Object detection using part based semantic segmentation

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*—Monocular vision systems are increasingly popular in driving assistance applications as they are easy to set up and do not require precise calibration or synchronization. The downside of monocular vision is the lack of 3D information, which makes the task of identifying individual objects that are close together in the image space difficult. The lack of 3D information must be compensated by high accuracy classification of the image data. This paper proposes a novel way of detecting objects using fully convolutional neural networks followed by lightweight geometric based post processing. The fully convolutional neural network has four semantic segmentation outputs corresponding to quarters of individual objects. Therefore, each pixel of the input image will be classified as either belonging to a top left, a top right, a bottom left, or a bottom right region of a whole object. If the object is occluded and only a few of the four regions are visible, the component pixels will still be labeled correctly. Based on the multiple outputs of the neural network, the pixels are grouped into connected regions using a clustering algorithm aware of the relations between the object's quarters. The accuracy of individual obstacle instances is similar to the accuracy of the results obtained from instance segmentation networks, while the demand of resources and the number of trainable parameters is significantly reduced.**

Keywords—computer vision, convolutional neural network, object detection, semantic segmentation

I. Introduction

Artificial intelligence and neural network based processing have facilitated great improvements and developments in multiple fields, most notably in computer vision. Applications of convolutional neural networks (CNNs) refer to medical image processing, autonomous robots and vehicles, surveillance and so on. The work proposed in this paper is closely related to autonomous robotic platforms or vehicles, using deep learning to process images of the surrounding scene.

In this paper we propose a CNN model that detects objects in a novel way, by using a semantic segmentation approach, followed by a geometric based post-processing step. The artificial neural network proposed in this work will provide four semantic segmentation outputs corresponding to quarters of the individual objects from the input image, meaning that each image pixel will be classified as part of the top left, top right, bottom left or bottom right region of the object. We then apply light post-processing to group the quarters into individual objects, similarly to a clustering algorithm. Partly occluded objects will still have the pixels labeled correctly, even if the quarters are not fully visible.

The solution proposed by us works similarly to instance segmentation approaches, but with a reduced network complexity.

II. Related work

Recent object detection approaches make use of convolutional neural networks that are composed of two modules: a backbone network that is usually pre-trained and a head that provides the prediction of the objects' bounding boxes and their corresponding class. Object prediction can be implemented with CNNs using either a one-stage approach, or a two-stage approach. One stage object detectors perform a regression of the bounding boxes and classes of the objects. Two-stage approaches use a region proposal network to generate regions of interest from the input image (first stage), that are then used to regress the bounding boxes and classification (second stage). These networks will feature a better accuracy than one-stage approaches, but are usually slower.

Yolo [1] and SSD [2] are one-stage detection architectures that have been widely popular. They can be trained with different backbone networks and Yolo usually performs better with Darknet [3] backbone. These networks make multiple bounding box predictions in a single step and also compute confidence and perform obstacle classification. Mostly these networks perform in real time achieving a high frame rate.

Another popular approach is Mask R-CNN [4] that performs instance segmentation (two stage approach). This approach has the advantage of predicting the individual, segmented pixel regions of obstacle instances. The main downside of such an approach is the network architecture complexity and the specfic pre-labeled datasets that need to be used during training.

Obstacle detection can also be performed using traditional image processing methods, using a single camera [5] or a stereo-vision setup [6]. This task can also be achieved with the help of additional sensorial data, such as RADAR [7] or LIDAR [8], or even both [9]. Additional sensors increase the accuracy of the system, but they require special calibration and synchronization, therefore adding extra complexity to such a system. The next step after detection is tracking, where the problem of occlusion appears. Some obstacles become partly occluded. Therefore, partial detection is essential for accurate traffic scene perception.

In this paper, we propose a solution that combines a CNN architecture with elements from classic geometric based obstacle reconstruction and detection. In this way, we are able to obtain results which are comparable to the ones obtained from instance segmentation networks, without having the complexity and higher computational resource requirements of these networks.

978-1-6654-0976-6/21/$31.00 ©2021 IEEE

III. System Overview

The proposed system features a unique method of extracting the individual object instances, based on generic semantic segmentation networks. Instead of using the network to classify the image pixels into different classes such as obstacle, free space, or even specific obstacles (cars, pedestrians, etc.), the network will label the image pixels with object parts (quarters) identifiers. The labeled parts are then grouped into individual obstacles based on their proximity and relative position as parts of the whole object.

The difference between the part-based semantic segmentation and the generic type-based semantic segmentation is shown in figure 1.

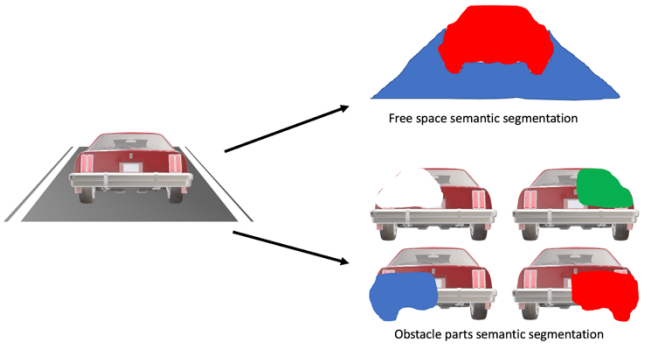


Fig. 1. The difference between free space segmentation vs. part-based semantic segmentation.

We propose a encoder-decoder convolutional neural network which uses as input a color image acquired on board of the vehicle, whereas the output consists of four channels encoding the binary status of the pixels as part of an individual object quarter (top left, top right, bottom left, bottom right). The algorithmic post-processing step consists of extracting the individual object instances as labeled regions of pixels and as rectangular bounding boxes, as illustrated in figure 2.

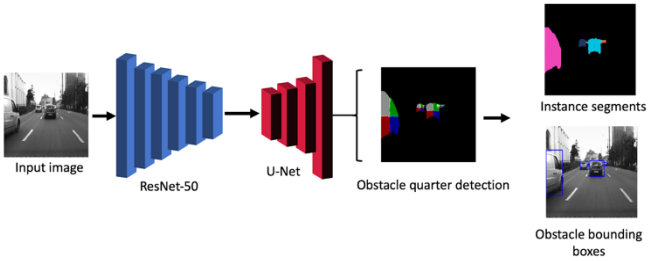


Fig. 2. System overview.

IV. Solution Description

A. Feature extraction

The encoder part of the neural network has the role of extracting relevant features from the input images that are fed into the decoder part. The encoder module is based on an existing CNN architecture: ResNet [10] that has won the ImageNet competition in the past and has proven its efficiency. In this paper we make use of a modified version called ResNet-50.

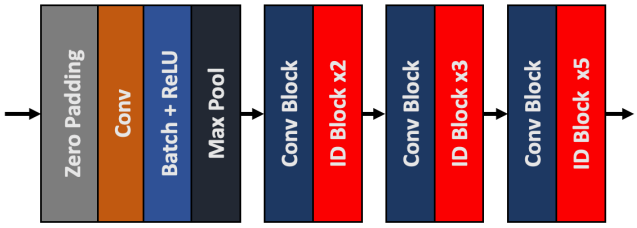


Fig. 3. The feature extractor based on ResNet-50.

B. Object quarter semantic segmentation

The structure of the U-Net based decoder is presented in figure 4. There are three concatenate operations with the corresponding layers from the encoder module. The final convolution operation will provide the network output.

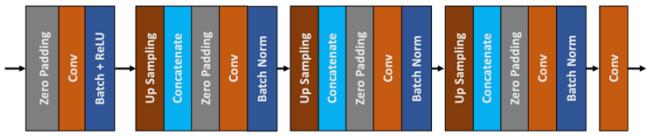


Fig. 4. The decoder module based on U-Net.

The output of the decoder will provide obstacles divided into four individual parts.

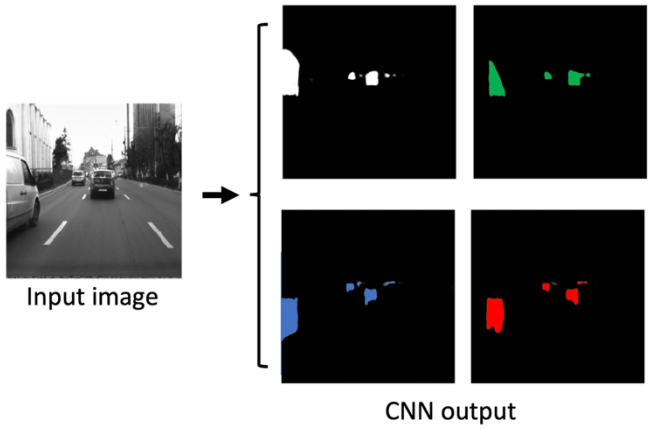


Fig. 5. The neural network outputs the obstacle quarters.

C. Object reconstruction

Based on the results of the neural network, the system must identify individual obstacles. The first step is to generate a 4-bit pixel encoding image, each bit corresponding to a type of quarter: bit 0 - top left quarter, bit 1 - top right, bit 2 - bottom left, and bit 3 - bottom right. Some pixels may belong to overlapping regions, and therefore they will have more than one bit set, as seen in figure 6.

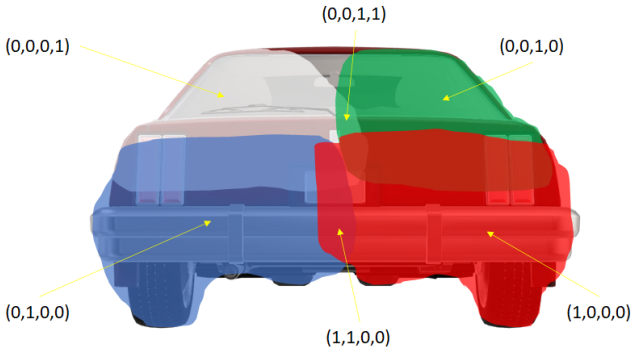


Fig. 6. Object pixel coding based on segmented quarters.

Based on the quarter coding, each pixel will have the value 0 if it is no obstacle point, or a value from 1 to 15, depending on the quarter overlap. We will generate 15 binary images, and label each one using connected components labeling. The labels from every set will be joined, and a region image will be generated, as seen in figure 7. The problem now becomes the problem of joining these regions into individual obstacles. A defining assumption is that each region will belong to only one object, because if adjacent objects are present in the scene their quarter codes will differ and will therefore split the regions. These regions are similar to the superpixels [11], but computed from semantic segmentation results.

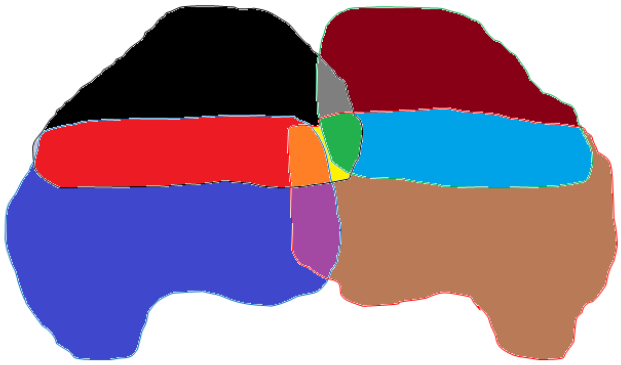


Fig. 7. Labeled regions based on connected pixels of the same quarter code.

The individual labeled regions will be clustered by assignment to rectangle hypotheses generated from quarter regions. Each quarter region can generate a complete object hypothesis, by expanding from the given region to the missing pieces of the object, using duplication of the region's bounding box along the horizontal and along the vertical axis. If an object is completely seen, it will have four quarters, and each quarter will generate a complete rectangle, as seen in figure 8. Each rectangle generated from each quarter of each object in the scene will receive a unique label.

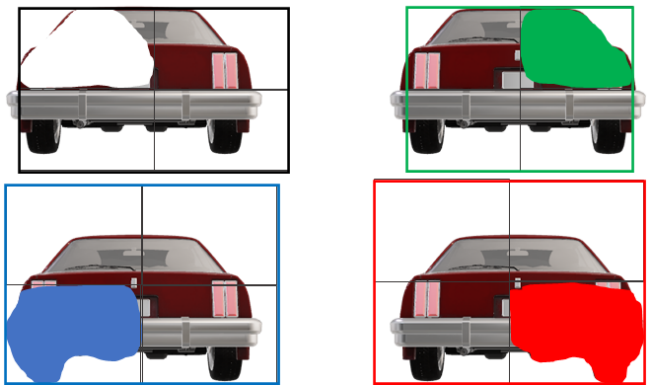


Fig. 8. Generation of rectangle hypotheses based on quarters.

Now we have, for each quarter region of each object, a rectangle which is a complete hypothesis of a bounding box of an obstacle. The hypotheses therefore outnumber the objects by a factor of almost 4 to 1. For this reason, the hypotheses will be merged based on an overlapping score and a pixel fitness score.

For each individual rectangle hypothesis *R*, a pixel score *S*(*R*) is computed as shown in figure 9. Each image pixel overlapped by the rectangle is checked to fit with the corresponding quarter of the rectangle: for example, if the quarter is top-left, the score is incremented if the image pixel has the top-left quarter bit set (Match). The score is then normalized with the rectangle area.

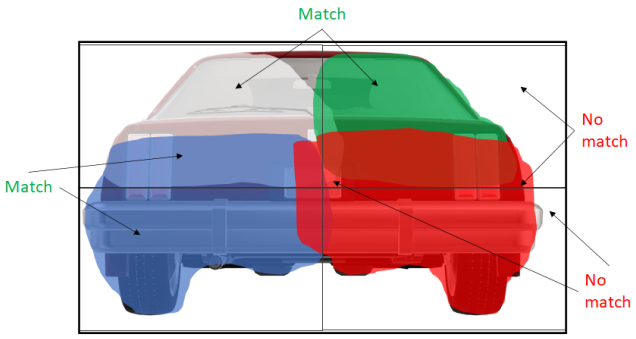


Fig. 9. Computing the rectangle score based on quarter matching.

The rectangles are subsequently re-labeled, based on their overlap with other rectangles, and based on their pixel score. Each rectangle Ri will be compared to every other rectangle Rj, and if they overlap significantly and S(Rj)>S(Ri), the rectangle Ri will be labeled with the label of Rj. The process is shown in figure 10.

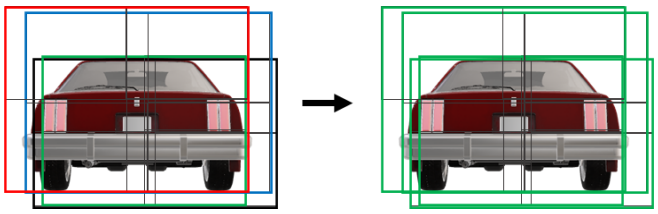


Fig. 10. Re-labeling the rectangle hypothesis by the best scored overlap.

The final step is to label each region shown in figure 7 with the label of the best scored rectangle overlapping the region. For each region, the overlapping pixels with every rectangle hypothesis are taken into consideration, and the rectangle score is added every time an overlapping pixel is found. The label of the highest scoring match is assigned to the region. If

the rectangle is re-labeled, as shown in figure 10, the final label is assigned to the region. The process is shown in figure 11, where we see two partially overlapping objects and their final label.

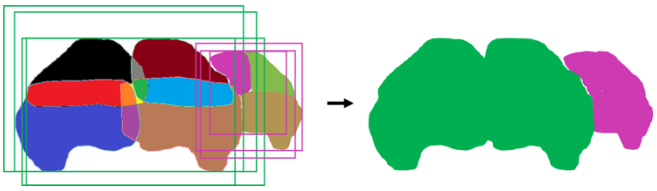


Fig. 11. Final result obtained by labeling the regions with the rectangle labels.

V. Training the Semantic Segmentation CNN

A. Automatic training data generation

For training we have used the following datasets: CityScapes [12], Berkeley Deep Drive [13] and KITTI [14]. These databases feature semantic segmentation examples and also obstacle detection examples, meaning that we can extract the four obstacle areas using the provided data. Each input image is split intro four different quarters using the existing bounding boxes from the datasets. Then, we mask each quarter with the data from the semantic segmentation maps in order to generate the top left, top right, bottom left and bottom right images of object instances.

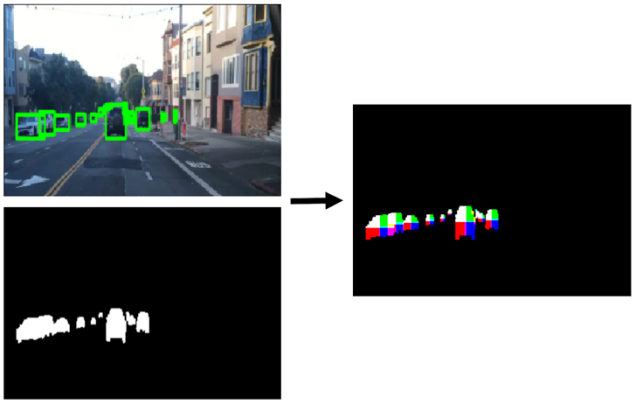


Fig. 12. Extracting the object quarters (right) from the dataset bounding boxes (top left) and semantic segmentation mask (bottom left).

B. CNN training

The neural network was trained on a system featuring 2 x 1080 Ti featuring 11 GB of memory, for a total of 500 epochs. We set the patience parameter to 50 epochs, meaning that if the training doesn't improve for 50 epochs, the process is stopped early (this usually happens after 200 epochs). Training one epoch takes around 50 seconds, meaning that we can obtain a fully working CNN model in less than 3 hours.

We perform data augmentation during training (random scaling, translation of the input data and color adjustments in the HSV color space). Training is performed using the Sorensen-Dice loss [15] and binary cross entropy.

VI. Results and Evaluation

Our proposed model was evaluated using well-known existing datasets and also the following dataset [6]. We compare our approach with the one proposed in paper [16]. The results are improved, especially on dataset [6] with images that were captured with camera systems that are different than the images included in the training datasets. We have also managed to train a Yolo V3 network on KITTI dataset and we compare with our proposed model. The results are presented in table 1.

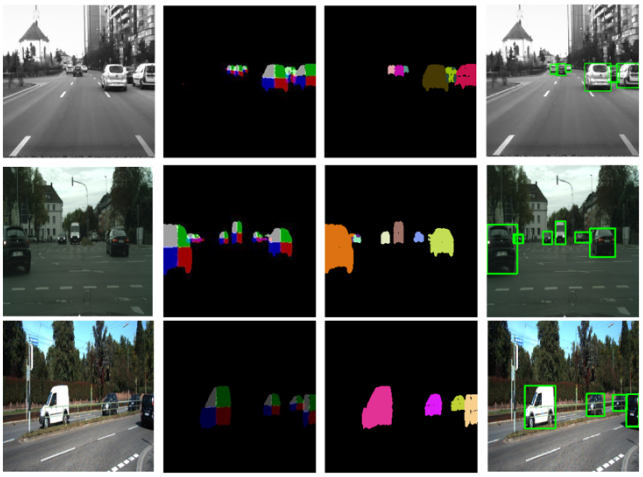
TABLE I. BOUNDING BOX EVALUATION

	Stereo dataset (IoU)	**Cityscapes dataset (IoU)**	**Kitti dataset (IoU)**
MONet [16]	0.74	0.80	0.55
Our system	**0.88**	**0.82**	0.70
DarkNet with Yolo V3	0.56	0.57	0.88

The Stereo dataset is composed of images acquired with a stereo-camera setup and we use only the left image from the setup. We then compared the results with the ground truth data obtained from a stereo tracking algorithm [17]. A stereo-vision system has a clear advantage in the case of partially overlapping objects, because they can be easily sorted out based on their depth. Still, this proposed monocular object detection solution obtains a high accuracy on this dataset.

Based on the evaluation on the CityScapes dataset, our solution has been proven highly accurate for detecting obstacles in complex city scenarios. For the KITTI dataset, the results are poorer due to the aspect ratio of the input images in the dataset, which impose to us two choices: we either resize the image as it is to our 256x256 pixels network input, or we crop the image and then resize. The first choice will severely deform the objects impacting the recognition process and the second choice will prevent possible objects from even being considered for detection.

Figure 13 illustrates some results of the proposed system in different scenarios and datasets.



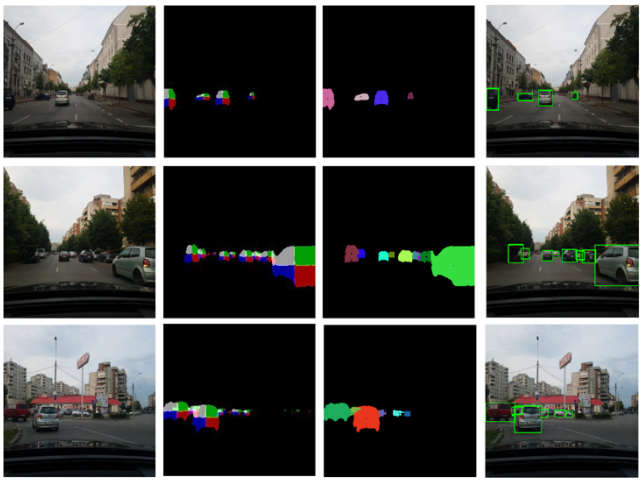


Fig. 13. Sample results: first column is the input image, the second column is the quarter semantic segmentation (CNN outputs merged), the third column shows the labels of individual instances, and the fourth column shows the extracted bounding boxes.

Mask R-CNN features 64 million parameters, whereas our model based on U-Net features 11 million parameters. The prediction time for Mask R-CNN is 0.27 seconds, compared to 0.043 seconds for our network. Labeling the images takes 0.014 seconds on average. Extracting the bounding boxes from the post-processed images takes an additional 0.0041 seconds on average, resulting a total average computational time of 0.0611 seconds for an individual frame. Testing was done on a desktop system using an Intel i7 CPU and equipped with two Nvidia 1080 Ti GPU boards that were used for training and prediction.

VII. Conclusion

The proposed system accurately detects object instances from road traffic sequences. We have obtained local geometric information using generic semantic segmentation networks and then we used this information for weak model based clustering of object parts. This way, our proposed system is able to successfully identify objects even if they are in close contact or partly occluded. The resulted system is lighter, faster, easier to train and has the accuracy comparable to networks that are much more complex.

The future work will be focused on the analysis of consecutive frames to improve the stability and performance of the segmentation and to track individual objects. The knowledge about the parts of the object will be very helpful when the object is tracked in the presence of occlusions.

Acknowledgment

The work was supported by a grant of Ministry of Research and Innovation, CNCS –UEFISCDI, project number PN-III-P4-ID-PCE2020-1700, within PNCDI III.

References

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

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

[3] J. Redmon, "Darknet: Open source neural networks in C", http://pjreddie.com/darknet/, 2013–2016.

[4] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN", 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980–2988.

[5] J. Zhou and B. Li, "Robust Ground Plane Detection with Normalized Homography in Monocular Sequences from a Robot Platform", 2006 International Conference on Image Processing, 2006, pp. 3017–3020.

[6] S. Nedevschi et al., "High accuracy stereo vision system for far distance obstacle detection", IEEE Intelligent Vehicles Symposium, 2004, pp. 292–297.

[7] W. Song, Y. Yang, M. Fu, F. Qiu, M. Wang, "Real-Time Obstacles Detection and Status Classification for Collision Warning in a Vehicle Active Safety System", in IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 3, pp. 758–773, 2018.

[8] W. Zhang, "LIDAR-based road and road-edge detection", 2010 IEEE Intelligent Vehicles Symposium, 2010, pp. 845–848.

[9] F. Homm, N. Kaempchen, J. Ota and D. Burschka, "Efficient occupancy grid computation on the GPU with lidar and radar for road boundary detection", 2010 IEEE Intelligent Vehicles Symposium, 2010, pp. 1006–1013.

[10] K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition", Computer Vision and Pattern Recognition, pp. 770–778, 2016.

[11] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua and S. Susstrunk, "SLIC Superpixels Compared to State-of-the-Art Syperpixel methods", IEEE Journal Transactions on Pattern Analysis and Machine Intellignece, vol. 34, no. 11, pp. 2274–2282, 2012.

[12] 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.

[13] 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.

[14] A. Geiger, P. Lenz and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite", 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3354–3361.

[15] 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, No. 4, pp. 1–34, 1948.

[16] R. Itu and R. Danescu, "MONet - Multiple Output Network for Driver Assistance Systems Based on a Monocular Camera", 2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP), 2020, pp. 325–330.

[17] R. Danescu, C. Pantilie, F. Oniga, S. Nedevschi, "Particle Grid Tracking System for Stereovision Based Obstacle Perception in Driving Environments", IEEE Intelligent Transportation Systems Magazine, Vol. 4, No. 1, 2012, pp. 6–20.