Modeling Traffic Scenes for Intelligent Vehicles using CNN-based Detection and Orientation Estimation

Carlos Guindel, David Martín, and José María Armingol

Intelligent Systems Laboratory (LSI) Research Group
Universidad Carlos III de Madrid, Leganés, Spain
{cguindel, dmgomez, armingol}@ing.uc3m.es

Abstract. Object identification in images taken from moving vehicles is still a complex task within the computer vision field due to the dynamism of the scenes and the poorly defined structures of the environment. This research proposes an efficient approach to perform recognition on images from a stereo camera, with the goal of gaining insight of traffic scenes in urban and road environments. We rely on a deep learning framework able to simultaneously identify a broad range of entities, such as vehicles, pedestrians or cyclists, with a frame rate compatible with the strong requirements of onboard automotive applications. The results demonstrate the capabilities of the perception system for a wide variety of situations, thus providing valuable information to understand the traffic scenario.

Keywords: Object detection · Viewpoint estimation · Deep learning · Intelligent vehicles

1 Introduction

Technology has adopted an increasingly important role in transportation systems over the past decades. Advanced Driver Assistance Systems (ADAS) have been introduced in an attempt to deal with the fact that wrong decision-making and distractions are the cause of a significant proportion of traffic accidents. These systems represent an increase in the degree of automation towards the future goal of fully autonomous driving, which is expected to lead to significant improvements in several issues associated with transportation systems.

While automated cars have been already successfully tested in urban environments [1], they are in most cases heavily dependent on off-line-built maps. Navigation with scarce or non-existent prior knowledge remains an open challenge due to the wide range of complex situations which is required to be handled (occluded landmarks, unexpected behaviors, etc.) within a highly dynamic, semi-structured environment.

Forthcoming self-driving systems will be demanded to understand complex traffic situations by themselves. This requirement relies on a robust inference of the position and motion of every traffic participant in the surrounding scene. Not

only the presence of obstacles must be accounted but also an accurate estimation of the class to which every obstacle belongs (i.e. car, cyclist, etc.) is essential in order to correctly understand and predict the traffic situation.

Vision-based approaches [2] have been proved to be highly cost-effective while enabling close-to-production assemblies given their compact size and ease of integration. Furthermore, video frames provide a rich data source from which additional information can be extracted.

In this paper, a vision-based approach enabling an enhanced onboard environment perception is introduced. It is targeted to the detection and localization of the different road participants present in the surroundings of a movable platform. Additionally, object detection is enriched through a viewpoint estimation, enabling high-level inference about short-term behaviors. A modern convolutional network-based framework is employed to perform the critical inference steps according to appearance features; later, stereo information allows introducing spatial reasoning into the system.

The remainder of this paper is organized as follows. In Section 2, we briefly discuss related works. Section 3 gives a general overview of the work. Section 4 describes the obstacle detection approach, while Section 5 introduces the scene modeling procedure. Results are reported in Section 6. Finally, Section 7 gives our conclusions about the work.

2 Related Work

Obstacle detection is an essential feature for automated driving systems. Consequently, a large number of algorithms have been historically developed to that end. Effort often focused on vehicle and pedestrian detection as these agents are the most commonly found ones in traffic scenes.

According to the sensing device in use, vision-based methods have traditionally fallen into two main categories: monocular-vision methods and stereo-vision methods. Stereo-vision provides depth information about the scene and thus is commonly used in driving applications [3].

Stereo-vision algorithms usually make some assumptions about the ground or the expected free space on it [4]. However, rich geometry information about the scene can be recovered, enabling the building of representations such as probabilistic occupancy maps [5], elevation maps [6] or full 3D models [7], where obstacles can be identified.

On the other hand, monocular obstacle detection is commonly based on appearance features. Selection of suitable features was traditionally the most crucial step in the performance of the detection pipeline and thus a lot of application-specific features, such as HOG-DPM [8], have been proposed to perform detection of traffic participants (e.g. cyclists [9]). Orientation estimation of the detected objects, while less frequent, has also been addressed [10].

Representation learning based on deep neural networks has delivered a paradigm shift in recent years, showing vast improvements over hand-crafted features in several kinds of recognition tasks. In particular, Convolutional Neural Networks

(CNN) can learn hierarchical data representations that have been shown useful in object classification [11].

Instead of using the classical sliding-window approach, detection with CNNs frequently relies on *attention mechanisms* to limit the number of proposals to be effectively classified. Within this tendency, Girshick et al. introduced the now widely known *recognition using regions* (R-CNN) paradigm [12]. These regions can be selected according to classical similarity-based segmentation methods; however, much effort has recently been devoted to end-to-end pipelines where every stage, including region proposal, can be effectively learned. Faster R-CNN [13] take advantage of a Region Proposal Network (RPN) which feeds the R-CNN responsible for the classification task.

CNNs have been applied in several tasks involved in autonomous driving, such as lane detection [14] and, certainly, object detection [15]. In some cases, orientation is also predicted to increase the information about the detected instances; thus, in [16], a CNN is used for object detection and viewpoint estimation. Viewpoint is also estimated in [17] through keypoint likelihood models.

3 System Overview

This work has been designed to become the core element of the perception module in the IVVI 2.0 (Intelligent Vehicle based on Visual Information) research platform [18]. IVVI 2.0 is a manned vehicle, equipped with cutting edge automotive sensors, which is meant for the development and testing of ADAS.

Visual sensing units in the IVVI 2.0 include a trinocular stereo camera covering the field of view in front of the vehicle, which is the source of the images used by the presented approach. The processing unit includes a high-performance GPU which enables high-parallel processing, such as that carried out in CNNs. Robot Operative System (ROS)[1] is used for inter-module cooperation.

The method presented here provides a step forward in vision-based detection and classification. The work consists of two main branches that are intended to run in parallel, as shown in Fig. 1:

1. Object detection and viewpoint estimation based on appearance. Features are extracted exclusively from the left stereo image.
2. Object localization, robust to changes in position and orientation of the vision system resulting from the vehicle movement. A stereo-based 3D reconstruction is performed, and the extrinsic camera parameters are extracted under a flat-ground assumption.

As usual in deep learning frameworks, our object detection approach is meant to be performed almost entirely in the GPU; on the other hand, the object localization pipeline is expected to make an intensive use of a typical multicore CPU during the step of 3D reconstruction. This twofold process has been designed to fill the available computing capability, in order to meet the time requirements inherent to the application.

[1] `http://www.ros.org/`

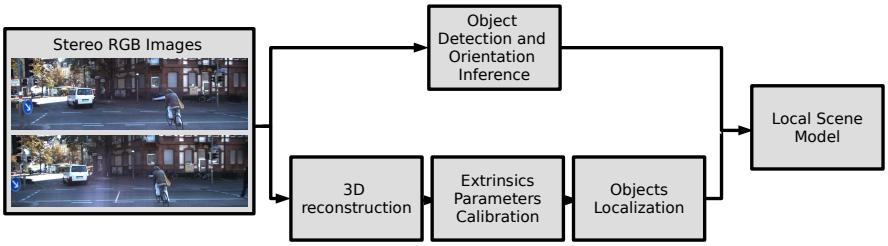


Fig. 1. Proposed system overview

4 Obstacle Detection

A wide variety of dynamic obstacles can be found in urban and road environments. Whereas object classification aims to classify predefined image regions, object detection also requires localizing every object within the image coordinates.

We adopt a state-of-the-art, CNN-based approach, Faster R-CNN [13], to perform object detection. Based on the popular R-CNN detector [12], Faster R-CNN provides an end-to-end trainable framework spanning from the image pixels to the final prediction. While outperforming classical detection pipelines, Faster R-CNN can deal with a large number of classes with no major impact on performance, making it particularly suitable for driving environments.

Faster R-CNN involves two different stages: a Region Proposal Network (RPN), which is responsible for identifying those image regions where objects are located, and an R-CNN, where image regions from the previous RPN are classified. Both components are based on CNN architectures and in fact, they share the same set of convolutional layers. For this reason, Faster R-CNN enables object detection at real-time frame rates.

We adopt the strategy introduced in [19] to incorporate the viewpoint inference into the detection framework. The basis of the idea is to benefit from the already computed convolutional features to obtain an estimation of the orientation of the objects with respect to the camera. Fig. 2 illustrates the approach. As with the region proposals from the RPN, viewpoint can be estimated at almost no cost during test-time given that convolutions are computed only once.

According to the requirements of the application, only the yaw angle (i.e. azimuth) from which objects are seen is to be estimated, since both relevant obstacles and the ego-vehicle are assumed to move on the same ground plane.

4.1 Discrete Viewpoint Approach

A discrete approach is adopted for the viewpoint estimation, so that the full range of possible viewpoints (2π rad) gets divided into N_b bins Θ_i; $i = 0, \ldots, N_b - 1$ of which only one is employed to represent the object viewpoint. Accordingly,

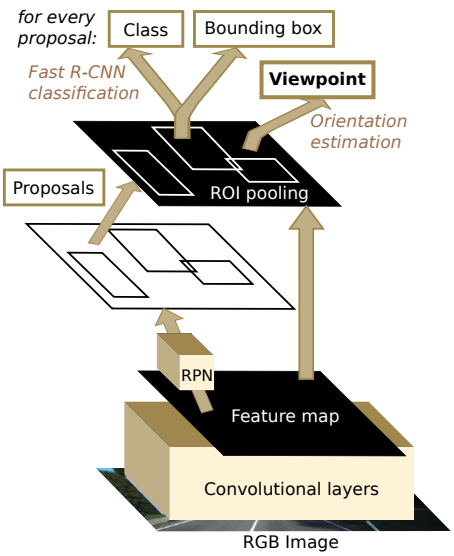


Fig. 2. Proposed object detection and viewpoint estimation approach.

objects with a ground-truth angle θ are assigned a viewpoint label i during the training step such that $\theta \in \Theta_i$:

$$\Theta_i = \left\{ \theta \in [0, 2\pi) \;\middle|\; \frac{2\pi}{N_b} \cdot i \leq \theta < \frac{2\pi}{N_b} \cdot (i+1) \right\} \tag{1}$$

The proposed object viewpoint estimation system is aimed to provide a viewpoint estimation consisting of the parameters of a categorical distribution over N_b possible viewpoints, $\boldsymbol{r}$. A single-valued $\hat{\theta}$ estimation can therefore be provided as the center of the bin b^* with the greatest probability according to $\boldsymbol{r}$:

$$\hat{\theta} = \frac{\pi(2b^* + 1)}{N_b} \tag{2}$$

4.2 Joint Detection and Viewpoint Estimation

In the R-CNN framework, image regions are propagated through the network and, finally, a fixed-length feature vector is extracted to predict the object class. We introduce the viewpoint estimation straightforwardly: it is inferred from the same feature vector that is also used to predict the class. This is motivated by the fact that appearance is highly affected by viewpoint, so a *good* set of features should be able to discriminate among different viewpoints.

Solutions introduced with Fast R-CNN [20] are adopted here, so the resulting feature vectors are fed into a sequence of fully connected layers that are finally split into three sibling layers. As in the original approach, the first two sibling layers provide a classification and a bounding box regression, respectively. On

the other hand, the new third layer is responsible for giving an estimation of the viewpoint, which is ultimately normalized through a softmax function. Given that classification is performed over K classes, the output of this branch is a vector $\boldsymbol{r}$ composed of $N_b \cdot K$ elements, representing K categorical distributions (one per class) over the N_b viewpoint bins:

$$\boldsymbol{r}^k = (r_0^k, \ldots, r_{N_b}^k) \text{ for } k = 0, \ldots, K \tag{3}$$

4.3 Training

According to the results reported in [13], we adopt an approximate joint training strategy, which has been shown to offer the best time-precision trade-off. Viewpoint estimation loss is introduced as a logistic loss that only adopts non-zero values for the ground-truth class; that is, from the N_bK-dimensional output $\boldsymbol{r}$, we only take into account the N_b elements belonging to the ground-truth class when computing the loss:

$$L_v = \frac{1}{N_B} \sum_{j \in B} L_{cls}(\boldsymbol{r}_j^{u^*}, b_j^*) \tag{4}$$

where N_B is the size of the batch B used to train the Fast R-CNN stage, and $L_{cls}(\boldsymbol{r}_j^{u^*}, b_j^*)$ is the multinomial logistic loss computed with the N_b elements from $\boldsymbol{r}$ corresponding to the ground-truth class u^* (i.e. the probability distribution of the angular bins for the ground-truth class) and the ground-truth label for the bin classification b^*.

This summation is added to the existing four components of the loss in the original Faster R-CNN framework, to get a five-component multi-task loss which is used for training the CNN. Although different weights might be applied to the components of the loss function, we let every (normalized) loss have the same contribution.

4.4 Implementation Details

As usual in classification tasks, convolutional layers are expected to be initialized from a model previously trained on the ImageNet classification dataset [21], while fully connected layers are given random values according to a Gaussian distribution. In this paper, eight evenly spaced viewpoint bins are considered for the viewpoint estimation ($N_b = 8$). Finally, a per-class non-maximum suppression (NMS) is applied to prune away the duplicated detections.

5 Scene Modeling

Object detection can be augmented with geometrical information in order to retrieve an instantaneous, local model of the traffic participants in front of the vehicle. To that end, we use the information from the two cameras in the stereo rig to build a dense 3D reconstruction of the environment.

Initially, the 3D point cloud of the scene is represented in camera coordinates. If the ground is assumed to be flat in a relatively small neighborhood from the vehicle, extrinsic parameters of the stereo system can be estimated in an online fashion. Accordingly, the effect of camera pose changes due to the vehicle movement (e.g. traveling on uneven road surfaces) can be properly removed.

Through this process, obstacles first detected through the object detection stage can be localized in *world* coordinates and assigned an absolute yaw angle.

5.1 Stereo 3D Reconstruction

We adopt a semi-global approach [22] to perform dense stereo matching. Despite this family of algorithms being more processing intensive than the traditional block-matching methods, challenges posed by road environments, e.g. lack of texture or changes of illumination, make them more suitable for the intended application. As an example, the disparity map obtained from the scene in Fig. 3a is shown in Fig. 3b.

As a result, a 3D point cloud is obtained (Fig. 3c). Then, a voxel grid down-sampling, with a grid size of 20 cm, is performed. In addition to reducing the amount of data to be processed, this filtering is aimed to normalize the point density along the depth axis.

5.2 Extrinsic Parameters Auto-Calibration

Coefficients defining the ground plane must be estimated as a first step to obtain the vision system extrinsic parameters. Two pass-through filters are applied in order to remove points outside a 0–2 m range along the vertical axis and a 0–20 m range along the depth axis. Within that ranges, flatness assumption is fulfilled with a high probability.

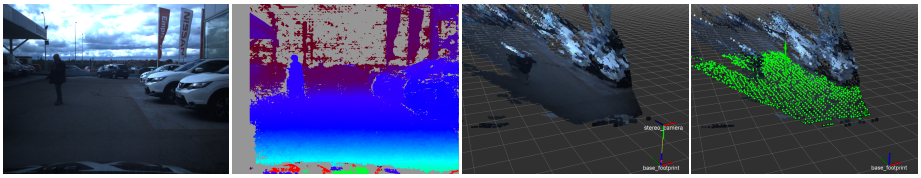
Points comprising the filtered point-cloud are then fitted to a plane using RANSAC [23] with a 10 cm threshold. Only planes perpendicular to a fixed direction, with a small angular tolerance, are considered. Since angles defining the camera pose are expected to be small, that axis is chosen as the vertical direction in camera coordinates. Fig. 3d illustrates the ground plane (shown in green) obtained from the voxel-filtered point cloud.

It can be shown [24] that, given a road plane defined by $ax_c+by_c+cz_c+d=0$, with (x_c, y_c, z_c) being the coordinates of a point belonging to the plane, roll (ψ), pitch (ϕ) and height (h) defining the camera pose can be obtained as:

$$\psi = \arcsin(a) \qquad \phi = \arctan\left(\frac{-c}{b}\right) \qquad h = d \tag{5}$$

Yaw angle cannot be extracted solely from the plane, and thus it is assumed to be nil. We choose not to translate the *world* coordinate frame along x and y camera axes, although that displacement may be arbitrarily chosen (e.g. the origin might be centered at the front end of the vehicle).

That set of extrinsic parameters defines a transformation which is then applied to the non-filtered point-cloud to get points in *world* coordinates.



(a) (b) (c) (d)

Fig. 3. Extrinsic parameters estimation pipeline: (a) left image; (b) disparity map; (c) point cloud; (d) inliers for the plane, in green, over the voxelized cloud (d).

5.3 Object Localization

To obtain the spatial location of the objects in the scene, the correspondence between points in the image and points in the 3D cloud, preserved within an organized point cloud structure, is exploited. Points belonging to the ground, as well as those too close to the camera (e.g. from the hood of the car), are removed beforehand. For every detection, the median values of the x, y and z coordinates for the set of 3D points corresponding to the 11 central rows of the object bounding box are computed and used as an estimation of the 3D location of the object. Yaw angle, expressed as the rotation around a local vertical axis, can be approximated taking into account the angle between the positive x axis of the world coordinate frame and the point given by the coordinates of the object.

By using all the inferred information about the obstacles, a top-view model of the vehicle surroundings is built, where every object in the field of view is included alongside their estimated orientation.

6 Results

Our joint object detection/viewpoint estimation pipeline was quantitatively evaluated according to the standard metrics on a well-established image benchmark, while the performance of the scene modeling stage and, eventually, the full system, have been tested in real traffic scenes using the IVVI 2.0 vehicle.

6.1 Object Detection and Viewpoint Estimation

Experiments for assess the object detection and viewpoint estimation branch have been conducted on the KITTI object detection benchmark [25], taking advantage of the available class and orientation labels. Since annotations for the testing set are not publicly available, the labeled training set has been divided into two splits, for train and validation, ensuring that images from the same sequence are not used in both subsets. Overall, 5,576 images are used in training whereas 2,065 are subsequently employed to test our algorithms.

Since our work focus on the simultaneous recognition of the different agents of the scene, our algorithm has been trained to detect the seven foreground classes

provided by the KITTI dataset. Special care was taken to avoid including regions overlapping with *DontCare* and *Misc* regions, neither as positive nor negative samples while training.

Given that our approach is independent of the particular architecture selected for the convolutional layers, we tested the two baseline architectures from Faster R-CNN in our application: ZF [26] and VGG16 [27]. On the other hand, even though all the models were obtained scaling the input images to 500 pixels in height during the training, different scales were evaluated at test-time. In all cases, training was carried out for 90k iterations, with a base learning rate of 0.0005, which was scaled by 0.1 every 30k iterations.

For the sake of brevity, we only evaluate Average Orientation Similarity (AOS), as introduced in [25], which is intended to assess the joint detection and viewpoint accuracy. Results for the different architecture/scale combinations are given in Table 1. Please note that results for the *Person sitting* and *Tram* classes are not reliable due to the low number of samples and were therefore excluded. Processing times are for our implementation using the Python interface of Caffe [28] and a NVIDIA Titan Xp GPU.

Table 1. Average orientation similarity (%) and run times (ms) on the test split for different scales and architectures

Net	Scale	Car	Pedest.	Cyclist	Van	Truck	mean	Time
ZF	375	44.2	35.6	16.1	8.5	3.2	21.5	46
	500	52.7	43.7	18.4	12.9	3.5	26.2	73
	625	51.6	40.7	22.7	15.1	5.3	27.1	90
VGG	375	64.8	54.7	25.0	22.9	8.5	35.2	79
	500	74.7	61.0	33.0	30.0	12.1	42.2	112
	625	75.7	60.9	35.2	31.1	15.4	43.7	144

As shown, precision does not grow significantly when the test-time scale is raised beyond the original train-time scale, i.e., 500 pixels. On the other hand, VGG16 considerably outperforms ZF for every analyzed class.

6.2 Scene Modeling

Tests for the scene modeling were performed using the IVVI 2.0 vehicle in real traffic situations. According to the results in the previous section, we chose the VGG16 architecture, and 500 as the image scale. Due to the generalization capability featured by CNN structures, models trained on the KITTI dataset were used without modifications. An ROI with 500 pixels in height, comprising the area where objects are typically present in the images, was extracted from the original 1024x768 images to be utilized by the CNN branch, while the full frame is employed to build the point cloud at the modeling branch.

Fig. 4 shows four examples of monocular detections (upper row) and their resulting scene models, where obstacles are represented as dots on a top view of

the reconstructed point cloud (lower row). Object orientation is represented by an arrow. Additionally, points belonging to the ground plane (RANSAC inliers) are projected on the image and colored green; they provide a rough estimation of the traversable area for the vehicle.

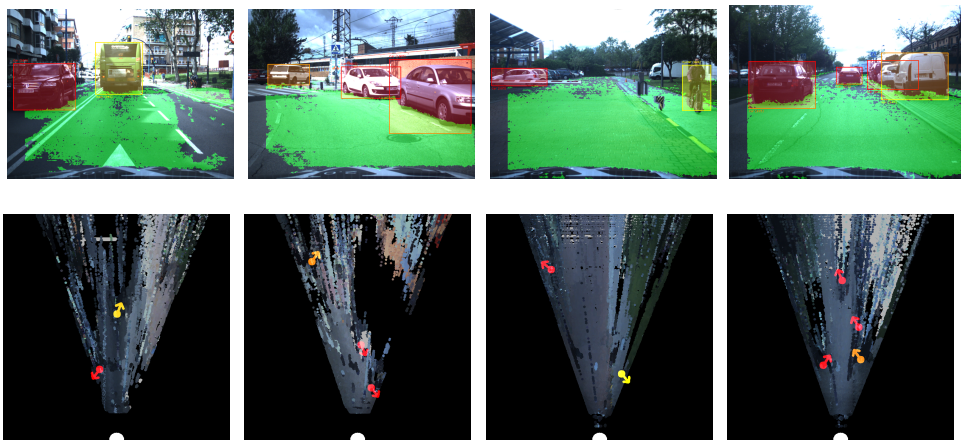


Fig. 4. Some examples of traffic scenes correctly identified by our system.

7 Conclusion

A computer vision framework designed to reach a full traffic scene understanding has been presented. Traffic participants are identified by a CNN-enabled method, showing the potential of this approach within automotive applications. Obstacle viewpoint estimation is introduced as an additional information source to endow the system with further insight into the scene features.

Because of the nature of the adopted approach, joint object detection and viewpoint estimation can be performed simultaneously over all classes. Since CNN parameters are shared across all categories and feature vectors computed by the CNN are low-dimensional, computation times are compliant with real-time requirements, yet achieving accurate results. This information can be further augmented using a stereo vision 3D reconstruction to gather an accurate situation assessment in complex traffic situations.

New categories of traffic elements, even those belonging to the infrastructure, may be subsequently added to enhance the scene understanding. The presented approach can be naturally extended to the time domain in order to make predictions about future behaviors of agents involved in the scene. In this regard, viewpoint estimation provided by the presented method plays a fundamental role to enable robust inference.

Additionally, the output provided by the system is suitable to be combined with information from other perception modules, e.g., semantic segmentation, to build an even further comprehensive model of the surroundings of the vehicle.

Acknowledges

Research supported by the Spanish Government through the CICYT projects (TRA2015-63708-R and TRA2016-78886-C3-1-R), and the Comunidad de Madrid through SEGVAUTO-TRIES (S2013/MIT-2713). The Titan Xp used for this research was donated by the NVIDIA Corporation.

References

1. Broggi, A., Cerri, P., Debattisti, S., Laghi, M.C., Medici, P., Panciroli, M., Prioletti, A.: PROUD-public road urban driverless test: architecture and results. In: Proc. IEEE Intelligent Vehicles Symposium (IV). (2014) 648–654
2. Zhu, H., Yuen, K.V., Mihaylova, L., Leung, H.: Overview of Environment Perception for Intelligent Vehicles. IEEE Transactions on Intelligent Transportation Systems (2017)
3. Franke, U., Pfeiffer, D., Rabe, C., Knoeppel, C., Enzweiler, M., Stein, F., Herrtwich, R.G.: Making Bertha See. In: IEEE International Conference on Computer Vision Workshops (ICCVW). (2013) 214–221
4. Musleh, B., de la Escalera, A., Armingol, J.M.: U-V disparity analysis in urban environments. In: Computer Aided Systems Theory - EUROCAST 2011. Springer Berlin Heidelberg (2012) 426–432
5. Badino, H., Franke, U., Mester, R.: Free space computation using stochastic occupancy grids and dynamic programming. In: IEEE International Conference on Computer Vision Workshops (ICCVW). (2007)
6. Oniga, F., Nedevschi, S.: Processing dense stereo data using elevation maps: Road surface, traffic isle, and obstacle detection. IEEE Transactions on Vehicular Technology **59**(3) (2010) 1172–1182
7. Broggi, A., Cattani, S., Patander, M., Sabbatelli, M., Zani, P.: A full-3D Voxel-based Dynamic Obstacle Detection for Urban Scenario using Stereo Vision. In: Proc. IEEE International Conference on Intelligent Transportation Systems (ITSC). (2013) 71–76
8. Felzenszwalb, P.F., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(9) (2010) 1627–1645
9. Tian, W., Lauer, M.: Fast Cyclist Detection by Cascaded Detector and Geometric Constraint. In: Proc. IEEE International Conference on Intelligent Transportation Systems (ITSC). (2015) 1286–1291
10. Pepik, B., Stark, M., Gehler, P., Schiele, B.: Teaching 3D geometry to deformable part models. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2012) 3362–3369
11. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks. In: Proc. Advances in Neural Information Processing Systems (NIPS). (2012) 1097–1105

12. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2014) 580–587
13. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(6) (2016) 1137–1149
14. Li, J., Mei, X., Prokhorov, D.: Deep Neural Network for Structural Prediction and Lane Detection in Traffic Scene. IEEE Transactions on Neural Networks and Learning Systems **28**(3) (2017) 690–703
15. Yang, F., Choi, W., Lin, Y.: Exploit All the Layers: Fast and Accurate CNN Object Detector with Scale Dependent Pooling and Cascaded Rejection Classifiers. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016) 2129–2137
16. Yang, L., Liu, J., Tang, X.: Object detection and viewpoint estimation with auto-masking neural network. In: Computer Vision - ECCV 2014. (2014) 441–455
17. Tulsiani, S., Malik, J.: Viewpoints and Keypoints. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2015) 1510–1519
18. Martín, D., García, F., Musleh, B., Olmeda, D., Peláez, G.A., Marín, P., Ponz, A., Rodríguez Garavito, C.H., Al-Kaff, A., de la Escalera, A., Armingol, J.M.: IVVI 2.0: An intelligent vehicle based on computational perception. Expert Systems with Applications **41**(17) (2014) 7927–7944
19. Guindel, C., Martin, D., Armingol, J.M.: Joint object detection and viewpoint estimation using CNN features. In: Proc. IEEE International Conference on Vehicular Electronics and Safety (ICVES). (2017) 145–150
20. Girshick, R.: Fast R-CNN. In: Proc. IEEE International Conference on Computer Vision (ICCV). (2015) 1440–1448
21. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision **115**(3) (2015) 211–252
22. Hirschmüller, H.: Stereo Processing by Semiglobal Matching and Mutual Information. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(2) (2008) 328–341
23. Fischler, M.A., Bolles, R.C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Communications of the ACM **24**(6) (1981) 381–395
24. de la Escalera, A., Izquierdo, E., Martín, D., Musleh, B., García, F., Armingol, J.M.: Stereo visual odometry in urban environments based on detecting ground features. Robotics and Autonomous Systems **80**(June) (2016) 1–10
25. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2012) 3354–3361
26. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Computer Vision - ECCV 2014. Springer International Publishing (2014) 818–833
27. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR **abs/1409.1** (2014)
28. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional Architecture for Fast Feature Embedding. In: Proc. ACM International Conference on Multimedia. (2014) 675–678