Semantic information based vehicle relative orientation and taillight detection

Flaviu Ionut Vancea
Computer Science Departmen
Technical University of
Cluj-Napoca, Romania
Email: flaviu.vancea@gmail.com

Sergiu Nedevschi
Computer Science Departmen
Technical University of
Cluj-Napoca, Romania
Email: sergiu.nedevschi@cs.utcluj.ro

***Abstract*—Vehicle taillight detection is an important topic in the fields of collision avoidance systems and autonomous vehicles. By analyzing the changes in the taillights of vehicles, the intention of the driver can be understood, which can prevent possible accidents. This paper presents a convolutional neural network architecture capable of segmenting taillight pixels by detecting vehicles and uses already computed features to segment taillights. The network is composed of a Faster RCNN that detects vehicles and classify them based their orientation relative to the camera and a subnetwork that is responsible for segmenting taillight pixels from vehicles that have their rear facing the camera. Multiple Faster RCNN configurations were trained and evaluated. This work also presents a way of adapting the ERFNet semantic segmentation architecture for the purpose of taillight extraction, object detection and classification. The networks were trained and evaluated using the KITTI object detection dataset.**

***Index Terms*—Vehicle detection, taillight detection, segmentation, deep learning, convolutional neural networks, vehicle orientation detection.**

I. Introduction

Analyzing the state of vehicle taillights is an important topic in the field of collision avoidance systems since it can inform drivers of the how other vehicles will behave in the near future. In autonomous driving systems, knowing the intention of other drivers can lead to better route planning and can reduce the number of avoidable accidents. In order to analyze their state, taillights must be first identified from an image.

Most existing methods focus on detecting taillights using explicit thresholds to extract red pixels or using a convolutional neural network (CNN) on extracted vehicle bounding boxes to segment taillight pixels. Since the methods using explicit thresholds are sensitive to illumination changes and are incapable extracting taillights from red vehicles, and the CNN based methods require the detection of vehicle bounding boxes before passing vehicle sub-images through a large network, this paper presents a convolutional neural network for extracting the taillights from vehicles during daytime scenarios using features computed for vehicle detection. The main contributions of the paper are the following:

- Proposed a CNN structure that detects vehicles and their orientation and uses the already computed features and the orientations to segment taillights.
- Different convolutional neural networks capable of taillight segmentation were trained and evaluated.
- The ERFNet model for semantic segmentation was modified for the purpose of object detection, object orientation detection and taillight segmentation.

This paper is organized in the following way: Section II presents other works on the topic of taillight detection, Section III provides a detailed description of the neural network architecture, Section IV details the training process and shows the experimental results of the system and section V concludes the paper.

II. Related Work

Many of the early systems for vehicle signaling detection [1]–[5] were capable of extracting taillights only during nighttime scenarios. Other systems capable of extracting vehicle taillights during daytime have been proposed in [6], [7] and [8]. Since taillights are predominantly red, most of those systems focus on extracting red pixels from images. This approach is very sensitive to illumination changes in the image and are incapable of extracting taillights from red vehicles.

Almagambetov et al. [6] have proposed a system for signaling detection in which they segment taillight pixels by first transforming an image into the Y'UV color space and then thresholding the obtained image to extract red and white pixels. The obtained regions are grouped into pairs if they passed two tests. The first test checks if two regions have similar shapes and sizes and if they are situated at a similar height in the image. The second test compares checks if two regions have similar colors by comparing their histograms. The authors also use Kalman filtering to track detected taillights in order to reduce false negatives.

In [7]–[10] the authors first detected vehicles and then searched for taillights within the obtained bounding boxes. The system described in [7] converting an image into the HSV color space, uses explicit thresholds to extract red pixels and then uses the OPTICS algorithm to extract the largest two clusters. In [8], the authors use the Y-distance test from [6] and compare the sizes and shapes of regions to detect pairs. In [10], the authors detect rear lights by using a lamp response function which measures the chance of a pixel to be a red component.

978-1-5386-8445-0/18/$31.00 ©2018 IEEE

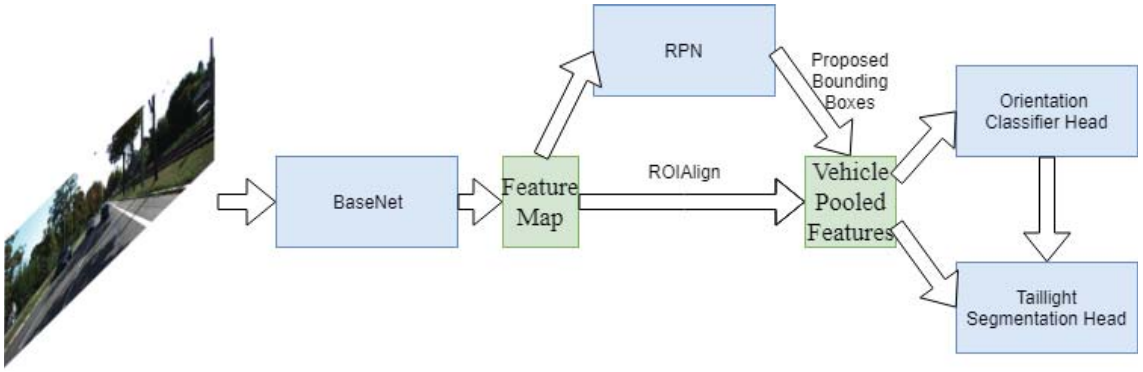

Fig. 1. Network architecture block diagram. An image is passed through a base network that produces a feature map. The feature map is used by a RPN to produce object bounding boxes which are refined by the classification subnetwork. The classification subnetwork also outputs an orientation of the detected objects relative to the camera. The rear-view cars are passed through a taillight segmentation head which extracts the taillight pixels.

More robust deep learning based methods for taillight extraction were proposed in [11] and [12]. G. Zhong et al. [12] first detect vehicles using fast RCNN [13] and then segment rear light regions using a fully convolutional network (FCN) [14]. J. G. Wang et al. [11] have fine tuned an AlexNet network with ”breaking” and ”normal” training samples to recognize if a vehicle is breaking or not.

The method described in this paper is a continuation of [15] in which the system first detects vehicles using a boosting classifier. Two methods are used to segment the taillight pixels from detected vehicle bounding boxes. The first method uses explicit thresholds for detecting red pixels, and the second method uses a FCN to segment the taillights. The main drawbacks of the first method are that it is incapable of detecting taillights from red vehicles and that it struggles with illumination changes in the scene. The second method is more robust to illumination changes and is able to extract taillight pixels from red vehicles but it has the disadvantage of needing to pass each detected vehicle sub-image separately through a large network. This work aims to extract taillight pixels from vehicles with their rear facing the camera by using the same features for both vehicle detection and taillight segmentation.

III. Proposed network architecture

Vehicle detection is an important part in taillight segmentation since it reduce the number of false positive detections and reduces the search space for taillights. Different neural networks were trained in order to detect vehicle taillights. CNNs such as Faster RCNN [16] and YOLO [17] have made a big impact in the field object recognition, being able to accurately extract bounding boxes of several object categories from images. Inspired by the work Mask RCNN [18], where the authors have extended a Faster RCNN to use the same features computed by a neural network for both object detection and segmentation in order to achieve instance level segmentation, this work uses one such network to both extract vehicle bounding boxes and segment taillight pixels.

To reduce the taillight search space and to decrease the number of false positives even further, only vehicles having their rear side facing the camera should be processed for taillight segmentation.

Based on the information above, this work uses a Faster RCNN to detect vehicle bounding boxes and classify them based on their orientation relative to the camera. A taillight segmentation head is then trained to extract taillight pixels from the detected bounding boxes classified as having their rear facing the camera. A block diagram of the network can be viewed in Fig. 1.

A. Vehicle detection

Vehicle detection and classification was done using a Faster-RCNN. The FasterRCNN is composed of a base network which outputs a feature map, a Region Proposal Network (RPN), and a classification network. The RPN proposes a set of anchor boxes at each feature point from the feature map, classifies the anchor boxes into one of two classes (object and non-object) and computes regression offsets for refined object detection. The classifier assigns different classes to the detected boxes and further corrects the bounding boxes using different offsets.

The anchors used to train the RPN have been constructed using a set of scales and aspect ratios from an original bounding box of size 16x16. The scales used were $\{1, 2, 3, 4, 5\}$ and the ratios used were $\{0.25, 0.5, 0.75, 1\}$ resulting in a total of $5 * 4 = 20$ anchors per point in the feature map.

The following base networks were trained for the purpose of object detection and classification: VGG16 [19], AlexNet [20] and a variation of the ERFNet encoder [21]. The problem with the original ERFNet encoder is that it has only 3 downsampler blocks, unlike VGG16 which has 4 pooling layer used to produce the feature map. ROIAlign is used to reduce the feature map corresponding to each detected box to a fixed size (7x7 for VGG16 and ERFNet and 6x6 for AlexNet). In case of the ERFNet encoder, ROIAlign does a much bigger downscale of the features which leads to overfitting the training data by having a mAP on the training set of over 0.7% and under 0.35% on the validation set. ERFNet encoder was modified in the following way: another downsampler block

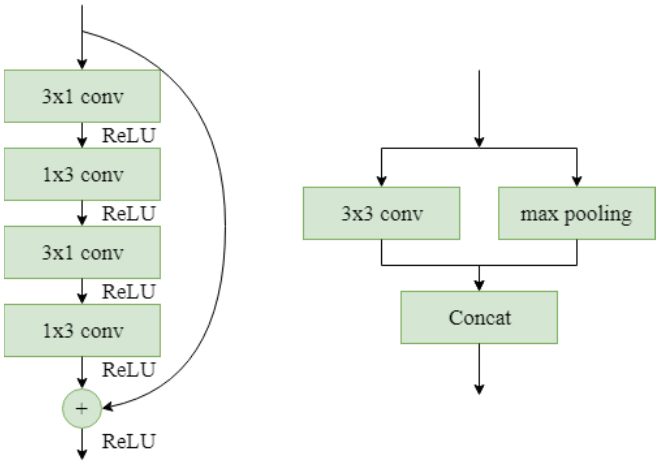

Fig. 2. Left: non-bottleneck-1d block. Right: Downsampler block; the outputs of the convolutional and max pooling layers are concatenated into a feature map.

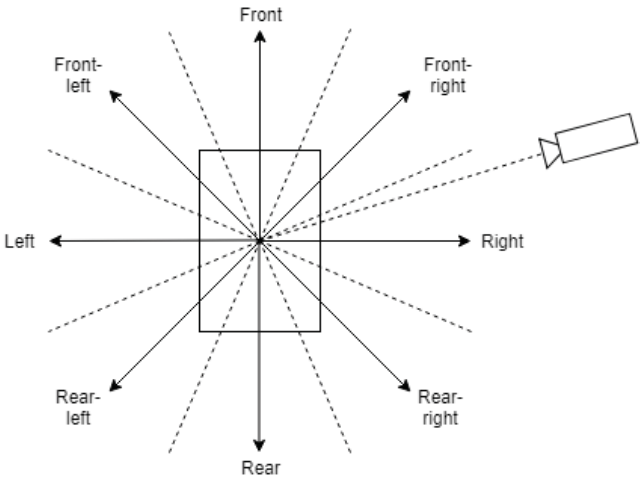

Fig. 3. The eight orientation classes relative to the camera.

that outputs 256 channels was added before the last 4 non-bottleneck-1d blocks and the number of filters in those last 4 blocks was doubled therefore it outputs a feature map of 256 channels instead of 128. The downsampler blocks and the non-bottleneck-1d blocks are illustrated in Fig. 2.

B. Relative orientation detection

The orientation of detected vehicles relative to the camera is obtained by the network classifier. The classifier can output one of 9 classes, 8 for orientation as shown in Fig. 3 and one for background. The classifier for AlexNet and VGG are the same as in the original networks. The classification layers for each of those networks are shown in Table I. For all networks, the fist two fully connected layers produce 4096 channel outputs and the third produces 9 channel outputs.

TABLE I
CLASSIFICATION LAYERS

BaseNet	AlexNet	VGG16	ERFNet
Layers	Dropout	Fully Connected	Fully Connected
	Fully Connected	ReLU	ReLU
	ReLU	Dropout	Dropout
	Dropout	Fully Connected	Fully Connected
	Fully Connected	ReLU	ReLU
	ReLU	Dropout	Dropout
	Fully Connected	Fully Connected	Fully Connected

C. Taillight segmentaiton

A taillight segmentation head was added to the network for the purpose of taillight segmentation. Since we are only interested in rear facing vehicles, only the vehicles classified as Rear, Rear-left or Rear-right are passed through the taillight segmentation head. The taillight segmentation heads used are shown in Table II. All segmentation heads contain 2 deconvolutional layers, also called transposed convolution layers, which are layers trained to upscale feature maps. The upscale block in the case of ERFNet is a deconvolutional layer followed by a batch normalization layer. The results of all segmentation heads are passed through a 1x1 convolutional layer that reduces the the number of channels to 1 and a sigmoid layer that transforms the values at each point between (0, 1) resulting in a grayscale image. This image is thresholded with 0.5 such that the value 0 represents background pixels and the value 1 represent taillight pixels. Since the taillight segmentation head outputs a fixed size image (28 x 28 for VGG16 and ERFNet and 24 x 24 for AlexNet), the output is then resized to the size of the detected bounding box.

TABLE II
TAILLIGHT SEGMENTATION HEADS

AlexNet and VGG16		ERFNet	
Layers	**num ch**	**Layers**	**num ch**
Deconv 2x	256	Upscale	128
Conv 3x3	256	Non-bt-1d	128
Conv 3x3	256	Non-bt-1d	128
Conv 3x3	256	Upscale	64
Conv 3x3	256	Non-bt-1d	64
Deconv 2x	256	Non-bt-1d	64
Conv 1x1			
Sigmoid			

IV. EXPERIMENTAL RESULTS

The models were trained using a subset of the KITTI object detection dataset [22]. Since the ground truth is needed to quantize the orientations, 80% of the training set was used as train data, 10% as validation data and the rest 10% as test data. The Faster RCNN is trained first on the train set and then the taillight segmentation head is added to the network. For taillight extraction, 811 images from the KITTI object

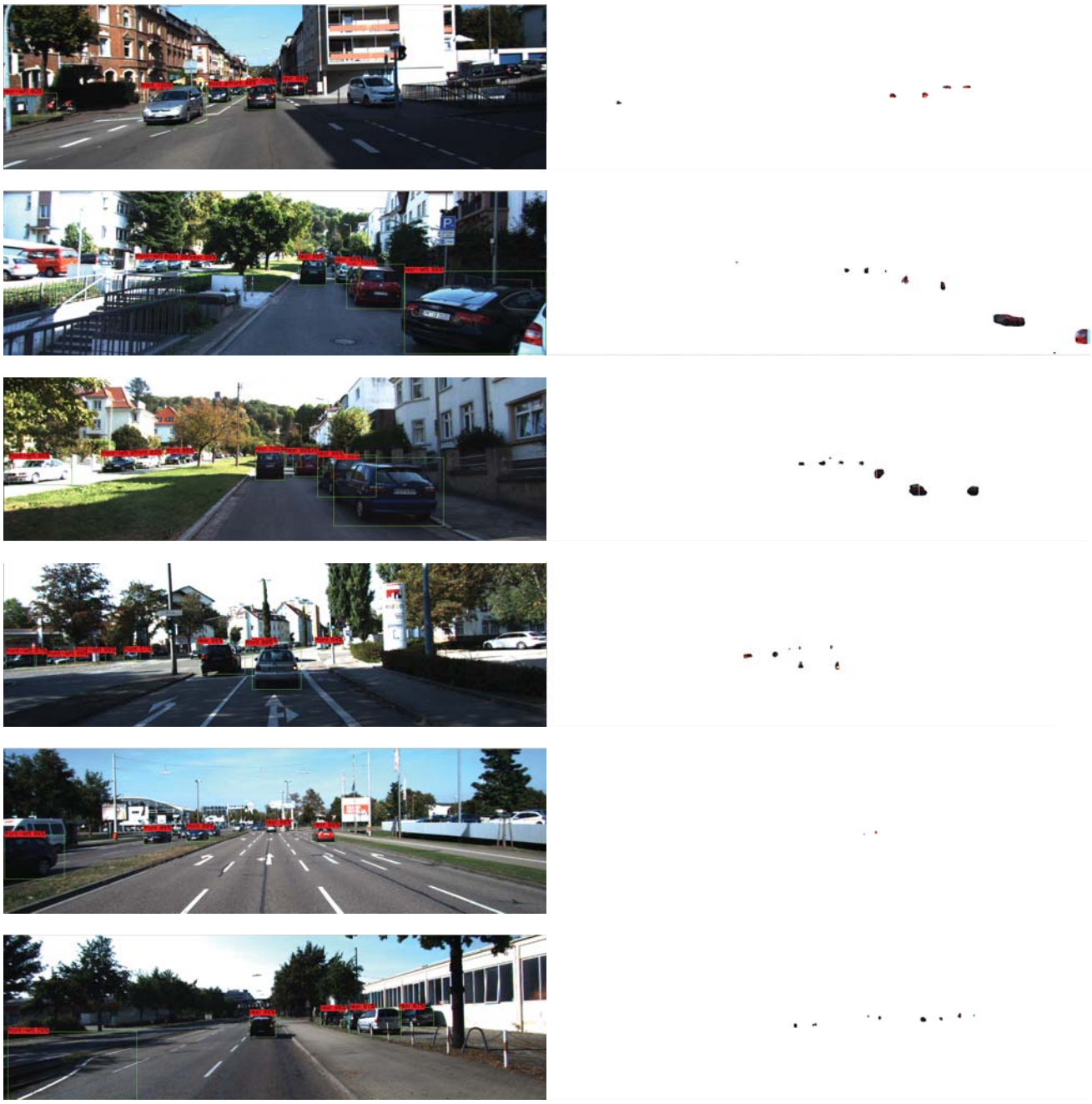

Fig. 4. Network results with ERFNet as a base network. Left column: vehicle and orientation detection. Right column: taillight segmentation.

detection dataset were manually labeled out of which 649 images were used for training, 81 for validation and the rest 81 for testing. Since the taillight set is just a small part of the KITTI dataset, the Faster RCNN part of the network is frozen. Fig. 4 presents the results of the network that uses ERFNet as a base. All training and testing was done on a system with an i5-2500 CPU, a GTX Titan Black GPU with 6GB VRAM and 8GB RAM.

A. Vehicle detection and relative orientation classification

The models that use VGG16 as base networks, were trained using stochastic gradient descent (SGD) [23] over 30 epochs with the learning rate set to 0.001 for the first 10 epochs, 0.0001 for the next 10 epochs and 0.00001 for the last 10 epochs. The model with AlexNet as a base network was trained in a similar manner but the learning rate was reduces every 30 epochs instead of every 10 and was trained over 150 epochs. The models with ERFNet as the base network

TABLE III
ORIENTATION DETECTION PERFORMANCE

Base Network	Right	Front-right	Front	Front-left	Left	Rear-left	Rear	Rear-right
VGG16	60.54%	58.24%	79.98%	67.69%	69.29%	68.64%	80.48%	69.83%
AlexNet	62.06%	60.02%	79.34%	69.77%	61.80%	70.19%	80.79%	62.59%
ERFNet	60.93%	46.70%	79.06%	69.02%	60.80%	67.55%	79.75%	70.60%

were trained using the Adam [24] optimizer over 100 epochs with a starting learning rate of 0.0005 set every epoch as $0.0005 * (1 - (epoch - 1)/100)^{0.9}$. The performance of the networks was measured using mean average precision (mAP) metric. A detection is considered valid if the intersection over union between the detected box and a ground truth box is greater than 0.5. Table IV shows the mAP on the validation and test sets and table III shows the average precision (AP) for all the orientation classes. It can be seen that when ERFNet is trained from scratch, instead of using weights pretrained on ImageNet [25], the mAP is almost 6% higher. Only the ERFNet model trained from scratch is illustrated in table III.

TABLE IV
ORIENTATION MAP

Base Network	Imagenet pretrained	validation mAP	test mAP
VGG16	yes	71.05%	69.22%
AlexNet	yes	71.83%	68.32%
ERFNet	yes	64.00%	63.33%
ERFNet	no	69.97%	66.81%

The loss of the FasterRCNN is computed as $L = L_{rpg\_cls} + L_{rpg\_reg} + L_{cls} + L_{reg}$ where . L_{rpg_cls} is the log loss over the vehicle and non-vehicle classes, L_{cls} is the log loss over the orientation classes, and L_{rpg_reg} and L_{reg} are the bounding box regression losses. The smooth L_1 loss, defined in (1), is used for computing the regression losses and it is used instead of L_2 loss since it is less sensitive to outliers.

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \quad (1)$$

B. Taillight Segmentation

After adding the taillight segmentation head and freezing the Faster RCNN, all models were trained over 60 epochs using the Adam optimizer with a starting learning rate of 0.001. The loss of taillight segmentation head is computed using Binary Cross Entropy. The learning rate is decreased by half every 10 epochs. The performance of the segmentation is evaluated using the Intersection over Union (IoU) metric shown below:

$$IoU = \frac{TP}{TP + FP + FN} \quad (2)$$

where TP is the number of true positives, FP is the number of false positives and FN is the number of false negatives. Table V shows the IoU on the validation and test sets as well as the average inference time of the networks.

TABLE V
TAILLIGHT SEGMENTATION IOU AND INFERENCE TIME

Base Network	validation IoU	test IoU	Inference time
VGG16	21.06%	19.74%	235ms
AlexNet	18.72%	21.55%	77ms
ERFNet	24.03%	24.45%	98ms

V. CONCLUSION AND FUTURE WORK

This work proposes a convolutional neural network structure capable of segmenting vehicle taillights using precomputed features. The system uses a Faster RCNN to detect vehicles and classify them based on their relative orientation to the camera along with a segmentation head that extracts taillight pixels. Three different network configurations were evaluated. The ERFNet encoder architecture was modified for the purposes of object detection by increasing the amount of filters in its last 4 blocks and adding a new downsampler block before them.

Future work will focus on improving the taillight segmentation IoU by exploring different architectures for the taillight segmentation head, increasing the number of images used for training the taillight segmentation head and using a Feature Pyramid Network (FPN) [26] as a base network in order to use features from earlier layers for taillight segmentation. A method for training the network end-to-end, instead of training the Faster RCNN first and then the taillight segmentation head, will also be explored.

ACKNOWLEDGMENT

This work was partially supported by the by the UP-Drive project (Automated Urban Parking and Driving), Horizon 2020 EU funded, Grant Agreement Number 688652 and partially supported by a grant of Ministry of Research and Innovation, CNCS - UEFISCDI, project number PN-III-P4-ID-PCE-2016-0727, within PNCDI III.

REFERENCES

[1] D. Y. Chen and Y. J. Peng, "Frequency-tuned taillight-based nighttime vehicle braking warning system," *IEEE Sensors Journal*, vol. 12, no. 11, pp. 3285–3292, Nov 2012.

[2] R. O'Malley, E. Jones, and M. Glavin, "Rear-lamp vehicle detection and tracking in low-exposure color video for night conditions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 453–462, June 2010.

[3] Y. L. Chen, B. F. Wu, and C. J. Fan, "Real-time vision-based multiple vehicle detection and tracking for nighttime traffic surveillance," in *2009 IEEE International Conference on Systems, Man and Cybernetics*, Oct 2009, pp. 3352–3358.

[4] N. Alt, C. Claus, and W. Stechele, "Hardware/software architecture of an algorithm for vision-based real-time vehicle detection in dark environments," in *2008 Design, Automation and Test in Europe*, March 2008, pp. 176–181.

[5] P. Thammakaroon and P. Tangamchit, "Predictive brake warning at night using taillight characteristic," in *2009 IEEE International Symposium on Industrial Electronics*, July 2009, pp. 217–221.

[6] A. Almagambetov, S. Velipasalar, and M. Casares, "Robust and computationally lightweight autonomous tracking of vehicle taillights and signal detection by embedded smart cameras," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3732–3741, June 2015.

[7] Z. Cui, S. W. Yang, and H. M. Tsai, "A vision-based hierarchical framework for autonomous front-vehicle taillights detection and signal recognition," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Sept 2015, pp. 931–937.

[8] H. T. Chen, Y. C. Wu, and C. C. Hsu, "Daytime preceding vehicle brake light detection using monocular vision," *IEEE Sensors Journal*, vol. 16, no. 1, pp. 120–131, Jan 2016.

[9] C. L. Jen, Y. L. Chen, and H. Y. Hsiao, "Robust detection and tracking of vehicle taillight signals using frequency domain feature based adaboost learning," in *2017 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW)*, June 2017, pp. 423–424.

[10] L. C. Chen, J. W. Hsieh, S. C. Cheng, and Z. R. Yang, "Robust rear light status recognition using symmetrical surfs," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Sept 2015, pp. 2053–2058.

[11] J. G. Wang, L. Zhou, Y. Pan, S. Lee, Z. Song, B. S. Han, and V. B. Saputra, "Appearance-based brake-lights recognition using deep learning and vehicle detection," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, June 2016, pp. 815–820.

[12] G. Zhong, Y.-H. Tsai, Y.-T. Chen, X. Mei, D. Prokhorov, M. James, and M.-H. Yang, "Learning to tell brake lights with convolutional features," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2016, pp. 1558–1563.

[13] R. Girshick, "Fast r-cnn," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 1440–1448.

[14] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *CoRR*, vol. abs/1605.06211, 2016. [Online]. Available: http://arxiv.org/abs/1605.06211

[15] F. I. Vancea, A. D. Costea, and S. Nedevschi, "Vehicle taillight detection and tracking using deep learning and thresholding for candidate generation," in *2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, Sept 2017, pp. 267–272.

[16] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015. [Online]. Available: http://arxiv.org/abs/1506.01497

[17] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," *CoRR*, vol. abs/1612.08242, 2016. [Online]. Available: http://arxiv.org/abs/1612.08242

[18] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017. [Online]. Available: http://arxiv.org/abs/1703.06870

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[21] E. Romera, J. M. lvarez, L. M. Bergasa, and R. Arroyo, "Erfnet: Efficient residual factorized convnet for real-time semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 263–272, Jan 2018.

[22] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 3354–3361.

[23] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, Dec 1989.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[25] 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)*, vol. 115, no. 3, pp. 211–252, 2015.

[26] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," *CoRR*, vol. abs/1612.03144, 2016. [Online]. Available: http://arxiv.org/abs/1612.03144