Guindel, C., Martin, D. & Armingol, J. M. (2018). Fast Joint Object Detection and Viewpoint Estimation for Traffic Scene Understanding. *IEEE Intelligent Transportation Systems Magazine*, 10(4), pp. 74–86.

DOI: 10.1109/mits.2018.2867526

Fast Joint Object Detection and Viewpoint Estimation for Traffic Scene Understanding

Carlos Guindel, David Martín *Member, IEEE,* and José María Armingol

***Abstract*—Environment perception is a critical enabler for automated driving systems since it allows a comprehensive understanding of traffic situations, which is a requirement to ensure safe and reliable operation. Among the different applications, obstacle identification is a primary module of the perception system. We propose a vision-based method built upon a deep 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 efficiency-oriented method achieves state-of-the-art accuracies for object detection and viewpoint estimation, and is particularly suitable for the recognition of traffic situations from on-board vision systems. Code is available at https://github.com/cguindel/lsi-faster-rcnn.**

***Index Terms*—Object detection, pose estimation, convolutional neural networks, computer vision, autonomous vehicles**

I. Introduction

DRIVER assistance systems and, especially, autonomous vehicles, rely on an array of complementary modules to perform the set of tasks involved in driving. As with human drivers, the perception function is probably the most critical one. In fact, a trustable situational understanding of the surroundings of the vehicle is a mandatory prerequisite for the rest of the subsystems down the pipeline. Among the responsibilities of the perception module, detection of dynamic obstacles around the car is almost always assumed as necessary as they may eventually interfere with the trajectory of the vehicle, with the attendant risk of collision.

Additionally, they are also increasingly expected to provide more detailed information about the type of agents involved in a hazardous situation. Classification allows a more accurate prediction of the immediate future behavior of the agents and, thus, improve the chances of success of an eventual 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 special attention in such situations.

While high-resolution laser rangefinders are increasingly used for obstacle detection, features provided by visual sensor systems are still usually needed to accurately identify the objects in the areas of interest around the vehicle. As an added benefit, video information can be employed by other driving-related applications where alternatives are impractical, such as lane departure warning or traffic sign recognition. However, traffic environments have traditionally posed a challenge to vision-based systems due to the lack of structure. Only recently have computer vision started to provide satisfactory accuracy rates thanks to the latest advances in deep learning techniques. In particular, convolutional neural networks (CNN) have proven to cope well with variations in poses, occlusions or illumination conditions [1], as required in traffic situations.

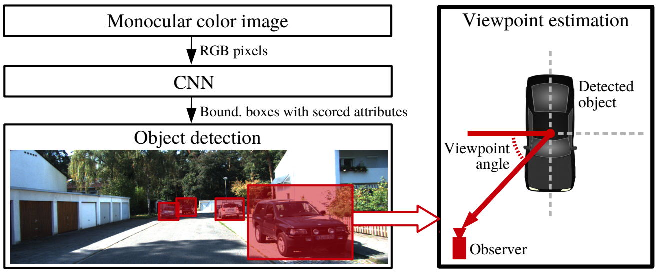


Fig. 1. Schematic overview of the proposed approach.

Despite robust detection of the dynamic agents in the scene being the core task of the perception system, the high maneuverability exhibited by typical road users makes it practically essential to provide the downstream modules with information on the pose of the objects to obtain accurate trajectory predictions and react accordingly, if necessary. In particular, the ability to estimate the orientation of the objects without relying on movement features ensures robustness against sudden changes in direction, thus reducing the uncertainty of short-term predictions. Conveniently, hierarchical representations of the appearance features provided by neural networks are suited for that purpose, as will be shown in this work.

Based on the widely used Faster R-CNN framework [2], we propose a joint detection and viewpoint estimation system especially suitable for onboard platforms, aimed to the identification of the road users in the field of view of a monocular camera. 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. An overview of the 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 per-instance estimation of the viewpoint, or observation angle, as described in the right part of Fig.

The authors are with the Intelligent Systems Laboratory (LSI) Research Group, at the Department of Systems Engineering and Automation, Universidad Carlos III de Madrid, 28911 Leganés, Madrid, Spain e-mail: {cguindel, dmgomez, armingol}@ing.uc3m.es.

1. Convolutional features are computed and shared for use in the tasks of region proposal, classification, and orientation estimation, for efficiency reasons.

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. Section IV is used to describe the details of the orientation estimation function. Experimental results are presented in Section V, and the conclusions of the paper are drawn in Section VI.

II. Related Work

For many years, onboard 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) or Haar-like features [3] are among the most traditional ones.

However, in recent years, representation learning has become the dominant approach in object recognition. Deep convolutional neural networks have emerged as a method enabling rich hierarchies of features [4], 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, as well as in real-life applications.

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 the ”regions with CNN features” philosophy introduced in R-CNN [5], where the convolutional network is applied over previously defined ROIs within the input image. Fast R-CNN [6] 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 [7]. Whereas 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, as in YOLO [8]. Thus, in Faster R-CNN approach [2], convolutional layers are shared between both proposal generation and classification, thus speeding up the process while achieving comparable, and sometimes 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 detections are required [9]. Some methods have been proposed in the literature to overcome this limitation; e.g., the recent ’recurrent rolling convolution’ architecture [10].

As an extension of the feature sharing idea, multi-task networks seek to exploit the same set of features to perform several tasks simultaneously. This paradigm is particularly attractive to onboard applications because of the typical time and processing restrictions. For instance, MultiNet [11] adds scene classification and road segmentation on top to the vehicle detection task, which is in turn based on a lightweight network. Similarly, in [12] semantic segmentation and road layout are inferred jointly.

While less frequent than detection, viewpoint estimation has been addressed previously in a significant number of works as a primary cue when understanding traffic environments. The pioneering work by Pepik et al. [13] extends the Deformable Part Model (DPM) scheme to handle different viewpoints with a 3D-aware loss function. More recently, methods based on CNNs take advantage of their enhanced detection accuracy [14], [15]. Pose-RCNN [16] extends Fast R-CNN to include viewpoint regression similarly to the present work, although they use proposals coming from stereo and lidar data, and a class-agnostic viewpoint estimation. Lately, viewpoint estimation has been embedded in the inference of the 3D detection and pose orientation of the objects in the scene [17], [18].

Our work makes use of the latest advances in object detection and extends them to include viewpoint estimation, following the tendency that is shown in the last group of works above. However, in contrast to them, we have designed a complete solution under the premise of achieving real-time performance with state-of-art accuracy levels for different traffic categories. This set of features makes our proposal particularly suitable for automotive applications.

III. Inference Framework

A variety of agents can be found in traffic environments, constituting potentially dangerous obstacles from the perspective of a moving vehicle. In this approach, we use a CNN-based approach to build a multi-class object detection and viewpoint inference system which is aimed to identify the objects within the image without significant prior constraints. The topology of the proposed network is depicted in Fig. 2.

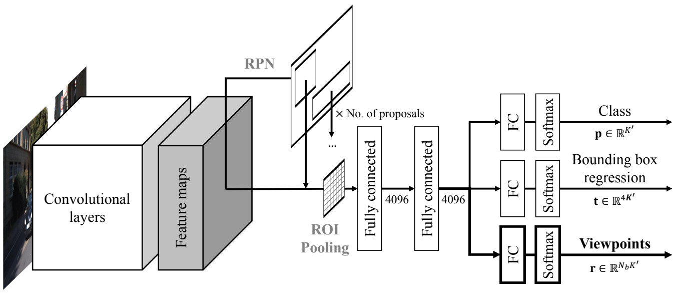


Fig. 2. Topology of the joint detection and viewpoint estimation network. $K^{'}$ is the number of classes, including a generic *background* class.

A. Faster R-CNN

Among the different CNN-based meta-architectures developed in recent years, we rely on Faster R-CNN, which has been found to be one of the most suitable ones to meet the real-time constraints posed in onboard applications [19]. Based as it is on feature learning, this method largely outperforms classic hand-crafted detectors, but it is also able to carry out the inference process efficiently, with the number of objects and classes not being a significant factor in performance.

Faster R-CNN is a fully trainable pipeline where detection happens in two stages. The first one is a region proposal network (RPN), which is responsible for selecting the potentially occupied image patches, while the second one is a Fast R-CNN subnet, where a set of box proposals coming from the first stage is classified. 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.

While the two-step procedure is slightly slower than single-shot approaches (e.g., SSD), its accuracy in the detection of small objects is considerably better. This feature is particularly desirable in onboard perception modules, where approaching objects must be detected ahead of time.

B. *Hyperparameter Tuning for Traffic Environments*

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. We keep the number of anchors at each position inalterable from the original model (nine) to not increase the complexity of the model, although it has been proven that adding new anchors would benefit the accuracy [15]. We have quantified that modifying the anchors' shape according to the application results in an improvement of some tenths of a percentage point in precision, with no impact on performance.

During our experiments, we also noticed that class imbalance in the training data was an important factor limiting detection accuracy. In particular, datasets aimed at autonomous driving typically have more car annotations than VRU annotations, even when those are particularly sensitive from a safety point of view. To mitigate this effect, classification loss, which is computed as a logistic loss in the regular Faster R-CNN, is replaced by an "information gain" (infogain) multinomial logistic loss. This way, the contribution of the various categories to the final loss can we weighted according to the frequency of the classes in the training set. Details of the training process, including the global loss function, are provided in Section IV-D.

IV. Viewpoint Estimation

Faster R-CNN uses a color image to provide a set of classified bounding boxes. As shown in Fig. 2, we extend this base topology to allow the network to predict the viewpoint from which each object is seen by the camera. We consider viewpoint a useful attribute to improve the understanding of the surrounding traffic scene since it enables a robust estimation of the objects' orientation.

As described in the previous section, one of the most appealing properties of the two-stage detection framework is that convolutional features can be shared across the different tasks to be performed by the network. We have found that the convolutional features used by the proposal and classification networks can be additionally exploited to provide a viewpoint estimation. Thanks to this approach, 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*

Due to the nature of the application, 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 may be considered to be negligible.

Viewpoint estimation methods can be divided into two groups: fine-grained pose estimators [14], able to infer arbitrary poses, or discrete pose estimators [20], which quantize the viewing sphere into a predefined number of bins and select the best one during inference. We adopt the discrete approach since, as will be demonstrated in Sec. V, it fits better into the Faster R-CNN design and has often been proven as adequate for high-level scene understanding [21].

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

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

where θ_o is an offset that can be used to control the start and end points of the bins. While this offset might be set to 0, we usually choose $\theta_o = \pi / N_b$ so the viewpoint bins correspond to well-defined orientations; i.e., front, rear, etc.

At training time, objects with ground-truth label θ^*_{j0} are assigned a viewpoint bin Θ_{l_0} such that $\theta^*_{j0} \in \Theta_{l_0}$. Similarly, viewpoint inference is designed to provide a bin $\Theta_{\hat{l}}$ representing the estimated pose for every object. The concept is illustrated in Fig. 3 for $N_b = 8$ and an offset angle $\theta_o = \pi/8$.

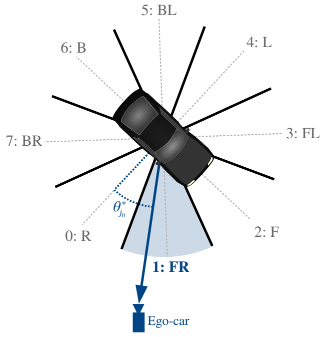


Fig. 3. Example of viewpoint quantization with $N_b = 8$ and $\theta^*_{j0} \in \Theta_1$ (i.e., $l_0 = 1$). Viewpoint bins are named using combinations of the four main orientations: front (F), back (B), left (L), and right (R).

For the introduction of the viewpoint estimation into the detection 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:

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

As will be described later, we use a class-aware prediction, so the output of the viewpoint branch of a network for K classes is made of $K+1$ different (N_b-1)-simplex sets, each describing the probability distribution for a class (plus the *background* class). Let us denote $\mathbf{r}^k$ the N_b-length prediction for the class k. Then, the estimated bin $\Theta_{\hat{l}_o}$ for a bounding box that has been classified as k_{j_o} is given by the index of the maximum element of $\mathbf{r}^{k_{j_0}}$; that is, $\hat{l} = \arg\max_i (r_i^k)$.

B. Pseudo-Continuous Viewpoint Estimation

We aim to provide an estimation of the viewpoint as a single value interpretable by higher-level applications. It is, therefore, necessary to convert the prediction from the discrete to the continuous domain. We could directly provide the center of the predicted bin; however, we have found that the estimation can be improved by means of an interpolation step. Following some ideas in [22], we use the probability values contained in r^k to provide an angular value within the range between two bin centers. The viewpoint is computed as the weighted average of two adjacent viewpoint bin centers, using their respective estimated probabilities provided by the network. In particular, we use the bin which has been given the highest probability, and its most probable neighbor. Given that the center of bin Θ_l, $Z(\Theta_l)$ is provided by:

$$Z(\Theta_l) = \frac{\pi(2l+1)}{N_b} - \theta_o \tag{3}$$

we obtain the viewpoint value for every object as:

$$\hat{\theta} = \frac{r_{\hat{l}}^k \cdot Z(\Theta_{\hat{l}}) + r_{\hat{l}_\pm}^k \cdot Z(\Theta_{\hat{l}_\pm}^k)}{r_{\hat{l}}^k + r_{\hat{l}_\pm}^k} \tag{4}$$

with $\hat{l}_\pm = \arg\max_{l \in \{\hat{l}-1, \hat{l}+1\}} (r_l^k)$. Obviously, bins 0 and $N_b - 1$ are assummed adjacent for this purpose.

Even though the objective vector for the class-aware distributions ($\mathbf{r}^k$) is given as a one-hot representation, interpreting the outcome as a manifold encoding all the viewpoints in $[Z(\Theta_{\hat{i}}) - \pi/N_b, Z(\Theta_{\hat{i}}) + \pi/N_b)$ allows increasing the resolution of the prediction and should be able to mitigate some instances of confusion.

C. 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, as shown in Fig. 2. Given the close relationship between orientation and appearance, we assume that the same set of features can be used for classification and pose estimation.

We use an RPN for selecting the proposals. This structure is responsible for assigning each anchor an *objectness* label (foreground/background) through a categorical probability distribution $\mathbf{a} = (a_0, a_1)$, as well as a refinement of its coordinates, expressed as an offset relative to the anchor box itself: $\mathbf{b} = (b_x, b_y, b_w, b_h)$.

At the classification stage, the resulting feature vectors (one for each proposal) are introduced into a sequence of fully connected layers which are finally divided into three sibling layers (instead of two as usual), responsible for the different inference tasks:

- Class. This layer applies the softmax function to get the categorical distribution $\mathbf{p}$ that describes the probabilities for the K available classes (and an additional catch-all *background* class): $\mathbf{p} = (p_0, \ldots, p_K)$
- 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 x and y coordinates and their width (w) and height (h) dimensions. The bounding box refinement is class-aware: $\mathbf{t}^k = (t_x^k, t_y^k, t_w^k, t_h^k), k = 0, \ldots, K$
- Viewpoint. We add a third layer for the estimation of the viewpoint, which is also obtained through a softmax function and given as a $N_b \cdot (K+1)$-length output representing $K+1$ categorical distributions (one per class plus a disposable one for the background proposals) over the N_b viewpoint bins: $\mathbf{r}^k = (r_0^k, \ldots, r_{N_b-1}^k), k = 0, \ldots, K$

D. 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 excellent trade-off between accuracy and training time.

Viewpoint is introduced into the loss function as a logistic loss that only adopts non-zero values for foreground classes. From the $N_b \cdot (K+1)$ elements in the output given by the viewpoint layer, $\mathbf{r}$, we only consider the N_b elements belonging to the ground-truth class during training time.

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

$$\begin{aligned} L =& \frac{1}{N_{B_1}} \sum_{j \in B_1} L_{cls}(\mathbf{a}_j, u_j) + \frac{1}{N_{B_1}} \sum_{j \in B_1} u_j L_{loc}(\mathbf{b}_j, \mathbf{b}_j^*) + \\ & \frac{1}{N_{B_2}} \sum_{i \in B_2} L_{inf}(\mathbf{p}_i, v_i) + \sum_{i \in B_2} [v_i \geq 1] \, L_{loc}(\mathbf{t}_i^{v_i}, \mathbf{t}_i^{v_i *}) + \\ & \frac{1}{N_{B_2}} \sum_{i \in B_2} [v_i \geq 1] \, L_{cls}(\mathbf{r}_i^{v_i}, w_i) \end{aligned} \tag{5}$$

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 fixed 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 via two parameters. As previously stated, outputs from the network are $\mathbf{p}_i$, $\mathbf{t}_i$ and $\mathbf{r}_i$, defined for each image region i in B_2, using the proposals given by $\mathbf{a}_j$ and $\mathbf{b}_j$, which are defined for each anchor j in B_1. In Eq. 5, u_j is the binary ground-truth class for the anchor j, v_i the true class of the region i and w_i the index of 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 [6], and L_{inf} is the infogain multinomial logistic loss:

$$L_{inf}(\mathbf{p}_i, v_i) = \sum_{k=1}^{K} H_{v_i,k} \log(p_{i,k}) \tag{6}$$

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, i.e., the element k of $\mathbf{p_i}$. 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 proportions of the different classes in the training dataset, such that less frequent classes have higher $H_{v_i,k}$ values. This way, the class loss element becomes:

$$L_{inf}(\mathbf{p}_i, v_i) = H_{v_i,v_i} \log(p_{i,v_i}) \tag{7}$$

which can be straightforwardly interpreted as a weighted multinomial logistic loss.

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

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.

V. Experimental Results

We performed our experiments on the KITTI object detection dataset [23], where viewpoint labels are available. Nine different categories representing dynamic agents are annotated; however, as usual, we performed our experiments using the *Car*, *Pedestrian* and *Cyclist* classes, for which a significant number of samples is available. Please note that we have also tested before the adequacy of the method using the whole set of classes [24]. Regions whose label belongs to other categories, including *DontCare* (the catch-all category for distant or unclear objects) are not used at training time, either as positive or negative samples. To that end, we only consider samples whose IoU (Intersection-over-Union) overlap with such non-valid regions is limited to 15% (for the proposals) and 25% (for classification).

We divided the KITTI training dataset, whose labels are publicly available, into two splits for training (5,415 images) and validation (2,065 images), ensuring that frames from the same sequence are not present in both training and validation sets. Samples in the dataset are separated into three difficulty levels, according to their size, and occlusion and truncation levels: *Easy*, *Moderate* and *Hard*.

We have performed a series of experiments to study the influence of some of the meta-parameters of the algorithm, as well as ablation studies to provide alternative use cases which may be useful depending on the specifications of the application.

Different well-established metrics for the assessment of detection and pose estimation algorithms will be used:

- Precision. 2-D object detection within the image is evaluated as the average precision (AP) for 11 different values of recall, following [25]. Unless otherwise specified, we follow the KITTI criteria regarding the minimum Intersection-over-Union (IoU) overlap required for true positives: 70% for *Car* and 50% for *Pedestrian* and *Cyclist*. Sometimes we summarize the AP across the three analyzed classes with the mean value (mAP).
- Orientation similarity. We use the measure introduced in [23] to assess the joint detection and orientation performance of the algorithm. This metric is a normalized variant of the cosine similarity where only the correctly detected samples are considered. The average value (AOS) is provided taking into consideration the same recall values used in the AP. Please note that this metric is upper-bounded by the average precision value. Similarly to the AP, AOS can be condensed into a mAOS metric.
- Recall for a fixed number of proposals. The recall achieved under different IoU requirements is a useful metric to determine the effectiveness of the region proposal methods. It can be averaged similarly to AP and AOS to provide the average recall value (AR).
- Mean Precision in Pose Estimation (MPPE). Discrete orientation approaches are sometimes evaluated using this metric introduced by [26], which is defined as the mean of the elements on the main diagonal of the confusion matrix obtained for the bin classification problem. While MPPE is usually computed for a fixed detection threshold, we apply here the same paradigm as with the precision and the orientation similarity and compute an MPPE value for each recall value.

It is important to note that, for evaluation purposes, detections are given the confidence score provided by the class prediction branch. The confidence in the viewpoint estimation, which is available from the predicted probability distribution $\mathbf{r}^k$, is not taken into account while computing the detection and orientation statistics. This design decision is intended to favor the detection over the viewpoint estimation. The classification score is also used to perform a non-maximum suppression (NMS) at the end of the pipeline in order to avoid redundant detections.

A. Training Parameters

Although the proposed method is agnostic to the architecture of the convolutional layers of the network, we use the VGG16 architecture from [27] 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 one of the parameters with higher impact on the final performance [9], 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 inference times tractable. The data are augmented by horizontal flipping. Training is performed for

50k iterations with a learning rate of 0.001, then for 50k iterations with 0.0001 and finally for another 50k iterations with 10^{-5}.

Other training parameters, e.g., batch sizes, are in line with the setup used for Pascal VOC at the original implementation of Faster R-CNN, except for the dropout regularization, which is not employed, and the extent of the weight-updating process, which affects now all the convolutional layers.

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

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

where f_{min} is the number of occurrences of the less frequent class and f_k the number of instances of class k. Background samples are assigned a unit weight. Values obtained from the train split are tabulated in Table I.

TABLE I
INFOGAIN MATRIX VALUES FOR THE EVALUATED CLASSES.

background	Car	Pedestrian	Cyclist
1	1.38	1.8	2

Total loss converges quickly during the first few epochs of training, which further proves the effectiveness of the proposed multi-task loss function.

B. Parameter Analysis

1) Information Gain Loss: In Table II we investigate the effect of utilizing the information gain loss for computing the class misclassification error at training time.

Infogain loss is mostly targeted at improving the precision performance in classification, but it also has an overall positive effect on the joint detection and viewpoint estimation accuracy. While the less frequent categories, i.e *Pedestrian* and *Cyclist*, were expected to be the most benefited, *Car*, whose contribution to the final loss is effectively reduced, improves the performance as well for *Moderate* and *Hard* difficulty levels.

2) Viewpoint Value Refinement: We have evaluated the effect of the interpolation technique described in Sec. IV-B, as well as the angle offset θ_o from Eq. 1. The interpolation method is a test-time modification, whereas the offset requires training a new model. For that reason, although these changes are focused on the viewpoint estimation performance, detection is also subject to variations when this option is selected. Results are summarized in terms of mAP and mAOS in Table III. As stated previously, θ_o is set to π/N_b to make bins coincidental with intuitive orientation directions.

Results show a non-negligible improvement in viewpoint estimation performance, but the angle offset also implies an improvement in the detection performance. Training probably benefits from the improved appearance separation among the different viewpoint bins when the offset is introduced.

3) Number of Viewpoint Bins: We introduced our approach for $N_b = 8$ [28] as a good compromise between the model complexity and the orientation estimation accuracy. In this work, we further study the influence of the number of bins in performance. Fig. 4 shows the results for viewpoint estimation regarding the orientation similarity and the MPPE for samples from the *Moderate* difficulty level. Tests have been performed for $N_b \in \{4, 8, 16, 32\}$. It is worthwhile to point out that resolution in the viewpoint estimation is limited to $2\pi/N_b$ rad, although our interpolation method is employed here to increase the resolution of the prediction.

AOS analysis in Fig. 4a shows that $N_b = 8$ offers a sweet spot in terms of orientation similarity, which is particularly noticeable for *Cyclist*, even when the underlying resolution is as coarse as $\pi/4$ rad. Overall, the loss in resolution for the smaller N_b values is counterbalanced by the improvement in MPPE due to the lower number of classes, as shown in Fig. 4b-4d. When using *Hard* samples, $N_b = 8$ outperforms its neighbor options (4 and 16) by around 1 percentage point (pp) in mAOS.

Nevertheless, $N_b = 16$ also offers a good trade-off, especially for *Car*, where viewpoint classification is usually based on a larger number of features, due to size concerns. For the same reason, *Easy* samples see a larger improvement of the mAOS. In contrast, for $N_b = 32$, the rise of the viewpoint classification loss at training time hurts the detection accuracy (-1.11 mAP for *Moderate*) and mAOS drops around 2.5 pp for all difficulty levels. From now on, we use $N_b = 8$.

C. Ablation Studies

Since our implementation is strongly aimed at onboard applications, we investigated the effect of one of the most critical parameters influencing inference time: the number of proposals from the RPN that get effectively classified in the Fast R-CNN stage. By default, features from 300 proposals are cropped from the feature maps, and each one goes through the chain of fully connected layers to obtain the final inference values. However, recent studies suggest that it is possible to obtain comparable accuracy using significantly fewer proposals [19], resulting in shorter computation times and thus higher framerates. The influence of the number of proposals is investigated in Fig. 5.

Results suggest that the effectiveness of the proposals are not affected as much as expected by reducing the number of proposals, up to a certain limit, as the average recall does not drop severely until less than 50 proposals are considered. This is also reflected in the mAP and mAOS values. The latter experiences a drop of no more than 1.33 pp when 50 proposals are used. Running time per frame gets reduced by a 9%, with 100 proposals, and a 13.6%, with 50 proposals, to reach 76 ms per frame using a Titan Xp GPU.

Overall, the performance of the method using 100 and 50 proposals is satisfactory enough to provide a faster alternative for critical applications. We also noticed that the largest impact was in the detection of smaller, and therefore distant, objects. To analyze this, we evaluated the variation of the total recall (i.e., considering all the proposals) with the Euclidean distance of the objects employed in the evaluation from the ego-car for 50, 100 and 300 proposals. Here we used the *Hard* difficulty level to take into account as many annotations as possible.

TABLE II
DETECTION AND VIEWPOINT ESTIMATION PERFORMANCE ON THE VALIDATION SET (%) WITH AND WITHOUT USING INFOGAIN LOSS.

infogain loss	Car			Pedestrian			Cyclist		
	Easy	Moder.	Hard	Easy	Moder.	Hard	Easy	Moder.	Hard
	Average Precision (AP)								
	89.24	77.95	60.65	**80.73**	68.78	62.96	64.38	49.39	47.15
✓	89.18	**78.67**	**61.21**	80.37	**69.03**	**62.98**	**68.98**	**51.50**	**49.99**
	Average Orientation Similarity (AOS)								
	87.64	76.37	59.11	71.95	61.13	56.00	**54.48**	**41.82**	40.03
✓	87.60	**77.11**	**59.70**	**74.52**	**63.35**	**57.37**	53.86	41.58	**40.55**

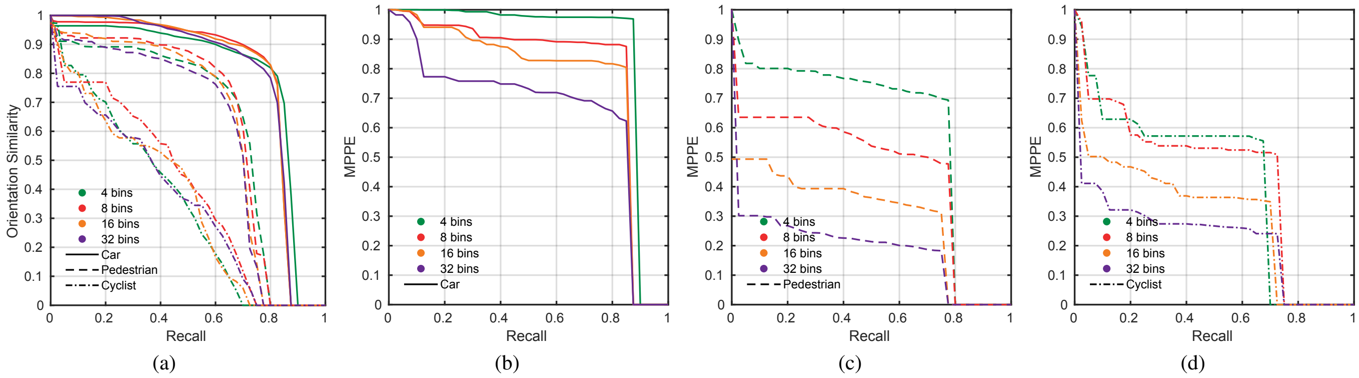


Fig. 4. Orientation similarity-recall and MPPE-recall curves on the *Moderate* validation set for different values of N_b.

TABLE III
DETECTION AND VIEWPOINT ESTIMATION PERFORMANCE ON THE VALIDATION SET (%) FOR THE VARIANTS IN VIEWPOINT REFINEMENT.

offs.	interp.	mAP			mAOS		
		Easy	Moder.	Hard	Easy	Moder.	Hard
		77.71	65.75	58.02	71.33	59.39	51.77
	✓	77.71	65.75	58.02	71.54	59.61	51.98
✓		**79.51**	**66.40**	**58.06**	71.75	60.48	52.34
✓	✓	**79.51**	**66.40**	**58.06**	**71.99**	**60.68**	**52.54**

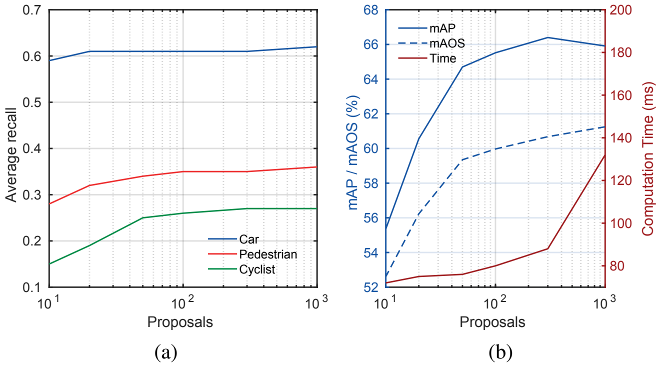


Fig. 5. Average recall, mean average precision and mean average orientation similarity versus the number of proposals on the *Moderate* validation set.

Results are provided in Fig. 6. They justify our hypothesis, with a particularly pronounced impact in *Cyclist*, and a far more limited one in *Car*.

Apart from the number of proposals, the most influential parameter on performance is the scale factor applied to the input images. However, in this case, we have found that the running time is highly sensitive to changes in the size of the image; thus, a 2× scale factor dramatically increases the mAOS in 7.39 pp, but running time also raises a 94.3% to reach 171 ms. Although powerful, this parameter is restricted to a limited range when pursuing real-time solutions.

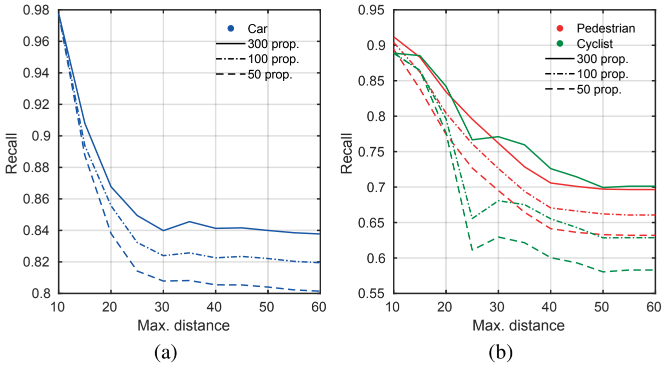
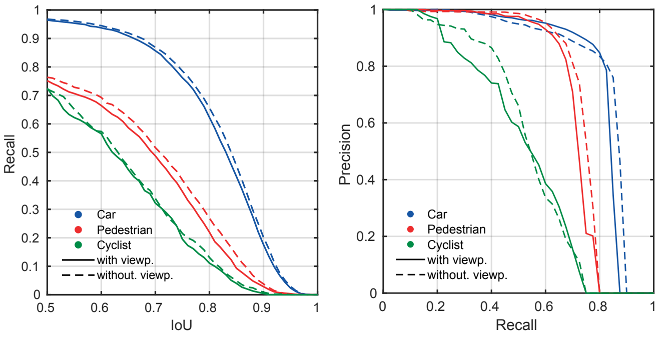


Fig. 6. Recall versus max. distance from the ego-car on the *Hard* validation set for different numbers of proposals.

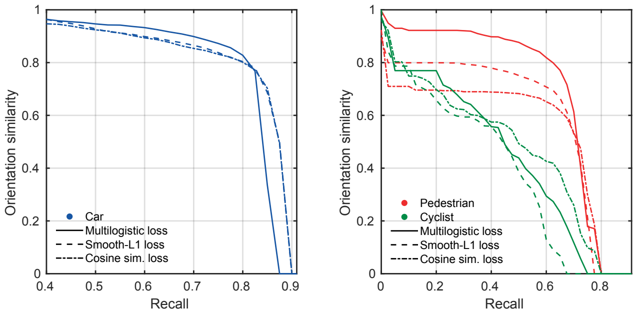
D. Evaluation

To prove the effectiveness of our multi-task approach, we compare it with the baseline Faster R-CNN approach, where only detection is performed, all other things being equal. The comparison is presented in Fig. 7.



(a) (b)

Fig. 7. Precision-recall and Recall-IoU (300 proposals) curves on the *Moderate* validation set with and without viewpoint estimation.



(a) (b)

Fig. 8. Orientation similarity versus recall on the *Moderate* validation set for three alternative viewpoint estimation methods. Our proposal uses the multilogistic loss.

According to the Recall-to-IoU metric, the viewpoint inference has a minor impact on the effectiveness of the proposals. On the other hand, the precision-recall curves demonstrate that the detection performance of both variants is comparable.

We also carried out an analysis of our method in comparison with two alternative approaches which pose the viewpoint estimation problem as a regression. We embedded these methods into our framework and performed experiments to assess the validity of the viewpoint estimation branch in particular.

The two alternative regression approaches that we have investigated are:

1) Smooth-L1 loss. As in other works, we used Smooth-L1 for orientation regression [29], [30]. The inference was posed as a class-aware regression to be performed by the viewpoint fully connected layer, with a $(K+1)$-length output vector (one element per class including the *background* class).
2) Cosine similarity loss. We also implemented the viewpoint estimation as a class-agnostic regression problem where the loss function was inversely proportional to the cosine similarity between the predicted angle and the ground-truth viewpoint. As mentioned before, the cosine similarity is the central component of the orientation similarity measure.

In both cases, the regression was performed by the viewpoint-dedicated fully connected layer on top of the common set of classification layers, as in our proposal. A comparison using orientation similarity is provided in Fig. 8.

Our discrete approach outperforms the other methods, even when they are continuous regression approaches, and therefore their estimations are not restricted by a fixed resolution. The overall improvement in mAOS is 3.86 pp from the AOS-based regression, and 4.23 pp from the Smooth-L1-based regression using the *Moderate* validation set.

Finally, we compare the performance of our approach against other algorithms on the KITTI benchmark. We submitted our results to the official website[1] in order to enable comparisons on a common test set. Average precision and average orientation similarity results for the top-ranked methods in the KITTI benchmark are tabulated in Table IV. All the methods in the table are based on DCNNs, except for the DPM-VOC+VP method, which we include as a baseline. Methods using stereo or lidar information, as well as single-class detection methods, are not considered.

As we rely on the results reported in the KITTI benchmark, processing times are dependent on the particular implementations and the hardware used in the measurements by the respective authors. However, it is a safe assumption that modern GPUs have been used in all cases except for DPM-VOC+VP, which is explicitly measured on a CPU. Therefore, our proposal, which spends 88 ms per frame, is significantly faster than the existing methods. Even when the accuracy of our method is close to the top-performing methods, we are confident that AP and AOS results could be improved by enlarging the scale of the input image as previously stated; however, here we decided to prioritize the framerate to provide a view of the results under close-to-market constraints.

Our implementation is based on the Python version of the Faster R-CNN code[2] and therefore uses Caffe [31]. Our times are measured on an NVIDIA Titan Xp GPU. Code has been made publicly available.

Fig. 9 shows some examples of detections on the KITTI test set for qualitative evaluation. As shown in Fig. 9c-9d, viewpoint provides a valuable insight into complex traffic situations, such as intersections. The influence of the pseudo-continuous estimation can be observed, for instance, in the leftmost car in Fig. 9e. Note that the effect of the scale can be seen in Fig. 9f, where some distant (and hence small) detections are missed.

VI. Conclusion

We have presented a monocular approach for object recognition, focused on traffic environments, which adds viewpoint inference on top of an highly-optimized CNN-based detection framework.

Results show that the performance of our proposal is on par with recent methods, although it is focused on efficiency and thus can perform all the inference tasks in less than 90 ms

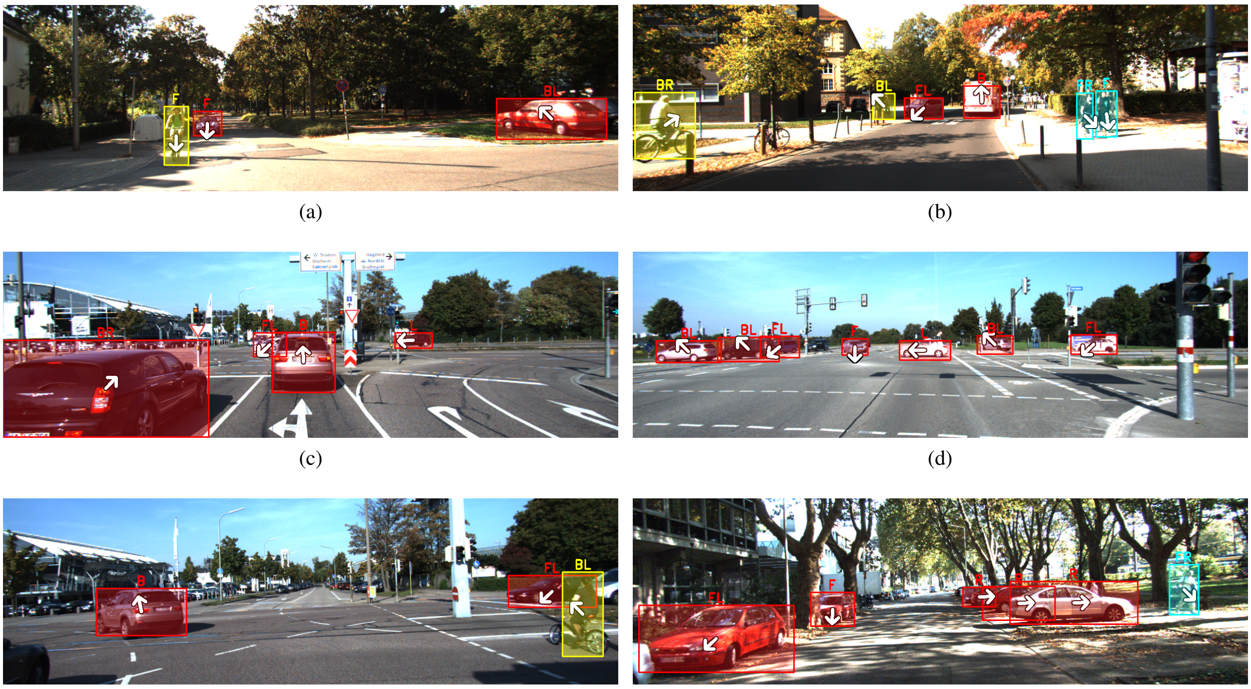
[1] http://www.cvlibs.net/datasets/kitti/eval_object.php

[2] https://github.com/rbgirshick/py-faster-rcnn

TABLE IV
COMPARISON WITH OTHER METHODS OF THE DETECTION AND VIEWPOINT ESTIMATION PERFORMANCE ON THE KITTI TEST SET (%). DATA OBTAINED FROM THE PUBLIC KITTI OBJECT BENCHMARK.

method	Car			Pedestrian			Cyclist			time (s)
	Easy	Moder.	Hard	Easy	Moder.	Hard	Easy	Moder.	Hard	
Average Precision (AP)										
DPM-VOC+VP [13]	80.45	66.25	49.86	59.60	44.86	40.37	43.65	31.16	28.29	8[1]
Mono3D [17]	90.27	87.86	78.09	77.30	66.66	63.44	75.22	63.85	58.96	4.2[2]
Deep3DBox [18]	90.47	88.86	77.60	-	-	-	82.65	73.48	64.11	1.5[2]
SubCNN [15]	90.75	88.86	79.24	83.17	71.34	66.36	77.82	70.77	62.71	2[3]
Ours (FRCNN+Or)	89.87	78.95	68.97	71.18	56.78	52.86	68.81	55.80	50.52	0.09[4]
Average Orientation Similarity (AOS)										
DPM-VOC+VP [13]	77.51	63.27	47.57	53.66	39.83	35.73	31.24	23.22	21.62	8[1]
Mono3D [17]	89.00	85.83	76.00	68.58	58.12	54.94	65.74	53.11	48.87	4.2[2]
Deep3DBox [18]	90.39	88.56	77.17	-	-	-	68.58	59.37	51.97	1.5[2]
SubCNN [15]	90.61	88.43	78.63	78.33	66.28	61.37	71.39	63.41	56.34	2[3]
Ours (FRCNN+Or)	88.52	77.61	67.69	66.84	52.62	48.72	63.41	50.91	45.46	0.09[4]

[1] CPU; [2] GPU 2.5 GHz; [3] GPU 3.5 GHz; [4] Titan Xp GPU (1.6 GHz)



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 9. Examples of detections with viewpoint estimation on the KITTI test dataset. Class is encoded using color: *Car*, *Cyclist* and *Pedestrian* classes are represented as red, yellow, and blue, respectively. The pseudo-continuous viewpoint estimation is depicted as an oriented white arrow at the center of the obstacle, while the predicted bin is provided above each bounding box as a combination of front (F), back (B), left (L) and right (R).

on a commercially-available GPU. We have made the source code available to ease the research reproducibility.

Viewpoint prediction will enhance the information made available to the decision-making systems in the vehicle, thereby increasing the understanding of the environment and improving the prediction of future traffic situations.

In future work, we plan to improve the recall in the detection of small and further objects by using features from shallower convolutional layers. On the other hand, we want to explore the use of single-shot meta-architectures (e.g., SSD) for detection in traffic environments. Finally, online hard example mining is also being tested to improve the management of the imbalance between foreground and background classes.

ACKNOWLEDGMENT

This work was 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). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPUs used for this research.

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. 39, no. 6, pp. 1137–1149, 2017.

[3] 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.

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

[10] J. Ren, X. Chen, J. Liu, W. Sun, J. Pang, Q. Yan, Y.-W. Tai, and L. Xu, "Accurate Single Stage Detector Using Recurrent Rolling Convolution," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5420–5428.

[11] M. Teichmann, M. Weber, M. Zoellner, R. Cipolla, and R. Urtasun, "MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving," *arXiv:1612.07695 [cs.CV]*, 2016.

[12] M. Oeljeklaus, F. Hoffmann, and T. Bertram, "A Combined Recognition and Segmentation Model for Urban Traffic Scene Understanding," in *Proc. IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 2292–2297.

[13] 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.

[14] 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.

[15] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Subcategory-aware Convolutional Neural Networks for Object Detection," in *Proc. IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017, pp. 924–933.

[16] M. Braun, Qing Rao, Y. Wang, and F. Flohr, "Pose-RCNN: Joint object detection and pose estimation using 3D object proposals," in *Proc. IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 1546–1551.

[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] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3D Bounding Box Estimation Using Deep Learning and Geometry," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7074–7082.

[19] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3296–3305.

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

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

[22] L. Yang, J. Liu, and X. Tang, "Object detection and viewpoint estimation with auto-masking neural network," in *Computer Vision - ECCV 2014*, 2014, pp. 441–455.

[23] 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.

[24] C. Guindel, D. Martín, and J. M. Armingol, "Modeling Traffic Scenes for Intelligent Vehicles Using CNN-Based Detection and Orientation Estimation," in *ROBOT 2017: Third Iberian Robotics Conference: Volume 2*. Springer International Publishing, 2018, pp. 487–498.

[25] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[26] R. J. López-Sastre, T. Tuytelaars, and S. Savarese, "Deformable part models revisited: A performance evaluation for object category pose estimation," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 1052–1059.

[27] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *CoRR*, vol. abs/1409.1, 2014.

[28] C. Guindel, D. Martin, and J. M. Armingol, "Joint object detection and viewpoint estimation using CNN features," in *Proc. IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, 2017, pp. 145–150.

[29] X. Chen and Y. Zhu, "3D Object Proposals for Accurate Object Class Detection," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2015, pp. 424–432.

[30] C. C. Pham and J. W. Jeon, "Robust object proposals re-ranking for object detection in autonomous driving using convolutional neural networks," *Signal Processing: Image Communication*, vol. 53, pp. 110–122, 2017.

[31] 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.

Carlos Guindel received a B.S. degree in Industrial Engineering in 2012, and an M.S. in Robotics and Automation in 2014, both from the Universidad Carlos III de Madrid (Spain), which also awarded him an Excellence Award in 2011 for his outstanding academic record. He is currently pursuing a Ph.D. degree as a member of the Intelligent Systems Laboratory (LSI), where he has been involved in several research projects since 2011. His research focus is on computer vision and artificial intelligence, with a special interest in their application to intelligent transportation systems and, particularly, autonomous vehicles.

David Martín graduated in Industrial Physics (Automation) from the National University of Distance Education (UNED, 2002) and Ph.D. degree in Computer Science from the Spanish Council for Scientific Research (CSIC) and UNED, Spain 2008. He was Ph.D. student at CSIC from 2002 to 2006. He was fellow at the European Organization for Nuclear Research (CERN, Switzerland, 2006-2008) and Post-Doc researcher in Robotics at CSIC (2008-2011). Currently, he is Professor and Post-Doc researcher at Carlos III University of Madrid and member of the Intelligent Systems Lab since 2011. His research interests are Computer Vision, Sensor Fusion, Intelligent Transportation Systems, Advanced Driver Assistance Systems, Autonomous Ground Vehicles, Unmanned Aerial Vehicles, and Vehicle Positioning and Navigation. In 2014, he was awarded with the VII Barreiros Foundation award to the best research in the automotive field. In 2015, the IEEE Society has awarded Dr. Martín as the best reviewer of the 18th IEEE International Conference on Intelligent Transportation Systems.

José María Armingol graduated from the Universidad Politécnica de Madrid (Spain) in Automation and Electronics Engineering in 1992. He received his Ph.D. in Robotics and Automation from the Universidad Carlos III de Madrid in 1997. He is Full Professor in the Systems Engineering and Automation Department at Universidad Carlos III de Madrid since the year 2012 and co-founded and member of the intelligent Systems Lab since the year 2000. His research interests focus on Computer Vision, Image Processing and Real-Time Systems applied to Autonomous Vehicles and Advanced Driver Assistance Systems. In 2014, he was awarded with the VII Barreiros Foundation award to the best research in the automotive field.