Guindel, Carlos; Martín, David; Armingol, José María; Stiller, Christoph (2018). Analysis of the Influence of Training Data on Road User Detection. *Proceedings of 2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES), September 12-14, 2018, Madrid, Spain,* pp. 1-6.

DOI: https://doi.org/10.1109/ICVES.2018.8519510

Analysis of the Influence of Training Data on Road User Detection

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

Christoph Stiller
Institute of Measurement and Control Systems
Karlsruhe Institute of Technology
Karlsruhe, Germany
stiller@kit.edu

***Abstract*—In this paper, we discuss the relevance of training data on modern object detectors used on onboard applications. Whereas modern deep learning techniques require large amounts of data, datasets with typical scenarios for autonomous vehicles are scarce and have a reduced number of samples. We conduct a comprehensive set of experiments to understand the effect of using a combination of two relatively small datasets to train an end-to-end object detector, based on the popular Faster R-CNN and enhanced with orientation estimation capabilities. We also test the adequacy of training models using partially available ground-truth labels, as a consequence of combining datasets aimed at different applications. Data augmentation is also introduced into the training pipeline. Results show a significant performance improvement in our exemplary case as a result of the higher variability of the training samples, thus opening a new way to improve the detection performance independently from the detector architecture.**

I. Introduction

Object detection is an issue frequently discussed in the computer vision literature, which has given rise over time to a plethora of methods aimed to solve the problem. The ITS community is not alien to this trend, given that the robust identification of road agents is widely recognized as one of the cornerstones of autonomous driving. In fact, safe operation of autonomous vehicles depends to a great extent on the quality of the data produced by the perception algorithms.

Research interest in semantic segmentation algorithms is experiencing enormous growth in recent years since they can arguably perform different functions which were previously divided into different modules [1]. Notwithstanding the potential of this kind of techniques, most methods are still unable to distinguish between different instances of the same category [2], which does not sit well with the requirements of autonomous navigation. Automated cars drive in a complex environment where individual agents diverge in behavior and therefore need to be appropriately identified and tracked. Recent developments in instance-aware semantic segmentation [3] show that this task is closely related to object detection. This is one of the reasons why object detection continues to be an active research topic, especially when applied to the particularly challenging road scenarios.

Computer vision, in general, and object detection, in particular, are nowadays unquestionably dominated by deep learning approaches, due to their superior performance. However, the optimization process involved in feature learning algorithms require a vast amount of data to obtain models with generalization abilities. Huge datasets, such as ImageNet [4] and COCO [5], have been developed to fulfill the demands of these data-hungry algorithms for the image recognition task. Nevertheless, traffic environments pose additional challenges which are not always covered by those generalistic datasets, such as occlusions and far objects. Although some autonomous-driving-oriented datasets, namely KITTI [6] and Cityscapes [7], are geared towards the requirements of these applications, the number of samples available in them is significantly lower, thus affecting the performance of learning algorithms. Additionally, labeled data differs between datasets, thus complicating the task of leveraging data from diverse sources.

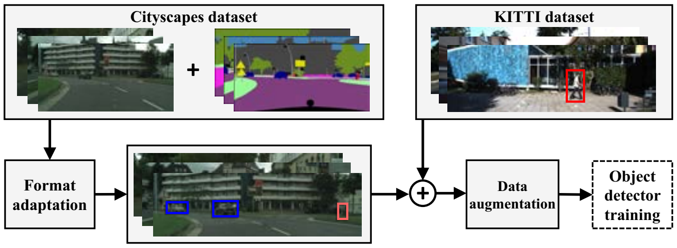


Fig. 1. Pipeline of the training procedure for object detection combining the KITTI and Cityscapes datasets and applying data augmentation.

Using these two datasets, we investigate in this paper the influence of training data on the performance of object detection algorithms. In particular, we focus on the improvement capabilities provided by introducing additional samples into the training process, as well as the possibility of using heterogeneous labels in a multi-task learning method. Tests are performed on our previously introduced object detection algorithm [8], which is based on the state-of-art architecture Faster R-CNN [9]. The procedure is sketched in Fig. 1.

The rest of this paper is structured as follows. In Section II, we provide a review of recent object detection works. In Section III, a brief description of the employed datasets, as well as the detection algorithm, is provided. Experiments and an interpretation of the results are given in Section IV, while the conclusions of the paper are drawn in Section V.

II. Related Works

Since a few years ago, practically all state-of-art object detection approaches on images are based on convolutional neural networks (CNNs). These structures provide end-to-end pipelines which can learn complex hierarchies of features, which enabled a breakthrough in the accuracy of the detection task; however, this comes at the cost of requiring much more extensive databases for training. Transfer learning alleviates this issue since, as was proved by Razavian et al. [10], image descriptors learned by a CNN are a generic representation that can be exploited to perform tasks different than those for which they were specifically trained.

Convolutional networks, initially used for image classification, were soon applied to object detection, first using classical sliding-window approaches [11], and later within tailored techniques, such as the "recognition using regions" (R-CNN) paradigm [12]. The latter proposed applying CNNs to previously selected region proposals to obtain features that could be finally classified by an SVM. While accurate, the approach was slow since it involved duplicate computations. Fast R-CNN [13] tried to overcome this problem by computing features over the whole image, then classifying only the object proposals by pooling the corresponding regions of the feature maps; the SVM classifier was also shown to be unnecessary.

The performance of the approaches based on this methodology was profoundly compromised by the quality of the incoming proposals. Particularly poor results were obtained on traffic environments due to the complexity inherent to those images, featuring objects at many different scales which are often occluded or truncated. Different region proposal methods were proposed to mitigate this shortcoming, such as 3DOP [14], which generates class-specific 3D proposals, and MS-CNN [15], which employs proposal sub-networks working at different scales in the feature pyramid.

However, the most influential approach for proposal generation was introduced by Ren et al. as Faster R-CNN [9]. They embed the region proposal stage into the learnable pipeline using a small region proposal network (RPN). This approach was able to provide high accuracy detection at online framerates, although not without problems, mainly caused by the fixed receptive field of the RPN.

Recent approaches aimed at detection in densely populated traffic scenes propose sophisticated paradigms to improve the results of the RPN, such as the scale-dependent pooling (SDP) by Yang et al. [16]. Although these solutions effectively improve the accuracy, they also introduce a significant overhead which prevents its practical application into onboard detection systems. Simpler detection meta-architectures have been proposed, including R-FCN [17], a variant of Faster R-CNN which defer the feature pooling step to the last layer prior to prediction.

Notwithstanding the interest of the architectural variants of object detectors, we argue that training data is a factor of paramount importance in the performance of object detectors, and is frequently overlooked in the literature.

III. Experimental Setup

A. Datasets

We rely on the KITTI Object Detection dataset [6] to validate our hypotheses. The KITTI dataset is made of 7,481 training images, profusely annotated and riddled with a variety of challenging instances with different sizes, poses, and occlusion status. Labels include the axis-aligned boxes of visible objects belonging to seven categories. This dataset is widely used in the literature to evaluate object detection algorithms in demanding scenarios. We use the training/validation subsets of [14], with 3,712 and 3,769 images, respectively. The validation subset is employed as the reference testbed throughout this work, enabling comparison between the different alternatives which will be analyzed.

We conduct experiments by including images from the Cityscapes dataset [7]. We employ the 2,975 frames available for training. Annotations from the Cityscapes dataset are pixel-wise as they are aimed at semantic segmentation algorithms. To use them in our object detection framework, we had to convert the annotations to the bounding box format featured by the KITTI dataset, suitable for object detection. We let each bounding box be the minimum enclosing rectangle of the set of polygons defining an instantiable object. Samples belonging to the *Rider* category are merged with the corresponding vehicle annotation, following the *Cyclist* class specification from the KITTI dataset. Besides, *Person* labels are assumed equivalent to the KITTI *Pedestrian* labels.

In order to leave the too heavily occluded or truncated samples out of the training procedure, we also provide each instance with an estimation of its occlusion and truncation, à la KITTI. We assume that occlusion takes place whenever a box belonging to a foreground object intersects with the bounding box of a background object. The degree of occlusion is estimated as the ratio between the size of the intersection and the area of the background box. On the other hand, truncation is considered to occur when any of the sides of the bounding box coincides with the image boundaries. Although both are rough estimations, we found them valid for our filtering purposes.

A region of interest of 2048 × 620 is extracted from the Cityscapes images to make them similar to the available KITTI frames regarding the vertical field of view; additionally, this cropping removes the hood of the ego-car, which is otherwise visible in the lower part of the images.

The KITTI dataset defines three difficulty levels: *Easy*, *Moderate* and *Hard*. We decided to ignore the samples not meeting the requirements imposed by the *Hard* level; that is:

1) Height: larger than 25 pixels.
2) Maximum occlusion: difficult to see, which corresponds to the level 2 in the KITTI annotations and 75% in the Cityscapes annotations.
3) Maximum truncation: 50%. Hence, we conservatively discard all the samples from the Cityscapes dataset which are adjacent to the image boundaries.

It is important to note that the ignored annotations are not used either a positive or negative samples to avoid confusion. Table I contains the number of samples that we will use in this paper for our experiments.

TABLE I
Number of Samples on the KITTI and Cityscapes Datasets

category	KITTI			Cityscapes
	train	val	total	total
Car	10 753	10 963	21 716	21 637
Pedestrian	2 104	2 172	4 276	15 788
Cyclist	594	600	1 194	1 481

B. Object Detection Method

We adopt Faster R-CNN as the object detection framework, with the configuration and adaptations described in [8]. Faster R-CNN is a two-stage detection method where a common set of feature maps, obtained through a particular feature extractor, is sequentially used by a region proposal network (RPN) to generate regions of interest, and by a classification stage to assign a category to those proposals. Proposals are generated from a set of predefined *anchors*, and the final bounding box is regressed during the classification step to provide a more accurate location.

Taking advantage of the multi-task nature of the approach, we introduced the viewpoint estimation task into the pipeline to infer the objects' observation angle along with the classified bounding boxes. We pose the problem as a multiclass classification where objects are assigned a viewpoint bin, spanning a discrete range. In this work, we use eight viewpoint bins. The whole pipeline is depicted in Fig. 2.

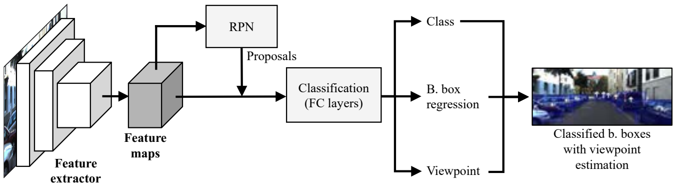


Fig. 2. Object detection and viewpoint estimation framework built upon the Faster R-CNN meta-architecture.

The three last branches (or *heads*) share almost all the processing except for one last specific fully connected layer (one per branch). Both the bounding box refinement and the viewpoint estimation are class-aware; i.e., independent predictions are given for each possible category, and the one corresponding to the inferred class is chosen.

We tried to keep the detector architecture as close to the original proposal as possible, to increase the generalizability of the conclusions of this study. Hence, we use the default set of anchors and most settings from [9]. Features corresponding to each proposal are sampled from the last convolutional layer using ROI pooling. In this paper, we only consider the configuration based on the VGG-16 architecture [18]. Input images are rescaled so that the shortest side is 500 pixels.

C. Multi-Task Training

We adopt the approximate joint training strategy from [9]. Therefore, we use a multi-task loss with five components: two from the RPN (*objectness* and bounding box regression), and the remaining three from the classification stage, accounting for the class, the bounding box refinement, and the viewpoint estimation. Regression losses are Smooth-L1 losses, whereas the objectness, class, and viewpoint are optimized through multinomial logistic losses. For the class estimation, we use a weighted multinomial logistic loss letting the underrepresented categories have a higher contribution to the final loss, as discussed in [8].

The training process is carried out using single-image batches, where proposals are extracted by the RPN to train the classification part. Whereas KITTI samples are endowed with all the ground-truth labels contained in the multi-task loss, Cityscapes lacks the viewpoint annotations. We handle this issue by letting the contribution of the viewpoint task be nil when processing a Cityscapes frame.

We use a standard SGD for training, with a step decay learning rate schedule. An initial learning rate of 0.001 is chosen. We drop the learning rate by a factor of 0.1 every 50k iterations. Unless otherwise specified, models are trained for 80k iterations.

IV. Analysis

We compare different alternatives for improving the object detection performance by modifying the training data. As previously above, results are obtained on the KITTI validation set. We use the measures established by the KITTI dataset [6]; namely, average precision (AP) and average orientation similarity (AOS). The former is obtained by averaging the precision values over different thresholds corresponding to equispaced recall values and is aimed to assess the detection accuracy; the latter is the comparable measure for joint detection and orientation assessment and is based on the cosine similarity. We limit our quantitative analysis to the commonly studied categories: *Car*, *Pedestrian* and *Cyclist*, for which a representative number of samples is available.

A. Combined Datasets

Firstly, we investigate the effect of adding the 2,975 training frames from Cityscapes on top of the KITTI training split, made of 3,682 images. In each training iteration, an image is randomly chosen from the mix of both datasets.

To decouple the analysis of the viewpoint estimation functionality, we first analyze the *vanilla* Faster R-CNN, without the viewpoint *head* (see Fig. 2), in the first place. In Table II, we study the change in average precision for the three difficulty levels when adding the Cityscapes data. We also include the results when using the Cityscapes dataset alone. The performance on the KITTI validation set is improved for all categories and difficulty levels by incorporating the Cityscapes frames. The overall effect is an increase of 4.24 points in mean AP (mAP) across the three classes on the *Moderate* data.

TABLE II
DETECTION PERFORMANCE ON THE KITTI VALIDATION SET (%, AVERAGE PRECISION) USING DIFFERENT SETS OF TRAINING DATA

category	tr. data	Easy	Mod.	Hard
Car	KITTI	90.05	79.32	70.04
	Cityscapes	81.37	63.66	53.47
	KITTI + CS	**90.31**	**84.94**	**70.33**
Pedestrian	KITTI	75.80	67.17	58.58
	Cityscapes	72.00	63.92	55.33
	KITTI + CS	**77.77**	**68.72**	**60.05**
Cyclist	KITTI	77.47	56.96	54.64
	Cityscapes	63.09	50.14	46.85
	KITTI + CS	**82.90**	**62.50**	**58.05**

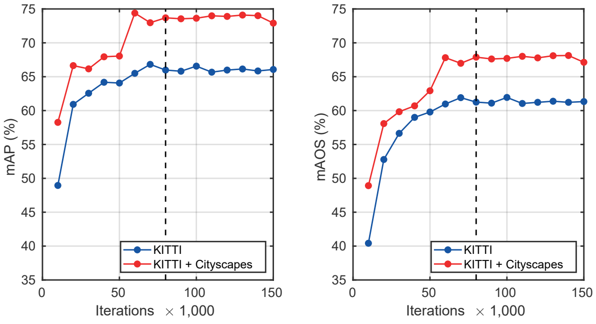
Next, we consider the full pipeline, including the viewpoint estimation branch. In addition to the 'pile-up' strategy, we also explore an additional approach to take advantage of the Cityscapes data: we pre-train a model using the Cityscapes set and transfer the weights as initial values of a new training process using the KITTI training split. Table III shows the results based on the AP and AOS measures, with 'KITTI + CS' representing the first strategy and 'KITTI (w. CS pret.)', the second one. Mixing both datasets proves more effective than using Cityscapes in a pre-training stage. Moreover, the enhancement is even more prominent than it was when evaluating detection exclusively. All categories experience a boost with an overall effect on the mAP of 7.71 points on the *Moderate* subset.

Average orientation similarity, responsible for jointly assess detection and orientation estimation, is also improved, even when, as stated before, Cityscapes samples do not have orientation annotations. Increase on mAOS reaches 6.64 points on the *Moderate* difficulty level. This effect will be further discussed in Sec. IV-C.

Finally, we test if the limited set of images from the Cityscapes dataset is enough to dispense with the commonly-employed ImageNet pre-training. Hence, we trained a model with the combined dataset, starting from weights initialized using a zero-mean Gaussian distribution with standard deviation 0.01 (except for the bounding-box regression and viewpoint estimation *heads*, initialized with standard deviation 0.001). We intentionally do not modify the training schedule. Results, in Table IV, show that initialization with a large, generalist dataset is still an essential requirement for the model to achieve a proper generalization ability.

B. Overfitting

The relatively low number of samples available, even when both datasets are combined, might arguably lead to overfitting. In Fig. 3, we analyze the evolution of mAP and mAOS on the KITTI *Moderate* validation set with the number of training iterations. We provide here data until 150k iterations, using the same step decay learning schedule described above. In both cases, performance plateaus shortly after the first learning rate decay at 50k, which justifies the choice of 80k iterations as stopping point. However, symptoms of overfitting are not observed.



(a) (b)

Fig. 3. Evolution of performance measures with the number of training iterations, with and without using Cityscapes samples: (a) mean average precision; (b) mean average orientation similarity.

Nonetheless, we also analyze the effect of using dropout regularization [19], with $p = 0.5$, on a model trained exclusively with the KITTI training split. Results, summarized in Table V, show no apparent benefit.

C. Heterogeneous Annotations

Results in Table III seems to suggest that using Cityscapes samples, which are not endowed with viewpoint annotations, benefits the orientation estimation performance. However, it is important to note that AOS evaluates the joint detection and orientation performance, and is, indeed, upper bound by the AP. Therefore, it is expected an increase in AOS when improving the AP, even when the viewpoint estimation performance remains constant. Nevertheless, it could be argued that alternating samples with and without orientation labels may hurt the viewpoint estimation performance itself.

As we are using a discrete approach, we analyze the performance as an isolated problem by using the 'mean precision in pose estimation' (MPPE) measure proposed by [20], which is the mean of the elements on the main diagonal of the confusion matrix of the viewpoint bin classification problem. We use the average MPPE across different recall values. As shown in Table VI, performance is comparable, or even better in some instances.

Following this line, we also proved the validity of training a model with an extended set of classes from both datasets, with the following categories: *Car*, *Truck* (including the KITTI's *Van* class), *Pedestrian* (including *Person_sitting*), *Cyclist*, *Train* (including *Tram*) and *Traffic Sign*. It is noteworthy that the latter is only available in the Cityscapes dataset, while it is considered as *background* in the KITTI samples; yet, the model can detect some well-visible instances, as will be shown in Sec. IV-E.

D. Data Augmentation Techniques

Data augmentation is frequently used to improve the robustness of deep learning methods. All models were trained with

TABLE III
Detection and Viewpoint Estimation Performance on the KITTI Validation Set (%) Using Different Sets of Training Data

category	tr. data	Detection (AP)			Orientation (AOS)		
		Easy	Mod.	Hard	Easy	Mod.	Hard
Car	KITTI	90.01	79.03	69.67	88.26	77.35	67.97
	KITTI + CS	**90.39**	84.59	70.21	**88.68**	82.79	68.57
	KITTI (w. CS pret.)	90.33	**86.16**	**70.58**	88.63	**84.43**	**69.01**
Pedestrian	KITTI	71.19	64.05	55.75	65.31	57.62	50.01
	KITTI + CS	**76.32**	**67.98**	**59.11**	**67.83**	**59.65**	**51.69**
	KITTI (w. CS pret.)	74.54	66.01	57.68	67.33	59.01	51.52
Cyclist	KITTI	77.33	54.87	52.89	69.73	48.79	47.06
	KITTI + CS	**86.11**	**68.49**	**63.46**	**77.66**	**61.23**	**56.83**
	KITTI (w. CS pret.)	83.18	60.37	57.35	75.55	54.36	51.74

TABLE IV
Detection and Viewpoint Estimation Performance on the KITTI Validation Set (%) With and Without Pre-Training on ImageNet

init.	tr. data	Detection (mAP)			Orientation (mAOS)		
		Easy	Mod.	Hard	Easy	Mod.	Hard
Yes	KITTI	**79.51**	**65.98**	**59.44**	**74.43**	**61.25**	**55.02**
No	K. + CS	53.80	42.99	37.25	47.93	39.13	33.00

TABLE V
Detection and Viewpoint Estimation Performance on the KITTI Validation Set (%) With and Without Dropout

dropout	Detection (mAP)			Orientation (mAOS)		
	Easy	Mod.	Hard	Easy	Mod.	Hard
No	**79.51**	**65.98**	**59.44**	**74.43**	**61.25**	**55.02**
Yes	79.20	65.34	58.43	73.77	60.73	54.16

random horizontal flipping, but we further study the effect of other augmentation techniques in this section. In particular, we focus on texture augmentation methods, affecting color and illumination. Four typical transformations are applied:

1) **Addition**. A random value between -40 and 40 is added to all pixels in the image.
2) **Multiplication**. All pixels in the image are multiplied by a factor randomly chosen from the range [0.5, 1.5].
3) **Additive Gaussian noise**. We add a small jitter to each pixel, following a Gaussian distribution with mean 0 and standard deviation between 0 and 5.1.
4) **Addition to 'hue' and 'saturation'**. A value randomly sampled in the range [-20, 20] is added to the H and S channels of the image expressed in HSV color space.

Not every transformation is applied to all frames, but instead a random number between 0, none of them, and 4, all of them, is chosen. We rely on the *imgaug*[1] library for the implementations of the augmentations. We conducted separated experiments with the KITTI training set and the combined set; the latter might especially benefit from this idea given the notable difference in contrast and color balance between both sources. However, as shown in Table VII, the impact of texture augmentations is reduced. The most relevant effect takes place on the *Hard* subset of the combined dataset (+0.9 pp mAP).

TABLE VI
Pure orientation performance on the KITTI Validation Set (%, MPPE) Using Different Sets of Training Data

category	tr. data	Easy	Mod.	Hard
Car	KITTI	**92.24**	80.93	69.29
	KITTI + CS	92.13	**83.40**	**71.72**
Pedestrian	KITTI	**59.03**	51.02	43.71
	KITTI + CS	57.74	**51.35**	**44.41**
Cyclist	KITTI	**70.71**	49.84	49.00
	KITTI + CS	64.95	**51.60**	**49.11**

TABLE VII
Detection and Viewpoint Estimation Performance on the KITTI Validation Set (%) With and Without Data Augmentation

tr. data	aug.	Detection (mAP)			Orientation (mAOS)		
		Easy	Mod.	Hard	Easy	Mod.	Hard
K.	No	79.51	**65.98**	**59.44**	74.43	**61.25**	**55.02**
	Yes	**80.39**	65.87	58.96	**74.56**	61.00	54.38
K. + CS	No	**84.27**	73.69	64.26	**78.06**	67.89	59.03
	Yes	83.96	**74.14**	**65.16**	77.95	**68.09**	**59.59**

E. *Qualitative Results*

Fig. 4 depicts a comparison, based on some selected examples from the KITTI test set, between the baseline model and a model trained with the combined dataset, as well as data augmentation. It can be noticed that some instances that were not detected using the baseline model are correctly identified using the enhanced training data.

On the other hand, examples in Fig. 5 shows the results with the mix of categories introduced in Sec. IV-C. As shown, some instances of traffic signs are correctly detected.

V. Conclusion

We have proved that modestly enhancing the training data can lead to notable improvements on the results obtained by a

[1] https://github.com/aleju/imgaug

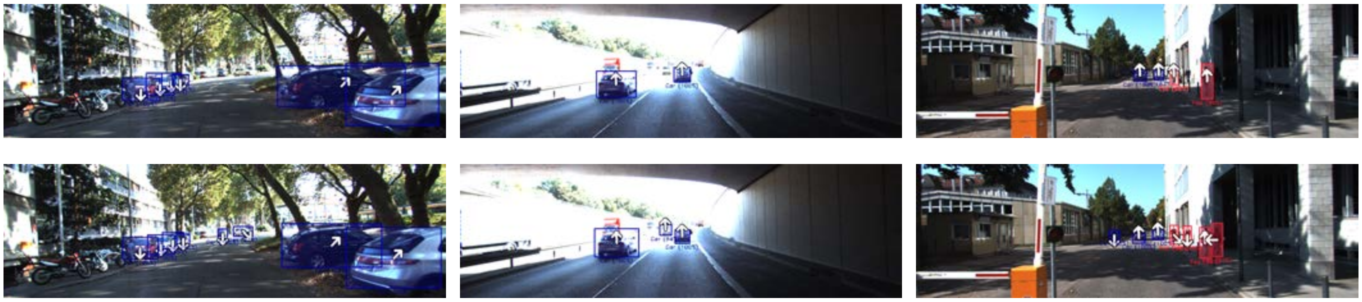


Fig. 4. Selected examples of object detection and viewpoint estimation results on the KITTI test dataset. Upper row: with a model trained on the KITTI training split; lower row: with a model trained on the combined dataset.

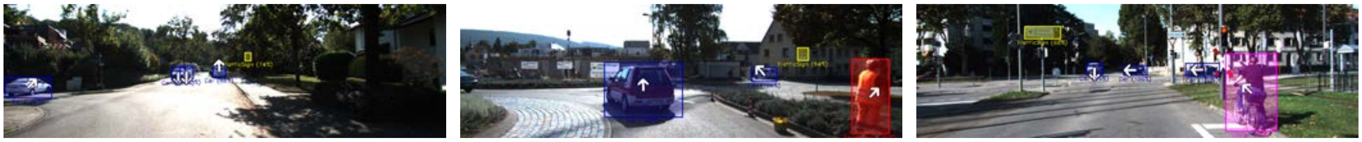


Fig. 5. Selected examples of object detection and viewpoint estimation results on the KITTI test dataset with additional categories.

CNN-based object detector, which is especially interesting on onboard applications, where scarcer data is usually available. The variability introduced by the reduced number of samples from the Cityscapes dataset can achieve a non-negligible improvement from a model trained on the KITTI dataset alone, even when tests are conducted on frames from the latter. Results of our experiments pave the way for future works taking advantage of multiple sources of data.

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. We would also like to acknowledge the contribution of Iñigo Barredo to the development of part of the software used in this work.

References

[1] E. Romera, L. M. Bergasa, and R. Arroyo, "Can we unify monocular detectors for autonomous driving by using the pixel-wise semantic segmentation of CNNs?" in *IEEE Intelligent Vehicles Symposium (IV) - DeepDriving Workshop*, 2016.

[2] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, "A Review on Deep Learning Techniques Applied to Semantic Segmentation," *arXiv:1704.06857 [cs.CV]*, 2017.

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

[4] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[5] T.-Y. Lin *et al.*, "Microsoft COCO: Common Objects in Context," in *Computer Vision – ECCV 2014*. Springer International Publishing, 2014, pp. 740–755.

[6] 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.

[7] M. Cordts *et al.*, "The Cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3213–3223.

[8] 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*, 2017, pp. 145–150.

[9] 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.

[10] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN Features off-the-shelf: an Astounding Baseline for Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2014, pp. 512–519.

[11] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. Lecun, "Pedestrian detection with unsupervised multi-stage feature learning," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 3626–3633.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 580–587.

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

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

[15] 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.*, 2016, pp. 354–370.

[16] F. Yang, W. Choi, and Y. Lin, "Exploit All the Layers: Fast and Accurate CNN Object Detector with Scale Dependent Pooling and Cascaded Rejection Classifiers," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2129–2137.

[17] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," in *Advances in Neural Information Processing (NIPS)*, 2016, pp. 379–387.

[18] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.1556 [cs.CV]*, 2014.

[19] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[20] 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.