End-to-End Single Shot Detector using Graph-based Learnable Duplicate Removal Supplementary Material

Shuxiao Ding[1,2][0000-0002-4040-5585], Eike Rehder[1][0000-0002-6255-0724], Lukas Schneider[1][0000-0001-9708-9248], Marius Cordts[1][0000-0001-8729-9233], and Jürgen Gall[2][0000-0002-9447-3399]

[1] Mercedes-Benz AG, Stuttgart, Germany
{shuxiao.ding,eike.rehder,lukas.schneider,marius.cordts}@mercedes-benz.com
[2] University of Bonn, Bonn, Germany
gall@iai.uni-bonn.de

A Comparison with other duplicate removal methods

Related works applied their methods on different backbones and implementations. To isolate their impact from the backbone's base performance we show it before and after applying the proposed methods as well as the relative changes of mean AP in percentage in Table 1. We compare our approach with the results provided in the respective publications. Note that all methods rely on a two-stage detector as a backbone. GossipNet [2] achieved a 0.8% higher AP for a standard Faster-RCNN model, corresponding to a 3.4% relative improvement. Relation Network [3] compared the changes for three different two-stage detectors with Relational Module (RM) in RCNN. We generally observe slight improvements with the highest relative improvement of 1.6% for FPN. Our approach achieves a relative improvement of 2.2% and 1.2% for EfficientDet-D0 and RetinaNet-ResNet50, respectively, which is less attractive compared to GossipNet. However, as shown in the previous section, the reported improvements of GossipNet can no longer be reproduced with SSD.

Table 1. Comparison of our approach and related works. We show improvements from using NMS to applying the learning duplicate removal method on MS-COCO *test-dev*. We also include the relative mean AP improvement in percentage for comparison.

method	e2e	model	AP	ΔAP in %
GossipNet[2]		Faster-RCNN	23.5 → 24.3	**+3.4%**
Relation Network[3]	✓	Faster-RCNN+RM	35.2 → 35.4	+0.6%
		FPN+RM	38.3 → 38.9	+1.6%
		DCN+RM	38.8 → 39.0	+0.5%
ours	✓	EfficientDet(SSD)	32.0 → 32.7	+2.2%
		RetinaNet(SSD)	34.2 → 34.6	+1.2%

B Additional Ablation Studies

Table 2. Comparison of the number of detection candidates K. We also compare the orignal setting $K = 600$ with our chosen $K = 300$ with a best performance for GossipNet.

Method	K	AP	AP_{50}	AP_{75}
ours	100	32.2	49.2	35.4
	200	32.2	49.1	35.2
	300	**32.7**	**49.9**	**35.9**
	400	32.5	49.6	35.5
	500	32.4	49.3	35.4
Gossip-Net	300	31.6	49.7	33.7
	600	31.3	49.6	33.3

Table 3. Ablation study of IoU threshold T_2, classification cost α and localization cost weight β.

T_2	α	β	AP	AP_{50}	AP_{75}
0.50	1.0	1.0	29.0	49.6	30.5
0.60	1.0	1.0	31.0	**50.3**	33.5
0.70	1.0	1.0	**32.7**	49.9	**35.9**
0.80	1.0	1.0	31.9	46.4	35.6
0.50	1.0	0.0	22.7	46.6	19.5
0.60	1.0	0.0	28.4	48.9	30.3
0.70	1.0	0.0	31.7	48.9	34.9
0.80	1.0	0.0	31.9	46.2	35.7
0.70	0.0	1.0	8.1	12.7	8.6

Table 4. Comparison of NMS and SoftNMS with our approach.

	hyper-	EfficientDet			RetinaNet		
	param	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
NMS	$N_t = 0.4$	31.4	**50.0**	33.1	34.2	52.1	36.6
	$N_t = 0.5$	31.6	50.0	33.3	**34.4**	**52.1**	36.8
	$N_t = 0.6$	**31.6**	49.2	33.7	34.3	51.4	37.1
	$N_t = 0.7$	31.2	47.4	**34.2**	33.9	49.8	**37.4**
SoftNMS	$\sigma = 0.2$	31.7	**49.9**	33.6	34.4	**51.9**	37.1
	$\sigma = 0.4$	**31.7**	49.2	**33.9**	**34.6**	51.8	**37.5**
	$\sigma = 0.6$	31.1	47.6	33.5	34.2	50.6	37.2
	$\sigma = 0.8$	30.3	45.9	32.9	33.5	49.2	36.6
ours	–	**32.7**	49.9	**35.9**	**34.7**	50.8	**38.7**

Table 5. Ablation study of the IoU threshold T_1 of the class-agnostic NMS for generating target of the pre-filtering head.

T_1	AP	AP_{50}	AP_{75}
0.95	32.3	49.5	35.3
0.90	**32.7**	**49.9**	**35.9**
0.80	32.5	49.6	35.6
0.70	32.2	49.1	35.2

Impact of the number of candidates We first vary the number of detection candidates as the input of the message-passing network K and compare their performance in Table 2. With $K = 300$, our approach achieves a highest performance. Reducing the number K to 200 leads to an AP drop of 0.5% but it doesn't decrease continuously when further reducing K to 100. On the other hand, increasing K causes a slighter performance drop. Similar to our approach, GossipNet also achieves a better performance by modeling the relationship between top 300 detection candidates, while the original setting $K = 600$ performs slightly worse.

Impact of matching parameters Next, we investigate the hyperparameters of the bipartite matching. The first four rows of Table 3 show the performance with different matching threshold T_2 that distinguishes positive and negative samples. The overall performance peaks at $T_2 = 0.7$ with an AP of 32.7%. In the next four rows, only classification cost is considered in matching by setting β to 0 so that the matching results are similar to the greedy matching that are used in related works [2], [3]. The model performs best when $T_2 = 0.8$, while the APs at different T_2 are worse than the one with localization cost. Different than Relation Network [3], we don't observe a strong correlation between T_2 and the IoU threshold for calculating AP in both settings. The reason might be that we used the refined boxes for matching while no box refinement are used in other works. The last row of Table 3 shows that the network fails when omitting the classification cost (i.e. $\alpha = 0$) because the matcher ignores the effect of the rescoring in this case.

B.1 Hyperparameters of heuristic algorithms

As the performance of the widely used heuristic duplicate removal algorithms, e.g. NMS and SoftNMS [1], strongly relies on hyperparameters tuning, we show a more comprehensive comparison between greedy NMS and SoftNMS to our approach on *val* in Table 4 by varying the IoU threshold N_t of greedy NMS and the normalizing parameter σ of SoftNMS respectively. Both, EfficientDet and RetinaNet perform best if SoftNMS with $\sigma = 0.4$ is used. Increasing σ of SoftNMS leads to a significant performance drop. As for greedy NMS, an IoU threshold N_t with 0.5 or 0.6 provides best performance but the highest AP_{75} is reached by setting N_t to 0.7. Our method outperforms the classical NMS and SoftNMS even for the optimal hyperparameters, especially on AP_{75}.

B.2 Supervision NMS in pre-filtering

We show an ablation study of the IoU threshold T_1 of the class-agnostic NMS that is used to supervise the pre-filtering head in Table 5. The overall performance can be reached by setting $T_1 = 0.9$. Increasing T_1 may lead to an insufficient filtering of highly duplicated boxes, which causes a performance drop by 0.4% if $T_1 = 0.95$. With a smaller IoU threshold T_1, the class-agnostic NMS filters more duplicates but also some potential true-positives. This can be interpreted by the slight performance drop with $T_1 = 0.8$ and $T_1 = 0.7$. After all, we don't seek a clean suppression of the most duplicates from the pre-filtering but a rough filtering that discards obvious duplicates. And we found that $T_1 = 0.9$ provides a good balance due to its best performance.

C Qualitative Evaluation

In this section, we show a more comprehensive qualitative evaluation of COCO *val* set. We use EfficientDet-D0 as the base model to generate detection results for evaluation.

C.1 Detections from different network stages

Our approach processes SSD raw detections in two stages: it first pre-filters all detections and then generates final rescored and refined detections. To show the effectiveness of each component, especially the pre-filtering, we illustrate the detection results from different intermediate stages. Each row of Figure 1 and 2 shows raw SSD detections, pre-filtered detections and final detections for one example. As most SSD raw detections are background, we only show the top 5000 raw detections for every category on the left side. The top 300 pre-filtered detections for every category are shown in the middle. We keep the top 100 detections among all categories as final detections following the COCO evaluation criterion. The box opacity indicates the predicted score and we show different categories in different colors. As shown in Figure 1 and 2, SSD predicts a large amount of very similar boxes with high scores which are easily recognized as duplication. The pre-filtering produces a sparser detection set by suppressing many highly overlapping boxes. The duplicate detections around larger objects are more likely suppressed by pre-filtering due to an obvious high IoU between each other. After rescoring by message passing, the network is able to produce only one high-scoring detection for every object.

C.2 Failure cases without pre-filtering

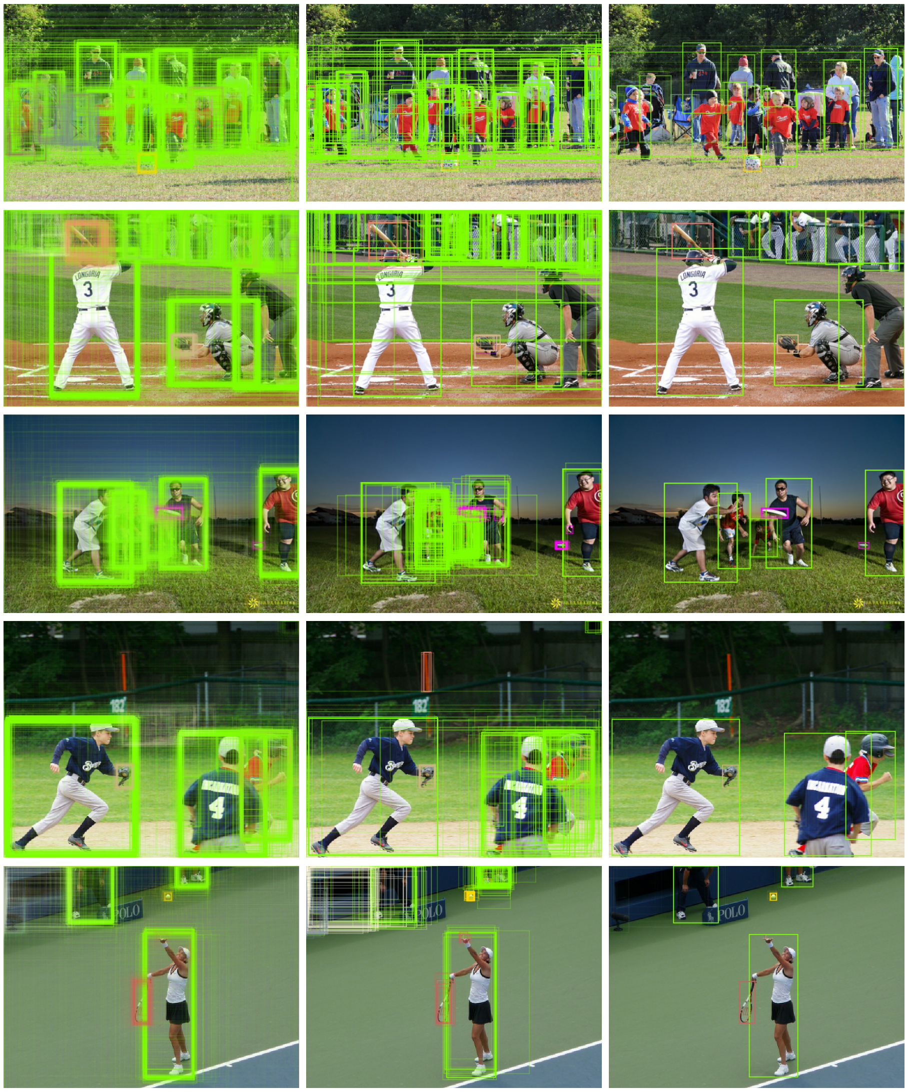
We show the effectiveness of our learning pre-filtering using two failure cases of the direct top-K sampling in Figure 3. In addition to showing the top 300 pre-filtered detections, we also show the top 300 SSD detections with highest scores in the second column that corresponds to the input of the GCN when the pre-filtering is disabled. In the first example, a football player in deep purple is highly occluded by the player in blue on the right side. Although the occluded player is detected by SSD (see the best 5000 detections), he is omitted by the direct top-K filtering but still kept by our learnable pre-filtering. The same issue can be observed in the second example even if the tennis player on the left side is not occluded. Using our learnable pre-filtering, the network re-ranks the detections, lifts the true-positives with relatively lower scores and thus keeps potential true-positives as much as possible.

C.3 Comparison with NMS

Figure 4, 5 and 6 show detection results of our approach and classical NMS for same examples on COCO *val* set. We keep the top 100 detection boxes among all categories of both networks and show the box opacity in proportion to their scores. In addition, we show ground-truths in opaque boxes with category labels for comparison. As discussed in the paper, our approach is able to suppress duplicates that have a lower IoU to the ground truth box or recover some low-scoring detections. Our approach performs especially well when objects with the same category appear together. Figure 7 shows a similar comparison on KITTI validation set that contains more crowded scenarios e.g. parking cars. In the first example, our approach is able to selects a better box with higher overlapping for the red car on the bottom right edge. Other examples also show a better performance of our approach in occlusion.

References

1. Bodla, N., Singh, B., Chellappa, R., Davis, L.S.: Soft-nms–improving object detection with one line of code. In: Proceedings of the IEEE international conference on computer vision. pp. 5561–5569 (2017)
2. Hosang, J., Benenson, R., Schiele, B.: Learning non-maximum suppression. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4507–4515 (2017)
3. Hu, H., Gu, J., Zhang, Z., Dai, J., Wei, Y.: Relation networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3588–3597 (2018)

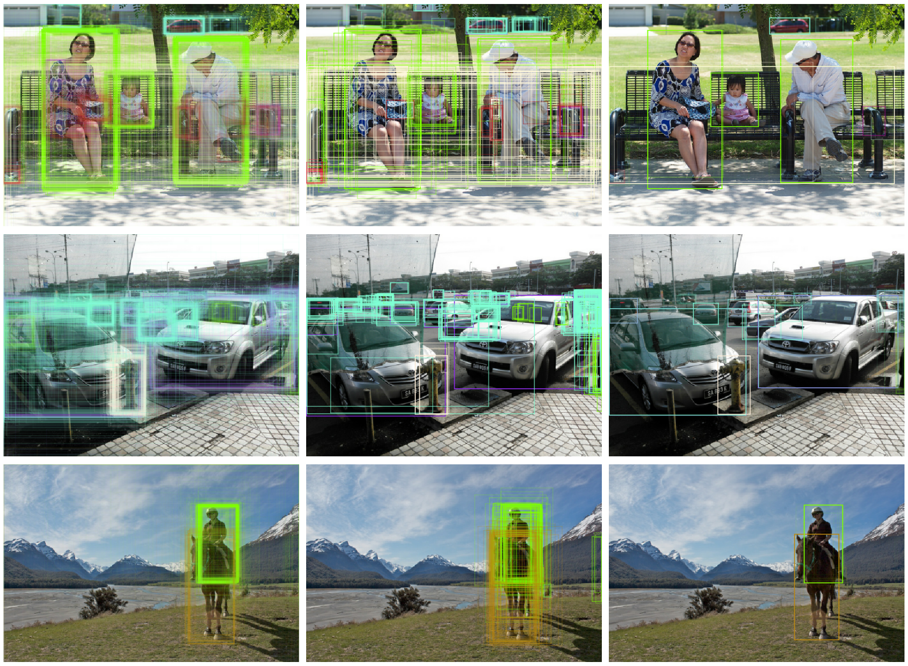


top 5000 SSD detections

top 300 pre-filtered detections

top 100 final detections

Fig. 1. Detection results from different network stages.

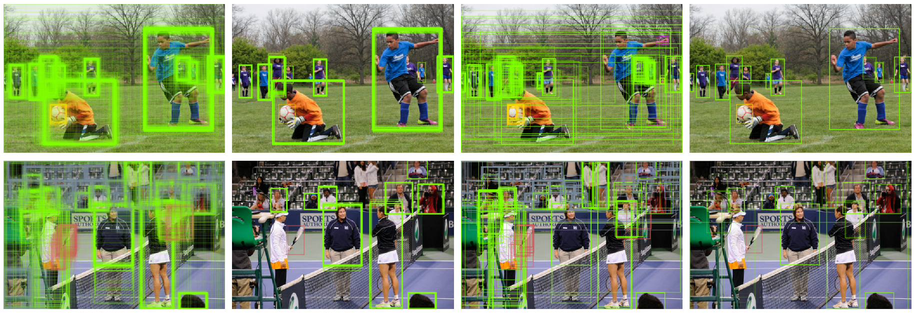


top 5000 SSD detections

top 300 pre-filtered detections

top 100 final detections

Fig. 2. Detection results from different network stages.



top 5000 SSD detections

top 300 SSD detections

top 300 pre-filtered detections

final detections (with pre-filtering)

Fig. 3. Failure cases when our learnable pre-filtering is disabled.

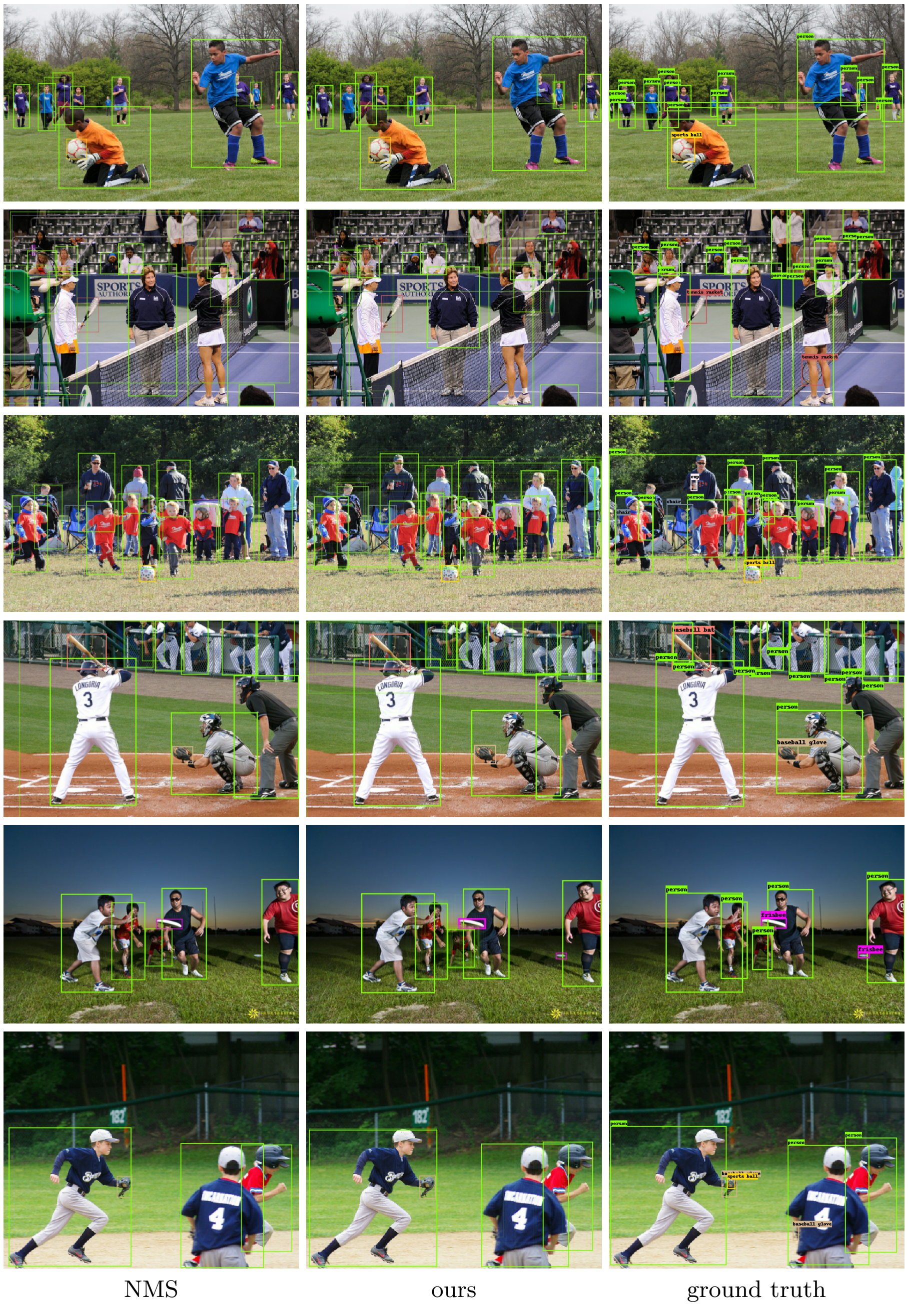


Fig. 4. Comparison with NMS and ground truth.

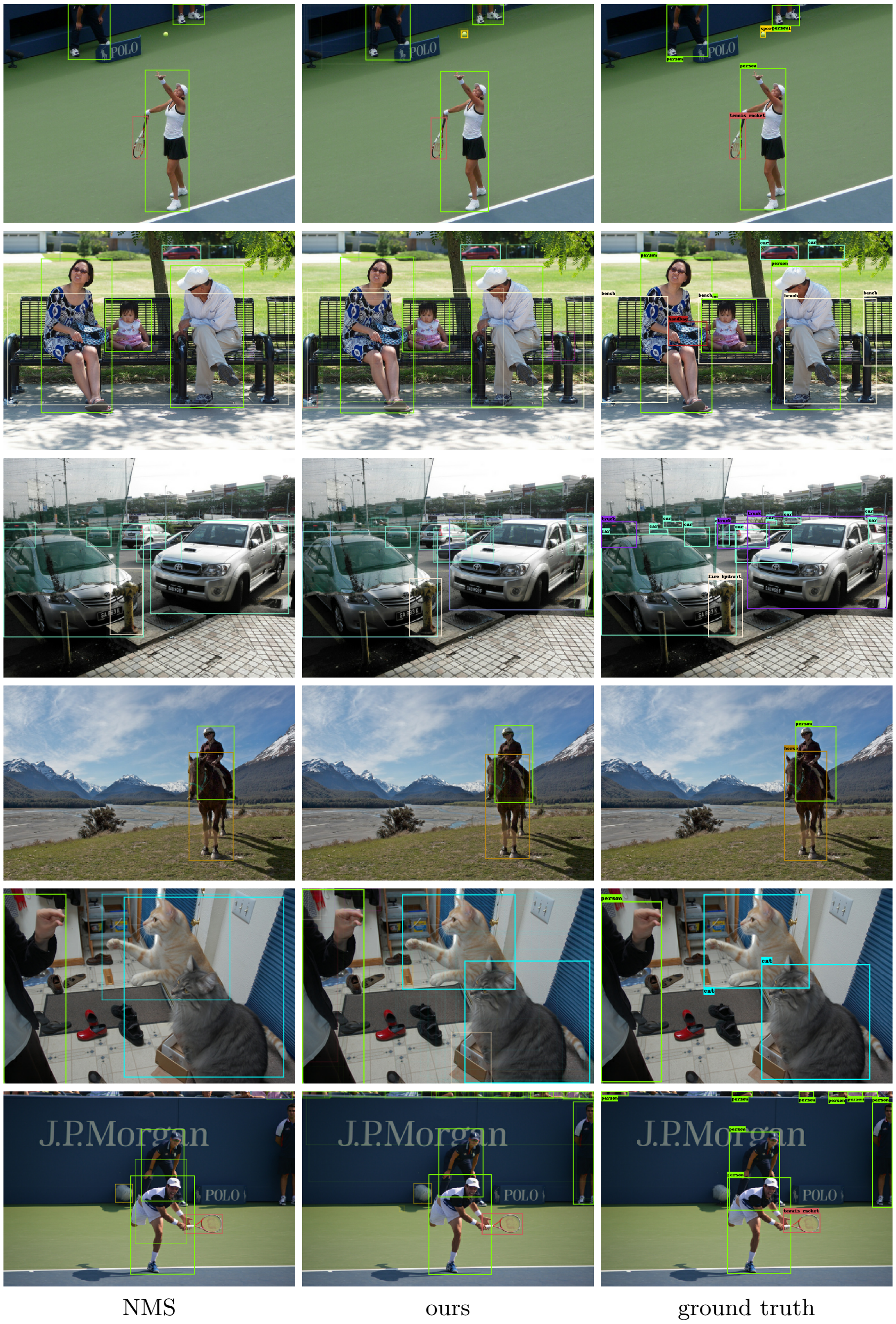
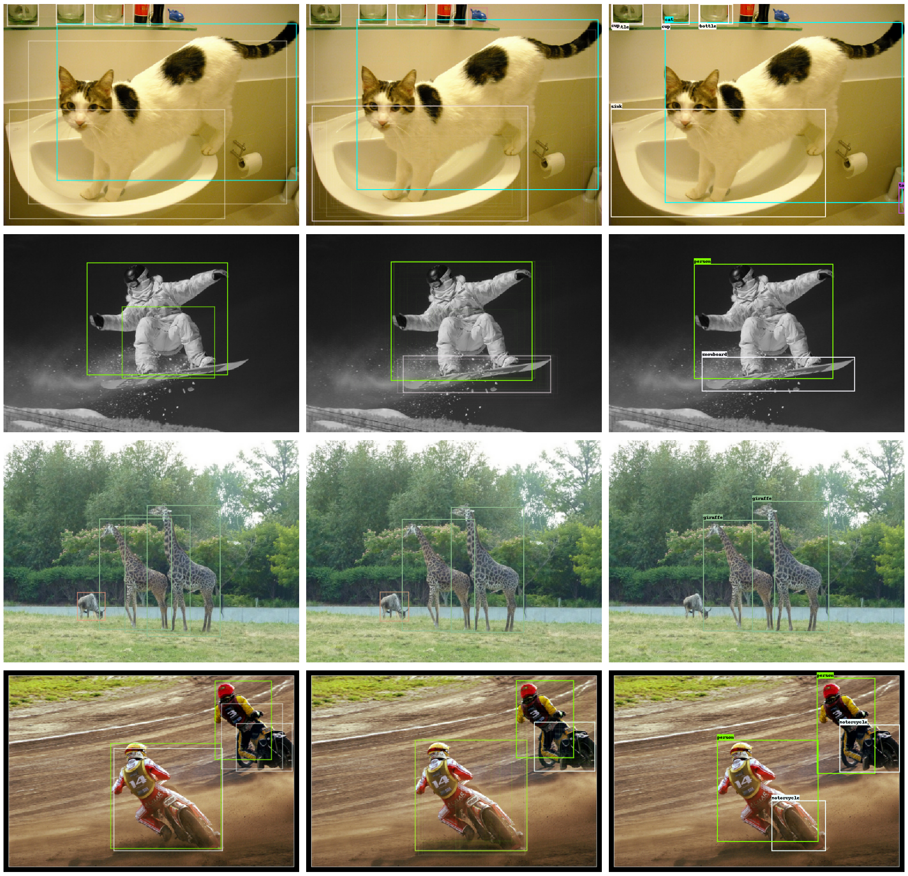
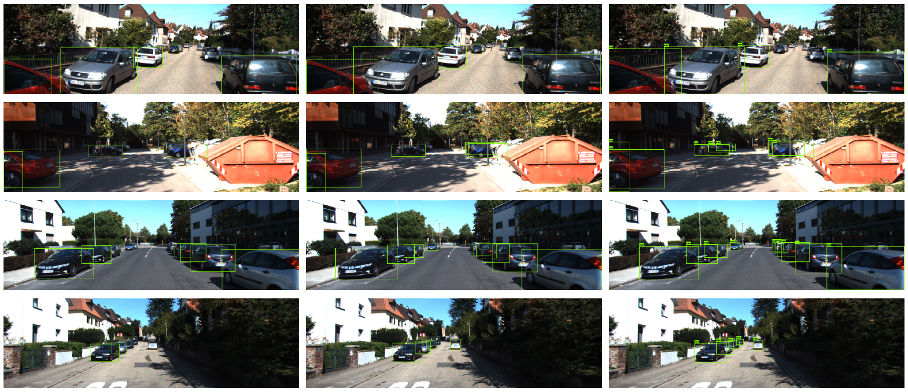


Fig. 5. Comparison with NMS and ground truth.



NMS ours ground truth

Fig. 6. Comparison with NMS and ground truth.



NMS ours ground truth

Fig. 7. Comparison with NMS and ground truth.