MONet - Multiple Output Network for Driver Assistance Systems Based on a Monocular Camera

Razvan Itu
Computer Science Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
razvan.itu@cs.utcluj.ro

Radu Danescu
Computer Science Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
radu.danescu@cs.utcluj.ro

Abstract—**Deep learning based image processing has become popular and approaches using convolutional neural networks (CNNs) have been widely used in recent years. In this paper we propose a multiple output convolutional neural network for road traffic scene understanding using a monocular camera. The color images are fed into the artificial neural network that produces multiple outputs. Our model performs three tasks: semantic segmentation, object detection and vanishing point computation. The semantic segmentation produces relevant data regarding the traffic scene, the obstacle detection module provides individual obstacles, whereas the vanishing point module will provide information that can be used to perform extrinsic camera calibration. We propose a novel obstacle detection approach and extend already published work by having a vanishing point detection module. The multiple outputs are predicted in a single-step and the information can be used as an initialization step for a 3D tracking system. Our network can extract individual dynamic objects and their correlation to the 3D space can be computed using the extrinsic parameters generated from the vanishing point module.**

Keywords—driver assistance, cnn, monocular, camera, semantic segmentation, obstacle detection, vanishing point, calibration

I. Introduction

Deep learning based image processing has become popular and approaches using convolutional neural networks (CNNs) have been widely used in recent years. The increase in processing power availability and new datasets has facilitated the development of CNNs that predict relevant information from road traffic images, such as vehicles or pedestrians, traffic signs, road and lane information and so on.

In this paper we propose a CNN model that is able to perform multiple tasks. This is achieved using a single encoder part and multiple decoders. The encoder is generally referred as "backbone" network, whereas the decoders are named "heads". The encoder part has the role of extracting the relevant features from images, whereas the decoders will generate the required outputs based on the features. We present a CNN architecture that is able to detect individual obstacles in the road scene, perform semantic segmentation and estimate the position of the vanishing point. The encoder features convolution layers. The semantic segmentation is composed out of upsampling layers that construct an image of the scene from the extracted features. The obstacle detection is implemented in a novel way by using the same structure as the semantic segmentation, instead of the traditional bounding box regression. The same idea is used also for the vanishing point computation. Predicting data related to vanishing point using a single backbone CNN is also a novel idea.

The neural network is trained using existing datasets and also by generating new data using our own algorithms that we previously published.

The artificial neural network presented in this paper can be used in a monocular camera based perception system, as an initialization step for a 3D tracker system: our system offers the dynamic and static obstacles from the sementic segmentation module, individual obstacle instances extracted from the detection module and their mapping into the 3D space can be performed using the extrinsic parameters computed from the vanishing point that is generated from the vanishing point module of the CNN.

II. Related work

Convolutional neural networks have been used for various tasks in recent years starting from image classification [1], object detection [2], semantic segmentation [3], [4], or even 3D data inference [5] or tracking [6]. Monocular depth estimation can also be achieved using CNNs and a survey is presented here [7]. A survey regarding deep learning based methods for autonomous driving is presented in [8].

The main difference of the semantic segmentation approaches compared to the instance segmentation ones, is that they are faster and they make use of the same extracted features. Instance segmentation models usually predict bounding boxes and then they try to apply the segmentation for each predicted box. Therefore they are mostly dependent on the quality of the bounding box prediction step, whereas the multiple output networks provide results that are completely independent (the segmentation does not interfere with the obstacle detector and vice-versa). Training multiple output networks proves to be a difficult challenge due to the fact that they make use of different loss functions that require different weights. Paper [9] proposes a multi-task neural network using images in the YUV color space. Our work is similar to MultiNet [10] and [11], with the main difference that we detect obstacles in a unique approach and we also predict information regarding the vanishing point, by using multiple databases, a different network structure and complexity. We also use small dimension images as input to reduce the prediction speed and facilitate a good performance on portable devices. We use the vanishing point to calibrate the monocular camera by computing the extrinsic parameters, more specifically the yaw and pitch rotation angles.

Multi-task artificial learning is used to improve computational efficiency. The main drawback is that it heavily relies on good training datasets that are usually harder to come by and multiple output networks are generally harder to debug. However, the main reason why these types of networks are becoming more popular is due to the scalability and extensibility nature, meaning that a multi-task network can be

978-1-7281-9080-8/20/$31.00 ©2020 IEEE

extended to predict other information such as: depth, optical flow or even tracking, by using the network's shared features.

III. Solution overview

The neural network proposed by us has a color image as input, whereas the output is composed out of an obstacle detection part, a segmentation part and a vanishing point detection part. We propose a novel structure of the multiple output network (MONet).

The input is based on an encoder part that has the role of extracting the relevant features from the input images. The network features three outputs based on the U-Net architecture. Each module is trained separately using the same loss function applied on different outputs. This approach is illustrated in figure 1.

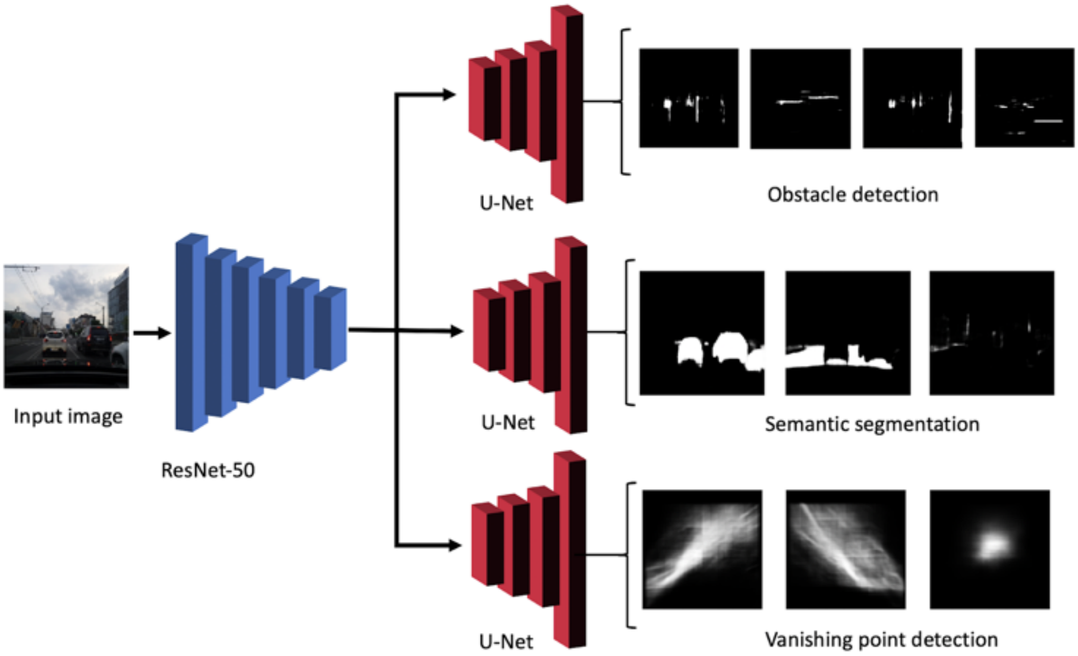

Fig. 1. The multiple output network architecture (MONet).

The obstacle detection part is based on a U-Net module that outputs the obstacle's 2D bounding box split into four layers (left, top, right and bottom). This original approach simplifies the network architecture by using the same generic layer structure for all detection tasks and also has the advantage of using direct connections with the layers from the decoder.

The main advantage of a solution based on multiple outputs is that the prediction for all the modules is done in a single step and the encoder part (the feature extraction) is shared between the modules. The model can be easily extended to infer other information.

A. *Feature extraction*

The first part of the network has the role of extracting the relevant information (features) from the input images and is based on the ResNet architecture [12], more specifically a modified version called ResNet-50. The ResNet is a popular approach that is widely used and won the ImageNet competition in 2015. This approach introduced the concept of the skip connections between the neural network layers that had a large impact on improving the performance of CNNs, especially those that feature a large number of layers.

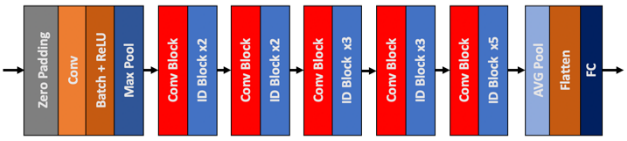

Fig. 2. The architecture of ResNet-50.

B. *Obstacle detection*

The CNN features an obstacle detection module that uses the same layers and structure as the semantic segmentation module. The main idea is to split the 2D bounding box of an obstacle into four parts in four different images: one contains the left delimiter line, the second image contains the top delimiter, the third features the right delimiter and the fourth has the bottom delimiter line. We make use of the existing 2D bounding box from the datasets and create the four independent images by drawing the lines with a thickness of 2 pixels, as illustrated in figure 3.

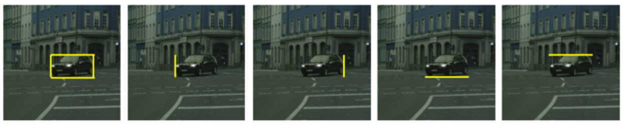

Fig. 3. Creating the training images for the obstacle detection module from the initial 2D bounding box. Image source: [15].

After the prediction from the convolutional neural network, the 2D bounding boxes can be reconstructed by combining the features from these four images as illustrated in figure 4.

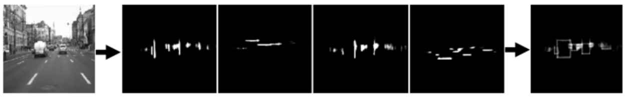

Fig. 4. Combining the obstacle prediction results (left, top, right and bottom) from the obstacle detection module using segmentation.

The individual obstacles are extracted by grouping the contours according to their relation to the objects body (figure 5).

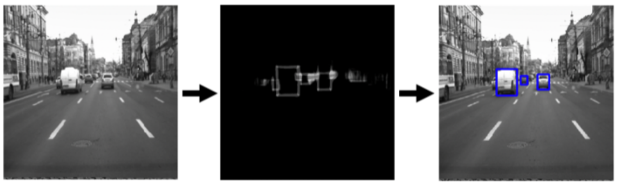

Fig. 5. Extracting the individual 2D bounding boxes from the CNN prediction.

An object is reconstructed by joining the left, top, right and bottom lines based on their proximity, which takes an additional 0.05ms to compute on average, after the prediction is done.

C. Semantic segmentation

The decoder part that constructs the semantic segmentation part is based on the U-Net approach [3], meaning that our proposed network features a central layer and three upsampling layers that are concatenated with their correspondent layers from the encoder (ResNet). The central layer features a convolution followed by a batch normalization, whereas the following three upsampling layers feature the following operations: upsampling, concatenation, zero padding and convolution followed by a batch normalization. The final layer from the semantic segmentation module features another convolution which will represent the segmentation map with the number of layers the same as the number of predicted classes.

Using the relevant features extracted using ResNet, the reconstruction layers of the U-Net will determine the output of the network: an image on 3 channels using the same dimension as the input, where each channel represents a different segmentation class. The 3 chosen classes are: road, dynamic objects and static objects. The first channel will represent the driveable road area, the second channel represents the moving (dynamic) objects such as: vehicles, buses, trucks, motorcycles, pedestrians, whereas the last channel (the static objects) represent the sidewalks and the lane delimiters (fences or barriers).

D. Vanishing point

The vanishing point can be computed by line intersections, or by analysing relevant features from images. We use our previously published work [13], based on computing the magnitute and orientation of the gradient, to compute three vote map images that can be used to extract the coordinates of the vanishing point. The first image represents the vote map of features from the left side of the input image, the second image contains the vote map of features from the right side of the image, whereas the third image represents the multiplication result of the previous two images (left and right). An example is presented in figure 6.

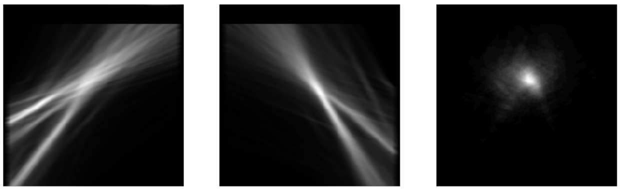

Fig. 6. Vanishing point relevant feature maps: the left side features, the right side features and the result of multiplying them.

The final coordinates in the input image of the vanishing point can be obtained either by a sliding window or by simply extracting the maximum value of the combined vote maps (the third image). We use the last approach and it takes an additional 0.09ms to compute.

The vanishing point is highly relevant because it can be used to compute the extrinsic parameters of the monocular camera. By knowing the focal distance and image size, the pitch and yaw rotation angles of the camera with respect to the world can be computed.

IV. Multi output CNN training

For training we have used 3 well-known datasets: CityScapes [14], Berkeley Deep Drive (BDD) [15] and Mapillary [16], due to the fact that they contain information regarding both segmentation and bounding boxes and their location in images. We prepared and processed the images in order to have them in the same input size and scale, meaning that we ended up using a total of 2975 images from CityScapes, 2759 images from Berkeley Deep Drive and 17109 images from Mapillary. The images were also filtered in order to keep those that feature a large number of road pixels. During training we used data augmentation techniques such as: random image translation and scaling and also random intensity and saturation adjustments in the HSV color space.

For the semantic segmentation we have used the binary cross entropy and Sorensen-Dice [17] loss function. The Sorensen-Dice loss function, also called Dice loss is a modified version of the Intersect over Union loss. The loss function used for the obstacle detection is the same.

The vanishing point module features the same loss function as the semantic segmentation (Sorensen-Dice loss). The training data for the vanishing point vote map feature images was generated using the approach described in [13].

The three loss functions are used simultaneously during training, each having a configurable weight. We trained with equal weights and using the same loss function for each module. We experimented with different weights for each loss function, and we ended up using the same for each.

V. Evaluation and results

The semantic segmentation part was evaluated on the validation set from the CityScapes dataset, which is composed of 500 images that were never before seen by the network during training. The result is similar to the one obtained in our previous published papers [18]: 0.906 IoU score for the road class. The score is smaller than the state of the art approaches due to the fact that we used smaller sized input images and a reduced number of classes during training. We have used 256 x 256 input images in order to favor a reduced prediction time.

The analysis of the prediction times for MONet on the validation set from CityScapes in presented in figure 7.

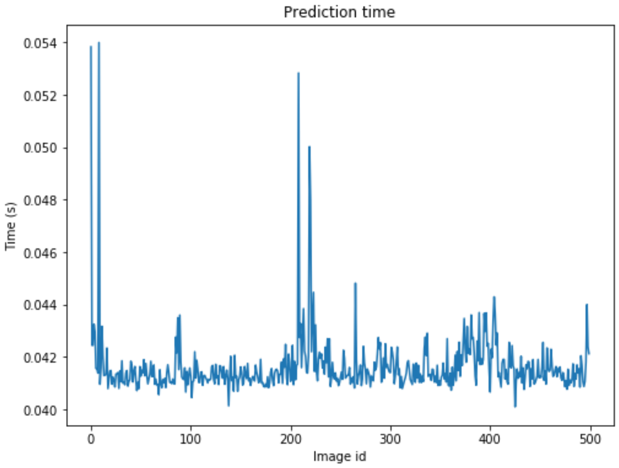

Fig. 7. Analysis of the prediction times (expressed in seconds) for the 500 images from the validation set from CityScapes dataset.

The prediction time analysis represents the number of seconds needed to make a complete prediction on a single image (meaning that it predicts all three outputs: semantic segmentation, obstacles and vanishing point). The average prediction time for the validation set of CityScapes was 41 ms, meaning that this approach represents a major advantage in terms of speed, especially when compared to other solutions that feature separate CNNs to segment images and detect obstacles, or solutions based on probabilistic algorithms to detect vehicles or obstacles in a scene. A faster execution time can be an advantage, even though the accuracy might not be so robust, especially if an approach like ours will be integrated with other tracking solutions. Extracting the vanishing point coordinates from the CNN output takes an additional 0.09ms on average (figure 8).

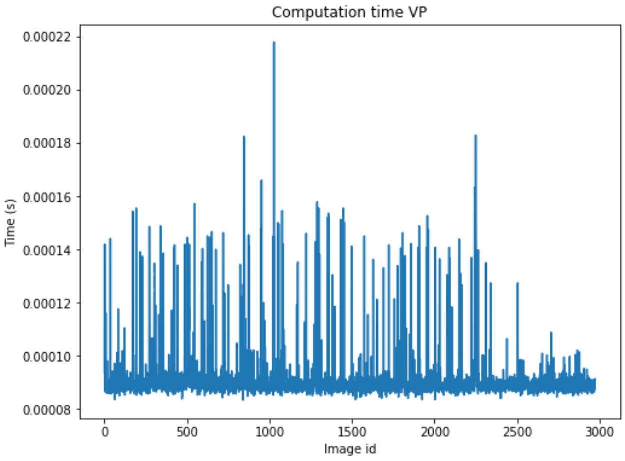

Fig. 8. Computing the coordinates of the vanishing point from the CNN output (the time is expressed in seconds), tested on 3000 images.

Qualitative results using our proposed network are illustrated in figure 9. The input images are completely new, unseen by the network during training (they are from our own dataset [18]).

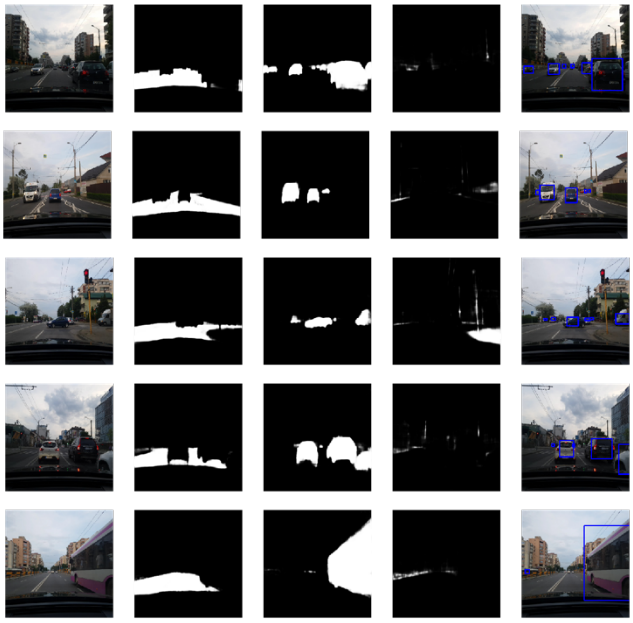

Fig. 9. Prediction results: the first column represents the input image, the second column is the driveable road area, the third column is the dynamic objects (vehicles, trucks, buses or pedestrians), the fourth represents the static objects (sidewalk, barriers or poles) and the last column represents the detected obstacles (bounding boxes).

We evaluated the detection module using one of our own datasets where the ground truth is considered the information from a stereovision camera setup. We obtained an IoU score of 0.74 on our own dataset featuring over 1000 images of various 3D objects extracted from a stereovision setup using particle filter tracking. We also performed an evaluation using the CityScapes validation dataset (500 images) and we obtained an IoU score of 0.80. We have also compared our approach with one using the same backbone (ResNet-50) but with YOLO [19] as output. This network was trained with the same datasets and by using the loss presented in the original YOLO paper.

The approach presented in this paper provides better results, as presented in table 1.

TABLE I. 2D BOUNDING BOX EVALUATION

	Our dataset (IoU)	**CityScapes (IoU)**
MONet with YOLO obstacle detection	0.64	0.63
MONet	0.74	0.80

Examples of 2D obstacle prediction results on our stereo-vision dataset are illustrated in figure 10. An additional video with results from the 2D box obtained from MONet is available at: https://vimeo.com/435661138.

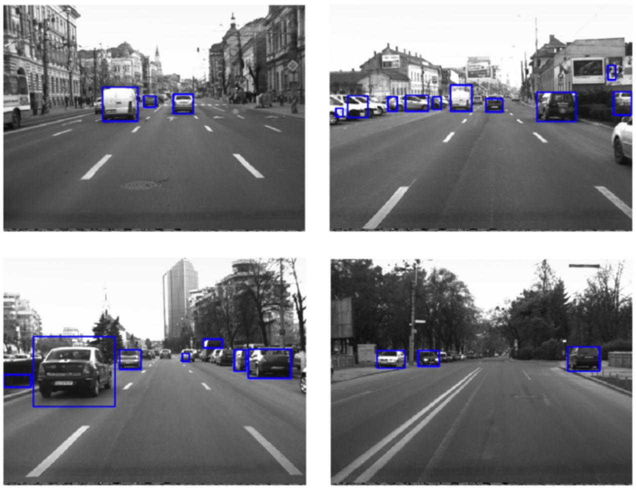

Fig. 10. 2D bounding box prediction on images from our own dataset.

Evaluating the vanishing point has been performed on a test set from CityScapes. We have also evaluated using our vanishing point dataset that is published at [20]. We have used the NormDist metric which is actually the RMSE (root mean squared error) divided by the input image diagonal. The evaluation results are presented in table 2.

TABLE II. VANISHING POINT EVALUATION

	CityScapes	**VP Highway**	**VP City**
NormDist (MONet)	0.016	**0.013**	**0.018**
NormDist [18]	-	0.050	0.026

Qualitative results of the vanishing point prediction are illustrated in figure 11.

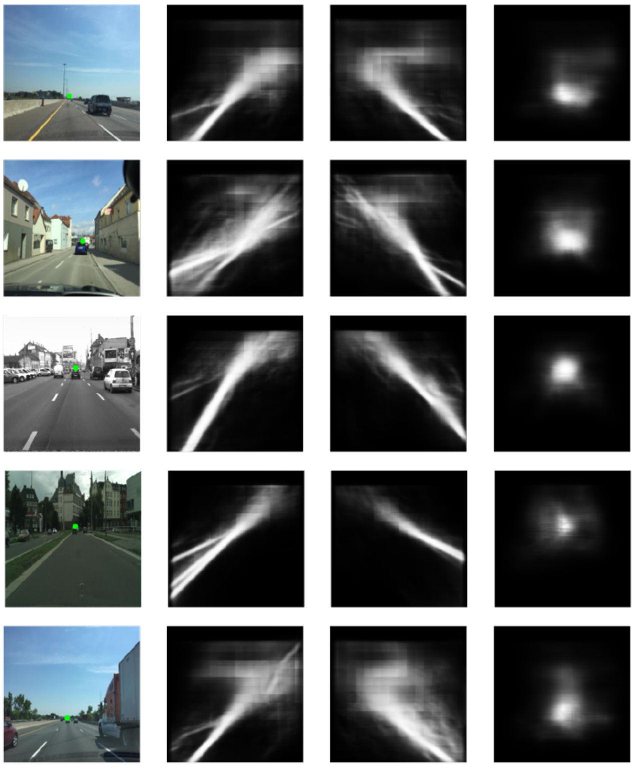

Fig. 11. Vanishing point prediction results (second, third and final column). The first image represents the input image with the extracted vanishing point colored in green. Images are taken from [14] and [20].

Our proposed CNN has a total median prediction time of 0.41ms and a total computational time of 0.55ms for an individual image. The total computation time contains the prediction time (0.41ms on average) and the time required to extract the VP (additional 0.09ms) and the individual bounding boxes (additional 0.05ms).

The semantic segmentation module is already integrated with our particle filter system that was previously presented in [18].

VI. CONCLUSION AND FUTURE WORK

In this paper we have presented a multi ouput network that performs three tasks relevant to the road traffic scene perception using a monocular camera. The CNN performs semantic segmentation, obstacle detection and vanishing point computation. Our solution favors execution speed by making multiple predictions in a single step. In addition to already published work, we presented a vanishing point detection module that can be used to calibrate the extrinsic camera parameters. We also presented a novel approach to detect obstacles based on semantic segmentation. The segmentation module that extracts the road, dynamic and static objects is currently integrated into our own road traffic processing system and in the future we plan to integrate also the 2D bounding box output in our framework. We plan to use the results presented here and fuse them our own tracking algorithm based on particle filters. The neural network can also be extended to perform other inference tasks related to road traffic scene perception such as depth or optical flow estimation.

ACKNOWLEDGMENT

This work was supported by a grant of Ministry of Research and Innovation, CNCS - UEFISCDI, project number PN-III-P1-1.1-TE-2016-0440, within PNCDI III.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, G.E. Hinton, “Imagenet classification with deep convolutional neural networks”, Advances in neural information processing systems, pp. 1097-1105, 2012.

[2] L. Wei, et. al., "SSD: Single Shot MultiBox Detector", European Conference on Computer Vision, pp. 21-37, 2016.

[3] O. Ronneberger, P. Fischer, T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer, Vol. 9351, pp. 234-241, 2015.

[4] A. Paszke, A. Chaurasia, S. Kim, E. Culurciello, "ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation", arXiv: 1606.02147, 2016.

[5] A. Mousavian, D. Anguelov, J. Flynn, J. Kosecka, “3D Bounding Box Estimation Using Deep Learning and Geometry”, Computer Vision and Pattern Recognition, pp. 5632-5640, 2017.

[6] H.-N. Hu, et. al, “Joint Monocular 3D Vehicle Detection and Tracking”, arXiv: 1811.10742, 2018.

[7] A. Amlaan Bhoi, “Monocular Depth Estimation: A Survey”, arXiv: 1901.09402, 2019.

[8] J. Ni, Y. et al., “A Survey on Theories and Applications for Self-Driving Cars Based on Deep Learning Methods”, Appl. Sci., Vol. 10, No. 2749, 2020.

[9] T. Boulay, “YUVMultiNet: Real-time YUV multi-task CNN for autonomous driving”, arXiv: 1904.05673, 2019.

[10] M. Teichmann, “MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving”, arXiv: 1612.07695, 2016.

[11] G. Sistu, I. Leang, S. Yogamani , “Real-time Joint Object Detection and Semantic Segmentation Network for Automated Driving”, arXiv: 1901.03912, 2019.

[12] K. He, X. Zhang, S. Ren, J. Sun, “Deep Residual Learning for Image Recognition”, Computer Vision and Pattern Recognition, pp. 770-778, 2016.

[13] R. Danescu, R. Itu, "Camera Calibration for CNN Based Generic Obstacle Detection", EPIA Conference on Artificial Intelligence, pp. 623-636, 2019.

[14] 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”, Computer Vision and Pattern Recognition, pp. 3213-3223, 2016.

[15] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, T. Darrell, “BDD100K: A Diverse Driving Video Database with Scalable Annotation Tooling”, arXiv: 1805.04687, 2018.

[16] G. Neuhold, T. Ollmann, S. R. Bulò, P. Kontschieder, "The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes", IEEE International Conference on Computer Vision, pp. 5000-5009, 2017.

[17] T. Sorensen, "A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons", Kongelige Danske Videnskabernes Selskab, Vol 5 (4), pp. 1-34, 1948.

[18] R. Itu, R. Danescu, "A Self-Calibrating Probabilistic Framework for 3D Environment Perception Using Monocular Vision", Sensors, Vol. 5, No. 20, 2020, Art. No. 1280.

[19] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection”, arXiv 1506.02640, 2016.

[20] R. Itu, D. Borza, R. Danescu, "Automatic extrinsic camera parameters calibration using Convolutional Neural Networks", 2017 IEEE 13th International Conference on Intelligent Computer Communication and Processing (ICCP 2017), pp. 273-278.