Vehicle Windshield Detection by Fast and Compact Encoder-Decoder FCN Architecture

A. Mountelos, A. Amanatiadis, G. Sirakoulis and E. B. Kosmatopoulos

Department of Electrical and Computer Engineering, Democritus University of Thrace, Xanthi, Greece
{alexmoun,aamanat,gsirak,kosmatop}@ee.duth.gr

***Abstract*—Vehicle semantic segmentation is critical in many advanced driving assistance systems, traffic management, and security surveillance systems. Most of such systems are deployed on low computational embedded systems located in the vehicles or in remote gantry and roadside poles. While fully convolutional networks have been proved to be a powerful classifier being able to make inference on every single pixel of the input image, they entail high computational costs even for the inference process. In this paper, a vehicle windshield semantic segmentation is proposed utilizing a fast and compact encoder-decoder architecture of a fully convolutional network implemented in a low-power embedded system. The performed qualitative and quantitative performance measurements exemplify a real-time portable embedded solution which is competitive in terms of performance and inference time.**

I. Introduction

Intelligent Transportation Systems (ITS) and Advanced Driving Assistance Systems (ADAS) are increasingly used for enhancing safety and driving experience. Since both systems can be associated with injuries or fatal accidents, their requirements are strictly defined by superior performance in a diverse set of weather conditions and hard real-time response constraints.

The front or rear windshield detection of a vehicle is a necessary processing stage for several intelligent systems such as high occupancy vehicle systems [1], carpooling, electronic tolls, driver cell phone usage detection [2], and safety systems [3], [4]. Most such systems utilize the infrared range making the necessary equipment costly for wide adoption, thus recent approaches utilize RGB cameras which are also robust for such a detection [5], as shown in Fig. 1. Vehicle windshield detection methods vary from simple image processing algorithms to recent machine learning techniques. A color image segmentation technique was implemented via an onboard automated camera system in order to detect the cars' windshield area from a live feed of roadside images [6]. Recent works on windshield detection are based on machine learning exploitation. Front windshield detection in [7] and [8] make use of appearance learning in deformable parts models [9]. Yuan et. al. [10], [11] suggest that using Hough transform to detect lines and a recommended method using maximum energy for candidate selection, windshield detection is possible in a system working with near-infrared images.

In this work, we propose a deep learning technique for front or rear windshield region semantic segmentation based on a fully convolutional network (FCN). Unlike convolutional neural networks (CNNs) which provide a bounding box for the detected region, the FCN provides accurate region segmentation which is much more efficient for the necessary post processing algorithms. The proposed network architecture is based on a fast and compact encoder-decoder approach to make effective use of the limited available resources in embedded platforms.

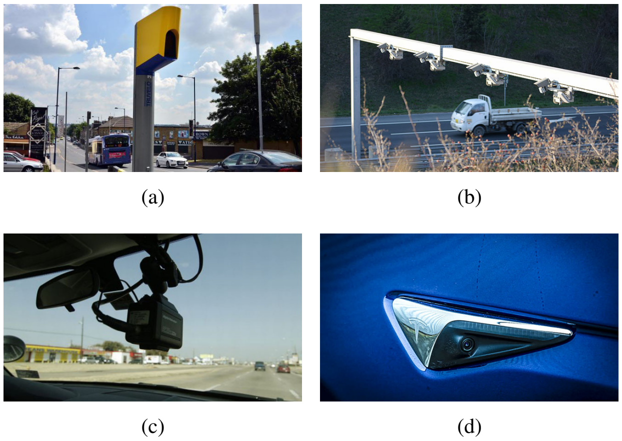

Fig. 1. Overview of different non-infrared camera setups used for intelligent transportation and advanced driving assistance systems. a) Roadside pole camera; b) Gantry mounted cameras; c) Dashboard camera; d) Vehicle built-in camera.

II. Proposed Method

Fully convolutional networks' high capability of dense inference from feature depiction [12], [13], derives from their similarity with deep neural networks which explains their popularity for image semantic segmentation purposes. Fully convolutional networks consist of two main parts; the encoder and the decoder, the first of which uses exclusively convolutional and pooling layers. The pooling layers are responsible for downsampling the image and have no trainable parameters, very much like in a convolutional neural network but excluding the last fully-connected layers to retain spatial coordinates [14]. The last key difference is of high importance because FCNs' training sets are made up of same resolution input and output images, in an effort to include the output probabilities in the resolution of the input image. To achieve

978-1-7281-1184-1/19/$31.00 ©2019 IEEE

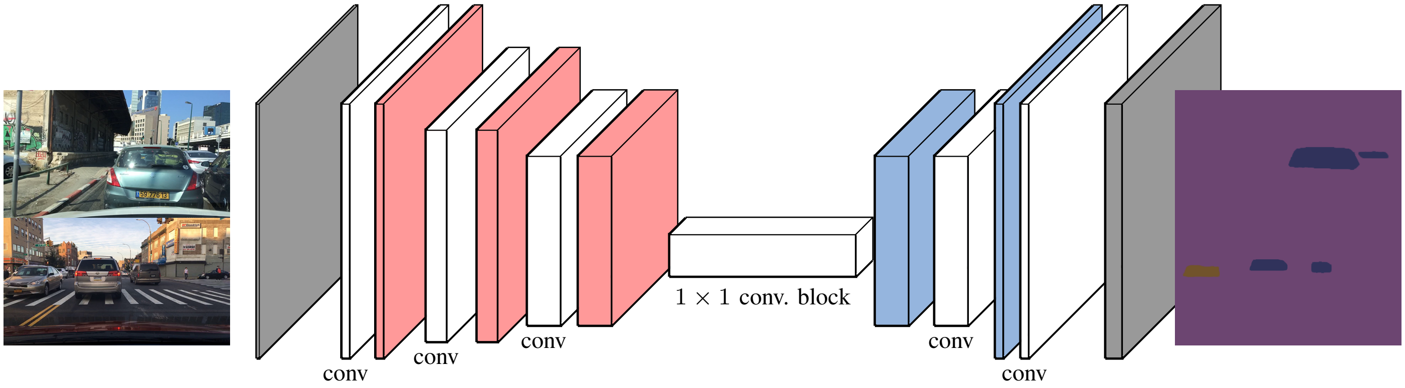

Fig. 2. The proposed fully convolutional network for vehicle windshield segmentation task. The red and blue layers denote the in-network downsampling and upsampling layers in the encoder and decoder, respectively.

the desired output, an FCN utilizes the innovative decoder to upsample the feature maps using convolutions in a backward strides fashion. However, intelligent vision dense tasks such as semantic segmentation need the spatial information to be retained in the output maps.

The overall architecture of the used network is shown in Fig. 2. It consists of three layers in the encoder and two layers in the decoder. In the encoder, the dimension is decreased while preserving spatial information of the image and adding depth to the model by increased parameters at a fairly low computational price. In the decoder blocks, the upsampling layer utilizes the interpolation kernel along with a concatenation step and finally the convolution step. In case different resolution input images may be used, to not limit the use of a network to one resolution a 1×1 convolutional layer lies between the encoder and the decoder. Three classes were chosen for the image segmentation namely for the rear windshield, the front windshield and the rest pixels.

Since there is no dataset with vehicle windshield segmentation, a subset of 1300 images from the BDD100K [15] dataset was fine-annotated utilizing the open-source scalabel annotation tool. The annotation utilized parametric Bezier curves for fitting the windshield labels with much higher accuracy and efficiency compared to rectangular bounding objects. The training was performed in a Titan Xp GPU and the selected training hyperparameters are listed in Table I. To shorten the training time and overall improve the network's performance, batch normalization was used, as it also results in faster gradient descent convergence and correspondingly to higher learning rates.

TABLE I
FCN HYPERPARAMETERS

Training Parameter	Tuned Value
Batch Size	64
Learning Rate	0.005
Number of Epochs	60
Steps per Epoch	100
Validation Steps	50
Workers (GPU)	120

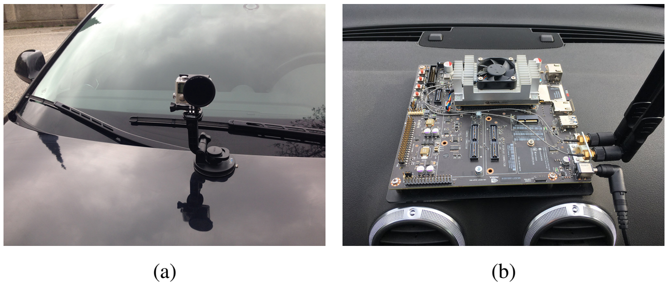
(a)
(b)

Fig. 3. The on-board system. a) The camera setup used for the on-board video streaming. b) The low-power embedded inference system.

III. REAL-TIME INFERENCE SYSTEM

The importance of the hood mounted, as shown in Fig. 3(a), camera's resolution and frame rate are vital for our method to yield satisfactory results. The camera chosen was a GoPro Hero4 Black which is capable of up to 2704×1520 with 60 frames per second video with a multiple layer polarizing filter to reduce windshield glare. The camera footage undergoes both; post calibration and rectification to further diminish barrel distortion even though the field of view was set to a $21mm$ equivalent to minimize distortion in the first place. The above camera balances resolution and sampling rate to achieve the required performance in our real-time scenario.

The on-board real-time inference was implemented on the NVIDIA Jetson TX1 Developer Kit [16], as shown in Fig. 3(b). While the inference stage is characterized by smaller computational workloads the predictions are highly constrained by time, memory, and power consumption due to the deployment on portable devices, robots [17] or autonomous vehicles. The NVIDIA Jetson TX1 is an embedded visual computing system with a 64-bit ARM A57 CPU, a 1 T-Flop/s 256-core Maxwell GPU and 4 GB of shared RAM.

Several design choices have been followed concerning the network architecture or model parameters. Since there is a strong reduction in resolution during the downsampling stages of the encoder, a loss of spatial information is apparent in

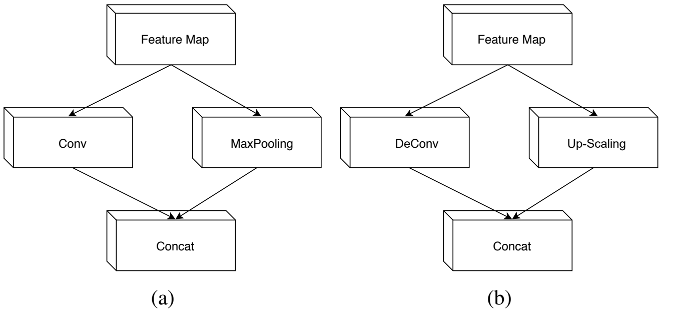

Fig. 4. a) Encoder parallelization of pooling operation together with a convolution operation resulting into a concatenated feature map. b) Decoder parallelization of up-scaling operation together with a deconvolution operation. The selected stride depends on the pooling or scaling factor.

the final output image. For this reason, we utilized sparse upsampled maps in the decoder from saved indices of the max pooling layers of the encoder, allowing us to reduce memory requirements compared to the widely used skip connections. For the upsampling, the nearest neighbor interpolation was used since it is the fastest fixed kernel with very low accuracy drop [18]. A fixed kernel was selected against the learnable ones for not introducing additional parameters in the network [19].

An early stage optimization of the network was performed by reducing the input size of the first two layers since the processing of the first large input frames is very expensive. However, this optimization is application-specific and cannot be utilized in all FCN implementations. More precisely, in our application, the windshield visual information is highly spatially redundant and thus can be compressed into a more efficient representation.

Many FCN implementations use the same architecture for the encoder and the decoder resulting in a symmetric architecture. In our case, the decoder is smaller than the encoder by one layer, since the decoder only upsamples the output of the encoder and fine-tunes the details. Finally, in the first step of the encoder, we performed pooling operation in parallel with convolution and concatenated the resulting feature maps. The same parallelization was applied in the last step of the decoder for the deconvolution together with the up-scaling operation, as shown in Fig. 4. This technique, inspired by the work in [20], allowed us to speed up the inference time.

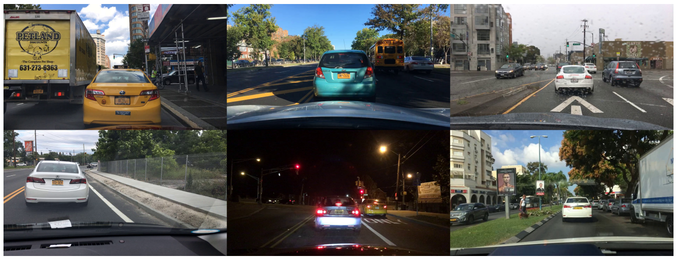

Fig. 5. A subset of the used video sequences from the BDD100K Dataset with diverse driving conditions.

TABLE II
PERFORMANCE COMPARISON

Method	NVIDIA TX1			
	640 × 360		1280 × 720	
	ms	fps	ms	fps
Elastic Deformation [7]	732	1.36	1152	0.86
Fully Convolutional Network	96	10.41	162	6.17

IV. Experimental Comparison Results

Apart from our custom dataset, experimental comparison results were performed also on a open and widely available dataset. Since image resolution is very critical for our method to perform adequately, the selected dataset for the comparison should meet this requirement. Among the many available autonomous driving datasets, the BDD100K [15] from Berkeley was the most appropriate one mainly because of its high camera resolution (1280 × 720 pixels) and high frame rate (30 fps). The dataset includes over 100K videos with significant diversity in weather, vehicle type, and time of day. We selected a subset of the database for the training, as discussed previously, and a subset for the testing while retaining the diversity of driving conditions, as shown in Fig. 5.

Since there is no similar work utilizing FCNs for windshield detection, our method was compared against a machine learning approach presented in [7] based on an elastic deformation model. For fair comparison, we added additional landmarks for allowing the method to locate frontal but also rear windshields without the need of including the two side view mirror landmarks.

In Table II, we report the inference timing results on the NVIDIA TX1 embedded system module for input image sizes of 640 × 360 and 1280 × 720, which are adequate for practical autonomous vehicle applications. The efficiency of the proposed method is evident, as its performance is one order of magnitude faster compared to the elastic deformation method. The storage model parameter requirements in half precision floating point format are only 0.7MB, making it efficient even for more limited edge devices, such as neural computation sticks [21], [22], since the whole network can fit in extremely limited and fast on-chip memory.

For the segmentation results, we use the standard Jaccard Index, commonly known as the PASCAL VOC intersection-over-union metric [23]:

$$IoU = \frac{Tp}{Tp + Fp + Fn} \tag{1}$$

where Tp, Fp, and Fn are the true positive, false positive, and false negative pixel numbers, respectively.

In Table III, the segmentation results for both the custom and the BDD100K datasets are reported for the three classes “front windshield”, “rear windshield” and “not windshield”.

TABLE III
DATASET SEGMENTATION RESULTS

Method	Class IoU	
	Custom	BDD100K
Elastic Deformation [7]	69.4	53.4
Fully Convolutional Network	78.1	67.3

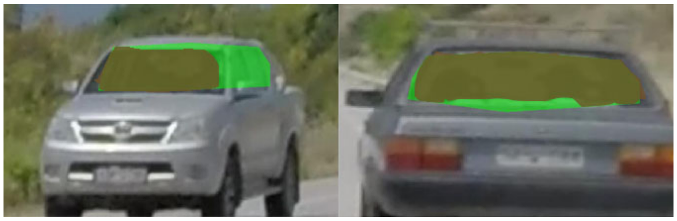

Fig. 6. Front and rear windshield predictions for FCN (brown color) and elastic deformation (green color).

The proposed method outperforms the elastic deformation method in class IoU for both datasets. Example predictions from both methods in the custom dataset are shown in Fig. 6. Apart from the more accurate results, it is apparent that bezier annotation during the training is more efficient allowing fitting curved boundaries against the line segments which are used in the elastic deformation method.

V. CONCLUSIONS

We have proposed a novel deep learning architecture based on a fully convolutional network for vehicle windshield semantic segmentation. Our main aim was to make efficient use of the limited resources of the embedded platforms for efficient deployment on autonomous vehicles, roadside poles or gantry mounted optical sensors. The proposed FCN model includes less parameters and requires less memory to operate. Furthermore, our work provides large gains in accuracy results compared to existing baseline models while maintaining the desired real-time performance on the NVIDIA TX1 portable embedded solution.

ACKNOWLEDGMENT

The current postdoctoral research was fully funded by the State Scholarships Foundation (IKY) through the “Strengthening Post-Academic Researchers” act from the resources of the Operational Program “Human Resources Development, Education and Lifelong Learning” with Priority Axes 6, 8, 9 and co-funded by the European Commission Social Fund - ESF and the Greek government.

The authors also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Jetson TX1 Developer Kit used for this research.

REFERENCES

[1] Y. Artan, O. Bulan, R. P. Loce, and P. Paul, “Passenger compartment violation detection in hov/hot lanes.” *IEEE Trans. Intelligent Transportation Systems*, vol. 17, no. 2, pp. 395–405, 2016.

[2] ——, “Driver cell phone usage detection from hov/hot nir images,” in *IEEE conference on Computer Vision and Pattern Recognition Workshop*, 2014, pp. 225–230.

[3] A. Amanatiadis, E. Karakasis, L. Bampis, S. Ploumpis, and A. Gasteratos, “ViPED: On-road vehicle passenger detection for autonomous vehicles,” *Robotics and Autonomous Systems*, vol. 112, pp. 282 – 290, 2019.

[4] A. Amanatiadis, K. Charalampous, I. Kostavelis, B. Birkicht, B. Andel, V. Meiser, C. Henschel, S. Baugh *et al.*, “Autonomous vehicle emergency recovery tool: a cooperative robotic system for car extraction,” *Journal of Field Robotics*, vol. 33, no. 8, pp. 1058–1086, 2016.

[5] B. Balci, B. Alkan, A. Elihos, and Y. Artan, “Front seat child occupancy detection using road surveillance camera images,” in *IEEE International Conference on Image Processing*, 2018, pp. 1927–1931.

[6] P. M. Birch, R. C. Young, F. Claret-Tournier, and C. R. Chatwin, “Automated vehicle occupancy monitoring,” *Optical Engineering*, vol. 43, no. 8, pp. 1828–1833, 2004.

[7] B. Xu, P. Paul, Y. Artan, and F. Perronnin, “A machine learning approach to vehicle occupancy detection,” in *IEEE International Conference on Intelligent Transportation Systems*, 2014, pp. 1232–1237.

[8] B. Xu and R. P. Loce, “A machine learning approach for detecting cell phone usage,” in *SPIE Video Surveillance and Transportation Imaging Applications*, 2015, pp. 94 070A–94 070A.

[9] X. Zhu and D. Ramanan, “Face detection, pose estimation, and landmark localization in the wild,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2879–2886.

[10] X. Yuan, Y. Meng, and X. Wei, “A method of location the vehicle windshield region for vehicle occupant detection system,” in *IEEE Int. Conference on Signal Processing*, vol. 1, 2012, pp. 712–715.

[11] X. Yuan, Y. Meng, X. Hao, H. Chen, and X. Wei, “A vehicle occupant counting system using near-infrared (NIR) image,” in *IEEE Int. Conference on Signal Processing*, vol. 1, 2012, pp. 716–719.

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

[13] 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 *Int. Conference on Learning Representations*, 2015.

[14] A. Amanatiadis, V. Kaburlasos, and E. Kosmatopoulos, “Understanding deep convolutional networks through Gestalt theory,” in *IEEE International Conference on Imaging Systems and Techniques*, 2018, pp. 312–317.

[15] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell, “BDD100K: A diverse driving video database with scalable annotation tooling,” *arXiv preprint arXiv:1805.04687*, 2018.

[16] N. Otterness, M. Yang, S. Rust, E. Park, J. H. Anderson, F. D. Smith, A. Berg, and S. Wang, “An evaluation of the NVIDIA TX1 for supporting real-time computer-vision workloads,” in *IEEE Real-Time and Embedded Technology and Applications Symposium*, 2017, pp. 353–364.

[17] A. Amanatiadis, “A multisensor indoor localization system for biped robots operating in industrial environments,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 12, pp. 7597–7606, 2016.

[18] A. Amanatiadis and I. Andreadis, “A survey on evaluation methods for image interpolation,” *Measurement Science and Technology*, vol. 20, no. 10, p. 104015, 2009.

[19] A. Amanatiadis, V. Kaburlasos, and E. Kosmatopoulos, “Interpolation kernels in fully convolutional networks and their effect in robot vision tasks,” in *IEEE International Conference on Imaging Systems and Techniques*, 2018, pp. 232–236.

[20] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[21] A. M. Khan, I. Umar, and P. H. Ha, “Efficient compute at the edge: Optimizing energy aware data structures for emerging edge hardware,” in *International Conference on High Performance Computing & Simulation*, 2018, pp. 314–321.

[22] J. Hochstetler, R. Padidela, Q. Chen, Q. Yang, and S. Fu, “Embedded deep learning for vehicular edge computing,” in *ACM/IEEE Symposium on Edge Computing*, 2018, pp. 341–343.

[23] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes challenge: A retrospective,” *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.