Semantic Segmentation of Road Profiles for Efficient Sensing in Autonomous Driving

Guo Cheng, Jiang Yu Zheng, and Mehmet Kilicarslan
Department of Computer and Information Science
Indiana University – Purdue University Indianapolis
Indianapolis, IN 46202, United States
{guocheng@iu.edu, jzheng@iupui.edu, mkilicar@indiana.edu}

***Abstract*—In vision-based autonomous driving, understanding spatial layout of road and traffic is required at each moment. This involves the detection of road, vehicle, pedestrian, etc. in images. In driving video, the spatial positions of various patterns are further tracked for their motion. This spatial-to-temporal approach inherently demands a large computational resource. In this work, however, we take a temporal-to-spatial approach to cope with fast moving vehicles in autonomous navigation. We sample one-pixel line at each frame in driving video, and the temporal congregation of lines from consecutive frames forms a road profile image. The temporal connection of lines also provides layout information of road and surrounding environment. This method reduces the processing data to a fraction of video in order to catch up vehicle moving speed. The key issue now is to know different regions in the road profile; the road profile is divided in real time to road, roadside, lane mark, vehicle, etc. as well as motion events such as stopping and turning of ego-vehicle. We show in this paper that the road profile can be learned through Semantic Segmentation. We use RGB-F images of the road profile to implement Semantic Segmentation to grasp both individual regions and their spatial relations on road effectively. We have tested our method on naturalistic driving video and the results are promising.**

Keywords— autonomous driving, road profile, temporal-to-spatial, semantic segmentation.

I. Introduction

Real time autonomous driving requires fast processing of sensor-fused data from all kinds of devices embedded in the vehicle. For example, if we have been driving a car for one hour with our sensor updating, LiDAR sends approximately 72GB data points, and a camera produces 2.6GB HD driving video. This brings a huge challenge to scene understanding and recognition with high accuracy and efficiency. The execution time in the road scene evaluation directly influences the subsequent decision making and path planning.

Reducing burden in computation while ensuring the accuracy of vision tasks is essential in real-time autonomous driving. There have been many researches aiming at solving this problem. In this paper, we replace *spatial-to-temporal* approach in traditional framework of frame recognition followed with tracking with a novel *temporal-to-spatial* approach. At the same position of 2D frames in driving video, we sample a 1D line of pixels, these consecutive lines piled along time axis in the video volume form a spatial-temporal image so that a driving video is reduced into a *road profile* image temporally.

The main vision sensing tasks for autonomous driving are identifying road area to follow and locating traffic to avoid collision. To achieve these goals using a single scanning line, we have to identify segments on the line occupied by lane, road, off-road, moving traffic, vertical objects, etc. Although one-pixel line does not provide sufficient spatial information of road and objects, the consecutive collection of lines as a spatial-temporal image [1] provides intrinsic spatial layout because of the continuous observation and smooth vehicle motion. There is an effort made to detect road edge in road profile [10].

Now the key is to divide different regions in the road profile to extract drivable area for the vehicle. This paper tackles this problem by using the *semantic segmentation* [3], which yields the direction of drivable road and between surrounding vehicles, but not influenced by visual appearances such as shadow, snow, highlight reflection, poor illumination, and shape deformation caused by ego-vehicle motion. The semantic segmentation not only identify unique patterns based on local features, but also constrains structural relation of different regions through maximum pooling and linear combination of local information. We will show that road profile is learnable with semantic segmentation. We also use RGB-F channel of road profile like RGBD image, where F is a channel describing features around the sampling line and is pre-computed as the sampling line is collected from video.

A. Related Works on Semantic Segmentation

Segmentation is a difficult task in the field of autonomous driving, which requires fast and accurate scene understanding. To implement pixel-wise classification, many methods of deep learning have pitched in and achieved compelling results. Long *et al* [4] replaced fully connected layer (FCN) in convolutional neural networks (CNN) with an architecture of fully convolution in an end-to-end and pixel-to-pixel model. In order to improve the learning performance in deep neural network, they also added skip layers as complementation. Since then, many novel neural networks have been put forward, He *et al.* [5] proposed ResNets by replacing the optimization objective with residuals blocks through shortcuts so that a network can converge easily with less computation. Zhao *et al.* [6] used a pyramid hierarchical architecture (PSPNet) with aggregated context from different regions to reinforce learning of global information. Pohlen *et al.* [7] enhanced the design of ResNets by separating the forward learning process in CNN into two streams: the pooling stream proceeds as basic FCN [4], while the residual stream functions are similar as residual blocks in ResNets [5]. The wave of deep learning comes along with the availability of

This is the author's manuscript of the article published in final edited form as:

Cheng, G., Zheng, J. Y., & Kilicarslan, M. (2019). Semantic Segmentation of Road Profiles for Efficient Sensing in Autonomous Driving. 2019 IEEE Intelligent Vehicles Symposium (IV), 564–569. https://doi.org/10.1109/IVS.2019.8814259

datasets such as KITTI and Cityscapes [8, 9], which brings a great stride toward the pixel-level visual understanding in driving environment.

B. *Temporal-to-Spatial Semantic Segmentation*

In this work, we implement the semantic segmentation in our temporal-to-spatial approach. There are two challenges we are facing: First, since our road profile temporally consists of lines from each frame of driving video, the spatial relations among different regions suffer from a certain degree of distortion due to unstable vehicle motion and projection. Second, low-level features learned in first several layers of a deep network are easily dismissed as network goes deeper. However, these features are important in identifying road edge, grass texture on roadside, etc. To solve the challenges above, we put forward a fully pyramid residual neural network, in which we adopt a *pooling stream* and a *residual stream* in a fully convolutional Encoder-Decoder model. By inserting a pyramid block into pooling stream to reduce context information loss among different temporal-spatial segmentations, the capability of networks to sense lower level features are much improved. Moreover, residual stream transmits low-level features back from previous layers so that the problem of degradation and gradient vanishing in very deep neural networks are eased.

In the following, we address the framework of sensing one-line for autonomous driving in Section II. Section III introduces the semantic segmentation applied to the road profiles. Section IV shows experiments of segmented road profiles in different weather and illumination conditions, and then a conclusion.

II. Road Profile for Navigation

A. *Road Profile from Driving Video for Data Reduction*

In autonomous driving, both a high accuracy of scene understanding and a fast speed of calculation in real time are required. The spatial-to-temporal strategy from 2D calculation of video frames to temporal tracking is time-consuming under resource constraints. This encourages us to explore an alternative approach to achieve a more efficient strategy in real-time sensing and driving.

Figure 1: The plane of sight through the sampling line in the frame captures road information for driving. The space below the current rays is previously scanned. The degraded shading in red denotes the past observation space.

A forward dashboard camera mounted on windshield captures driving scenes. As the vehicle is moving, the video has frame rate of 30 fps capturing road surface. A sampling line is fixed under the horizon in the video frame. The plane of sight reaches 15m ahead on the ground or scans vertical objects on roadside and on moving vehicles (Fig. 1). The moving plane of sight covers the entire space below the current plane of sight without redundancy as the vehicle moves forward.

To facilitate video processing, we first convert long driving videos to congregate images called *road profile* [2] for visualization and data reduction. From each frame as Fig. 2, we sample a line ***l*** with a fixed distance from the horizon in the image to capture temporal road scenes 15m ahead, according to its coverage of lanes and road, as well as the vehicle responding time to danger. This is also a proper distance to plan short range driving path. The lane width is calibrated from this setting; the total width of the frame covers about four lanes by the sampling line. If we can sense vehicle and lane on the current sampling line (Fig. 2a), we can estimate the vehicle occupied region using the lane information (Fig. 2b). The vehicle moving direction can be planned instantly (Fig. 2c). The depth and TTC to vehicles and vertical obstacles can also be calculated by tracking their widths and changes [17].

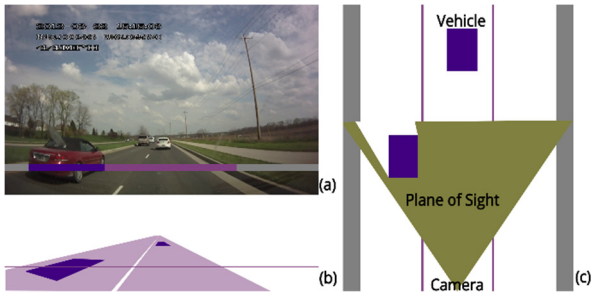


Fig. 2: Using a scanning line to plan vehicle moving direction. (a) the horizontal sampling line ***l*** in driving view, capturing the road surface and possible vehicles. It covers the road side, road for driving area and obstacles to avoid. (b) Vehicle occupied region projected from the sampling line. (c) the plane of sight of the sampling line. The colors have semantic meaning as: **grey**-roadside, **purple**-vehicle, **pink**-road.

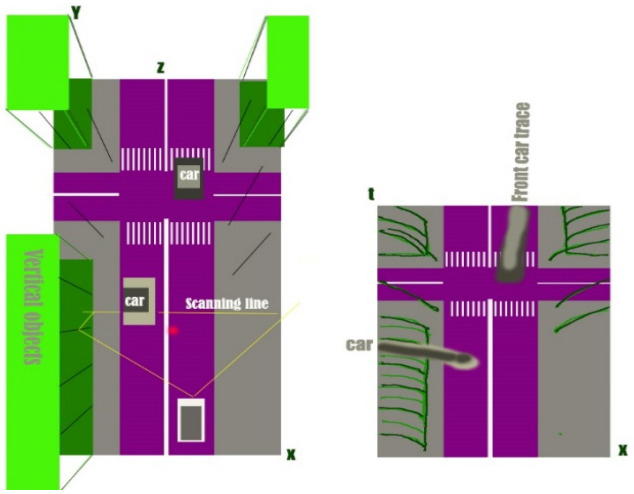


Figure 3: Projection of road profile. (a) Helicopter view in the perspective projection of a road segment. The scanning line is about 15m ahead vehicle camera. (b) Road profile in parallel-perspective projection, where the ground is similar to the road of bird-eye view, but the vertical features such as building rims and poles appearing as hyperbolas. Dynamic vehicle traces are added as smooth trajectories. The time length of road profile is vehicle speed-adjusted.

By copying pixels on line ***l*** into another image, sampled lines from consecutive frames are piled in the image along the time axis such that a *road profile* is created [2]. A five-minute HD video yields a road profile of 1280×9000 pixels, which reduces video to 1/720th in size but keeps all the road and roadside colors over four-lane width.

Although the road profile is sampled in one-line per frame, which contains spatial information in *x* direction (latitudinal), the consecutive collection of lines also reveals spatial information longitudinally. As shown in Figure 3, the scanning

line captures road surface at the interaction with the plane of sight through the line. In addition, the scenes under this plane have been scanned in previous frames. The temporal contexture of road is recorded already in the road profile.

B. Spatial and Temporal Appearance in Road Profile

Ideally, if the ego-vehicle/camera moves at a constant speed straight forward, the road profile employs a parallel-perspective projection; parallel for those planes of sight, and perspective on the plane of sight. The ground with the same height from the camera has the same shape layout as in a perspective projection of a helicopter view (Fig. 3). Despite of minor waves of road and lanes due to limited vehicle rolling and pitching, the road profile records lane, road, and off-road regions along road.

Unlike horizontal features on the ground, vertical features in the 3D space such as trees, poles, and vehicle rims are repeatedly scanned by the plane of sight. Their traces are hyperbola curves if the vehicle moves straight in a constant speed and being further adjusted by ego-vehicle speed. By locating these objects and calculating their tangent, we can identify whether they are approaching to the camera (zero-flow) during the vehicle motion [10]. For an identified vehicle boundary, its location with respect to the camera can be briefly estimated as Fig. 3 illustrated.

Temporally, if the ego-vehicle has a yaw change (turning), all the road structures move inversely in the opposite direction in the road profile. If the vehicle stops, background scenes appear as purely parallel lines along the time axis. Passing vehicles on side lanes leave short trajectories inward, while passed vehicles by the camera has short traces outward. A front vehicle has its trace appearing in the road profile for while if it is closer than 15m, and its trace width is squeezed and expanded according to the headway space ahead. This can also yield TTC and allow the vehicle to adjust the distance to front car or avoid collision. In order to avoid collision, we should stay alert when a front vehicle stops at the traffic lights and gets closer. Figure 4 shows examples when vehicles appear in the road profile. We can find that the direction of a vehicle trace implies its relative speed with the camera, i.e., a passing or passed vehicle.

If the road profile is classified to road, roadside, and other vehicles, short-range path planning (road following and speed control) be done in road profile. In principle, if the vehicle ego-motion is known from its control, two consecutive lines in the road profile can be mapped toward the road space. The road portion cut off by road edges or vehicle rims provides a drivable area for keeping the vehicle on road. High vertical objects such as vehicles, poles, and trees are always captured by the sampling line if they come closer than 15m. Their latest positions in the road profile can be used for avoidance directly. Only an obstacle much lower than the camera height (mostly in a vehicle prohibit area) can be under the current plane of sight if the vehicle moves close to it. To avoid collision onto it, the path planning uses its previously detected positions in earlier frames.

For the path planning and autonomous driving on normal roads, we classify pixels on the latest lines in the road profile into six semantic regions according to surface materials and vehicle motion styles. The pixels are labeled with RGB values:

- **Road** (128,64,128): the road surface in temporal space.
- **Roadsides** (128,128,128): adjacent to the road on two sides, including the sidewalk, grass, buildings area, etc.
- **Vehicles** (64,0,128): moving or stopped vehicle seen from the driving view.
- **Lane marks** (255,255,255): include either yellow solid line or white dashed line on the road.
- **Vertical obstacles** (0,128,64): vertical objects on road side, including buildings, telegraph poles and so forth.
- **Stopping period** (192,128,64): the whole period in road profile while ego-vehicle is stopping temporally.

Figure 4: Road profile and labels. The time axis is upward. The horizontal axes are the *x* axis in the image. Vehicle traces are marked with **V** signs. (a) Vehicle traces in motion and stopping are visible in the road profile. (b) We labeld six semantic classes from materials and motion styles for road profile.

The road appearance in the images is determined from the surface material reflectance both on-road and off-road. Under different illuminations, the color changes dramatically in the road profile [2]. The semantic segmentation is trying to detect drivable area such as road, and obstacles with semantic meaning, but not influenced from visual appearances such as shadow and highlight on road.

III. Semantic Segmentation on Road Profile

A. Training Dataset and Pre-Processing

We use naturalistic driving video to generate road profiles.

A 5-min driving clip generates 9000 frame road profile. Across five different weather categories, 25 videos are selected for training and additional 5 videos are used for testing. We manually label the road profiles with original resolution into regions with the color defined in previous section, and then spatially scaled to the width of 256 pixels by selecting maximum value in the scaled regions. Similar to the RGB-D dataset NYUDv2 [11], we add an additional F channel for each RGB road profile. Channel F is pre-calculated features around sampling line in driving video, and here is the edge linearity [10] to include local structural information. We know the linearity provides a strong cue in finding lane marks according to our previous study. Our input dataset is a collection of RGB-F road profiles. The information in F channel is a compensation to the temporal space, and it helps detecting lane marks.

Semantic segmentation [3,18] not only learns local features, but also memorizes the global relation of features through linear combination of local features and pooling. One-line contains latitudinal spatial relation, but not longitudinal relation and temporal. However, our semantic segmentation uses a short history patch to provides correct layout of segments.

- Even if the road profile lacks height information of objects in frames, it contains both spatial layout (e.g., road, roadside, cars, pedestrians) and temporal event (e.g., camera turning, stopping, changing lane, traces of surrounding vehicles).
- Our temporal events have continuity due to a smooth motion of vehicles and camera, rather than arbitrary appearing/disappearing. Therefore, the earlier information provides cues for current region recognition. This temporal continuity is preserved by semantic segmentation.

Therefore, we use small patches with time length T to perform semantic segmentation. To generate instant output for vehicle control, this patch window shifts t lines along the road profile to predict latest segments. We can thus generate results at a short time interval t (≥ 1 *frame*). On the other hand, a short T requires less computing time in deep learning and overlaps in temporal shifting but may miss necessary longitudinal layout of road. We first resize input road profile of 1280 pixels in width to 256 pixels horizontally for using an existing software. Then, we carry out two experiments to observe the learnability of road profiles: (1) T is set to 256 frames and t shifts 9 lines along time axis, causing adjacent patches overlapping 256-9 lines; (2) T is set as 2 lines including t as one line, which is the extreme condition in semantic segmentation that obtains knowledge of the last line. Though it accelerates training process, the testing accuracy may decrease due to less temporal information.

B. Architcture of Road Profile Semantic Segmentation

A general concern in road profile semantic segmentation is the accuracy in learning road edges and other region boundaries, as those curves along the time axis reflect vehicle motion. Based on their precise location detected, we can plan an accurate path and avoid vehicle collision. The pooling operation in down-sampling (encoding) can enlarge the receptive field and reduce the risk of over-fitting. But it also performs at expense of resolution loss significantly each time. By adopting the architecture as displayed in Fig. 5, we made the following improvements in the semantic segmentation:

The residual stream [5] carries residual information of feature maps in full image resolution, making precise segmentation at boundaries and edges. In road profile semantic segmentation, we parallelly calculate the convolutional and pooling results inside the residual block and then concatenate with the pooling stream at each layer. The residual stream can be presented as

$$x_m = x_n + \sum_{i=n}^{m-1} F(x_i; \omega_{i+1}) \quad (1)$$

where x_m denotes the output of residual block at m^{th} layer, $F(x_i; \omega_{i+1})$ is the residual with parameters ω_{i+1} learned in the backpropagation of network training.

The pooling stream [7] is responsible for learning global relationship of image elements, which results in correct segmentation of different regions. By utilizing residual and pooling streams together, the learned spatial and temporal features are more enriched along the boundary; this is significant to road profile semantic segmentation, since the ego-vehicle will be controlled according to the boundaries of road and other vehicle in the road profile.

A pyramid pooling module [6] consists of four small pooling filters of 1×1, 2×2, 3×3, and 6×6 pixels. Each filter forms a different feature map after pooling. Thereby, such a hierarchical architecture contains information in different scales. By up-sampling each feature map into identical map size via bilinear interpolation, they can be concatenated to form a multiple-layer feature representation. This carries both global and local context information.

Fully convolutional layer [4] replaces fully-connected

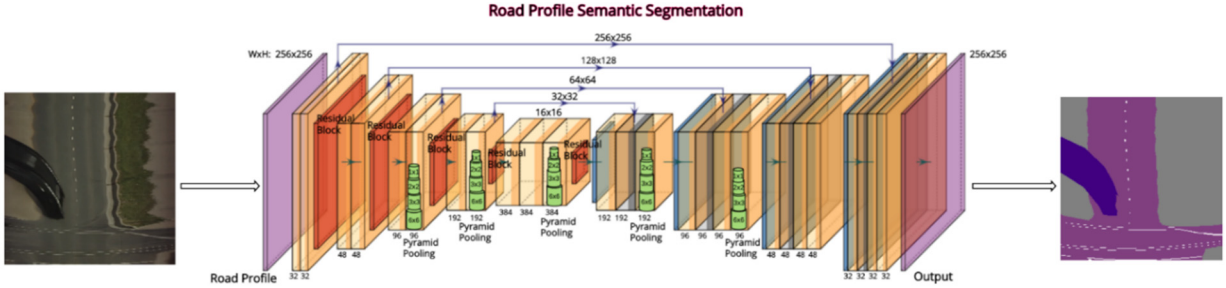


Figure 5: Architecture of Road Profile Semantic Segmantation. The input image goes through an 11-layer fully convolutional nueral network. The pyramid pooling modules (green) are inserted into the pooling stream; the results of residual stream with residual blocks (red) are concatenated with pooling stream in each layer. In the down-sampling/up-sampling process, the image size is reduced/enlarged by half while the image depth is increased/decreased.

layer in the end of neural network, such an Encoder-to-Decoder architecture breaks restriction of fixed input image size and achieves classification at pixel level.

By combing above models, our model becomes a deep fully convolutional network with high resolution. As displayed in Fig. 5, every input patch will be calculated in pooling and residual streams. Residual stream enables hierarchical features to be transmitted as residual, and the pyramid pooling module in pooling stream re-concatenates the hierarchical features after the convolution. The results of two streams will be merged after up-sampling. Finally, a pixel-wise classification is generated through Softmax in the fully convolutional layer.

IV. EXPERIMENTS

Our work implemented temporal-to-spatial approach and achieved nice results on road profile semantic segmentation in two experiments as described in Table I.

Table I Patch size and moving steps in experiments of segmenting road profiles

	Patch size for training and testing	Out patch size in testing	Training moving step	Patch width
Ex. 1	256 lines	Last 9 lines	9 lines	256
Ex. 2	2 lines	Last 1 lines	1 lines	256

Augmentation To avoid risk of destroying temporal information and distorting vehicle motion records, we do not perform data augmentation.

Filter Size: our neural networks adopt small kernel size as pooling filters: 1×1, 2×2, 3×3, and 5×5 pixels. It has been proved to be a good choice in ResNet [5] that the pooling performance by using smaller filters multiple times is better than using a large filter once. As receptive filed increases, image resolution decreases and the loss is unrecoverble, a larger pooling filter introduces more resolution loss.

Number of layers: we fine-tune all layers by backpropagation through the whole network. Take Ex. 1 as example, the down-sampling process reduces the input image size from 256×256 to 16×16; the up-sampling process increases the size of feature map from 16×16 to 256×256, followed by a fully convolutional layer. In the semantic segmentation of road profiles, there are six layers for encoder, and five layers for decoder.

Implementation All models are trained and tested with Tensorflow [13] on a single NVIDIA GTX 780Ti. For Ex.1, the training time is 18hr using 970 patches × 25 road profiles. The testing time is 0.3 second to generate a patch of 9 lines in the road profile. It is approximately a real-time sensing with three-time prediction of road scenes per second. For Ex. 2, it takes 0.8s to finish one second video (30lines), but temporal resolution of prediction is detailed to frame level. The vehicle can make an instant response to the input data for path planning.

Optimization In this experiment, we also use RMS in Tenserflow optimizer to minimize the loss function, and set a decayed learning rate started from 0.0001.

We carry out the experiment on road profile dataset, which contains five weather categories, and the semantic labeling with six classes. In this end-to-end neural network, we trained data all from scratch; for each weather category, we select one road profile for testing, the testing results are presented in Fig 6.

To evaluate the testing results, there are many standard semantic segmentation metrics in benchmark datasets KITTI and Cityscapes for autonomous driving [13], such as Intersection-over-Union (IoU), Pixel Accuracy (PA), Mean Pixel Accuracy (MPA), Mean Intersection over Union (MIoU) and Frequency Weighted Intersection over Union (FWIoU) [14]. In our experiments, we adopt two metrics, PA and IoU, as

$$PA = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

$$IoU = \frac{Intersection}{Union} = \frac{TP}{TP + FP + FN} \tag{3}$$

Where TP/TN denotes true positive/negative pixels, FP/FN denotes false positive/negative pixels for each semantic class. Per-class PA and IoU of semantic classes are tested on 5 videos and shown in Table I. Though in a small set of one video clip (9000 frames), weather category-based mIoU are also tested and summarized in Table II.

Table I Pixel Accuracy of semantic segmentation across semantic classes

PA IoU	Road	Road-side	Vehicle	Lane Mark	Stopping Period	Vertical Objects
Ex1-PA	0.947	0.949	0.995	0.996	0.976	0.9726
IoU	0.889	0.905	0.441	0.609	0.122	0.593
Ex2-PA	0.921	0.932	0.993	0.995	0.967	0.962
IoU	0.842	0.872	0.246	0.568	0.223	0.550

Table II Pixel Accuracy of semantic segmentation across weathers

PA (mIoU)	Sunny facing sun	Sunny back to sun	Rainy	Shadow	Cloudy
Ex1 mPA	0.991	0.975	0.9734	0.9868	0.933
mIoU	0.8054	0.6768	0.4709	0.7500	0.4684
Ex2 mPA	0.9882	0.9725	0.9664	0.9827	0.893
mIoU	0.6153	0.5524	0.4890	0.5827	0.3950

The experiment results displayed in Fig. 6 is an intuitive way to evaluate the accuracy of segmentation. First, road profile semantic segmentation has stunning performances in reducing the influence of shadow, highlight and other illumination changes across different weather. For example, semantic segmentation removed wiper traces in the road profile in a raining day, and leaved no imprints of rain drops from vehicle glass. Second, the testing accuracy in Ex. 2 is lower than Ex. 1 either on semantic classes or weather categories, mainly because the smaller input patch contains less related temporal-spatial information. However, by reducing the height of patch into two pixels in Ex. 2, we find road profile is still learnable, this has great significance in real time prediction of driving scene by just scanning the latest line in the road profile.

In the future work, we will continue segmenting sub-classes on road profile. For example, we will add pedestrian in our labeling work, because the detection of pedestrian is helpful to avoid traffic accident in autonomous driving. Furthermore, in the training dataset, we can change the temporal overlapping pixels between consecutive patches, which has an influence on the learning of some local consecutive curves. Third, there are still space between Ex. 1 and 2 on the selection of patch length, we will try other time lengths between 2 and 256 pixels as a tradeoff. Fourth, in addition to the experiments on those weather above, we can try on some extreme weather categories such as night and dark lit. Last, to enhance the spatial relation of pixels by embedding some post-processing steps such as CRF-based refinement into the end-to-end streaming of networks [15, 16].

V. Conclusion

We apply a temporal-to-spatial approach for real-time autonomous driving with to pixel-wised semantic segmentation. The experimental results are convincing while the cost in calculation is much less than that of frame by frame. Semantic segmentation can be implemented on driving scenes if a spatial 2D view is reduced into a temporal 1D view. In the reduced temporal-spatial space, an accurate and fast semantic segmentation will avoid vehicle collision at a close range. Through the training of road profiles under all kinds of weather and illumination, our model can filter the disturbance of shadow, highlight, rainy drops, and sunny to find drivable regions and objects with collision danger.

References

[1] G. Cheng, J. Y. Zheng, H. Murase, "Sparse coding of weather and illuminations for autonomous driving and ADAS", Proc. IEEE Intell. Veh. Symp., pp. 2030-2035, 2018.

[2] G. Cheng, Z. Wang, J. Y. Zheng, "Modeling Weather and Illuminations in Driving Views Based on Big-Video Mining", Intelligent Vehicles IEEE Transactions on, vol. 3, no. 4, pp. 522-533, 2018.

[3] V. Badrinarayanan, A. Kendall, R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation", IEEE Trans. Pattern Anal. Mach. Intell., 39(12), 2481-2495, Dec. 2017.

[4] J. Long, E. Shelhamer, T. Darrel, "Fully convolutional networks for semantic segmentation", IEEE CVPR, pp. 3431-3440, 2015.

[5] K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition" in IEEE CVPR, 770-778, 2016.

[6] H Zhao, J Shi, X Qi, X Wang, J Jia, "Pyramid scene parsing network", IEEE CVPR, 2017.

[7] T. Pohlen, A. Hermans, M. Mathias, B. Leibe, "Fully-resolution residual networks for semantic segmentation in street scene", IEEE CVPR, 2017.

[8] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The kitti vision benchmark suite. IEEE CVPR, June 2012.

[9] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, "The cityscapes dataset for semantic urban scene understanding", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[10] Z. Wang, G. Cheng, J. Y. Zheng, "Road Edge Detection in All Weather and Illumination via Driving Video Mining", IEEE Trans. Intelligent Vehicles, 2019 (to appear).

[11] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. ECCV, 2012.

[12] M. Abadi, et. al. TensorFlow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint, 1603.04467, 2016. arxiv.org/abs/1603.04467. Software available from tensorflow.org.

[13] B. Wang, V. Fremont, S. A. Rodriguez, "Corlor-based road detection and its evaluation on the kitti road benchmar" in Intelligent Vehicles Symposium Proceedings, 2014 IEEE, 31-36, 2014.

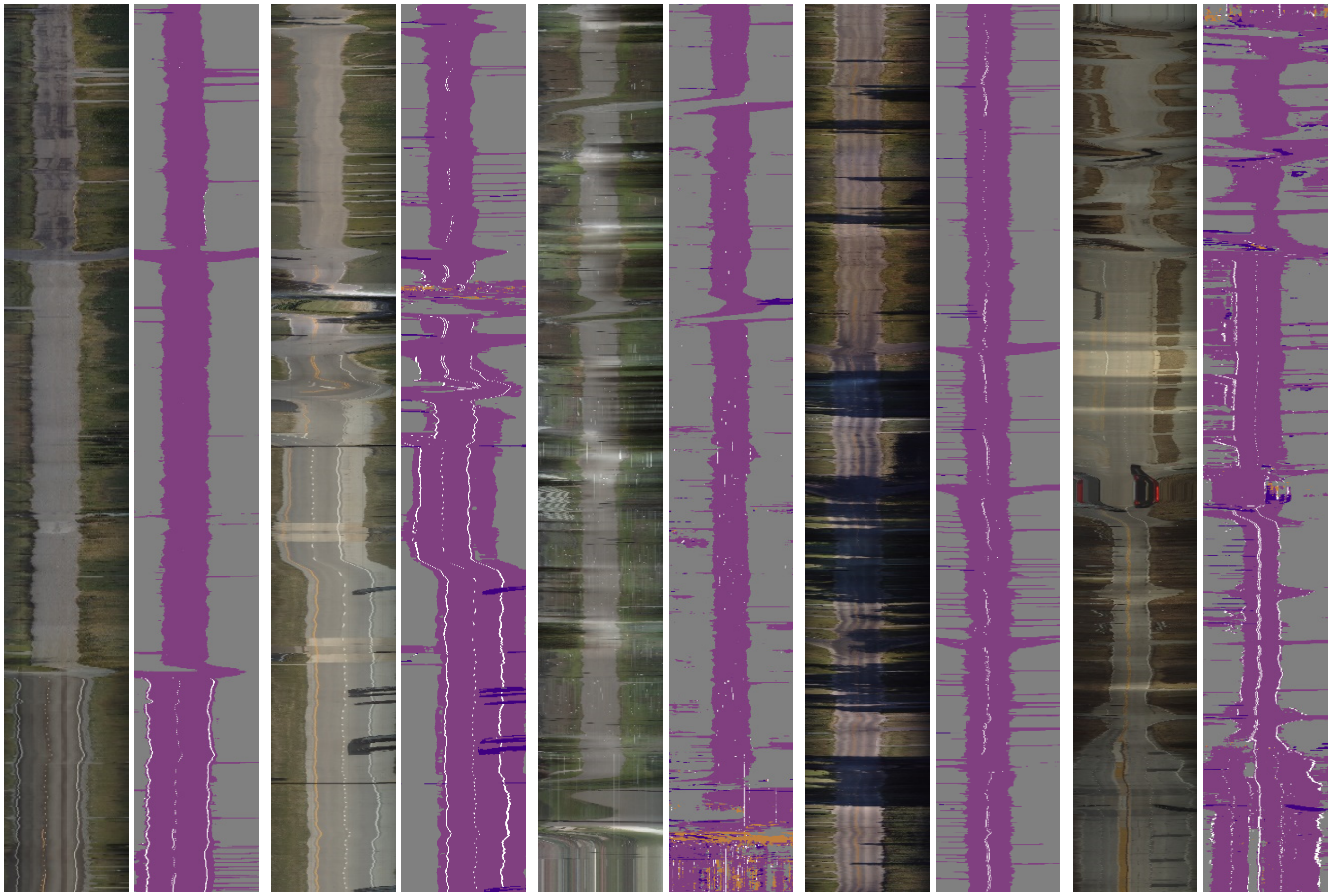
[14] M. Berman, A. R. Triki, M. B. Blaschko. "The Lovasz-Softmax Loss: A Tractable Surrogate for the Optimization of the Intersection-Over-Union Measure in Neural Networks" IEEE CVPR, 2018.

[15] X. Ma, and H. H. Eduard "End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF." CoRR abs/1603.01354 (2016)

[16] P. Knobelreiter, C. Reinbacher, A. Schekhovtsov, T. Pock, "End-to-end training of hybrid CNN-CRF models for stereo," IEEE CVPR, 2017, 2339-2348.

[17] M. Kilicarslan, J. Y. Zheng, "Predict Vehicle Collision by TTC From Motion Using a Single Video Camera", IEEE Trans. on Intelligent Transportation Systems, 1-12, May, 2018.

[18] M. D. Sulistiyo, et. al. Attribute-aware Semantic Segmentation of Road Scenes for Understanding Pedestrian Orientations, IEEE ITSC, 1-6, 2018.



Sunny Facing the Sun | Sunny Back to the Sun | Rainy | Shadow | Cloudy

Figure 6: The road profile semantic segmentation on five weather categories in Ex. 1. The time axes are upward.