
SEE PROFILE

Fast Road Scene Segmentation using Deep Learning and Scene-based Models

Vijay John
Toyota Technological Institute
Email: vijayjohn@toyota-ti.ac.jp

Chunzhao Guo
Toyota Central R & D Labs
Email: czguo@mosk.tytlabs.co.jp

Seiichi Mita
Toyota Technological Institute
Email: smita@toyota-ti.ac.jp

Kiyosumi Kidono
Toyota Central R & D Labs
Email: kidono@mosk.tytlabs.co.jp

Hossein Tehrani
Denso Corporation
Email: hossein_tehrani@denso.co.jp

Kasuhiza Ishimaru
Nippon Soken
Email: kazuhisa_ishimaru@soken1.denso.co.jp

***Abstract*—Pixel-labeling approaches using semantic segmentation play an important role in road scene understanding. In recent years, deep learning approaches such as the de-convolutional neural network have been used for semantic segmentation, obtaining state-of-the-art results. However, the segmentation results have limited object delineation. In this paper, we adopt the de-convolutional neural network to perform the semantic segmentation of the road scene using colour and depth information. Moreover, we improve the network's limited object delineation within a computationally efficient framework using novel features that are learnt at the pixel-level and patch-level for different road scenes. The patch-level features represent the road scene geometry. On the other hand, the learnt pixel-level features represent the appearance and depth information. The features learnt for the different road scenes are indexed with the scene's pre-defined label. Following the indexing, the random forest classifier is trained to retrieve the relevant geometric and appearance-depth features for a given road scene. The retrieved features are then used to refine identified error regions in the initial semantic segmentation estimate. Our proposed algorithm is evaluated on an acquired dataset and compared with state-of-the-art baseline algorithms. We also perform a detailed parametric evaluation of our proposed framework. The experimental results show that our proposed algorithm reports better accuracy.**

I. Introduction

In recent years, the development of intelligent vehicle and autonomous driver assistance systems (ADASs) have received significant attention from various researchers. In ADAS, the semantic segmentation of road scenes is an important research problem for road environment perception. Semantic image segmentation involves the assignment of labels to all the pixels in an image. However, the robust and accurate estimation of the pixel labels are not straightforward. Some of the challenging issues include the variations in the illumination and appearance, apart from occlusions.

In recent years, deep learning methods, which have demonstrated state-of-the-art results on image classification tasks, have been modified and used for pixel-labeling [1]. These modified deep learning algorithms such as the deconvolutional neural network [2], while reporting state-of-the-art results, have limited object delineation [3]. In this paper, we address this issue within the deconvolutional neural network using features learnt at the patch-level and pixel-level for different road scenes. The de-convolution neural network is trained to obtain an initial estimate of the pixel-level labels using depth and colour information, which are represented using a novel hue, saturation and depth-based feature representation. Novel geometric features are learnt at the patch-level, and are extracted from the manually labelled road scene images. On the other hand. the appearance-depth features are learnt at the pixel-level, and are extracted from the deconvolutional network's feature maps. Both these learnt features are indexed using the road scene's pre-defined scene label. We term the geometric and appearance-depth features as the refinement features. For a given road scene, to perform the refinement, we retrieve the relevant refinement features using a trained random forest scene classifier. The random forest classifier is trained using the road scene features, extracted from the deconvolutional network, and their scene labels. The retrieved refinement features are used to refine the identified error regions in the initial semantic segmentation estimate. The refinement is restricted to identified error regions to enhance the computational efficiency. We validate our proposed algorithm using an acquired dataset. We perform a comparative analysis of our algorithm with baseline algorithms, and report better semantic segmentation accuracy with fast processing time. We also perform a detailed parameter analysis of our algorithm and report our observations. To the best of our knowledge, our main contribution to the literature are as follows: (a) application of the deconvolution network for road scene semantic segmentation; (b) feature representation of the colour and depth information within a novel hue, saturation and depth image; (c) utilization of novel scene-based refinement features to address the limited object delineation in deep learning-based semantic segmentation frameworks. We structure the remainder of the paper as follows. In Section II we report the literature review. We present our algorithm in Section III, and the experimental results in Section IV. Finally, we conclude the paper in Section V.

II. Related Work

Semantic segmentation being an important research topic has received significant attention from the research community.

Typically, two-stage patch-level classification approaches are utilized to predict the class label for the central pixel [4]–[6]. In the first step, classifiers such as the random forest [5] or boosting [6], generally, use appearance-based features extracted from the image patches to obtain an initial estimate of the pixel-level labels. These labels are then refined using models such as the Conditional Random Field (CRF). For example, in the work by Sturgess et al. [6], appearance and structure-for-motion features are used within a boosting framework to generate the initial pixel-level label estimates, which are then refined using the CRF. In an alternative approach, researchers employ the geometric scene constraints, or context, to obtain robust pixel-level label estimates [7], [8]. A survey of the different context-based segmentation approaches are presented in the work of Galleguillos et al. [7].

To enhance the robustness and accuracy of the aforementioned approaches, researchers have started to adapt the deep learning framework for semantic segmentation [1]–[3], [9]. In the work by Long et al. [1], the CNN-based image classifier is adapted to perform semantic segmentation using color images by replacing the fully connected layers with the fully convolutional layers. Noh et al. [2] extend this approach by proposing the de-convolutional and unpooling layers. Both approaches report state-of-the-art segmentation accuracies, but are limited by the CNN's object delineation. These limitations are subsequently addressed by utilizing the dense CRF to model the spatial relationship between the image pixels [3], [9], and refine the pixel-level label estimates. Chen [9] use the CRF to refine the CNN-based segmentation as a post-processing step. While, Zheng at al. [3] formulate a deep learning framework, where both the CNN and CRF-based recurrent neural network are combined and learnt end-to-end. The CRF-based approaches report significant improvements on the semantic segmentation results, but are limited by their computational complexity.

Compared to the literature, we utilise the de-convolution neural network framework with both color and depth images to obtain an initial estimate of the pixel-level labels. These labels are then refined using the refinement features.

III. Algorithm

Given a road scene represented with colour and depth images with N pixels, the semantic segmentation algorithm assigns a pixel label k, from a set of K pre-defined pixels labels, to each pixel. In our paper, we have four pixel labels representing the *road*, *sky*, *ceiling* and *obstacles* in the road scene. The *obstacles* class includes the vehicles, pedestrians, buildings, pavements and other vertical objects in the road scene.

In our proposed algorithm, we first, utilize the de-convolution neural network (Deconvnet) [2] to perform an initial semantic segmentation estimate of the road scene. The initial estimates are refined in the subsequent steps using patch-level and pixel-level refinement features. The refinement features are learnt for different road scenes. Each road scene along with the extracted refinement features are indexed with pre-defined scene labels s. The scene labels are obtained from a set of S pre-defined scene labels. In this paper, we define 6 road scene labels, namely, *cityroad*, *highway*, *backalley*, *parking*, *tunnel*, and *intersection*. The patch-level refinement feature represent the road scene geometry, and are extracted from the manually annotated road scene labels. The pixel-level refinement features encapsulate the appearance-depth information of the road scene, and are extracted from the Deconvnet's feature maps. In order to reduce the computational complexity, we restrict the refinement to error regions in the initial estimate. The error regions or blobs are identified at the inter-class boundaries of the initial estimate. To refine the error blobs, for a road scene indexed with a scene label, we first retrieve the refinement features with the same scene label using the random forest classifier. Using the retrieved scene label, the refinement features are then used within a layered scheme to generate the final estimate of the pixel-level labels. We next describe the learning and testing phases of our algorithm in greater detail.

A. Learning Phase

The learning phase of the proposed algorithm consists of the (a) fine-tuning the deconvolution network, (b) training the random forest scene classifier and (c) extracting and indexing the refinement features according to the pre-defined scene labels. An overview of the learning phase of the algorithm is presented in Fig 1.

1) Deconvolutional Network: In recent years, the traditional CNN framework used for image recognition has been modified for pixel-level labeling, achieving good results. In this paper, we propose to adapt the deconvolution neural network (Deconvnet) proposed by Noh et al. [2] to obtain an initial estimate of the pixel labels using both the color and depth information. The Deconvnet contains the unpooling and the deconvolutional layers. The unpooling layers represent the reverse operation of pooling. The pooling layers are modified to store the locations of the maximum activations using switch variables. These variables are used by the unpooling layers to regenerate the activations to their original size and pooling location. The sparse unpooling layer output is densified by the deconvolutional layers. This is achieved by performing convolution-like operation with learnt filters. Similar to the convolutional layers, the lower deconvolutional layers learn the low-level feature representation, while the higher layers learn the high-level feature representation.

In this paper, unlike the original Deconvnet which uses only the color information to perform semantic segmentation, we use both the color and depth information. A stereo camera mounted on our intelligent vehicle is used to acquire the left-right color image pair from the road scene. The depth map is then estimated from the image pair using the multipath Viterbi algorithm proposed by Long et al. [10]. The input to our Deconvnet is derived from the left color image and the estimated depth map. The RGB color space of the left image is transformed to the HSV color space, and the hue and saturation channels are extracted. These channels are then combined with

the depth map to form a three channel Hue, Saturation, Depth (HSD) Deconvnet input with dimension $224 \times 224 \times 3$. The three channel input is then used to fine-tune the Deconvnet, which has been pre-trained on the 20-channel PASCAL VOC 2012 segmentation benchmark [2]. The fine-tuned Deconvnet is used to perform the semantic segmentation of the road scene.

The architecture of our fine-tuned Deconvnet is similar to the original Deconvnet, apart from the final output layer. More specifically, 21 neurons in the original output layer are replaced with 4 neurons, corresponding to our 4 object classes. The learning for the convolutional (C1-C5), fully convolutional (FC6, FC7), deconvolutional-fully convolutional layer (DC-FC) and deconvolutional (DC1-DC5) layers of the architecture (Fig 1) are initialised with the corresponding pre-trained weights and biases of the pre-trained Deconvnet. To account for pre-trained initialisation, the learning rates are set to a low value 0.001, while the momentum and learning multipliers for the pre-trained layers are set to $(0.9, 1, 2)$. On the other hand, the final output layer is initialised without the pre-trained weights. Consequently, the momentum and learning multiplier is set to 10 and 20 to facilitate faster learning. Finally, the number of training iterations is set to 5000. Since we fine-tune the pre-trained Deconv network with a three channel input (RGB), we are constrained to using a three channel input. Thus, we encode the hue, saturation and depth information within three input channels to fine-tune the Deconv network.

2) Random Forest-based Scene Classifier: In our algorithm, the random forest framework is used, as a scene classifier, to predict the scene label associated with the given road scene. The random forest classifier is represented using an ensemble of decision trees. The decision trees are trained to perform classification using the techniques of tree bagging and feature bagging. Tree bagging is used to train each decision tree on a random subset of the training dataset. While, feature bagging is used to perform the split at each node in a given tree using a random feature subset, selected from the training subset [11]. The scene classifier is trained using pairs of road scene features and scene labels. The road scene features are represented using the set of feature maps F_{P5} generated by the $P5$ layer in our fine-tuned Deconv network. The $P5$ feature maps at the end of the convolution layers encode the entire road scene, as their local receptive field corresponds to the complete input image.

3) Geometric Features: The geometric features are used within the first layer of our layered refinement scheme. The geometric feature vector encodes the geometrical and spatial relationship between the various pixel labels in a given road scene or image. These spatial relationships can be used to refine or re-assign wrongly labeled pixels. For example, in a typical road scene, the sky region is always above the road region. This relationship can be used to refine or re-assign wrongly labelled sky pixels present below the road labelled pixels. In this paper, we represent this spatial relationship using the distance and orientation between the different pixel labels in the image. To account for computational efficiency, the geometric feature vector is computed at the patch-level using non-overlapping square image patches P and square receptor Q image regions. The receptor image regions are defined to be bigger than the image patches (q=32×32, p=16×16). In the learning phase, the proposed geometric feature for different patches and different road scenes is extracted from the ground truth labeled image. The geometric feature, $\mathbf{g^s}(p)$, for a given scene label s and image patch p is represented as,

$$\mathbf{g_p^s} = [\omega_p^1, \ldots, \omega_p^K] \tag{1}$$

where ω_p^k encodes the inter-class and intra-class spatial relationship in patch p between the k-th pixel label and all other K pixel labels across the Q non-overlapping receptor image regions. The detailed representation of ω_p^k is given as,

$$\omega_p^k = [\mathbf{v}(p,1)_k^1, \ldots, \mathbf{v}(p,q)_k^1, \ldots, \mathbf{v}(p,1)_k^K, \ldots, \mathbf{v}(p,q)_k^K] \tag{2}$$

, where each 2-D entry represents the magnitude and orientation of the mean spatial vector generated between patch p and receptor region q. The mean spatial vector is computed from a set of spatial vectors, which are generated between the centroid pixel coordinate of patch p with dominant pixel label k and the centroid pixel coordinates of the K pixel label regions in q. The orientation of the mean spatial vector corresponds to the orientation between the horizontal axis and the mean spatial vector. An illustration of the geometrical feature vector is given in Fig 2. Note that if a particular pixel label is absent in either p or q, we set the corresponding 2-D entry to zero. For computational efficiency in Eqn 2, we consider all non-dominant pixel labels in patch p to be absent. A size threshold is used to identify the dominant pixel label.

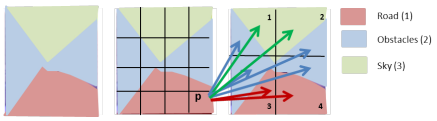


Fig. 2. An illustration of the geometric feature for a road scene label (left) with P patches (middle) and Q receptor regions (right). We represent the mean spatial vector between the patch p and the Q receptor regions. For the patch p the road is the dominant label $k = 1$. Thus, only ω_p^1 in (Eqn 1) has non-zero entries. For the vector ω_p^1 in Eqn 2, in this illustration, we only represent the non-zero mean spatial vectors (examples:$\mathbf{v}(p,1)_1^2$ $\mathbf{v}(p,1)_1^3$.

4) Appearance-Depth Feature: The appearance-depth features are used in the final layer of our layered refinement scheme. The appearance-depth features are extracted from the $C1$ feature maps in our fine-tuned Deconv network. We represent the feature maps as $F_{C1}(h,w,d)$, where h, w and d represent height, width and the number of feature maps. In F_{C1}, for each h and w index, the d-dim vector, $\lambda(h,w)$, across the d feature maps represents the features extracted from its local receptive field in the HSD input (Fig 3). Each $\lambda(h,w)$ is indexed using the centroid pixel coordinate n of the HSD input local receptive field. For a HSD input image with scene label s, the appearance depth feature for the n-th pixel coordinate with pixel label k is represented by $\mathbf{a_n^s}(\mathbf{k})$ using the

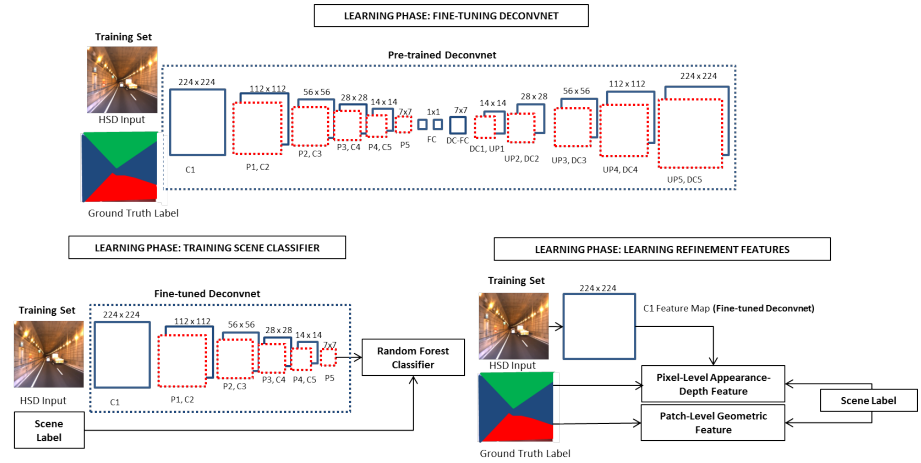
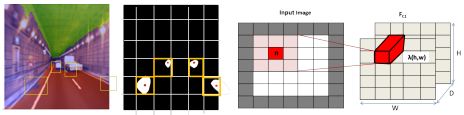


Fig. 1. An overview of the learning phase. In the Deconvnet architecture, P1-P5 and UP1-UP5 represent the max pooling and unpooling layers, respectively, with size 2 and stride 2.

corresponding pixel-indexed $\lambda(h,w)$ in F_{C1}. An illustration of the appearance-depth feature is shown in Fig 3-b.



(a) (b)

Fig. 3. An illustration of the (a) binary error blobs identified from the initial estimate. The yellow boxes and red circle correspond to the error image patches and the error centroids. In (b), we illustrate the appearance-depth feature at pixel n is represented using the corresponding pixel-indexed feature $\lambda(h,w)$ in F_{C1}. Note that the pixel n represents the centroid of the local receptive field.

B. Testing Phase

The testing phase of the proposed algorithm is categorized into the initialization step, pruning step and refinement step. We next explain the different steps in detail (Fig 4).

1) Initialization Phase: First, the fine-tuned Deconv network is used to obtain an initial semantic segmentation estimate for a HSD test input. The Deconv network outputs a pixel-level segmentation score for each pixel label and the pixel label corresponding to the maximum score is assigned to each pixel. Next, we identify the error blobs within the semantic segmentation estimate using the segmentation scores. An empirical segmentation score-based threshold is used to identify the T error blobs at the inter-class boundary location. Finally, we estimate the scene label for the given HSD input by using the corresponding $P5$ layer feature map from the Deconv network within the trained random forest classifier. Utilizing the scene label, the initial estimates are then refined in the layered scheme.

2) Geometric Feature-based Pruning: In the first layer, we perform a patch-based pruning of the pixel labels associated with the T error blobs. As an initial step, for each error blob t, we first identify the image patch p, where it lies. Subsequently, for each identified patch p, we generate a set of K candidate geometric features $\rho_p^s(k)$, from the initial semantic segmentation estimate. Each k candidate geometric feature is generated by considering the k-th pixel label to be the dominant label for the patch p. We then compute the Euclidean distance between the k-th candidate geometric feature and the learnt scene-categorized geometric feature $\mathbf{g_p^s}$. This is given as,

$$D(k) = dist(\rho_p^s(k), \mathbf{g_p^s}) \tag{3}$$

Given the estimated distances $D = \{D\}_{k=1}^{K}$, we then prune the set of pixel labels using a distance-based threshold.

3) Appearance-Depth Feature-based Refinement: In the final refinement layer, we first estimate the centroid pixel coordinates N of the T error blobs. Next, for each centroid pixel coordinate n, we generate the candidate appearance-depth feature, σ_n^s, using the corresponding pixel-indexed λ in the test feature maps F_{C1}. The test feature maps are obtained by convolving the HSD input with the learnt $C1$ filters. An Euclidean distance-based nearest neighbour classifier is used to obtain the final semantic segmentation estimate for each error blob centroid n using the learnt appearance-depth features $\mathbf{a_n^s}(k)$ (Eqn 4) with estimated scene label s and pruned pixel labels K. The pixel labels identified for each error blob centroid is applied to all pixels within the error blob.

$$\hat{k} = \underset{k}{\operatorname{argmin}}\ dist(\sigma_n^s, \mathbf{a_n^s}(k)) \tag{4}$$

IV. Experimental Results

We validate our proposed algorithm using an acquired dataset containing 921 frames with manually annotated ground truth. To validate the algorithms and perform the parameter analysis, we use the 5-fold cross validation technique. Example frames from the dataset are shown in Fig 5. Finally, we implement the algorithm on a Linux machine with Nvidia

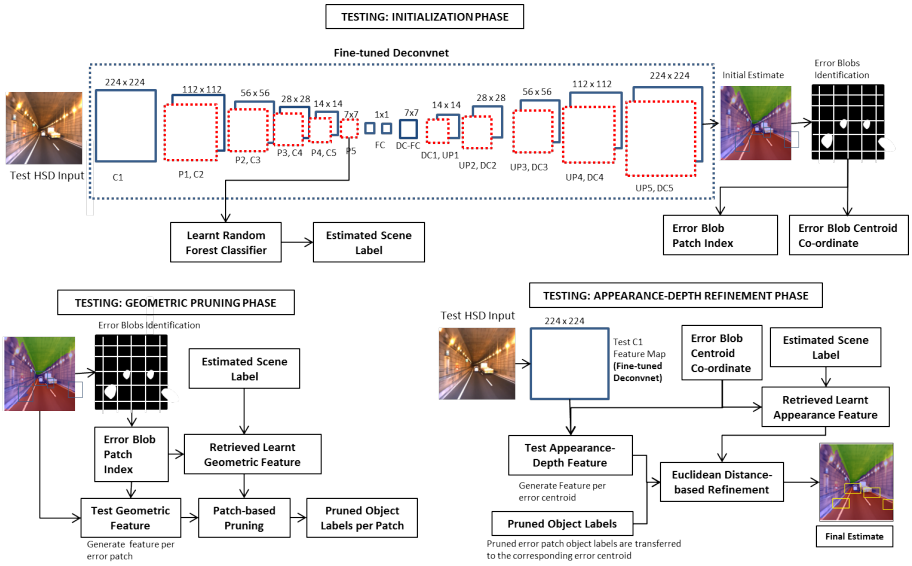


Fig. 4. An overview of the testing phase.

Geforce GTX960 graphics card using the GPU-based Caffe software tool [12] and Matlab.

A. Comparative Analysis

We perform a comparative analysis of our proposed algorithm with the FCN [1] and the original Deconv network [2]. The comparative results are computed from the final estimate of each algorithm's pixel-level labels. An estimated pixel-level label is considered to be detected, if the estimated label is same as the ground truth label. The results as shown in Table I, demonstrate our algorithm's semantic segmentation result, which is better than the baseline algorithm. We report an average computational time of 200ms per frame with 75ms for the Deconvnet estimate, 8ms for the error region identification, 10ms for the scene classification, 25ms for the geometric feature extraction, 75ms for the geometric feature pruning, 1ms for the appearance-depth based refinement.

TABLE I
THE MEAN AND STD. DEV OF THE SEMANTIC SEGMENTATION ACCURACY.

Algorithm	Detection Accuracy
Proposed Algo.	98.9% ± 0.05
Deconv. [2]	98.1% ± 0.02
FCN. [1]	89.1% ± 3.2

B. Parameter Analysis

a) Fine-tuning the Deconv network: In our proposed framework, we fine-tune the original Deconv network with the 5-fold training partitions of the acquired dataset. The fine-tuning mechanism is evaluated by performing a comparative analysis between the fine-tuned Deconv network (FT) and the direct-trained Deconv network (DT). In the direct-trained Deconv network, the architecture of the fine-tuned Deconv network is retained, but the learning is performed without using the pre-trained weight and bias. For the direct training, the learning rate was increased to 0.01 and the iterations was also increased to 10000. The results as tabulated in Table II, show the fine-tuned Deconv network reports better semantic segmentation accuracy than the direct-trained Deconv network. The detection accuracies correspond to segmentation results of the Deconv networks without any refinement.

TABLE II
VARIED NETWORK TRAINING.

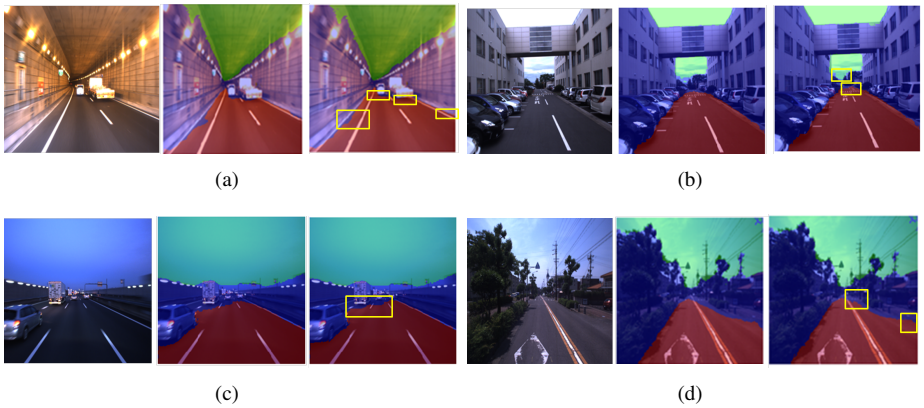
Prop Algo.	Det. Accuracy
FT Deconv	98.7%±0.06
DT Deconv	78.7%±0.3

TABLE III
VARIED NETWORK INPUT.

Prop Algo.	Det. Accuracy
HSD	98.9%±0.05
HSV	98.5%±0.06
RGB	98.5%±0.02

b) Depth-based Feature: Here we validate the proposal to incorporate the depth information along with the color information for the semantic segmentation. A comparative analysis is performed by varying the input to the Deconv network. More specifically, we report the detection accuracies for the HSD input, HSV color space and RGB color space-based input. The semantic segmentation results given in Table III show that incorporating depth information improves the detection accuracies.

c) Layered Refinement: The advantage of the layered refinement framework is validated by comparing the detection accuracies of the proposed algorithm with and without the refinement step. Additionally, a similar experiment is performed for the proposed algorithm with the RGB input and HSV input also. The results in Table IV-V show that the refinement step improves the detection accuracies across the different inputs. The HSV input reported a similar performance to the RGB input.



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

Fig. 5. (a)-(d) Examples of the results of our proposed algorithm. In each figure, we represent the left stereo image (left), the Deconvnet's initial estimate (middle) and our algorithm's final estimate (right). The yellow rectangle in the final estimate images corresponds to the improved boundary segmentation.

TABLE IV
REFINEMENT EVALUATION WITH HSD INPUT.

Prop Algo.	Det. Accuracy
Refine.	98.9%±0.05
No refine.	98.7%±0.06

TABLE V
REFINEMENT EVALUATION WITH RGB INPUT.

Prop Algo.	Det. Accuracy
Refine.	98.5%±0.02
No refine.	98.1%±0.02

TABLE VI
SCENE CLASSIFIER ACCURACY

Algo.	Det. Accuracy
Random Forest	100%±0
Naive Bayesian	98.7%±0.4

TABLE VII
PARAMETER EVALUATION OF THE SCENE-BASED REFINEMENT

Prop Algo.	Det. Accuracy
With Scene	98.9%±0.06
No Scene	98.7%±0.05

d) Scene Classification: By utilizing a 5-fold cross validation on the acquired dataset, we evaluate the proposal to use the random forest for scene classification. A comparative analysis is performed with the naive Bayesian classifier [13]. The results tabulated in Table VI show that the random forest reports 100% accuracy on the acquired dataset, unlike the Naive Bayesian classifier.

e) Scene-based Layered Refinement: The scene-based refinement of the pixel-level labels is an important component of our proposed algorithm. This component is validated by comparing the detection accuracies of the layered refinement step of the proposed algorithm with and without scene-categorized features. For the refinement step without the scene-categorized features, the appearance, depth and geometric refinement features are learnt over the entire dataset. Consequently, the random forest-based scene classifier is not used for this algorithm variation. The results tabulated in Table VII show that utilizing scene-categorized features improves the semantic segmentation accuracy.

V. CONCLUSION

In this paper, we propose a deep learning-based semantic segmentation algorithm for road scenes. The deconvolutional network is utilized to provide an initial semantic segmentation estimate using color and depth inputs. These initial estimates are then refined using scene-categorized appearance, depth and geometric features in a layered framework. The scene label associated with each road scene is retrieved using the random forest-based scene classifier. We evaluate our proposed algorithm on an acquired dataset and perform a comparative analysis with baseline algorithms. We show that our Deconvnet-based algorithm with scene-categorized features report better semantic segmentation results than the baseline algorithms, especially at the inter-class boundaries. We also report a fast computational time of 200*ms* per frame. In our future work, we will evaluate our algorithm with a bigger dataset with more object classes and work towards the real-time implementation of the algorithm.

REFERENCES

[1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *in CVPR*, Nov. 2015.

[2] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation." *CoRR*, vol. abs/1505.04366, 2015.

[3] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr, "Conditional random fields as recurrent neural networks," in *ICCV*, 2015.

[4] A. Ess, T. Mueller, H. Grabner, and L. J. V. Gool, "Segmentation-based urban traffic scene understanding." in *BMVC*, 2009.

[5] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation." in *in CVPR*, 2008.

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

[7] C. Galleguillos and S. Belongie, "Context based object categorization: A critical survey," *CVIU*, vol. 114, no. 6, pp. 712–722, Jun. 2010.

[8] J. Yao, S. Fidler, and R. Urtasun, "Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation," in *CVPR*, 2012.

[9] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," in *ICLR*, 2015.

[10] Q. Long, Q. Xie, S. Mita, H. Tehrani, K. Ishimaru, and C. Guo, "Real-time dense disparity estimation based on multi-path viterbi for intelligent vehicle applications," in *BMVC*, 2014.

[11] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees." *Machine Learning.*, vol. 63, no. 1, pp. 3–42, 2006.

[12] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guaddarrame, and T. Darrel, "Caffe: Convolutional architecture for fast feature embedding," in *arXiv preprint arXiv:1408.5093*, 2014.

[13] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Mach. Learn.*, vol. 29, no. 2-3, pp. 131–163, Nov. 1997.