

# Vehicle Pose Estimation using G-Net: Multi-Class Localization and Depth Estimation

Javier García López<sup>a,b</sup>, Antonio Agudo<sup>b</sup> and Francesc Moreno-Noguer<sup>b</sup>

<sup>a</sup>*FICOSA ADAS SLU, Barcelona, Spain*

<sup>b</sup>*Institut de Robòtica i Informàtica Industrial, CSIC-UPC, 08028, Barcelona, Spain*

**Abstract.** In this paper we present a new network architecture, called G-Net, for 3D pose estimation on RGB images which is trained in a weakly supervised manner. We introduce a two step pipeline based on region-based Convolutional neural networks (CNNs) for feature localization, bounding box refinement based on non-maximum-suppression and depth estimation. The G-Net is able to estimate the depth from single monocular images with a self-tuned loss function. The combination of this predicted depth and the presented two-step localization allows the extraction of the 3D pose of the object. We show in experiments that our method achieves good results compared to other state-of-the-art approaches which are trained in a fully supervised manner.

**Keywords.** Pose estimation, Depth estimation, Vehicle detector, Deep learning.

## 1. Introduction

Estimating the 3D pose of rigid objects like vehicles has been a challenge for the last years, e.g., [1,2,3,4,5]. In order to be effective and accurate, though, these approaches normally require large amounts of training images annotated with the ground truth 3D pose and these training data are not always easy to obtain. In this work we contribute with an intuitive, straightforward and practical method of estimating the 3D pose of vehicles by using only 2D images and our own training pipeline consisting in multi-class detection with Region-based CNNs, bounding box refinement and depth prediction. By customizing the loss function of the depth estimator as it will be explained in section 3.2 we get to use only one simple RGB image as input to infer its depth map. Combining the predicted depth map with two phase localization, first vehicle and then wheel detection, we are able to extract the 3D pose of the detected vehicle, being this the main objective of our work.

In this work, we have considered the wheels as one of the most characterizing features of a vehicle for the pose extraction. There are other works such as [6] that are based on a wheel location for extracting the pose but, as the mentioned work obtained the wheels by image processing for ellipse finding and wheel recognition, we present a method based on deep learning that combines several known network architectures tuned for this specific approach.

## 2. Related Work

The recent advancement of region proposal methods have established new successful CNNs architectures. In this section, we review previous studies related to this paper. In particular, we review recent localization methods and 3D pose extractors.

### 2.1. FASTER R-CNN

The proposed method by Girshick *et al.* in [7] for object localization, so called region-based CNNs (R-CNNs) implied a big step forward into robust and accurate networks for object localization using Deep Learning due to its good results. It starts with a pre-processing region proposal network (RPN) that outputs the proposals in the training image to be the object to classify, secondly comes a final classification of each region with a category-specific linear support vector machines and with the results they fine tune the CNN end-to-end for detection. Each detection phase of the proposed pipeline in the G-Net is followed by a post-processing algorithm to refine the predicted bounding boxes [8].

### 2.2. Pose estimator with Deep Learning

There is a large literature on 3D pose extraction and many researchers have been published to solve this issue. One of the latest state-of-the-art methods is Deep Manta [1] which is based on first step of vehicle localization using a Region-Proposal-Network for extracting the first proposals of vehicle in input images, after two refinement steps in which the visible parts of each detected vehicle are localized and used in the last step, the inference. In this last part, the extracted information is combined with a dataset of the 3D models from the vehicles that appear in the training data for 2D-3D matching and therefore estimate the 3D model that corresponds to the detected 2D vehicle.

Many other approaches have been published and presented also good results in vehicle 3D pose extraction such as [9,10]. They have also shown accurate results using similar approaches as the Deep Manta, the first object proposal extraction through CNNs followed by pose extraction through an energy minimization approach that places object candidates in 3D using the fact that objects should be on the ground-plane.

In this work, we show good results of the presented network architecture, the G-Net, compared to other known methods. The main achievement of this work is the obtention of trustful 3D information of the vehicles (such as their pose) by designing a network that only needs planar images.

## 3. The G-Net

In this section we will detail the pipeline presented in this work. We have defined a two-step method based on planar RGB images with a first vehicle localization step followed by the second step consisting of a wheel extraction and depth estimation running in parallel. As it will be described in this section, this approach bases its implementation on a region-proposal-extraction for both vehicles and wheels and a second classification from proposed detected objects (similar to [7]) that allows a fast and accurate detection of the searched classes on the image. As defined in 3.2, we obtain depth from detected classes

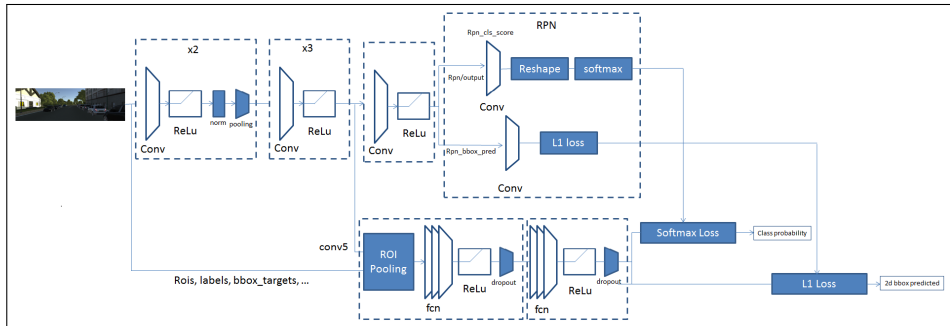
by following a pipeline formed by image general context extraction, image gradient calculation and final refinement network. The main contribution of our method is the obtention of 3D information following an heuristic and straightforward method with weakly supervised training that only needs RGB input images, their notations and depth ground truths. Having an easy trainable network that does not need multiple input resources, such as 3D render models of the detected objects or multiple keypoint labeling that could be difficult to generate, is one of the main issues we wanted to avoid with the presented G-Net.

### 3.1. Vehicle detection and wheel localization

Vehicle detection is a well-known topic in autonomous driving. There have been many studies with different approaches on how to handle this problem in the most appropriate way in terms of accuracy, resource consumption and execution time.

We base our so-called G-Net on a vehicle detection followed by a wheel localization phase on the cropped image of the vehicle, both based on a recurrent neural network [7, 11] with weakly supervised training [12], which performs first a region proposal and stores these proposed sections for each image. After this, a set of 7 convolutional layers with linear rectification phases and pooling treat the images so that the proposed regions or the candidates are classified as vehicles or wheels or are discarded.

The net architecture shown in Fig. 1 has been trained with vKITTI [13] and KITTI [14] in order to obtain a robust detector:

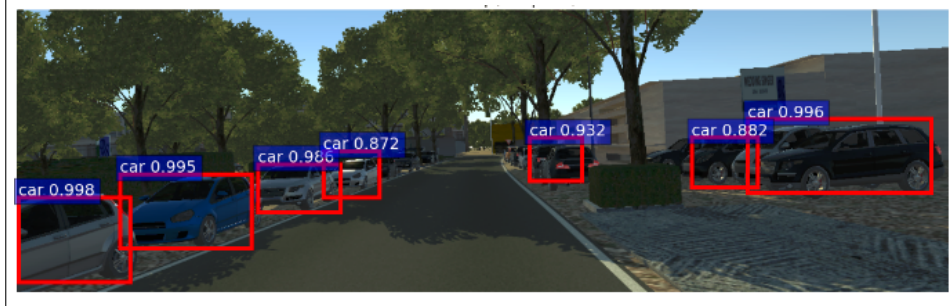


**Figure 1.** First step of G-Net: Detection and 2D BBox drawing. This architecture is formed by multiple convolutional layers followed by ReLU and poolings for the first RPN. Lastly there are two stages of fully convolutional layers with ReLU and dropouts which output will be fed into the L1-loss function to calculate the predicted bounding boxes of the detected features.

If the selection was successful two variables are predicted, the upper left corner and the bottom right corner of the two-dimensional bounding box that would contain the found vehicle.

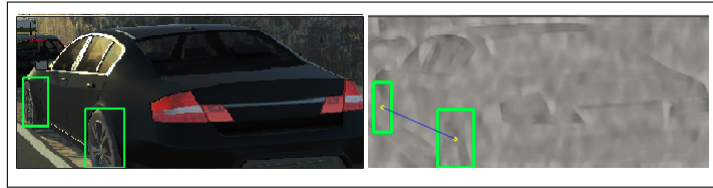
The last layer of our net architecture is a soft-max layer that calculates the error between the predicted bounding box and the ground truth, which is the same input image with bounding box annotation (Fig. 1 and Fig. 2).

The framework used to run train and validation stages through the data is *Caffe* [15] due to its simplicity to use and we validated our algorithm in two different datasets in order to check the effectiveness of this first step of G-Net: vKITTI and Synthia dataset [16].



**Figure 2.** Vehicle detection on vKITTI [13] dataset. Inference of first step of the presented G-Net on a validation image part of the vKITTI dataset.

Once the vehicles have been detected and an accurate bounding box has been created around them, we needed a detection of some points that would characterize the 3D pose of the vehicle, in our case, the wheels.



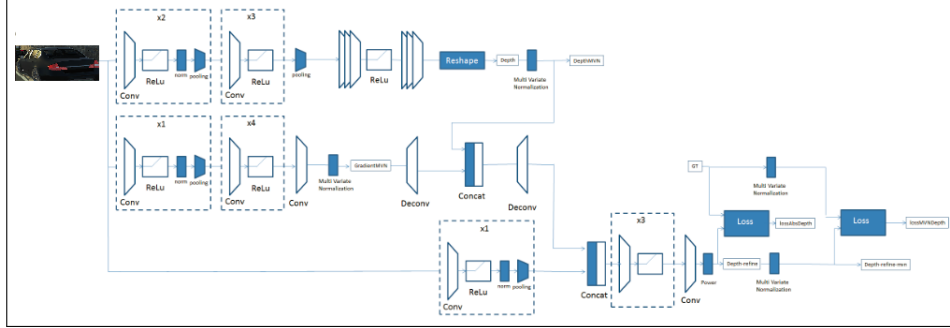
**Figure 3.** Wheels detected on the cropped image around detected vehicle (left) and its predicted depth map (right). This is the output of the second step of the presented G-Net consisting of parallel wheel localization and depth prediction from a single RGB image, as explained in 3.2.

In order to avoid inaccurate localization due to uncertainties in the detection process, the two detection phases of this work are followed by a non-maximum suppression [8] as post processing algorithm that avoids to have multiple detection of the same class and merges all detections that belong to the same object.

### 3.2. Depth estimation

In this last part of the method, we need to estimate the depth of the key-feature (wheels in our case) with respect to the camera so that we will be able to reconstruct the shape of the vehicle to estimate the pose. We train the architecture based on [17] as shown in Fig. 4 to extract the depth-map based on the steps defined in Fig. 5. In the proposed pipeline G-Net the depth prediction runs in parallel with the wheel detector as both have as input RGB cropped image around the detected vehicles.

The output of this part of the G-net will be the predicted depth for each pixel. By defining a normalized scale invariance loss function in the similar way like in [18] (see Eq. (1)) in which the mean value of the depth ground truth and the mean from estimated depth map are divided by their respective maximum values to obtain a scale invariant loss function that outputs similar loss for input images that represent same world structure but from different distance [18]. With this calculation of the loss function we produce the same loss even when the input RGB image is not identical but depicts the same real-world structure viewed from different distance.



**Figure 4.** Last step of G-Net: Depth estimation based on [17]. This architecture has a common part with the detection network (Figure 1) formed by several convolutional layers, ReLu and pooling. Its output is then fed into a two-step FCN layers for the general context extraction, into a convolution-deconvolution sequence for the gradient calculation or into the refinement network formed by the proposed loss function (equation 1).

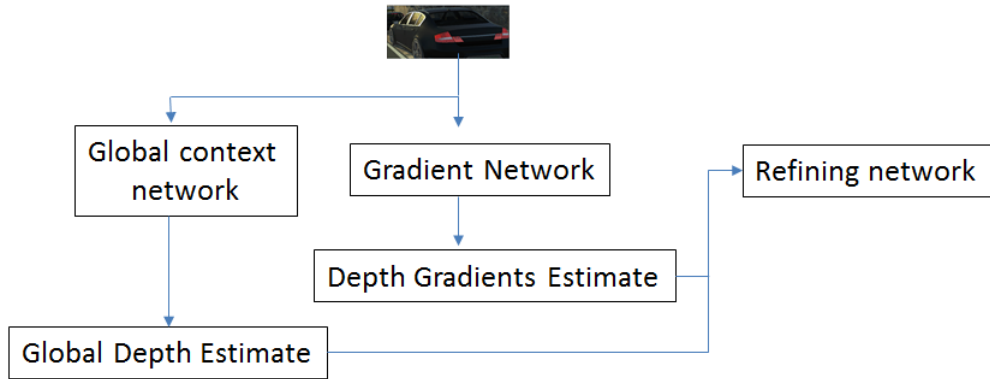
$$Loss(y, y^*) = \frac{1}{n} \sum_i \left( \frac{y_i - y_m}{\sqrt{y_v}} - \frac{y_i^* - y_m^*}{\sqrt{y_v^*}} \right), \quad (1)$$

where  $y$  is the predicted depth map,  $y_m$  is the mean of the predicted depth map,  $y^*$  and  $y_m^*$  are the depth map ground truth and its mean and lastly  $y_v$  and  $y_v^*$  are the variances of these depth maps [18].

This method for depth extraction consists of three steps:

1. General context network based on [17] that makes use of fully-connected layers with full field of view as entrance to estimate the scene's global context.
2. Gradient network to extract vertical and horizontal gradient of calculated depth maps.
3. Refinement network to obtain more accurate results as the ones provided by the general context network making use of the calculated image gradients.

Taking the extracted features from the previous part, we will then have for each image the predicted depth, measured in a camera coordinate system, such as we represented in Fig. 6.



**Figure 5.** High level architecture for Depth estimation [18]. As explained in section 3.2 this high level pipeline is formed by a parallel global context network and gradient extraction from input RGB image, followed by a final refining network [18].

#### 4. Pose extraction

In this paper we have presented a new method for 3D object pose estimation based on planar images (see Fig. 3) with region-based CNNs and weakly supervised learning. Following it will be presented how the pose from the vehicles can be extracted from the already predicted bounding boxes and depths explained in Section 3.

Once the net is working, for the extraction of the pose the intrinsic and extrinsic parameters of the camera will be needed, to obtain 3D points from the detected wheels based on mathematical operations done with the camera parameters:

$$x_{2D} = [K] * [R_t] * X_{3D} \quad (2)$$

where  $x_{2D}$  are the coordinates of the point in camera coordinate system (image coordinates) and  $X_{3D}$  are the coordinates in world coordinate system. The matrix  $K$  are the intrinsic parameters and  $[R_t]$  are the extrinsic of the camera.

As the training dataset is vKITTI [13], the intrinsic parameters of the camera are known (see Eq. (3)), and the extrinsic parameters are given per frame.

$$K = \begin{pmatrix} 725 & 0 & 620.5 \\ 0 & 725 & 187.0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (3)$$

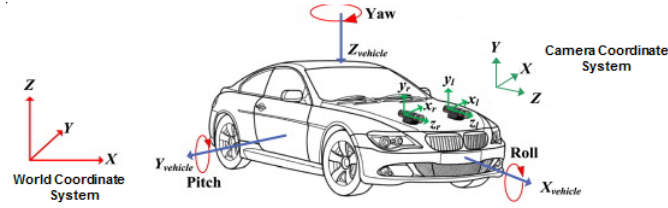
Having these, the 3D points of the detected wheels will then be calculated (see Eqs. (4) and (5)):

$$X_{3D} = K^{-1} * [R_t]^{-1} * x_{2D} \quad (4)$$

Once the wheels have been detected the coordinate system shall be rotated and translated from Pixel Coordinates to Camera Coordinate System (CCS), see Fig. 6. For that, the extrinsic parameters of the camera will be needed:

$$X_{CCS} = [Rt] * x_{PCS} \quad (5)$$

The Camera Coordinate System (CCS) is represented as shown in Figure 6:



**Figure 6.** Camera and vehicle coordinate system being the Z-axis of CCS the optic axis of the camera [19].

Making use of the inferred depth map, it will be estimated if the located wheels belong to the same side of the car or to the front/rear part of the vehicle. The order of the projection of the vector between wheels over the vehicle planes changes if the detected wheels belong to one left/right side or front/rear part. In the case that wheels from same side are detected, the vector defining the orientation of the vehicle will be calculated as follows:

$$V(x, y, z) = P1(x, y, z) - P2(x, y, z), \quad (6)$$

being P1 and P2 the center of two bounding boxes in CCS.

The yaw, pitch and roll angle to determine the 3D orientation of the vehicle will be calculated based on the projection of the predicted angle to the vehicle planes as follows:

$$yaw = \arccos(V_{xz} \cdot V_z / |V_{xz}| |V_z|), \quad (7)$$

$$pitch = \arccos(V_{yz} \cdot V_y / |V_{yz}| |V_y|), \quad (8)$$

$$roll = \arccos(V \cdot V_{xz} / |V| |V_{xz}|). \quad (9)$$

Being  $V_{xz}$  and  $V_x$ , the projection of  $V$  (see Eq. (6)) over  $XZ$ -plane and over  $X$ -plane respectively and  $V_{yz}$  and  $V_y$  the projection of  $V$  over  $YZ$  and then over  $Y$  respectively.

## 5. Experimental Results

In this section we will present some results of the described method and its comparison with other methods. As specified before in this paper, the used dataset for training and validation has been a set of 6300 planar images from KITTI [14] and vKITTI [13] datasets.

Method	Time (s)	Easy	Moderate	Hard
Deep Manta [1]	2	97.44	90.66	82.35
3DVP [20]	40	65.73	54.60	45.62
SubCNN [21]	2	83.41	74.42	58.53
3DOP [10]	3	91.45	81.63	72.97
DPM [22]	-	47.27	55.77	43.59
OC-DPM [23]	-	73.50	64.42	52.40
AOG [24]	-	43.81	38.21	31.53
Mono3D [9]	4.2	91.01	86.62	76.84
G-Net (Ours)	2.3	93.21	86.33	80.90

Table 1.: Comparison of average predicted pose of vehicles in KITTI dataset using different methods for 3D pose extraction. These results are presented as the percentage of well detected poses. Our approach has been applied to a subset of 3420 images in which a minimum of two wheels are visible. Our experiments show good and trustfull results although methods like Deep Manta [1] obtain better performance. This is due to the two phase detection-depth estimation pipeline. In the case of the Deep Manta, there is one first vehicle localization phase similar to ours, but in our case there is a second network to predict the depth at pixel level of the input image. This second step produces an increasment in the false positive rate that leads to an unaccurate pose estimation.

These results compare the calculated pose for each detected vehicle through the dataset with its correspondent labelled pose. Important to note is the definition of easy, moderate and hard in terms of percentage of the object occluded [14]:

- Easy: Min. bounding box height: 40 Px, Max. occlusion level: Fully visible.
- Moderate: Min. bounding box height: 25 Px. Partially visible.
- Hard: Min. bounding box height: 25 Px, Max. occlusion level: Difficult to see.

## 6. Conclusions

Here it is presented a straightforward method that makes use of other state-of-the-art techniques for localization and depth estimation for vehicle pose estimation. We show that obtaining a robust training and by using the proposed Loss Function for the treatment of input image data we obtain very acceptable performance results. Tuning these networks for this purpose allows us to infer the 3D pose of the detected vehicles with good results as seen in Table 1 and Table 2.



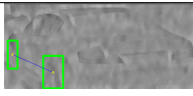
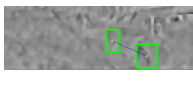

Image	Angles( $^{\circ}$ )	Accuracy(rad)
	yaw: 0.0045 pitch: 0.007 roll: -1.566	0.002 0.06 0.06
	yaw: 0.004 pitch: 0.006 roll: -1.566	0.002 0.06 0.0197
	yaw: 0.004700 pitch: 0.007 roll: -1.566	0.002 0.065 0.0198

Table 2.: Accuracy of our algorithm on vKITTI images [13] having roll, pitch and yaw calculated based on (6), (7) and (8). For simplicity reasons only three images have been taken to show the performance of the algorithm and the obtained accuracy. We show to get good results as the accuracy, calculated as the difference between the predicted angle and the labelled angle, remains close to zero (predicted and ground truth values shall be as similar as possible).

The presented method performs as good as many state-of-the-art methods (see Table 1) in terms of accuracy and execution time and presents an optimized usage of training data to avoid needing multiple training resources. The main contribution of this work is the implementation of a new pipeline based on region-based CNNs for detection and depth extraction using weakly supervised learning, labeling a subset of the training data, that only needs planar RGB images as input for extracting robustly 3D information from only 2D input resources.

### Acknowledgements

This work was supported by the Catalan Government inside the program "Doctorats Industrials" and by the company FICOSA ADAS S.L.U. J. García López is supported by the industrial doctorate of the AGAUR.

### References

- [1] Florian Chabot , Mohamed Chaouch , Jaonary Rabarisoa , Celine Teuliere, Thierry Chateau, "Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle," *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [2] Víctor Vaquero, Alberto Sanfeliu, Francesc Moreno-Noguer, "Deep lidar cnn to understand the dynamics of moving vehicles," *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.
- [3] Michael Hodlmoser, Branislav Micusik, Marc Pollefeys, Ming-Yu Liu and Martin Kampel, "Model-based vehicle pose estimation and tracking in videos using random forests," *International Conference on 3D Vision - 3DV*, 2013.
- [4] Víctor Vaquero, Ivan del Pino, Francesc Moreno-Noguer, Joan Sola, Alberto Sanfeliu and Juan Andrade-Cetto, "Deconvolutional networks for point-cloud vehicle detection and tracking in driving scenarios," *European Conference on Mobile Robots (ECMR)*, 2017.
- [5] Wenhao Ding, Shuaijun Li, Guilin Zhang, Xiangyu Lei, Huihuan Qian, Yangsheng Xu, "Vehicle pose and shape estimation through multiple monocular vision," *International Conference on Intelligent Robots and Systems (IROS)*, 2018.

- [6] Marcus Hutter and Nathan Brewer, "Matching 2-d ellipses to 3-d circles with application to vehicle pose identification," *International Conference Image and Vision Computing New Zealand (IVCNZ)*, 2009.
- [7] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems (NIPS)*, 2015.
- [8] Jan Hosang, Rodrigo Benenson, Bernt Schiele, "Learning non-maximum suppression," *Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [9] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3d object detection for autonomous driving," *Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [10] X. Chen, K. Kundu, Y. Zhu, A. Berneshawi, H. Ma, S. Fidler and R. Urtasun, "3d object proposals for accurate object class detection," *Advances in Neural Information Processing Systems (NIPS)*, 2015.
- [11] Ming Liang, Xiaolin Hu, "Recurrent convolutional neural network for object recognition," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [12] Thibaut Durand, Nicolas Thome, Matthieu Cord, "Weldon: Weakly supervised learning of deep convolutional neural networks," *Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [13] Adrien Gaidon, Qiao Wang, Yohann Cabon, Eleonora Vig, "Virtual world as proxy for multi-object tracking analysis," *Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [14] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research (IJRR)*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [15] Berkeley, "caffe." [Online]. Available: <http://caffe.berkeleyvision.org/>
- [16] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," *Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [17] David Eigen, Christian Puhrsch, Rob Fergus, "Depth map prediction from a single image using a multi-scale deep network," *Advances in Neural Information Processing Systems (NIPS)*, 2014.
- [18] Jan Ivanec, "Depth estimation by convolutional neural networks," Master's thesis, Brno University of technology, 2016.
- [19] M. S. Iftekhar, N. Saha, and Y. M. Jang, "Stereo-vision-based cooperative-vehicle positioning using occ and neural networks," *Optics Communications*, vol. 352, pp. 166 – 180, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0030401815003569>
- [20] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Data-driven 3d voxel patterns for object category recognition," *Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [21] Y. Xiang, W. Choi, Y. Lin, S. Savarese, "Subcategory-aware convolutional neural networks for object proposals and detection," *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017.
- [22] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, Sept 2010.
- [23] B. Pepik, M. Stark, P. Gehler, and B. Schiele, "Occlusion patterns for object class detection," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, June 2013, pp. 3286–3293.
- [24] B. Li, T. Wu, and S.-C. Zhu, "Integrating context and occlusion for car detection by hierarchical and-or model," *European Conference for Computer Vision (ECCV)*, 2014.