

Map-Supervised Road Detection

Ankit Laddha, Mehmet Kemal Kocamaz, Luis E. Navarro-Serment, and Martial Hebert
Robotics Institute
Carnegie Mellon University

aladdha@cs.cmu.edu, kocamaz@cmu.edu, lenscmu@ri.cmu.edu hebert@ri.cmu.edu

Abstract—We propose an approach to detect drivable road area in monocular images. It is a self-supervised approach which doesn't require any human road annotations on images to train the road detection algorithm. Our approach reduces human labeling effort and makes training scalable. We combine the best of both supervised and unsupervised methods in our approach. First, we automatically generate training road annotations for images using OpenStreetMap¹, vehicle pose estimation sensors, and camera parameters. Next, we train a Convolutional Neural Network (CNN) for road detection using these annotations. We show that we are able to generate reasonably accurate training annotations in KITTI data-set [1]. We achieve state-of-the-art performance among the methods which do not require human annotation effort.

I. INTRODUCTION

Recognizing the obstacle-free road region to drive in front of the vehicle (e.g. Figure 1) is a very important information. It is essential for autonomous driving and useful for advanced driver assistance systems (ADAS). Researchers have used various sensors such as laser scanner [2], range scanner, stereo camera pair [3] and monocular camera [4] for this. In this paper, we are interested in using images from a monocular camera to detect the collision-free road area.

The top performing methods [5], [6], [7] which use the publicly available KITTI benchmark [1] for the performance evaluation follow the human supervised learning paradigm. They collect images by driving a vehicle and ask humans to outline the drivable road area. These labeled images are used to train a classifier. This classifier is then used to predict free road space in images at test time.

Adapting these methods to new scenarios is hard because they require considerable human effort to produce new training annotations. If the labeled examples for training can be automatically generated, we can mitigate two major problems: scalability and cost. In this paper, we are interested in addressing the following questions: Can we automatically generate the road annotations without any human intervention and use them to train a road classifier?

A map of the area is essential for navigation while driving, even for humans. We use this necessary and already publicly available information along with other localization sensor information to automatically label images for training. However, the map usually is very coarse and rife with errors. This problem is compounded by the presence of dynamic objects, such as the cars and pedestrians for which there is



Fig. 1: Sample image with the annotated drivable road area. Note that out of the two parallel road in the image only one is labeled as drivable. Also, the road region occluded by the car is not labeled as drivable.

no information available in the map. Also, the localization sensors employed on the vehicle might be noisy.

The problems outlined above lead to errors in the labeling. We exploit the appearance features of the image to reduce the errors in the automatic annotation process. During testing, we only use image information which allows us to generalize to areas without GPS or with poor signal quality. Figure 2 shows an overview of our approach and compares it to the traditional human-supervised paradigm.

Contributions: Our main contributions can be summarized as follows:

- We propose a novel, scalable and cost effective method to automatically generate drivable road area annotations using localization sensors (GPS and IMU) on the vehicle and publicly available noisy OpenStreetMap data.
- We train a CNN using these noisy labels for road detection and outperform all the methods which do not require human effort for image labeling.

II. RELATED WORK

Monocular Road Detection: The majority of the image based drivable road detection approaches follow one of three paradigms: Human-supervised, Self-supervised, and Unsupervised.

The human-supervised approach is followed by [5], [4], [7]. They use human-generated annotations to learn a road model using powerful supervised machine learning algorithms. [5], [7] use Convolutional Neural Networks (CNN), and [4] uses a 1-D graphical model and mixture of Gaussians. These approaches are typically the best performing ones. However, they are costly and unscalable due to the human effort involved. In our approach, we also use a CNN, but we reduce the cost of training by removing humans in the image labeling step.

¹<https://www.openstreetmap.org>

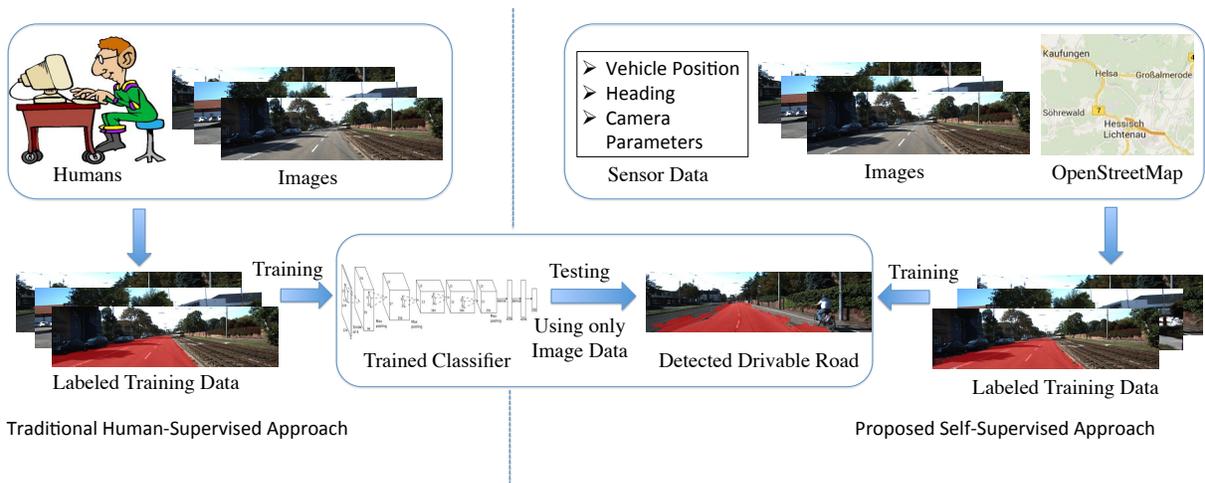


Fig. 2: The pipeline of human-supervised and our proposed map-supervised road detection approaches. Human-supervised approaches use human to label the training data. Whereas, we use publicly available OpenStreetMap data, vehicle pose, camera parameters and pixel appearance features to label the training data. During testing we only use the image data.

Unsupervised approaches use only a single image at any given time. These approaches work either by finding road borders using color and texture information [8], [9] or by building a color-based model of the road points [10]. Road boundary detection based methods usually make assumptions about the shape of the road. Pixel based model methods assume that the center-bottom part of the road belongs to road. These approaches work well in highway scenes, but their performances leave a lot of room for improvements in urban areas.

Self-supervised algorithms [11], [12], [13] rely on the past predictions to adapt an existing model or train a model for the current image. They can be seen as operating between the human-supervised and unsupervised approaches. These methods employ very simple models because of the need to rapidly adapt the model on the fly. [13] uses color based template matching, [12] and [11] uses color based mixture of Gaussians. They also suffer from model drifts, so they require resetting the algorithm after some time. Our approach can be considered as a part of this category but we train a CNN in our offline training step.

Using Maps: Maps provide rich information about static man-made elements of the scene. However, this information is approximate and noisy. Therefore, [14], [15] used map as a prior in their scene labeling algorithms. They require human labeled images for training. [16] uses maps as prior for online road detection, but their algorithms also require some human annotations. [17] uses maps to label aerial images and require map information at the test time. Whereas, we use maps to label images taken from ground and our classifier does not need any map information during the testing.

Training with Machine Generated Labels: [18] uses structural information of a scene predicted by [19] for road detection. They first segment the image into horizontal and vertical surfaces and sky. They use these categories to aid their online road detection. In the experiments, we show that our approach compares favorably with theirs.

III. APPROACH OVERVIEW

Our goal is to recognize pixels denoting the drivable road area in a monocular image. As shown in Figure 2, our approach includes two steps:

- First, we automatically build a set of noisy labeled images using maps, localization sensor data and camera parameters. We reduce the annotation noise using pixel appearance features.
- In the following step, we train a Fully Convolutional Network using these automatically generated labels.

The main difference between the traditional supervised learning paradigm and our approach is that we use machine generated labels to train the classifier, thereby eliminating the human effort involved.

IV. AUTOMATIC ROAD ANNOTATION

A. Overview

Training a statistical model requires to have labeled instances which are representative for the distribution of data. We generate the training samples by exploiting information from maps in the following steps:

- First, we use vehicle pose and maps to reconstruct the 3D scene around the vehicle.
- We then get an initial labeling of the images by projecting the reconstructed scene onto the image plane using a calibrated camera. This initial labeling is too noisy due to occlusions, and errors in the map data, vehicle pose and calibration parameters.
- In the third step, we refine this labeling based on pixel appearance to reduce the error.

We use the publicly available OpenStreetMap (OSM) as the source of map information.

B. Initial Labeling using OpenStreetMap (OSM)

OSM provides information about static structures existing in the scene. We use this data for two types of objects: Roads and buildings. We first reconstruct the 3D scene of a

100x100m² area around the vehicle. To populate the scene with roads and buildings, we use the GPS coordinates of the boundaries of buildings and coordinates of the center line of the roads. Other properties such as number of floors in the building, height of each floor, type of road, number of lanes, width of each lane etc are also extracted from the map database. However, since these are not always present, we make assumptions about them based on the geographic knowledge (e.g. residential roads have one lane and are 3m wide). Subsequently, we project this reconstructed scene onto the image plane using a calibrated camera.

This projection results in each of the pixels in the image being labeled with: road, building or none. The pixels labeled as none or building are combined to form the non-road class. The projected labels are very noisy mainly due to sensor errors, presence of dynamic objects, and erroneous or absent lane width data.

Approximate vehicle pose estimation causes errors in the relative position of the scene with respect to the vehicle. It causes the projection of labels to be displaced (Figure 3a) from their actual position. OSM provides information about the static elements in the scene, but it does not provide information about the dynamic objects, such as cars and pedestrians (Figure 3b). The map projection mislabels dynamic objects that occlude the road. Erroneous or absent lane width data causes over/under estimation of the extent of a road (Figure 3c).

C. Label Refinement

To reduce the noise in the initial labeling step, the images present in the training set are relabeled such that pixels with similar appearance are assigned to the same label.

We use the following approach: First, we cluster the pixels based on appearance and then we assign a label to each cluster based on the statistics of the pixels in that cluster. Assume we have K clusters, for each cluster i , nr_i denotes the number of road pixels, nnr_i denotes the number of non-road pixels after the initial labeling step.

One way to label a cluster i is to define a ratio r_i such that $r_i = nr_i/nnr_i$ and label a cluster i road if $r_i \geq 1$ and non-road if $r_i < 1$. However, it will fail if the number of non-road pixels in a data set is much greater than road pixels, since most of the clusters will be assigned to non-road in this case. Therefore, in order to normalize the number of pixels for both classes, we modify the ratio r_i to r_i^{mod} as follows and use it:

$$r_i^{mod} = \frac{nr_i / \sum_{i=1}^K nr_i}{nnr_i / \sum_{i=1}^K nnr_i} \quad (1)$$

We use color features to represent the appearance of pixels. In particular, we use HSI and YCbCr color spaces. To achieve robustness from shadows, we only use the H and S channels from HSI space, and Cb and Cr channels from YCbCr space.

To reduce the false positives for the road class, we restrict the candidate road pixels. We divide the image into superpixels by thresholding the gPb [20] field, using a very low threshold value (0.01 in our case). We observe that the true road pixels are close to the OSM projected road pixels. Therefore, we consider a superpixel as road candidate if more than 10% of its area was predicted as road in initial labeling. We consider all the pixels present in the road candidate superpixel as road candidate pixels.

Some images contain multiple roads separated by a divider. We are looking for the drivable road area, so detecting any other road except our current road is a false positive. As drivable road area, we select the largest connected component in the image labeled as road. Figure 4 shows results after each step of label refinement for a sample image. It demonstrates that we are able to reduce the noise in the road annotations.

D. Determining Number of Clusters (K)

We use K -means clustering which requires the number of clusters (K) as input. We cannot use the standard cross validation approach: Select K which maximizes the validation set performance, because we do not have any human labeled ground truth data. We use the L method [21] to determine K . This method finds the knee of the K vs clustering evaluation metric graph by fitting a pair of straight lines to it. We use the Sum of Squared Error (SSE) as the evaluation metric.

Assume that there are n candidates $\{k_1, k_2, \dots, k_n\}$ for K , such that $k_1 > k_2 > \dots > k_n$. The L method determines the knee in the graph of K vs SSE as follows: For each $i = \{2, \dots, n-2\}$, it splits the set of candidates into two parts $L_i = \{k_1, \dots, k_i\}$ and $R_i = \{k_{i+1}, \dots, k_n\}$. Then, it fits separate lines to the parts of graph belonging to these sets and calculates the Root Mean Squared Error (RMSE) for those lines. Lets denote RMSE for L_i and R_i as $rmse_{L_i}$ and $rmse_{R_i}$ respectively. The the total RMSE at pivot k_i ($rmse_{k_i}$) is a weighted sum of two RMSEs:

$$rmse_{k_i} = \frac{k_i - k_1}{k_n - k_1} rmse_{L_i} + \frac{k_n - k_{i+1}}{k_n - k_1} rmse_{R_i} \quad (2)$$

The knee point is the k_i which minimizes the $rmse_{k_i}$.

$$K = \underset{k_i}{\operatorname{argmin}} rmse_{k_i} \quad (3)$$

V. ROAD DETECTION

A. Model

We use a Fully Convolutional Network (FCN) for road detection. A FCN only contains convolutional layers so it could produce output at each pixel. Fully connected layers in a network can be easily converted into convolutional layers. Each node in the fully connected layer can be treated as a filter whose spatial size is same as the number of edges incident on that node. For example the first fully connected layer in [22] has 4096 nodes and each node has 49 incident edges. So this can be considered as a convolutional layer with 4096 filters with spatial size of 7x7.

We use the FCN proposed by [23] for our purpose. They modify and convert the 16-layer network from [22] to a FCN.

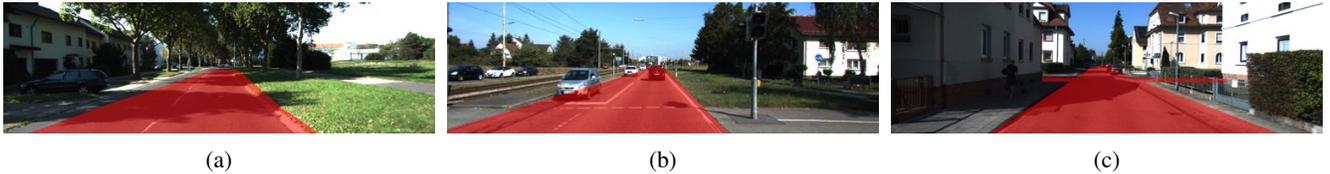


Fig. 3: Errors in the initial road labeling. Examples of errors: (a) due to sensor noise (road region is shifted to the right). (b) due to dynamic objects (mislabeled cars). (c) due to erroneous lane width data (mislabeled sidewalk).



Fig. 4: (a) Initial labeling using OpenStreetMap. (b) Label Refinement using K-Means. (c) Improved Labeling after restricting the allowed road pixels. (d) Final Labeling after removing the non-drivable parallel road.

We provide a brief description of the modifications. See [23] for further details.

The FCN we get from directly converting the network from [22] produces a very coarse output with stride of 32. To increase the resolution [23] skip subsampling after the first three max-pooling layers in the network and modify the convolutional filters in the layers that follow them by introducing zeros to increase their size. The size of filters of the last three convolutional layers is increased by a factor of 2 and size of the fully connected layers is increased by 4.

After converting the network of [22] to a fully convolutional one, the first fully connected layer has 4096 filters of large spatial size of 7×7 . This becomes a computational bottleneck. Therefore, [23] reduces the number of filters to 1024 and spatial size to 3×3 by subsampling. The filter size is reduced to 3×3 such that the spatial receptive field of the network remains the same.

The final layer of our adaptation of the network has 2 channels (one for road and another for non-road). The final output has a stride of 8 so we use bi-linear interpolation to increase its resolution to the image size.

B. Training

We use the publicly available code² by [23] for implementing and training the network. We use soft-max loss at every pixel for back propagation. The network by [22] is already trained on image classification using ImageNet data [24]. We fine-tune the network with initial learning rate of 0.001 and batch size of 5. After every 5 epochs we multiply the learning rate by 0.1. We use a momentum of 0.9. For data augmentation we use mirroring and cropping.

VI. EXPERIMENTS

A. Dataset

We use the KITTI data set [25], [1] to evaluate our approach. It contains a diverse set of road annotations of various different scenes taken in a span of various days. It consist of two sets: *train set* containing 289 image and *test set* containing 290 images.

²<https://bitbucket.org/deeplab/deeplab-public/src>

We use the evaluation protocol defined by [1]: Results are evaluated in birds eye view (BEV) using per pixel metrics. We use Precision (PRE), Recall (REC) and F-measure (F) to evaluate the performance of automatic labeling. In addition to these metrics, we also use Average Precision (AP), False Positive Rate (FPR) and False Negative Rate (FNR) to evaluate road detection.

B. Analyzing Performance of Automatic Labeling Method

The evaluations in this section are done by comparing with the available human annotated road labels for *train set*.

1) *Determining Number of Clusters K*: The first step in our algorithm is to find the number of clusters (K) required for the clustering algorithm. Figure 6 shows the graph of K vs Sum of squared error (SSE). To find the knee of this graph, we plot the graph of K vs Total RMSE using the L method [21] (See Figure 6). Based on this, we select $K = 70$ since it has the minimum RMSE.

Method	F	PRE	REC
Our Approach	85.51	84.16	86.90
Co-Labeling [26]	84.74	88.02	81.69
Map Projection	78.93	74.97	83.34

TABLE I: Quality of the machine generated labels.

2) *Results*: Table I shows that the label refinement step is able to reduce the noise in the initial projection of OpenStreetMap data. We also compare our label refinement method with the co-labeling approach of [26] which extends the fully connected conditional random field of [27] to label a large set of images simultaneously.

We use the initial labeling obtained by projecting the map data onto the image plane as the unary term. To set the pairwise term, we use the same appearance features as we used in our approach. We also select the largest connected component as the final drivable road region which is the same as our approach. We give the co-labeling approach a slight advantage by setting the model parameters such that they maximize the F-measure on *train set*. Both methods

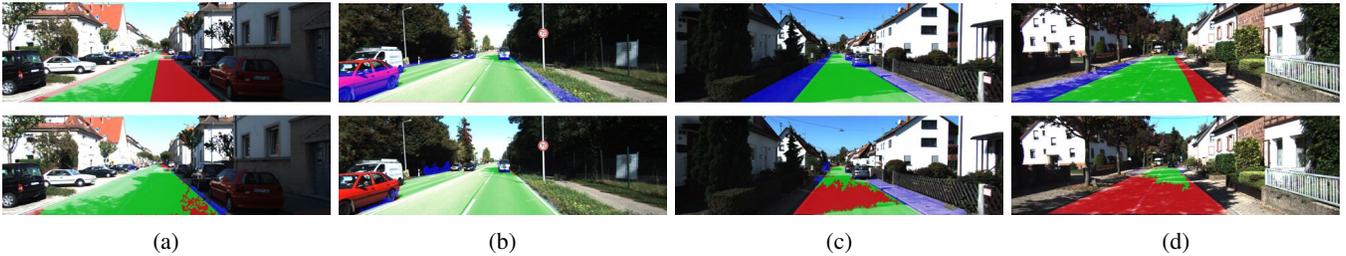


Fig. 5: Qualitative results for the automatic labeling of images. Top row displays results after initial map projection and bottom row shows results after our label refinement approach. In (a), our approach is able to extend the road annotation to cover the whole road. In (b), we successfully removed the incorrectly labeled cars from the road class. Figures (c) and (d) illustrate failure cases where our approach incorrectly removed most of the road pixels and kept the non-road pixels. The color coding is as follows: Green - True Positive, Red - False Negative, Blue - False Positive for road.

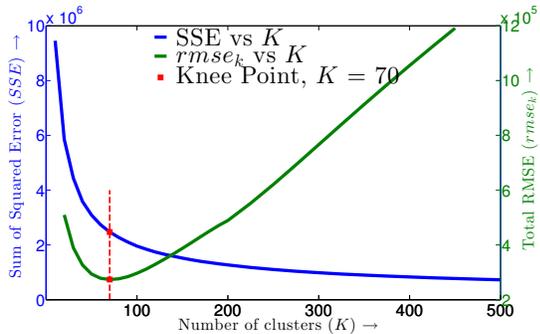


Fig. 6: Plots on *train set* to determine number of clusters. We can see that knee of the K vs SSE graph occurs at $K = 70$ because the $rmse_k$ is minimum at that point.

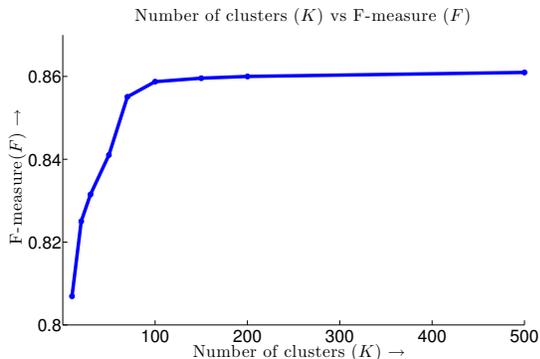


Fig. 7: Effect of different values of K on our proposed automatic labeling approach.

are based on the assumption that similar pixels should be assigned to the same label. We see that our method and co-labeling have similar performances. However, our method does not require any ground truth to select model parameters.

3) *Qualitative Results*: We show some successful cases where we are able to increase the quality of the initial annotations by map projection using appearance features in Figure 5a and 5b. Figure 5c and 5d shows some of the failure cases. The main reasons for the failures are: Extreme shadows, and similar appearance of the road and sidewalk.

4) *Ablative Analysis*: There are four steps in the algorithm: Projection of map data onto image plane (Map),

Step	F	PRE	REC
Map	78.93	74.97	83.34
Cluster	76.41	63.90	95.01
GPB	84.62	81.79	87.65
Component	85.51	84.16	86.90

TABLE II: Quality of machine generated labels after each step of the proposed algorithm. For details see VI-B.4.

label refinement using clustering (Cluster), restricting road candidate pixels using gPb (GPB) and finding the largest connected component (Component). Table II shows the results after each step. From this, we observe that the clustering step provides a boost in true positives. The number of false positives are reduced by computing the road candidate pixels and finding the largest connected component.

5) *Effect of K* : Figure 7 shows the performance of our automatic labeling approach with respect to the number of clusters K . We can see that for $K \geq 70$ the F-measure is very similar for all K . This indicates that after a specific value of K , which is in the order of 100, our approach is robust to the number of clusters.

C. Analyzing Performance of the Road Detection

We train the Fully Convolutional Network (FCN) using the images from *train set* and automatically generated road labels. We evaluate on the *test set*. During testing we only use monocular color images.

In Table III, we can see that the FCN trained with refined labels (With Refinement) perform better than the one with labels we directly get from map projection (Without Refinement). This is because the refined labels are of better quality. We can also see that we outperform [18]. They also train a CNN using machine generated labels and uses only monocular image for testing. However, their CNN is trained to predict the geometric structure (sky, vertical and horizontal) in the image which is then used to detect road. To build a road model for each image they assume that the middle bottom part of the image is road.

Next, we compare our approach with other methods which do not require human supervision but use other sensors. In Table III, RES3D+VELO [2] uses laser data from Velodyne sensor for this task and GRE3D+SELAS uses a stereo pair.

Method	AP (\uparrow)	MaxF (\uparrow)	PRE (\uparrow)	REC (\uparrow)	FPR (\downarrow)	FNR (\downarrow)
Testing with Monocular Images						
Proposed - With Refinement	89.96	87.80	86.01	89.66	10.34	10.34
Proposed - Without Refinement	82.20	83.37	80.03	86.99	13.01	13.01
CN [18]	78.80	79.02	76.64	81.55	13.69	18.45
Testing with Other Sensors						
GRES3D+SELAS	86.86	85.09	82.27	88.10	10.46	11.90
RES3D+VELO [2]	78.34	86.58	82.63	90.92	10.43	9.08
Training with Human Annotations						
Fully Supervised (Proposed)	90.96	91.61	91.04	92.20	5.00	8.71

TABLE III: Road detection performance of various methods on the *test set*. Note that, \uparrow denotes higher is better and \downarrow denotes lower is better.

We achieve higher performance than these methods, thus establishing a new state of the art among approaches which do not require human supervision.

Finally, in Table III (Fully Supervised) we report the upper-bound performance of our classifier (FCN). In this case, the training is done with human annotated, ground truth labels. We can see that the FCN trained with automatically generated labels is able to achieve close to the upper-bound performance in terms of average precision and recall.

VII. CONCLUSION

We presented a map-supervised, monocular image-based drivable road area detection system. Our approach reduces the human image labeling effort and makes the supervised road detection algorithms scalable and cost effective. It can automatically generate training labels for drivable road recognition using the noisy data from publicly available OpenStreetMap and other localization sensors on the vehicle.

In the future, we plan to investigate the use of temporal information present in videos to improve the labeling step. We could use the motion information present in the videos to remove the false positives on the moving vehicles. This would be very useful in cluttered scenes where large part of road is occluded by moving vehicles. We would also like to extend our system for extreme weather conditions such as rain and snow, which are currently not handled by our system. For this, we plan to investigate 3D features from videos and CNN features to distinguish between the road and background pixels.

REFERENCES

- [1] J. Fritsch, T. Kuehnl, and A. Geiger, "A new performance measure and evaluation benchmark for road detection algorithms," in *IEEE ITSC*, 2013.
- [2] P. Y. Shinzato, D. F. Wolf, and C. Stiller, "Road terrain detection: Avoiding common obstacle detection assumptions using sensor fusion," in *IEEE IV*, 2014.
- [3] L. Xiao, B. Dai, D. Liu, T. Hu, and T. Wu, "Crf based road detection with multi-sensor fusion," in *Intelligent Vehicles Symposium (IV)*, 2015.
- [4] J. Yao, S. Ramalingam, Y. Taguchi, Y. Miki, and R. Urtasun, "Estimating drivable collision-free space from monocular video," in *IEEE WACV*, 2015.
- [5] D. Levi, N. Garnett, E. Fetaya, and I. Herzlyia, "Stixelnet: A deep convolutional network for obstacle detection and road segmentation," in *IEEE WACV*, 2015.
- [6] D. Munoz, J. A. Bagnell, and M. Hebert, "Stacked hierarchical labeling," in *ECCV*, 2010.
- [7] R. Mohan, "Deep deconvolutional networks for scene parsing," in *arXiv:1411.4101*, 2014.
- [8] H.-Y. Cheng, B.-S. Jeng, P.-T. Tseng, and K.-C. Fan, "Lane detection with moving vehicles in the traffic scenes," *Intelligent Transportation Systems, IEEE Transactions on*, 2006.
- [9] H. Kong, J.-Y. Audibert, and J. Ponce, "General road detection from a single image," *Image Processing, IEEE Transactions on*, 2010.
- [10] J. M. Alvarez and A. Lopez, "Road detection based on illuminant invariance," *Intelligent Transportation Systems, IEEE Transactions on*, 2011.
- [11] L. M. Paz, P. Piniés, and P. Newman, "A Variational Approach to Online Road and Path Segmentation with Monocular Vision," in *ICRA*, 2015.
- [12] D. Lieb, A. Lookingbill, and S. Thrun, "Adaptive road following using self-supervised learning and reverse optical flow," in *RSS*, 2005.
- [13] C. Tan, T. Hong, T. Chang, and M. Shneier, "Color model-based real-time learning for road following," in *IEEE ITSC*, 2006.
- [14] S. Wang, S. Fidler, and R. Urtasun, "Holistic 3d scene understanding from a single geo-tagged image," in *IEEE CVPR*, 2015.
- [15] K. Irie and M. Tomono, "Road recognition from a single image using prior information," in *IROS*, 2013.
- [16] J. M. Alvarez, A. M. Lopez, T. Gevers, and F. Lumberras, "Combining priors, appearance, and context for road detection," *Intelligent Transportation Systems, IEEE Transactions on*, 2014.
- [17] V. Mnih and G. Hinton, "Learning to label aerial images from noisy data," in *ICML*, 2012.
- [18] J. M. Alvarez, T. Gevers, Y. LeCun, and A. M. Lopez, "Road scene segmentation from a single image," in *ECCV*, 2012.
- [19] D. Hoiem, A. Efros, M. Hebert, *et al.*, "Geometric context from a single image," in *IEEE ICCV*, 2005.
- [20] P. Dollar and C. Zitnick, "Fast edge detection using structured forests," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2015.
- [21] S. Salvador and P. Chan, "Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms," in *IEEE ICTAI*, 2004.
- [22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.
- [23] 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.
- [24] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, 2015.
- [25] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.
- [26] J. M. Alvarez, M. Salzmann, and N. Barnes, "Large-scale semantic co-labeling of image sets," in *IEEE WACV*, 2014.
- [27] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *NIPS*, 2011.