HAL
open science

Improving semantic segmentation in urban scenes with a cartographic information

Abdelhak Loukkal, Vincent Frémont, Yves Grandvalet, You Li

To cite this version:

Abdelhak Loukkal, Vincent Frémont, Yves Grandvalet, You Li. Improving semantic segmentation in urban scenes with a cartographic information. 15th International Conference on Control, Automation, Robotics and Vision (ICARCV 2018), Nov 2018, Singapore, Singapore. pp.400-406, 10.1109/ICARCV.2018.8581165 . hal-01875096

HAL Id: hal-01875096

https://hal.science/hal-01875096

Submitted on 16 Sep 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Improving semantic segmentation in urban scenes with a cartographic information

Abdelhak Loukkal[1,2] , Vincent Fremont [3] , Yves Grandvalet [2] , You Li [1]

***Abstract*— This paper presents three different approaches to inject a location information in semantic segmentation Convolutional Neural Networks (CNN) applied to urban scenes. The assumption that a location information would improve semantic segmentation performance emerges from the idea that some elements of urban scenes are located in a predictable manner. This assumption is confronted to realistic data on the CARLA autonomous driving simulator, which is used to create our own synthetic dataset with images, depth maps and bird-eye-view cartographic images. Simulators circumvent the difficulties due to the scarcity of publicly available synchronous labeled images and location information. We consider the location information as a cartographic image as we assume it is the simplest option to include it in a CNN. We assess the relevance of injecting the cartographic information in three different manners: as a CRF potential, as an additional task and as an additional encoder input of a CNN. The three methods are evaluated and compared with a state of the art CNN with regards to the pixel-wise accuracy, mean intersection over union and intersection over union of some important classes. The multi-encoder approach improves the intersection over union of the pedestrians, vehicles and traffic signs classes by respectively 4%, 1.6% and 9 %.**

I. INTRODUCTION

Semantic segmentation has been drawing a lot of attention from the computer vision and autonomous driving communities for many years because in addition to detecting key elements in the scene, it adds semantic information to the global scene understanding problem.

Urban scenes are very challenging environments for autonomous driving because of the dynamics of observed objects. However, some objects' locations in these scenes are quite predictable: for example it is impossible to have a building in the middle of a road and cars are more likely to be found on the road rather than on the sidewalk. The intuition about using a location information in the form of a digital high definition map in perception modules seems reasonable. The objective of this paper is to prove that adding the cartographic information to a semantic segmentation neural network can improve its accuracy regarding some important class of objects like vehicles, pedestrians or traffic signs.

For a neural network to be able to take advantage of some cartographic information, it would be necessary to have a dataset containing labeled images with their corresponding cartographic information. In the publicly available datasets for semantic segmentation, GPS data is available but we lack a precise cartographic information. For this reason, we choose to work on synthetic data produced by CARLA autonomous driving simulator [1] which provides a complete platform with semantic segmentation, depth, LIDAR and a precise map of a virtual town.

For each frame obtained from the simulator, we extract the part of the map that corresponds to one hundred meters ahead of the vehicle. These extracted portions of the map come in a bird-eye-view configuration. We apply an inverse perspective mapping to project the map portion in the camera plane, so as to ease matching with camera frames. We propose three neural networks designs for processing this cartographic information in a CNN. First, we add the map as a pairwise potential in a CRF on top of a state of the art CNN. Second, we design a multi task network that outputs semantic segmentation, depth and the cartographic map. Finally, we add it as an additional input in a network with multiple encoder streams.

II. RELATED WORK

The success of deep CNNs for image classification [2] has encouraged researchers to explore the effectiveness of these networks for dense predictions tasks like semantic segmentation or depth estimation. The current state of the art approaches in semantic segmentation leverage the idea of fully convolutional neural networks [3] keeping the spatial information, that is usually lost in regular CNNs, by avoiding fully connected layers. These networks come in two parts: the encoder extracts features from the input image and the decoder up-samples the encoded feature map to match the size of the input image. State of the art networks for pixel wise semantic segmentation take also advantage of spatial pyramid pooling [4] and dilated convolutions [5] to segment objects at different scales and enlarge the receptive fields, hence the context, of the convolution filters.

Semantic segmentation is a very complex task for which satisfying results were obtained thanks to public datasets like KITTI [6], CamVid [7] and Cityscapes [8] which contains much more labeled images than the two previous ones. More recently, thanks to a new labeling pipeline, Apolloscapes [9], a huge dataset of more than 100k labeled images was released.

gm random fields (CRF) are graphical models that are commonly used for semantic segmentation. Their energy function is composed of unary potentials and pair-wise potentials, see (2), on neighbors (can be pixels or patches of pixels). Originally, the main limitation of this model was its inability to capture long range dependencies, with pixels that are in different regions of the image. Region

[1]Renault S.A.S, 1 av. du Golf, 78288 Guyancourt, France.
[2]Sorbonne universités, Université de technologie de Compiègne, CNRS, Heudiasyc, UMR 7253, Compiègne, France.
[3]Ecole Centrale de Nantes, LS2N, UMR CNRS 6004, Nantes, France.

(a) The car is located on the map of the virtual town

(b) Extracted portion of the map corresponding to 100m ahead

(c) The map portion is projected in the camera plane

(d) Camera image

Fig. 1: Inverse perspective mapping pipeline

based approaches that incorporate hierarchical connectivity and higher-order potentials tried to improve on the original CRFs but lacked accuracy due to the unsupervised segmentation that produces regions. Fully connected CRFs are more expressive models, which present the advantage to have pair-wise potentials between all pixels in an image but the difficulty of inference hindered their potential. A mean field algorithm was proposed to learn efficiently fully connected CRFs [10]. CRFs have been successfully applied as a post processing or refinement step in CNNs in order to improve boundaries delineation for segmentation tasks. [11] proposed an approach that includes the CRF in an end-to-end CNN-CRF where the Mean Field approximate inference is formulated as a recurrent neural network and the Mean Field algorithm steps reformulated as convolution operations.

Autonomous driving involves different tasks like semantic segmentation, object detection or depth estimation. Having a single neural network that solves all these tasks is an important asset. Multi-task learning is a learning strategy that consists in learning different tasks simultaneously, sharing knowledge between tasks. For example, a shared representation improves learning efficiency and prediction accuracy. Multi-task learning is an interesting framework for autonomous driving and mobile robotics in general because it combines different systems in one and eases real-time computing. The multi-task learning loss is a weighted sum of the tasks' individual losses. This weighting can be either uniform or tuned manually. Recently, [12] provided a new weighting approach based on homoscedastic uncertainty, an uncertainty that captures the relative confidence of the different tasks.

Dense prediction networks used for semantic segmentation and depth estimation come in the encoder-decoder configuration. Using several streams for the encoder has been studied in several papers, among the first is [13] that fuses the features extracted from the image and the depth at different scales in the encoder to improve semantic segmentation.

Using a contextual information to improve semantic segmentation has been explored in previous papers. [14] augment pair-wise CRFs potentials with higher order potentials defined on sets of pixels that are determined with unsupervised segmentation algorithms. [**?**] More recently [15] introduced an approach where street layouts are estimated then used to compute spatial prior maps that are used as additional potentials in a CRF to improve semantic segmentation.

III. Synthetic dataset

Real world semantic segmentation datasets are very expensive to annotate and large-scale public datasets do not provide precise cartographic information. Therefore, simulation was the designated solution for our problem. Among the available autonomous driving simulators [16], [17], [1], we chose CARLA as image, depth, semantic and cartographic map are easy to access. The map of the virtual town is provided as a PNG image encoded in three layers with one layer giving information about roads, another one about intersections and the last one about lanes. The car can be positioned in the 2D image map through its world location, using the transformation (provided by the authors) from world coordinates to pixel coordinates in the 2D image map.

All the methods investigated in this paper rely on the fact that the cartographic map is projected in the camera plane. So first, for each camera frame extracted from the simulator, using the coordinates of the vehicle, a portion of the image map corresponding to one hundred meters ahead of the vehicle is extracted. Then, using the transformation between the bird-eye-view plane and the camera plane, the points of the bird's-eye view map are projected in the camera plane with inverse perspective mapping, see Fig.1. The transformation between the two planes is the following:

$$p \sim K[R|T]w \tag{1}$$

where K is the intrinsic matrix of the camera, R the rotation matrix, T the translation matrix, $p = (u, v, 1)^T$ the coordinates in the camera plane and $w = (x, y, 1)^T$ the coordinates in the bird-eye-view plane.

This projected map is homogeneous to a segmentation label. We assign labels to road, intersection and "other" pixels. Alignment of the projected map and the road in images is not perfect because of the vehicle dynamics that change the rotation and translation matrices that define the inverse perspective mapping. This could be compensated using the vehicle's inertial sensors but we chose not to do it to be more realistic.

IV. Proposed methods

Our approach is based on the idea that the cartographic image is considered as a prior to inject in a CNN. Here we use Deeplab V2 [18], a state of the art network for pixel-wise semantic segmentation. We have explored three ways of incorporating the cartographic information in the network:

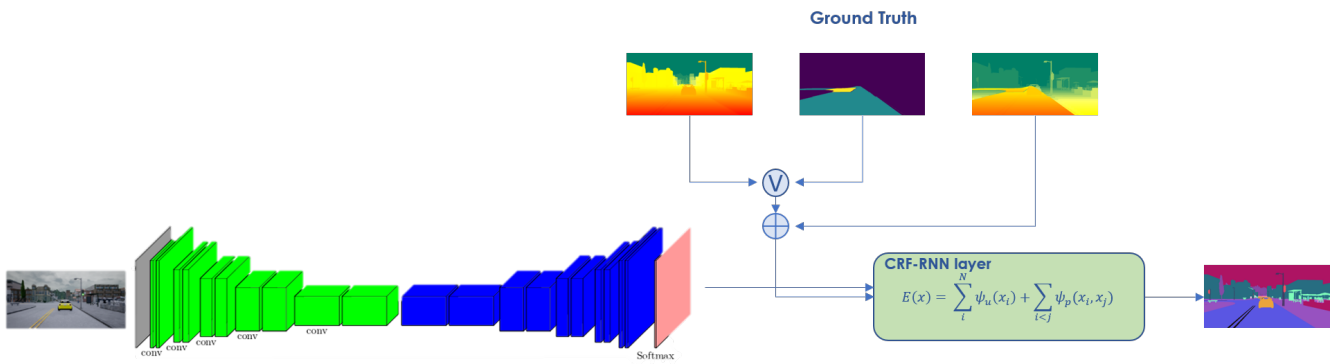

Fig. 2: Method 1: Different combinations of pairwise potentials have been tried to improve the semantic segmentation results. Trials using the cartographic map and the depth map as separate appearance kernels and a single kernel based on the element-wise fusion of depth and cartographic map have been performed.

- The first method consists in adding the ground truth cartographic and depth maps as additional entries in the pairwise potentials of the CRF-RNN layer on top of a CNN (see Fig.2).
- The second method is similar to the previous one regarding the CRF except that the depth and cartographic maps that are used for the CRF are predicted by a multi task network instead of being ground truth (see Fig.3).
- The last method is a CNN with three encoder streams: image, depth and cartographic maps. The features extracted from the three streams are fused by element-wise operations and fed to a decoder that outputs a semantic segmentation map (see Fig.4).

A. Deeplab with CRF-RNN layer

We design our network using Deeplab V2 as our basis. This network takes advantage of dilated convolution to enlarge the size of the feature maps and the receptive field without increasing the number of parameters and a trous spatial pyramid pooling to aggregate multi scale information. We add an additional CRF-RNN layer at the output of the network to allow end-to-end training of the CRF with the CNN. The Gibbs energy of the fully connected CRF is the following:

$$E(x) = \sum_{i=1}^{N} \psi_u(x_i) + \sum_{i<j} \psi_p(x_i, x_j) \tag{2}$$

The unary potential $\psi_u(x_i)$ is computed for each pixel by the CNN that produces a distribution over the label assignment x_i. The pair wise potentials are a linear combination of Gaussian kernels:

$$\psi_p(x_i, x_j) = \mu_p(x_i, x_j) \underbrace{\sum_{m=1}^{K} w^{(m)} k^{(m)}(f_i, f_j)}_{\mathrm{k}(f_i, f_j)} \tag{3}$$

where $k^{(m)}$ is a Gaussian kernel, $w^{(m)}$ a weight, f_i a feature vector and μ a compatibility function that can be given by a simple potts model $\mu_p(x_i, x_j) = [x_i \neq x_j]$.

The original paper on fully connected CRFs [10] defines the following pairwise potentials that have been successfully applied since:

$$k(f_i, f_j) = w^{(1)} \underbrace{exp(-\frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|c_i - c_j|^2}{2\theta_\beta^2})}_{\text{appearance kernel}} + w^{(2)} \underbrace{exp(-\frac{|p_i - p_j|^2}{2\theta_\gamma^2})}_{\text{smoothness kernel}} \tag{4}$$

where c_i and p_i are respectively the vector of RGB values for pixel i and the position in the image.

We explore here different combinations of potentials. We consider adding the cartographic map and the depth map as separate appearance kernels and we also evaluated adding a kernel based on the fusion of both depth and cartographic map that we call focus map:

$$k^{focus}(f_i, f_j) = exp(-\frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|F_i - F_j|^2}{2\theta_\beta^2}) \tag{5}$$

where:

$$F = (1/D) * M \tag{6}$$

and F is the obtained 2D focus map, D the 2D depth map, M the 2D cartographic map (analogous to a segmentation map with label 2 for intersections, label 1 for the road and label 0 everywhere else) and the division and multiplication are element-wise operations.

The weights of the different Gaussian kernels are learned end to end through back-propagation of the gradients in the network.

B. Multi task network

To design the multi task network, we built it on Deeplab to which we add two decoder branches, one for the depth estimation and one for the map estimation. The Deeplab

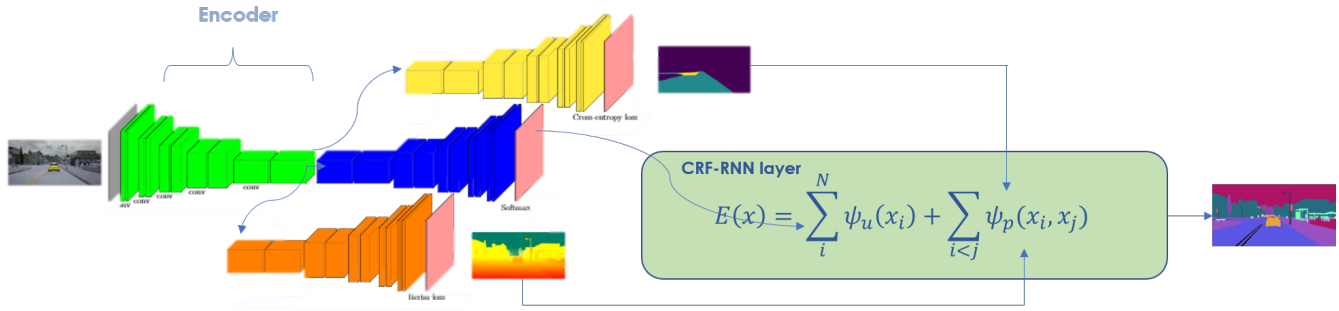

Fig. 3: Method 2: Multi task network with semantic segmentation, cartographic map and depth outputs. The output of the semantic segmentation branch is used as a unary potential in the CRF layer and depth and map outputs are used as pair-wise potentials in the CRF.

network outputs feature maps eight times smaller than the original input image, so bilinear sampling is used for up-sampling. The map estimation being considered as another segmentation task, the decoder we use is similar to the one used in the original Deeplab. For the depth estimation task, we added 3 up-convolution layers to resize the output to the input size and obtain a smoother depth estimation. To weight the different losses, we did manual tunning of the weights. The best results were obtained with the following configuration:

$$Loss_{total} = 10 * Loss_S + 0.1 * Loss_D + Loss_M \quad (7)$$

where $Loss_S$, $Loss_D$ and $Loss_M$ are respectively the loss functions for semantic segmentation, depth estimation and map segmentation. This particular weighting was obtained empirically.

The loss function for the semantic segmentation and cartographic map estimation tasks is the cross entropy loss. For the depth estimation task, we have chosen the BerHu loss which was shown to yield better results than the L1 loss [19]. The reverse Huber loss is defined as:

$$B(x) = \begin{cases} |x| & if\ |x| \leq c \\ \frac{x^2+c^2}{2c} & if\ |x| \geq c \end{cases} \quad (8)$$

For every gradient descent step where we compute $B(y-\hat{y})$, c is defined as:

$$c = \frac{1}{5} max_i(|\hat{y}_i - y_i|) \quad (9)$$

where y_i and $\hat{y}_i$ are respectively the ground truth depth and predicted depth. c is in other words twenty percent of the maximal per batch error. This loss function is equivalent to L1 loss when $x \in [-c, c]$ and to L2 loss otherwise.

We add the CRF-RNN layer at the end of the semantic segmentation branch. The output of the depth estimation and cartographic map estimation branches are then used as additional pair-wise potentials in the CRF, see (3).

C. Multi encoder streams network

In this section, we use a neural network with multiple encoder streams. We build from Deeplab network to which we add two encoder streams, one for the depth and another one for the map. The feature maps extracted from the three streams are fused to obtain one feature map that is passed to the semantic segmentation decoder. The fusion strategy that achieved the best results is done first by element-wise multiplication of the image and map features, then the result of the multiplication is added element-wise to the depth features.

V. Experiments

We run the experiments on a set of two NVIDIA Titan X. We use ADAM optimizer [20] with an initial learning rate of 10^{-4} and exponential decay. We compare the three networks with the original Deeplab without CRF. We compare with different CRF pair-wise potentials and encoder streams to highlight the impact of the cartographic map. All networks are fine-tuned from the pretrained solution fitted on Cityscapes and Pascal VOC. All the networks containing the CRF-RNN layer were trained with a batch size of one because of implementation limits of the custom CRF-RNN layer, other networks were trained with a batch size of three , due to limited computational resources. In this paper, Tensorflow has been used to design and train the CNNs. For the CRF part of the code, the keras/tensorflow code given by [11] was used. The metric used to evaluate the performance of the network is the IoU, intersection over union, defined as:

$$IoU = \frac{TP}{TP + FP + FN}$$

where TP, FP and FN are respectively the true positive, false positive, and false negative pixel counts on the set of test images. This metric is evaluated for each class and the mean over all classes is computed as a general summary.

A. Dataset

We generated 4700 labeled images using the CARLA autonomous driving simulator. We ran 10 episodes of 470 frames each, each episode starting from a different position in the virtual world. We used virtual town number one for all our experiments. The number of vehicles in the world

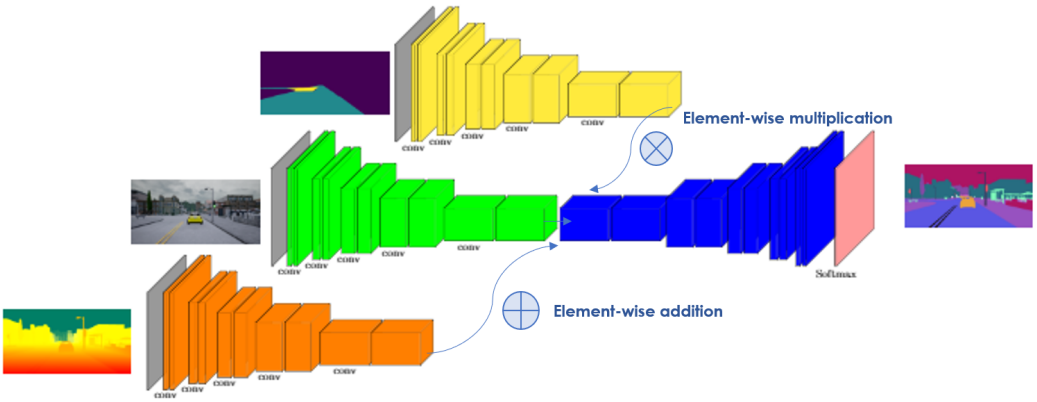

Fig. 4: Method 3: Multi encoder stream network. The features extracted from the depth and map encoders are fused, by element wise multiplication for the map and element wise addition for the depth, with the features extracted from the RGB image.

was fixed to 140 and the number of pedestrians to 120. For each episode, the weather was randomly selected among 4 possibilities: Clear noon, cloudy noon, clear morning, cloudy morning. During the simulation, the vehicle was controlled by the autopilot included in CARLA simulator code. The dataset was split in a training set composed of 4230 images and a test set of 470 images (one episode).

B. Deeplab with CRF-RNN layer

We have first fine-tuned a pretrained Deeplab V2 network on our dataset for 10 epochs. We train a Deeplab network with an additional CRF-RNN layer and compare the results with those of the regular Deeplab network that was fine-tuned on the training data. We tested different additional pairwise potentials. The results are shown in Table I. We observe that the CRF with the cartographic map and depth map as separate additional potentials has the best results regarding the fences and traffic signs IoU improving the IoU by respectively 2.2 % and 3.3% compared to the regular Deeplab and the focus map additional potential has the best results for the vehicles, pedestrians and poles IoU improving the IoU by respectively 0.9% , 3.9% and 0.5 %. This gives a hint on the fact that adding the cartographic information in the CRF adds an information on the location of some important classes like pedestrians, vehicles or traffic signs that are more likely to be found respectively on or outside the road. The Deeplab network with CRF and no additional potential has the best overall performance regarding the classes of interest and achieves the best mIoU and pixel wise accuracy and the best road and road lines IoUs.

C. Multi task model

For the multi task model, we have used the pretrained weights for the encoder and trained the network for ten epochs first. Once the network has learned to predict depth and cartographic map, the outputs of these two branches have been injected in a CRF-RNN layer at the end of the semantic segmentation branch and the network was trained for ten more epochs. Therefore, results with the different CRF pairwise additional potentials are compared with the regular Deeplab and the multi task network without CRF both trained during 20 epochs. The results are shown in Table II. The multi-task network is referred to as MT for brevity. The networks with additional pair-wise potentials are referred to as $MT + CRF + additional - potentials$. The CRF with the focus map, here the focus map is obtained by fusing the output of the depth and map branches of the network, has the highest mean IoU out performing the regular Deeplab by 1.2 % but regarding the classes of interest, the regular Deeplab has the highest IoUs. When comparing the multi task network with and an without CRF, we observe that in this case, the CRF doesn't improve the results in the main classes of interest.

D. Multi encoder model

For this approach, we have used the pretrained weights for the image branch and trained from scratch the depth and map branches. We have tested different fusion strategies and different fusion combinations. The best results were obtained by multiplying the image features by the cartographic map features and then adding the depth features. The results are shown in Table III. The multi-encoder network is referred to as ME to simplify notation. The networks with additional encoders are referred to as $ME : Operations$. This method achieves the best results among the three tested methods and even in ten epochs has better results in the important classes than the CRF, regular Deeplab and multi task approaches in respectively ten, twenty and twenty epochs. The multi stream network where we multiply by the map and add the depth improves the IoU of the pedestrians, vehicles and traffic signs classes by respectively 4%, 1.6% and 9 %. Multiplying by the features of the cartographic map insures a stronger relationship with the features of the image and enforces the relation between the classes and there location in the map.

TABLE I: Results of the first method. Best results are shown in bold. All networks are trained on 10 epochs.

	Deeplab	Deeplab + CRF	Deeplab + CRF + depth	Deeplab + CRF + depth & map	Deeplab + CRF + focus map
mIoU	50.2	**52.4**	48.2	50.5	51.1
Pixel accuracy	88.3	**88.8**	86.5	87.8	88.6
Vehicle IoU	86.3	87.00	86.3	86.5	**87.2**
Pedestrians IoU	18.0	20.7	16.6	19.8	**21.9**
Traffic sign IoU	57.0	60.2	57.2	**60.3**	60.0
Fences IoU	37.1	38.7	34.7	**39.3**	**39.3**
Poles IoU	29.7	**30.2**	28.0	30.1	**30.2**
Road IoU	89.7	**90.7**	88.6	89.2	89.7
Road lines IoU	22.6	**39.4**	35.2	36.5	24.9

TABLE II: Results of the second method. Best results are shown in bold. Regular Deeplab and MT without CRF are trained on 20 epochs, the MT CRF networks are trained first on 10 epochs without CRF then on another 10 epochs with CRF.

	Deeplab	MT	MT + CRF	MT + CRF + depth	MT + CRF + depth & map	MT + CRF + focus map
mIoU	52.1	52.3	53.2	52.4	51.9	**53.3**
Pixel accuracy	**88.8**	**88.8**	88.5	88.2	88.4	88.7
Vehicle IoU	**87.4**	87.3	87.2	87.0	87.1	86.9
Pedestrians IoU	**20.8**	18.9	19.0	20.0	19.3	18.9
Traffic sign IoU	61.7	63.3	62.2	**64.2**	63.4	63.5
Fences IoU	**39.4**	38.4	37.7	38.5	38.2	38.5
Poles IoU	**30.6**	30.1	30.4	29.7	30.0	30.2
Road IoU	89.9	89.9	**90.8**	89.5	89.8	89.3
Road lines IoU	22.4	23.4	**39.0**	29.6	22.5	35.4

TABLE III: Results of the third method. Best results are shown in bold. The networks evaluated in this table are trained on 10 epochs

	Deeplab	ME: Image + depth	ME: Image + focus map	ME: Image * map	ME: Image* map + depth
mIoU	50.2	**51.6**	50.7	49.6	50.7
Pixel accuracy	88.3	88.5	88.6	88.55	**88.6**
Vehicle IoU	86.3	87.5	87.5	87.6	**87.9**
Pedestrians IoU	18.0	19.7	20.9	20.6	**22.0**
Traffic sign IoU	57.0	63.5	62.8	62.6	**66.0**
Fences IoU	37.1	**40.4**	39.8	38.3	39.9
Poles IoU	29.7	30.2	30.1	30.1	**30.6**
Road IoU	89.7	89.8	89.93	**90.0**	**90.0**
Road lines IoU	22.6	23.1	21.7	22.1	**23.5**

E. Discussion

This paper confirms the idea that the predictability of some objects' location in a road scene can help improving the semantic segmentation and presents encouraging results regarding the use of a cartographic information as an image in CNNs. We hope that this line of works will encourage the release of publicly available real world datasets with synchronous semantic segmentation labels and precise cartographic information. The maps would need to have a high precision at centimeter-level. This kind of maps is already popular for autonomous driving and is called High Definition maps. In this paper, the only available information in the map is about the road boundaries and intersections. HD maps contain richer information: about the lanes which can potentially improve the accuracy of the segmentation even more, especially road lines IoU. Whatever format these HD maps come in, it would be possible to rasterize them to use them as proposed in this paper.

VI. Conclusion

The location information yields important information about a vehicle surroundings. In this paper, we have investigated how to inject a location information in a CNN, considering it as another image input to the network. We have explored adding the map image as a pair wise potential in a CRF, as an additional task in a multi task framework and as an additional encoder branch. After comparison with the regular Deeplab network, adding a map based pair-wise potential in the CRF can improve slightly the intersection over union of important classes but the regular CRF still has a better overall performance. Training a multi task network with a cartographic map prediction branch fails to improve the performance regarding the most important classes. Finally, adding the map information as an additional encoder branch of a segmentation network insures the best results improving significantly the intersection over union of important classes like vehicles, pedestrians or traffic signs.

Acknowledgement

This work has been conducted within SIVALAB, a joint research laboratory between Renault and Heudiasyc, which targets issues pertaining to integrity questions in autonomous driving.

References

[1] Alexey Dosovitskiy, Germán Ros, Felipe Codevilla, Antonio López, and Vladlen Koltun. CARLA: an open urban driving simulator. In *1st Annual Conference on Robot Learning, CoRL 2017, Mountain View, California, USA, November 13-15, 2017, Proceedings*, pages 1–16, 2017.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, pages 1106–1114, 2012.

[3] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):640–651, 2017.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part III*, pages 346–361, 2014.

[5] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.

[6] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *I. J. Robotics Res.*, 32(11):1231–1237, 2013.

[7] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009.

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 3213–3223, 2016.

[9] Xinyu Huang, Xinjing Cheng, Qichuan Geng, Binbin Cao, Dingfu Zhou, Peng Wang, Yuanqing Lin, and Ruigang Yang. The apolloscape dataset for autonomous driving. *CoRR*, abs/1803.06184, 2018.

[10] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain.*, pages 109–117, 2011.

[11] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. Conditional random fields as recurrent neural networks. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1529–1537, 2015.

[12] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *CoRR*, abs/1705.07115, 2017.

[13] Caner Hazirbas, Lingni Ma, Csaba Domokos, and Daniel Cremers. Fusenet: Incorporating depth into semantic segmentation via fusion-based CNN architecture. In *Computer Vision - ACCV 2016 - 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part I*, pages 213–228, 2016.

[14] Pushmeet Kohli, Lubor Ladicky, and Philip H. S. Torr. Robust higher order potentials for enforcing label consistency. *International Journal of Computer Vision*, 82(3):302–324, 2009.

[15] Jeonghyeon Wang and Jinwhan Kim. Semantic segmentation of urban scenes with a location prior map using lidar measurements. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017, Vancouver, BC, Canada, September 24-28, 2017*, pages 661–666, 2017.

[16] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017.

[17] Andrew Best, Sahil Narang, Daniel Barber, and Dinesh Manocha. Autonovi: Autonomous vehicle planning with dynamic maneuvers and traffic constraints. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017, Vancouver, BC, Canada, September 24-28, 2017*, pages 2629–2636, 2017.

[18] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, 2018.

[19] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *Fourth International Conference on 3D Vision, 3DV 2016, Stanford, CA, USA, October 25-28, 2016*, pages 239–248, 2016.

[20] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.