Joint Object Detection and Viewpoint Estimation using CNN features

Carlos Guindel, David Martín, José María Armingol
Intelligent Systems Laboratory
Universidad Carlos III de Madrid
Leganés, Madrid, Spain.
{cguindel, dmgomez, armingol}@ing.uc3m.es

***Abstract*—Environment perception is a critical enabler for automated driving systems since it allows a comprehensive understanding of traffic situations. We propose a method based on an end-to-end convolutional neural network that can reason simultaneously about the location of objects in the image and their orientations on the ground plane. The same set of convolutional layers is used for the different tasks involved, avoiding the repetition of computations over the same image. Experiments on the KITTI dataset show that our method achieves state-of-the-art performances for object detection and viewpoint estimation, and is particularly suitable for the understanding of traffic situations from on-board vision systems.**

I. Introduction

Next generation driver assistance systems and autonomous vehicles rely on trustable perception systems to provide a situational understanding of the surroundings of the vehicle. These systems must be capable of detecting obstacles that may interfere with the trajectory of the vehicle to avoid a possible collision, but they are also increasingly expected to be able to identify the type of agent involved in a hazardous situation. This particular feature will allow a more accurate prediction of their immediate future behavior and, thus, improve the chances of success of an avoidance maneuver. The primary beneficiaries would be the group of Vulnerable Road Users (VRU), such as pedestrians or cyclists, who are more severely affected by traffic accidents and could be treated with distinguished consideration in such situations.

Despite the widespread use of high-resolution laser rangefinder systems in obstacle detection applications, vision-based approaches are attracting research interest lately due to the emergence of a set of machine learning techniques encompassed under the name of *deep learning*. In particular, Convolutional Neural Networks (CNN) have proven to cope well with variations in poses, occlusions or lighting conditions [1], which are typically found in driving environments, thus becoming a powerful tool for object detection in the context of Intelligent Transportation Systems. Moreover, the use of image sensing devices is particularly attractive because they are usually more cost-effective than the alternative technologies and provide additional information for different driving-related applications, such as lane departure warning or traffic sign recognition systems.

In addition to the identification of the dynamic objects in the scene, additional comprehensive information may be extracted from the perception system in order to provide the decision makers with a complete situational awareness. The orientation of objects in the environment is one of the most significant sources of information that can be used to anticipate future events and react accordingly.

Conveniently, hierarchical data representations provided by convolutional features are suited for that purpose, given that appearance features can be used to discriminate among different viewpoints.

Based on the widely used Faster R-CNN framework [2], we propose an object detection approach especially suitable for urban environments, designed to identify the different road users present in front of the vehicle. In addition to studying the influence of the multiple parameters of the algorithm for this particular application, we introduce a new inference task into the existing paradigm, aimed to determine the orientation of the objects.

The rest of this paper is structured as follows. In Section II, we provide a brief review of related works with similar goals. In Section III, an overview of the inference system is presented. Sections IV and V are used to describe the details of the detection and orientation estimation functions, respectively. Experimental results are presented in Section VI, and the conclusions of the paper are drawn in Section VII.

II. Related work

For many years, on-board object detection research has been focused on the design of sophisticated features, specialized in the identification of a single type of agent; usually, vehicles or pedestrians. Histograms of gradients (HoG) [3] or Haar-like features [4] are some of the most frequently used in driving applications.

However, in the last five years, feature learning has become the dominant approach in object recognition. Convolutional Neural Networks (CNNs) have emerged as a method enabling rich hierarchies of features [5], which are impossible to build, in practice, using hand-crafted features. Since the first appearance of modern CNNs, its compelling performance has been widely proven in the most challenging recognition challenges, such as the ILSVRC competition [6].

Even though CNNs were first applied to object identification within a fixed-size input image, multiple approaches were soon proposed to integrate these structures into a complete object detection framework. One of the most popular ones nowadays is R-CNN [7], where the convolutional network is applied over previously defined ROIs within the input image. Fast R-CNN [8] introduced extensive improvements over the original implementation, but the detection accuracy, as well as the computation time, was still highly dependent on the algorithm used for the generation of proposals.

Therefore, considerable research effort is currently devoted to *attention* mechanisms generating these proposal windows [9]. While first approaches were based on classic segmentation techniques, most recent developments are geared towards applying the feature learning paradigm in an end-to-end fashion, spanning from the input image to the classification result [10]. In Faster R-CNN approach [2], convolutional layers are shared between both proposal generation and classification, thus speeding up the process while achieving comparable, or even better, detection performance.

Nonetheless, the fixed receptive field inherent to Faster R-CNN feature maps has been shown to be suboptimal when low-area object detections are required, such as in highway environments [11]. As a matter of fact, methods currently leading the KITTI object detection benchmark use indeed different approaches to overcome this limitation [12], [13].

On the other hand, object orientation estimation has frequently been identified in the literature as a primary cue when understanding traffic environments [14]. Nevertheless, the majority of monocular detection approaches aimed to on-board platforms are still limited to object localization within the image. Some notable exceptions are [15], which extends the Deformable Part Model (DPM) to handle different viewpoints with a 3D-aware loss function, and [16], where a detection scheme based on AdaBoost is introduced.

Current efforts in this field are focused on the use of convolutional features for object viewpoint estimation [17]. Our work falls into this category, although it is particularly tailored to automotive systems since considerable efforts have been made to comply with the real-time requirements expected in such applications without sacrificing accuracy.

III. System overview

A schematic view of the joint detection and viewpoint estimation system is presented in Fig. 1.

Using a single RGB image as an input, we aim to provide bounding boxes representing object detections in image coordinates, as well as a viewpoint estimation for each of these instances.

Convolutional features are computed and shared for use in the tasks of region proposal, classification, and orientation estimation, for efficiency reasons. Two well-differentiated structures are present in the architecture in order to handle these features and apply them to region proposal (RPN) and classification (Fast R-CNN). As will be introduced later, viewpoint inference is embedded into the latter in our approach.

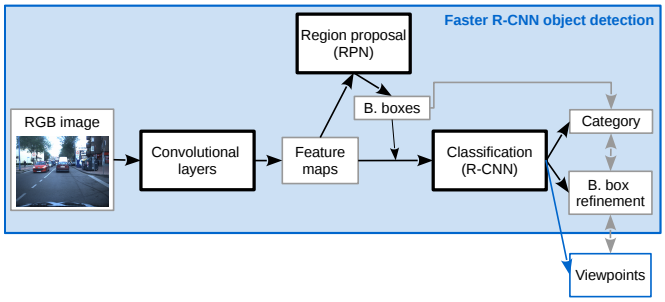


Fig. 1. Overview of the proposed approach for detection and viewpoint estimation.

IV. Object detection

Traffic environments are populated with a variety of agents which constitute potentially dangerous obstacles from the perspective of a moving vehicle. We aim for a robust object detection algorithm which should be able to not only classify every object into one of the multiple categories but also to localize every object within the image without significant prior constraints. It is, therefore, desirable to utilize a method which is not negatively affected by the increase in the number of classes.

A. Faster R-CNN framework

Given the constraints posed by the application, we rely on the popular Faster R-CNN approach for object detection. Being based on feature learning, this method outperforms classic detectors using hand-crafted features, but it is also able to carry out the detection stage under real-time constraints, with the number of objects (or classes) not being a major factor in the performance.

Faster R-CNN combines the two typical stages of the R-CNN detection methodology, i.e. proposal and classification, in the same end-to-end trainable pipeline. As previously stated, two different structures are still present in the architecture: a Region Proposal Network (RPN), which is responsible for selecting the potentially occupied image patches, and a classical R-CNN, where image regions from the previous stage are classified. Nevertheless, the main advantage of the method over similar approaches is given by the fact that both structures rely on the same convolutional features; in other words, they share the same set of convolutional layers, and thus the convolution filters only need to be applied once during inference.

B. Fine-tuning for traffic environments

In Faster R-CNN, RPN proposals are parametrized relative to some fixed reference boxes, called *anchors*. We have modified the anchors to fit better the objects in the environment. As a result of an analysis of the shapes of the objects in traffic scenarios, we use three scales with box areas of 80^2, 112^2 and 144^2 pixels and three aspect ratios (height/width) of 0.4, 0.8 and 2.5.

To mitigate the effects of highly unbalanced training sets (e.g. cyclists being much less frequent than cars), classification loss is computed as an “information gain” (infogain)

multinomial logistic loss, as opposed to the commonly used logistic loss. This way, different losses can be applied to the various categories during the training process, thus increasing the relative importance of the less common classes in the loss function. Details of the training process, including the global loss function, are provided in Section V-C.

V. Viewpoint estimation

Faster R-CNN accepts a color image as input and uses it to provide an output consisting of a set of bounding boxes where objects have been predicted to be found, as well as its classification into one of the available categories. For the purpose of improving the understanding of the surrounding traffic scene, we propose an additional task to be performed by the object detection pipeline: object viewpoint estimation.

We have found that the convolutional features used by the proposal and classification networks can be additionally exploited to that end. Using that particular philosophy, viewpoints can be estimated at almost no cost during test time, as was also the case with the RPN proposals, given that they make use of the already computed convolutional features.

A. Viewpoint inference problem

According to the requirements of the environment, viewpoint estimation is limited to the yaw angle from which objects are perceived. The potentially concerning obstacles and the ego-vehicle are assumed to move on the same ground plane, and thus the relative pitch and roll angles are accounted as negligible.

Viewpoint estimation methods can be divided into two groups: fine-grained pose estimators [18], able to infer arbitrary poses, or discrete pose estimators [19], which quantize the viewing sphere into a predefined number of bins and select the best one during inference. We adopt the discrete approach since it fits better into the Faster R-CNN design and has often been proven as adequate for high-level scene understanding [20].

In our approach, the full circle of possible viewpoints (2π radians) is divided into N_b bins. Each bin $\Theta_i; i = 0, \ldots, N_b - 1$ encompasses a range of viewpoints:

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

During training, objects with ground-truth label θ_{i_0} are assigned a viewpoint bin Θ_{i_0} such that $\theta_{i_0} \in \Theta_{i_0}$, as showed in Fig. 2 Similarly, viewpoint inference is designed to provide a bin $\hat{\Theta}$ representing the estimated pose for every object.

For the introduction of the viewpoint estimation into the Faster R-CNN framework, we pose the problem as the inference of the parameters of a categorical distribution over N_b possible outcomes. Thus the viewpoint estimation provides a prediction $r \in \Delta^{N_b - 1}$ with Δ^N being the N-simplex:

$$\Delta^N = \left\{ x \in \mathbb{R}^{N+1} \;\middle|\; \sum_{i=1}^{N+1} x_i = 1 \;\wedge\; \forall i \colon x_i \geq 0 \right\} \quad (2)$$

As a single angle $\hat{\theta}$ is expected to be generated for use in higher level applications, we take the center of the bin which has been given the maximum probability according to r; that is, Θ_{i^*} with $i^* = \arg\max_i (x_i)$:

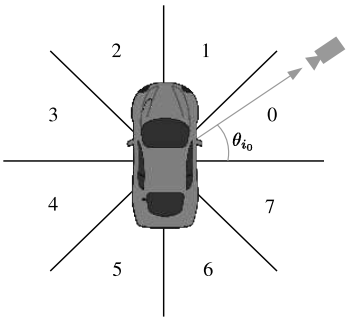


Fig. 2. Example of viewpoint quantization with $N_b = 8$ and $\Theta_{i_0} = 0$

$$\hat{\theta} = \frac{\pi(2i^* + 1)}{N_b} \quad (3)$$

B. Joint detection and viewpoint inference framework

Under the R-CNN framework, image patches are fed into the CNN to extract a fixed-length feature vector, which is then used in the class inference and also in the bounding box regression. Following this strategy, we use the final fixed-length feature vector for the additional task of viewpoint estimation. This approach is based on the intuition that appearance is heavily affected by the point of view, especially for the objects which are often present in traffic environments (i.e. vehicles and pedestrians). Therefore, features used for discriminating among different classes should also be able to distinguish among the full range of viewpoints.

In our approach using Faster R-CNN, image regions are previously selected by the region proposal stage in the RPN. This structure is responsible for assigning a binary class label $a \in \{0, 1\}$ and a bounding box refinement, expressed by a vector representing the coordinates of the predicted bounding box $b = (b_x, b_y, b_w, b_h)$, to each predefined anchor. These coordinates are relative to the anchor box itself.

Regarding the classification stage, we adopt the improvements introduced in Fast R-CNN [8]. Therefore, the resulting feature vectors (one for each proposal) are introduced into a sequence of fully connected layers which are finally divided into 3 sibling layers (instead of 2 as usual), responsible for the different inference tasks:

- Class. This layer applies the softmax function to get the categorical distribution p that describes the probabilities for the K available classes (and an additional background class):
$$p = (p_0, \ldots, p_K) \quad (4)$$
- Bounding box refinement. The second layer performs a bounding box regression to provide an output with four real values per class, representing the offset to be applied to the bounding boxes in their of x and y coordinates and their width (w) and height (h) dimensions:
$$t^k = (t_x^k, t_y^k, t_w^k, t_h^k) \text{ for } k = 0, \ldots, K \quad (5)$$
- Viewpoint. We add a third layer for the estimation of the viewpoint, which is also obtained through a softmax

function and given as a $N_b \cdot K$ output representing K categorical distributions over the N_b viewpoint bins:

$$r^k = (r^k_0, \ldots, r^k_{N_b}) \text{ for } k = 0, \ldots, K \tag{6}$$

C. Loss function and training

Among the three different strategies proposed in [2] for training networks with features shared, we adopt the approximate joint training strategy, which has been shown to offer an optimal trade-off between accuracy and training time [21].

Viewpoint is introduced into the loss function as a logistic loss that only adopts non-zero values for foreground classes. From the N_bK-dimensional output given by the viewpoint layer, r, we only consider the N_b elements belonging to the ground-truth class during training.

Therefore, region proposal and classification stages are trained simultaneously with a multi-task loss L with five components:

$$\begin{aligned} L = &\frac{1}{N_{B_1}} \sum_{j \in B_1} L_{cls}(a_j, u_j) + \frac{1}{N_a} \sum_{j \in B_1} u_j L_{loc}(b_j, b^*_j) + \\ &\frac{1}{N_{B_2}} \sum_{i \in B_2} L_{inf}(p_i, v_i) + \sum_{i \in B_2} [u \geq 1] \, L_{loc}(t^v_i, t^{v*}_i) + \\ &\frac{1}{N_{B_2}} \sum_{i \in B_2} [u \geq 1] \, L_{cls}(r^u_i, \Theta_i) \end{aligned} \tag{7}$$

In the training process, each Stochastic Gradient Descent (SGD) step is performed over two mini-batches randomly sampled from an image, B_1 and B_2. B_1 is composed of a predefined number of predefined anchors employed during the RPN training, while B_2 is made of a set of labeled regions of interest and used to train the R-CNN classification stage. The proportion of foreground samples in every mini-batch is controlled. As previously stated, network outputs are p_i, t_i and r_i, defined for each image region i in B_2, using the proposals given by a_j and b_j, which are defined for each anchor j in B_1. In Eq. 7, u_j is the ground-truth class (foreground/background) for the anchor j, v_i the true class of the region i and Θ_i the ground-truth bin representing the orientation of the object. Ground-truth values for the bounding box coordinates are indicated with a '$*$' superindex.

On the other hand, L_{cls} are logistic losses, L_{loc} are smooth-L1 losses, as introduced in [8], and L_{inf} is the infogain multinomial logistic loss:

$$L_{inf}(p_i, v_i) = \sum_{k=1}^{K} H_{v_i,k} \log(p_{i,k}) \tag{8}$$

with $H_{v_i,k}$ being the element (v_i, k) of the infogain matrix H and $p_{i,k}$ the predicted probability of sample i belonging to the class k. We choose H to be a diagonal matrix, and its elements (i.e. $H_{v_i,k}$ with $v_i = k$) are selected according to the expected proportions of the different classes in the real-traffic environment, such that less frequent classes are assigned lower $H_{v_i,k}$ values.

Finally, per-element losses are aggregated and normalized by the size of their respective mini-batches N_{B_1} and N_{B_2}, and the total number of anchors within the limits of the image, N_a. Iverson bracket indicator function $[u \geq 1]$ is used to exclude background examples ($u = 0$) in bounding box refinement and viewpoint estimation.

Although different weights might be assigned to the five components of the loss function to control the balance between them, we let every loss have the same contribution.

VI. Experimental Results

We evaluate our approach on the challenging KITTI object detection dataset [22], which is composed of images captured in real traffic environments and profusely annotated. Nine different categories, representing typical road agents, are identified in the dataset. Regions with the *DontCare* label, which is assigned to distant or unclear objects, and the *Misc* label, applied to objects not fitting the rest of categories, are not used. To that end, we select image regions during training so that IoU (Intersection-over-Union) overlap with that non-valid regions is limited to 15% (for the proposals) and 25% (for classification). As we want to test the performance of the method in complex environments with a broad diversity of classes, the remaining seven classes are used in training, even though, as usual, evaluation is limited to the most populated categories, i.e. *Car*, *Pedestrian* and *Cyclist*. The KITTI training dataset, whose annotations are publicly available, was divided into two splits for training (5,415 images) and validation (2,065 images), ensuring that images from the same sequence are not present in both training and validation sets.

Following the KITTI setup, we use the Average Precision (AP) metric for the object detection task and Average Orientation Similarity (AOS) for assessing the performance of joint object detection and viewpoint estimation. IoU overlappings of 70% for *Car*, and 50% for *Pedestrian* and *Cyclist* over ground-truth bounding boxes are required for the detections.

A. Training parameters

Although the proposed method is agnostic to the network architecture, we use the VGG16 architecture from [23] to perform the evaluation. As is standard practice, we use an ImageNet pre-trained model to initialize the weights in the convolutional layers.

The selection of the image scale has been identified as the parameter with the greatest influence on the final performance, with larger scales improving the accuracy. We resize the images by a factor of approximately 1.33 (to a fixed height of 500 pixels) to keep detection times tractable. Training is performed for 50k iteration with a learning rate of 0.001, then for 50k iterations with 0.0001 and finally for another 50k iterations with 10^{-5}. The remaining parameters are selected following the baseline Faster R-CNN tuning for PASCAL VOC, including the number of RPN proposals (up to 300).

On the other hand, eight viewpoint bins are considered in the viewpoint inference process (N_b = 8), so the resolution of the orientation estimation is $\pi/4$ rads.

Finally, infogain matrix values are selected according to the frequencies observed for the different categories in the training set, using to the following equation:

$$H_{k,k} = 2 \cdot (f_{min}/f_k)^{1/8} \tag{9}$$

where f_{min} is the number of occurrences of the less frequent class and f_k the number of instances of class k.

Fig. 3 illustrates the contribution of the different components in the multi-task loss during training. A moving average of 20 iterations is applied. It can be observed that the weight of the viewpoint loss is predominant during the first iterations, but it converges quickly to fall under the classification and bounding box regression losses, which further proves the effectiveness of the proposed loss function.

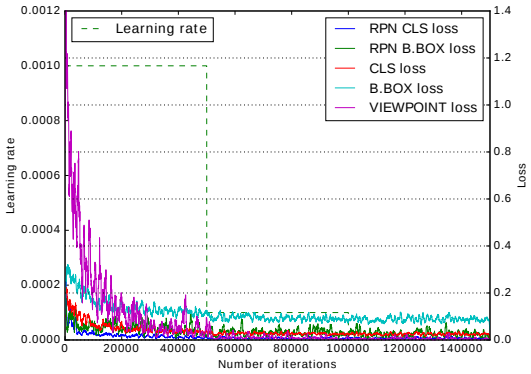


Fig. 3. Evolution of the losses during training. Best viewed in color.

B. Evaluation

Results are reported in Table I in terms of the selected metrics. Row designated as *mean* shows the mean Average Precisions (mAP) and mean Average Orientation Similarity (mAOS) values across the three categories. Our proposal is focused on efficiency and thus can perform all the inference tasks in 390 ms using a Tesla K40 GPU. Our implementation uses Caffe [24].

For the sake of comparison, mAP and mAOS results reported by [13] (SubCNN) and [17] (Mono3D) in the public KITTI object detection ranking[1] are also included in Table I, since they are currently the top-ranked comparable methods. Please note that running times for GPU implementations of these approaches are reportedly around 2 seconds and 4.2 seconds per frame, respectively. Moreover, it has to be considered that slightly different training and evaluation sets are used (as that results are evaluated in the privately annotated KITTI test set). As is also shown in Table I, our method exceeds the accuracy of the baseline L-SVM approach [20] by a large margin.

For further analysis, Figs. 4 and 5 show the (monotonically decreasing) precision-recall and orientation similarity curves obtained with our method. In the current implementation, detection scores for evaluation are computed from the predicted class probability only, so the confidence in the viewpoint estimation (which is available from the predicted probability distribution r^k) is not considered while computing the detection and orientation statistics. This design decision, which is intended to favor the detection over the viewpoint estimation, is probably the reason behind the early drop in precision shown in Fig. 5b.

[1]http://www.cvlibs.net/datasets/kitti/eval_object.php

TABLE I
DETECTION AND VIEWPOINT ESTIMATION PERFORMANCE ON THE VALIDATION SET (%) AND COMPARISON WITH OTHER METHODS

	Detection (AP)			Orientation (AOS)		
	Easy	Moder.	Hard	Easy	Moder.	Hard
Car	88.26	77.72	60.48	87.53	76.75	59.40
Pedestrian	78.73	67.12	60.73	72.75	61.45	55.39
Cyclist	62.42	44.82	43.56	49.48	34.49	33.52
mean	76.47	63.22	54.92	69.92	57.56	49.44
SubCNN	*84.52*	*77.14*	*69.44*	*80.37*	*72.85*	*65.45*
Mono3D	*82.91*	*73.90*	*67.09*	*75.91*	*66.58*	*60.18*
L-SVM	*50.27*	*41.11*	*35.45*	*46.13*	*37.78*	*32.49*

VII. CONCLUSIONS

We have presented a monocular approach for object detection focused on traffic environments which is based on a state-of-the-art CNN framework but also enables viewpoint inference to enhance the information provided by the on-board perception system.

According to the results, its performance is comparable to more sophisticated approaches not intended for real-time operation. Our efficient method based on the sharing of convolutional features enables real-time processing times, but it also constitutes a scalable framework where performance may be improved in the presence of higher-performance hardware (e.g., by enlarging the scale of the images).

Decision-making systems would benefit from the further insight into the environment provided by the viewpoint inference, thereby increasing the understanding of the environment and improving the prediction of future traffic situations.

In future work, we plan to extend our method to incorporate fine-grained orientation inference, such that viewpoint would not be represented by a single angle bin, but by the full probability distribution of bins. The final viewpoint inference could be performed by interpolation between the most probable bins in the estimation. That approach may exploit a cross-entropy logistic loss for the viewpoint loss component.

ACKNOWLEDGMENT

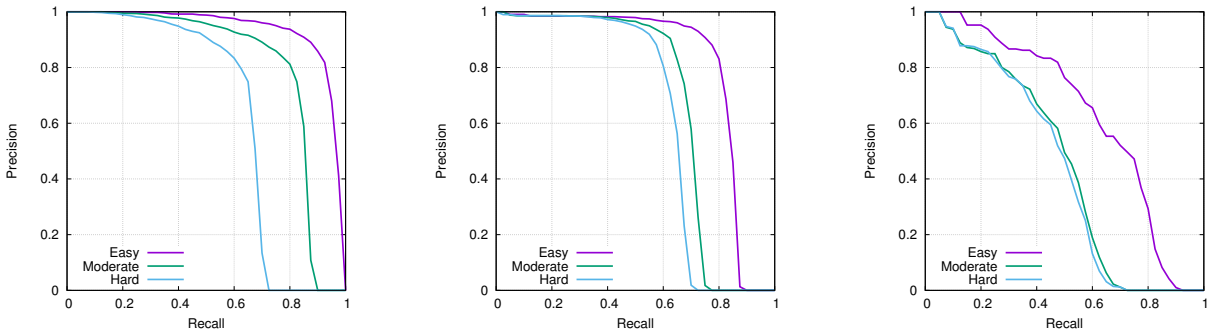
Research supported by the Spanish Government through the CICYT projects (TRA2013-48314-C3-1-R, TRA2015-63708-R and TRA2016-78886-C3-1-R), and the Comunidad de Madrid through SEGVAUTO-TRIES (S2013/MIT-2713). The Tesla K40 used for this research was donated by the NVIDIA Corporation.

REFERENCES

[1] A. Mahendran and A. Vedaldi, "Understanding Deep Image Representations by Inverting Them," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 5188 – 5196.

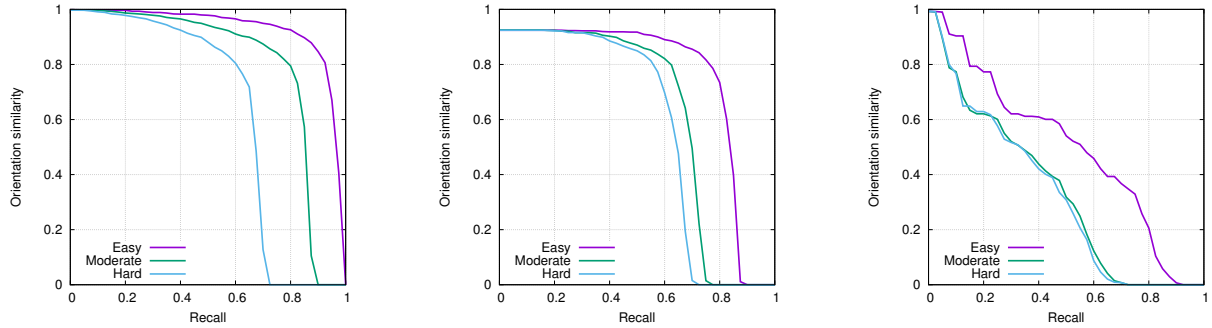
[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, 2016.

[3] A. Prioletti, A. Mogelmose, P. Grisleri, M. M. Trivedi, A. Broggi, and T. B. Moeslund, "Part-based pedestrian detection and feature-based tracking for driver assistance: Real-time, robust algorithms, and evaluation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1346–1359, 2013.



(a) Car (b) Pedestrian (c) Cyclist

Fig. 4. Precision-recall curves for detection.



(a) Car (b) Pedestrian (c) Cyclist

Fig. 5. Orientation similarity curves for detection and viewpoint estimation.

[4] F. García, D. Martín, A. de la Escalera, and J. M. Armingol, "Sensor Fusion Methodology for Vehicle Detection," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 123–133, 2017.

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[6] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based Convolutional Networks for Accurate Object Detection and Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 142–158, 2016.

[8] R. Girshick, "Fast R-CNN," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[9] J. Hosang, R. Benenson, P. Dollar, and B. Schiele, "What Makes for Effective Detection Proposals?" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 4, pp. 814–830, 2016.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[11] Q. Fan, L. Brown, and J. Smith, "A Closer Look at Faster R-CNN for Vehicle Detection," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2016, pp. 124–129.

[12] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection," in *Computer Vision - ECCV 2016. Lecture Notes in Computer Science, vol 9908*, 2016, pp. 354–370.

[13] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Subcategory-aware Convolutional Neural Networks for Object Detection," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017.

[14] A. Barth and U. Franke, "Estimating the driving state of oncoming vehicles from a moving platform using stereo vision," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 4, pp. 560–571, 2009.

[15] B. Pepik, M. Stark, P. Gehler, and B. Schiele, "Multi-View and 3D Deformable Part Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 11, pp. 2232–2245, 2015.

[16] E. Ohn-Bar and M. M. Trivedi, "Learning to Detect Vehicles by Clustering Appearance Patterns," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2511–2521, 2015.

[17] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3D Object Detection for Autonomous Driving," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2147–2156.

[18] C. B. Choy, M. Stark, S. Corbett-davies, and S. Savarese, "Enriching Object Detection with 2D-3D Registration and Continuous Viewpoint Estimation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 2512–2520.

[19] C. Gu and X. Ren, "Discriminative Mixture-of-Templates for Viewpoint Classification," in *Computer Vision - ECCV 2010*, 2010, pp. 408–421.

[20] A. Geiger, C. Wojek, and R. Urtasun, "Joint 3D Estimation of Objects and Scene Layout," in *Advances in Neural Information Processing Systems (NIPS)*, 2011.

[21] R. Girshick, "Training R-CNNs of various velocities," in *ICCV 2015 Tutorial on Tools for Efficient Object Detection*, 2015.

[22] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3354–3361.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1, 2014.

[24] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding," in *Proc. ACM International Conference on Multimedia*, 2014, pp. 675–678.