UnionDet Supplementary Material

Bumsoo Kim[1⋆], Taeho Choi[1∗], Jaewoo Kang[1†], and Hyunwoo J. Kim[1†]

Korea University, Seoul 02841, Republic of Korea
{meliketoy,major1965,kangj,hyunwoojkim}@korea.ac.kr

Summary. In this supplement, we provide more detailed analysis and experimental evidence that has not been provided in the main paper due to the limited space. This includes 1) Detailed explanation for our Foreground Focal Loss, 2) More discussion for the experimental settings on HICO-DET dataset, 3) More precise analysis of time vs. performance including the end-to-end time, and 4) More qualitative analysis for UnionDet.

1 Foreground Focal Loss

As introduced in the main paper, one major issue of union-level detection is that union regions often overlap over each other. This causes noisy training since a single anchor is assigned with one ground-truth label at most during training (see Fig.1). Different pairs with exactly identical union regions can be dealt with ease by simply merging the overlapping ground-truth labels. However, it is very difficult to merge all the ground-truth labels based on the portion they overlap with each other. To this end, we propose foreground focal loss to address this issue.

Reminding Target Object Classification Loss. Since it is not suitable to use only the standard IoU for Union-level Detector, we propose the novel union anchor labeling function $U_{ij} \in \{0, 1\}$ which indicates whether i_{th} ground-truth label $\breve{g}_i$ and j_{th} anchor a_j are associated or not. As briefly mentioned in the main paper, only one ground truth with the largest IoU is associated with anchor a_j when multiple ground truths are matched. We define $i^\star$ as the index of the ground-truth index associated with anchor box a_j.

$$i^\star = \underset{i}{\operatorname{argmax}}(U_{ij} \cdot \mathrm{IoU}(a_j, \breve{g}_i^{loc})). \tag{1}$$

Note that $i^\star$ is valid only in the positive anchor set $A_+ = \{a_j | \sum_i U_{ij} > 0\}$ because all of the U_{ij} are zero in negative anchor set $A_- = \{a_j | \sum_i U_{ij} = 0\}$. Based on Eq.1, the detailed union anchor labeling function U_{ij} is given as

$$U_{ij} = \begin{cases} 1 & \text{if i==}i^\star \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

⋆ equal contribution, †corresponding author

For using ground-truth index $i^\star$, the union loss function $\mathcal{L}_u(\breve{\theta})$ with target object classification loss (Eq.3 in the main paper) is rewritten as

$$\begin{aligned}\mathcal{L}_u(\breve{\theta}) &= \sum_{a_j\in A_+}\sum_{\breve{g}_i\in\breve{\mathcal{G}}} U_{ij}\left[\mathcal{L}^{act}_{ij}(\breve{\theta})+\mathcal{L}^{loc}_{ij}(\breve{\theta})+\mathcal{L}^{cls}_{ij}(\breve{\theta})\right]+\sum_{a_j\in A_-}\mathcal{L}^{bg}_{j}(\breve{\theta})\\ &= \sum_{a_j\in A_+}\left[\mathcal{L}^{act}_{i^\star j}(\breve{\theta})+\mathcal{L}^{loc}_{i^\star j}(\breve{\theta})+\mathcal{L}^{cls}_{i^\star j}(\breve{\theta})\right]+\sum_{a_j\in A_-}\mathcal{L}^{bg}_{j}(\breve{\theta}).\end{aligned} \tag{3}$$

Foreground Focal Loss in Union Branch As mentioned in our main paper, the union action classification loss is given as

$$\mathcal{L}^{act}_{i^\star j}(\breve{\theta}) = \mathrm{FL}(\breve{a}^{act}_{j}, \breve{g}^{act}_{i^\star}, \breve{\theta}), \tag{4}$$

where FL denotes focal loss [10]. Predicted union action vector in j_{th} anchor $\breve{a}^{act}_{j} \in \mathbb{R}^T$ is trained with the $i^\star_{th}$ ground-truth union action vector $\breve{g}^{act}_{i^\star} \in \mathbb{R}^T$. In Fig. 1, a single anchor box is associated with only one ground-truth box $\breve{g}_{i^\star}$ and action vector per each anchor is trained with the corresponding ground-truth vector $\breve{g}^{act}_{i^\star}$. Under ordinary focal loss, positive action labels for the ground-truth $\breve{g}_{i'}$ ($i' \neq i^\star$) are mistakenly treated as negative even though the anchor box sufficiently contains visual features of $\breve{g}_{i'}$. To solve this issue, we propose **Foreground Focal Loss** that ignores negative action labels for anchors in A_+. The final loss function in Union Branch is given as

$$\mathcal{L}_u(\breve{\theta}) = \sum_{a_j\in A_+}\left[\breve{g}^{act}_{i^\star}\cdot\mathcal{L}^{act}_{j}(\breve{\theta})+\mathcal{L}^{loc}_{j}(\breve{\theta})+\mathcal{L}^{cls}_{j}(\breve{\theta})\right]+\sum_{a_j\in A_-}\mathcal{L}^{bg}_{j}(\breve{\theta}). \tag{5}$$

2 More Analysis of our HICO-DET results.

The main issue in HICO-DET evaluation is the large amount of objects that are left unlabeled. As the evaluation metric (AP) is directly affected by the performance of the base object detector, fair evaluation is hindered when comparing architectures that leverage different object detectors (Faster-RCNN vs RetinaNet). Even though performances in the standard COCO evaluation of the two networks are comparable (37.9 vs 37.4), our one-stage dense object detector (RetinaNet) suffers from a more severe performance drop in HOI detection when used without fine-tuning. To overcome this issue, we followed the experimental setting of previous works [1] in our main paper and fine-tuned the base object detector to reduce the impact of these false-negative detection. In this section of our supplement, we show the evaluation results when eliminating this fine-tuning stage in our architecture, and provide deeper analysis related to the experimental setting in HICO-DET dataset.

Results without Fine-tuning Detectors. Fig.2 shows the false-positive predictions caused by our object detector (red boxes are the ground-truth label, blue

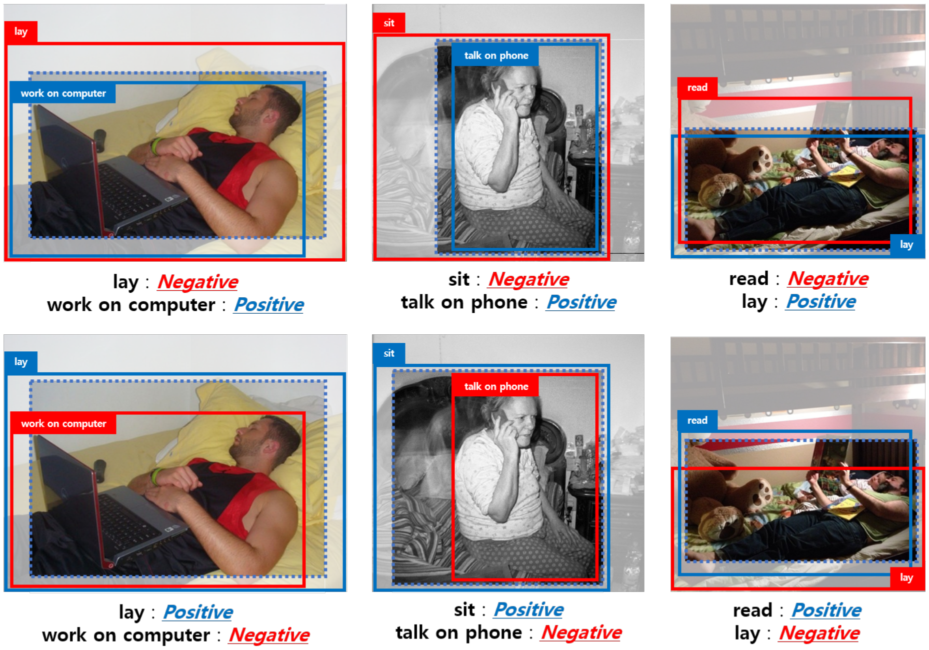

Fig. 1. Examples of overlapping union regions causing noisy training. The dotted blue box represents the anchor, the blue box represents the ground-truth box that has the highest IoU with the anchor, and the red box represents the overlapping ground-truth box that has high IoU with the corresponding anchor but not been mapped as a label. Despite the two anchors (top vs bottom) have nearly identical visual appearances, ordinary focal loss results in contradictory learning objectives (e.g. for the first row, 'lay' is trained as a negative once and as a positive once for the two anchors with almost identical visual features).

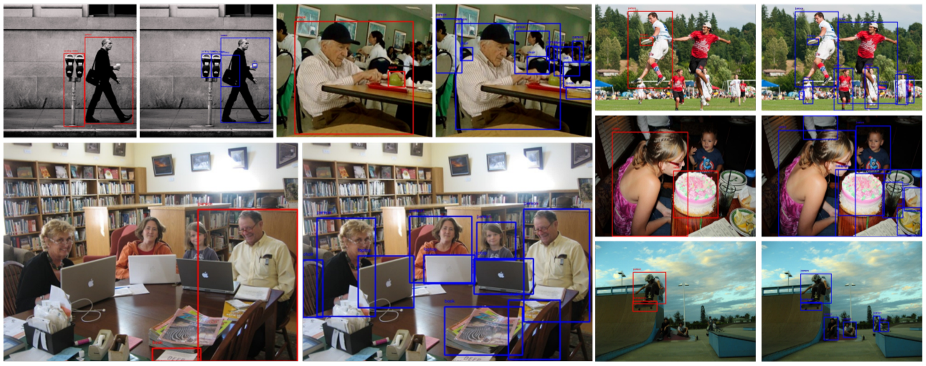

Fig. 2. Detection results on HICO-DET dataset. Red is the ground-truth. Blue is our prediction. As you can see, the loose ground-truth annotation results in many false-positive detections in this dataset.

boxes are our prediction). It can be seen that the majority of the false-positives are caused because somewhat obvious objects are not properly labeled. Since the final HOI detection performance is measured with AP, high confidence detection on these regions will eventually harm the final HOI detection performance.

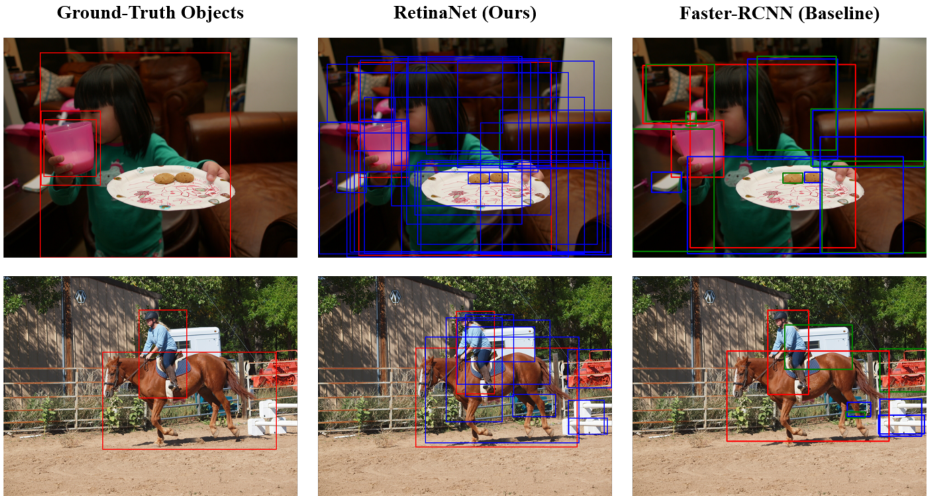

Fig. 3. Detection results on HICO-DET dataset. Red, Blue, Green each denote detection with confidence over 0.8, 0.5 and 0.2, respectively. As you can see, our base detector (RetinaNet) suffers from a notable amount of false-positive detection compared to Faster-RCNN.

Fig.3 shows that our base object detector (RetinaNet) ends up with much more false-positive predictions compared to the base detector in previous HOI detection models (Faster-RCNN). Therefore when the fine-tuning step is eliminated, our model suffers from a performance drop caused by the false-positive predictions of RetinaNet (see Table.1). The performance drops across both "Default" and "Known Object" setting. Note that we still achieve state-of-the-art performance in the "Known Object" setting despite this performance gap in the object detector.

Performance Drop in Default Evaluation. In the "Default" evaluation setting in HICO-DET (see Table.1), the performance is dependent on both the object detection results and the accuracy of interaction prediction. Note that in the default setting of HICO-DET, the more prevalent source of error comes from recognizing the objects [2]. In Table.1, it can be seen that our one-stage dense object detector suffers from a large set of false-positive sets caused by the dense prediction of RetinaNet and incomplete labeling.

Table 1. Performance and additional inference time comparison in HICO-DET. Models with † are the ones that have fine-tuned the base object detector.

		Default			**Known Object**			
Method	Ext src	Full	Rare	Non Rare	Full	Rare	Non Rare	*t(ms)*
*Models with **external** features*								
Functional Gen. [1]†	[11]	21.96	16.43	23.62	-	-	-	
Functional Gen. [1]	[11]	16.96	11.73	18.52	-	-	-	
Models with original comparison								
VSRL [7]	✗	9.09	7.02	9.71	-	-	-	-
HO-RCNN [8]	✗	7.81	5.37	8.54	10.41	8.94	10.85	-
InteractNet [6]	✗	9.94	7.16	10.77	-	-	-	55
GPNN [12]†	✗	13.11	9.41	14.23	-	-	-	40
iCAN [4]	✗	14.84	10.45	16.15	16.26	11.33	17.73	75
TIN ($\mathrm{RC_D}$) [9]	✗	13.75	10.12	15.45	15.34	10.98	17.02	70
DCA [14]	✗	16.25	11.16	17.75	17.73	12.78	19.21	130
***Ours*†**	✗	**17.58**	**11.72**	**19.33**	**19.76**	**14.68**	**21.27**	**9.06**
Ours	✗	14.25	10.23	15.46	**18.30**	**13.57**	**19.72**	**9.06**

State-of-the-art Performance in Known-Object Evaluation. In the "Known Object" setting, objects are less of a source of error. Therefore in this setting, tuning for verbs (V), not objects, gives the best result [2]. As our main challenge is to improve *interaction* detection and speed-up *interaction* prediction, improvement in the "Known Object" setting is considered significant. Table 2 in the main paper shows that our model gives the best performance in this setting: showing that our UnionDet is well-tuned to capture interaction.

3 Time vs Performance Analysis

End-to-End Inference Time. We measured inference time on a single Nvidia GTX1080Ti GPU. Our model achieved the fastest end-to-end inference time (**77.6 ms**). Since most multi-stage pipelines use different heavy networks at different stages, it causes additional latency to switch models and save/load intermediate results. Considering a real-world application on a single GPU, the gain from our approach is much bigger than the gap in end-to-end inference time analysis shown in Table 2.

iCAN, RC_D, and DCA requires object detection results from Faster-RCNN in Detectron [5] for its input. On our machine, this takes 83ms inference time *excluding* the i/o time for intermediate results. **iCAN** takes **173 ms** and **158 ms** in total that include 83 ms for object detection phase using Faster-RCNN, and feature fusion 90 ms (early) and 75 ms (late). RC_D [9] was implemented based on [4] and takes **153 ms** that includes the same object detection inference time for Faster-RCNN. **GPNN** takes **138 ms** which includes only the average graph neural network inference time 40ms and object detection time 98 ms using Deformable Convolution Network. For the models that have no official code, we compared them with the inference time reported in the literature. **InteractNet** takes **135ms** [6] and **Deep Contextual Attention (DCA)** [14] takes **213**

Table 2. Comparison of performance and ***end-to-end*** inference time on V-COCO test set. $\cdot_{\#1}$, $\cdot_{\#2}$ each refers to the performance with Scenario#1 and Scenario#2.

Method	Feature backbone	External Resources	$\mathrm{AP}_{\mathrm{role}}$	t(ms)
Models with original comparison				
VSRL [7]	ResNet50-FPN	✗	31.8	-
InteractNet [6]	ResNet50-FPN	✗	$40.0_{\#2}$	135
BAR-CNN [8]	ResNet50-FPN	✗	43.6	-
GPNN [12]	**ResNet152**	✗	44.0	138
iCAN [4]	ResNet50	✗	$44.7_{\#1}$	158
TIN (RC_{D}) [9]	ResNet50	✗	$43.2_{\#1}$	153
DCA [14]	ResNet50	✗	47.3	213
UnionDet (Ours)	ResNet50-FPN	✗	$\mathbf{47.5}_{\#1}$ $\mathbf{56.2}_{\#2}$	**77.6**

ms which is the sum of 83 ms for Faster-RCNN, and 130 ms for interaction prediction.

However, the end-to-end inference time is insufficient for precise comparison since the inference time heavily depends on the inference time of the backbone object detector (e.g., Faster-RCNN [13], Deformable Convolutional Network [3]) which can vary depending on benchmark settings (e.g., the library, CUDA, CUDNN version or hyperparameters). Therefore, our main comparison on time vs performance will be covered by the ***additional*** time for interaction prediction, excluding the time for object detection.

Additional Inference Time for Interaction. Our approach adds the minimal inference time on top of a standard object detector thanks to the parallel architecture directly detecting union bounding boxes within a one-stage detector. This enables our model to achieve minimal time for the HOI prediction phase. In Table 1, Table 2 of our main paper and Fig. 4 above, we compared the additional inference time of the HOI interaction prediction model excluding the time of the object detection phase. For InteractNet [6] that did not provide component-wise time analysis for interaction prediction in literature and has no implementation code released, we measured the additional inference time by subtracting the benchmark inference time for backbone RPN (**80 ms** [5]) from the end-to-end inference time (135 ms).

The additional interaction inference time of our model is calculated by subtracting the inference time of RetinaNet [10] from the total inference time of *UnionDet*. We present the time measure with a 640×480 scale image as baselines have done [6, 4, 9, 14], and present the mean and standard deviation for 5 runs. The final additional time for interaction prediction of ours has been calculated as the difference between 77.62±0.178 ms of our entire pipeline and 68.56±0.075 ms of our base detector *without* any interaction prediction, ***9.06***±0.193 ***ms***.

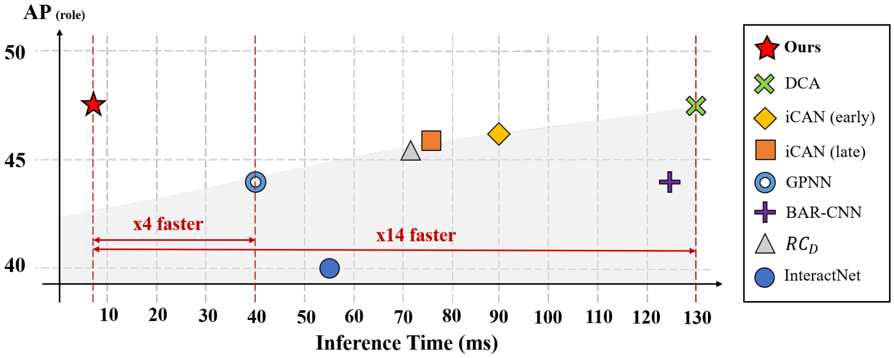

Fig. 4. Time vs Performance analysis based on additional interaction inference time.

4 More Qualitative Analysis for UnionDet

Qualitative Analysis for Target Object Classification. Fig. 5 presents more qualitative analysis to show the difference between the models that have been trained with/without our Target Object Classification loss. The red boxes show the union region prediction that matches the ground-truth interaction over a score threshold of 0.3. The first row shows visual examples of union-level detection with vanilla union detection branch, without both Anchor Labeling and Target Object Classification loss. The second row shows results of union-level detector that has been implemented with Anchor Labeling but without Target Object Classification loss. As our Anchor Labeling ensures ground-truth labels to cover a sufficient portion of the target object, the large bias towards humans is somewhat alleviated. However, it can be observed that the prediction is still suffering from the bias towards human regions: thus failing to cover the target object. The bottom row in Fig. 5 shows the results after applying our Target Object Classification Loss. We can see that our Target Object Classification loss encourages the union-level detector to capture the union region that correctly encloses the target object.

Target Object Classification loss improving HOI. Since the target object classification loss guides the union region to have a specified target, it resolves ambiguous union-level predictions. Detailed examples are shown in Fig. 6. In both subfigures, the union-level detector trained without Target Object Classification loss fails to capture the region that tightly encloses both the *human* and *target object*. Even if the IoU is substantial, the Union Matching function μ_u appears to be relatively low if either the target object or human is not properly enclosed within the predicted area, which leads to a low μ_u. Our Target Object Classification loss enables the final union-level detector of our UnionDet to capture tight regions that have enhanced matching scores with the correct $\langle human, object\rangle$ pair with interaction. This leads to an improvement in our final inference score for the appropriate $\langle human, object \rangle$ pair, and thus enables more accurate HOI detection.

Target Object Classification loss Suppressing False-Positives. Another delightful property of our Target Object Classification loss is that it eventually suppresses false-positive union-level detections that don't properly include the $\langle human, object \rangle$ pair with interaction (boxes in orange in Fig. 9). In the first and third columns in Fig. 9, the vanilla detector captures region that does not correctly contain the *human* region. In the second and fourth columns of Fig. 9, the Target Object Classification loss guides the union-level detector that captures excessive areas to focus on the actual union area with the target object.

More Union-level Detection in Complex Scenes. In Fig. 8 we present a few more qualitative analysis on union-level detection results in complex scenes that includes multiple unions that has the same target object or multiple unions overlapping over each other. In the left and right subfigure of Fig. 8, more than one person is interacting with the same object (soccer ball, frisbee). Our union-level detector successfully captures the precise union region of the interaction between each $\langle human, object \rangle$ pair. In the middle subfigure of Fig. 8, you can see that our union-level detector successfully captured both union regions of $\langle human - look - frisbee \rangle$ and $\langle human - look - human \rangle$.

More Results in UnionDet Capturing Remote Target Objects. Here, we present more qualitative analysis on union-level detection results that have remote target objects. As mentioned in our main paper, union-level detection gets more difficult as the target object of the interaction becomes small and remote. In Figure 9, more examples of our UnionDet successfully capturing remote and small target objects with the correct interaction is presented.

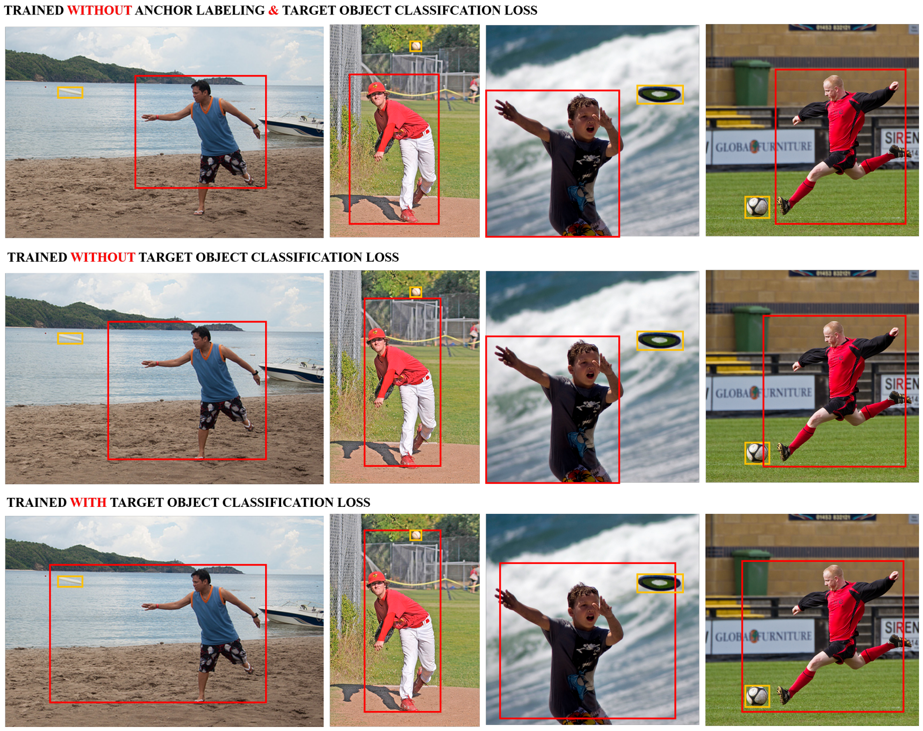
TRAINED WITHOUT ANCHOR LABELING & TARGET OBJECT CLASSIFCATION LOSS
TRAINED WITHOUT TARGET OBJECT CLASSIFICATION LOSS
TRAINED WITH TARGET OBJECT CLASSIFICATION LOSS

Fig. 5. Target Object Classification loss successfully guiding union-level detectors to enclose target objects. Predicted union-level region with the correct interaction type in red, and target object in yellow.

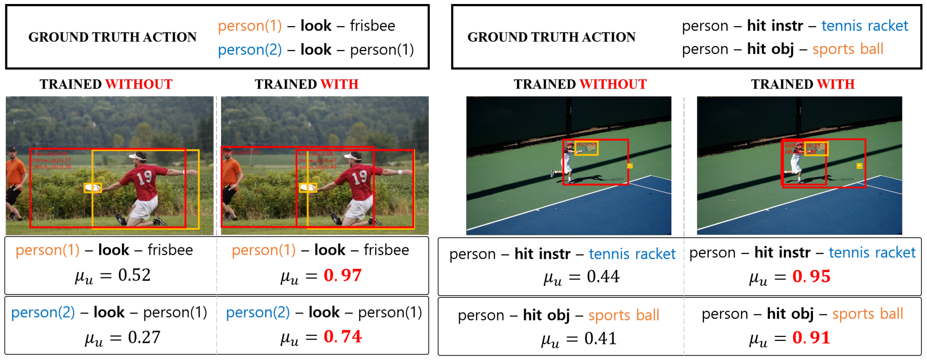
GROUND TRUTH ACTION
person(1) – look – frisbee
person(2) – look – person(1)
GROUND TRUTH ACTION
person – hit instr – tennis racket
person – hit obj – sports ball
TRAINED WITHOUT
TRAINED WITH
TRAINED WITHOUT
TRAINED WITH
person(1) – look – frisbee
μ_u = 0.52
person(1) – look – frisbee
μ_u = 0.97
person – hit instr – tennis racket
μ_u = 0.44
person – hit instr – tennis racket
μ_u = 0.95
person(2) – look – person(1)
μ_u = 0.27
person(2) – look – person(1)
μ_u = 0.74
person – hit obj – sports ball
μ_u = 0.41
person – hit obj – sports ball
μ_u = 0.91

Fig. 6. Target Object Classification loss successfully resolving complicated union regions with overlap. The left and right image for each section is the union-level detection by UnionDet trained without/with Target Object Classification loss, respectively. As the Target Object Classification resolves ambiguous union regions, the Union Matching Score μ_u gets improved for the correct $\langle human, object \rangle$ pair of interaction. Predicted union-level region with the correct interaction type in red, and target object in yellow.

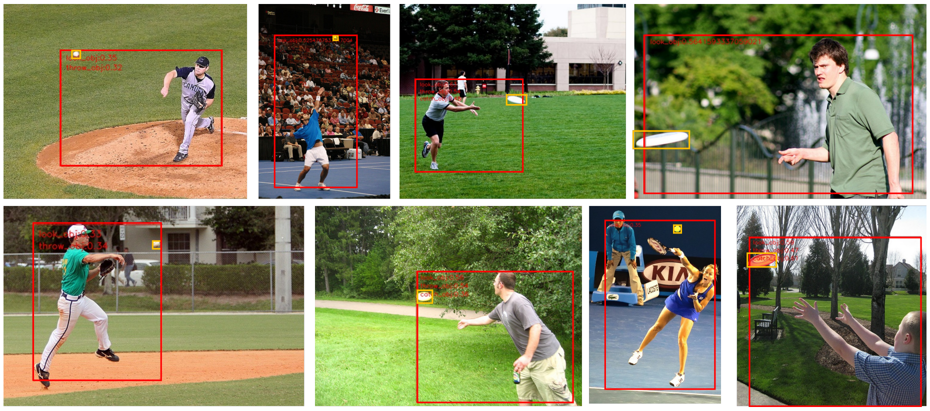

Fig. 7. Union-level detection successfully capturing small and remote target objects. Predicted union-level region with the correct interaction type in red, and target object in yellow.

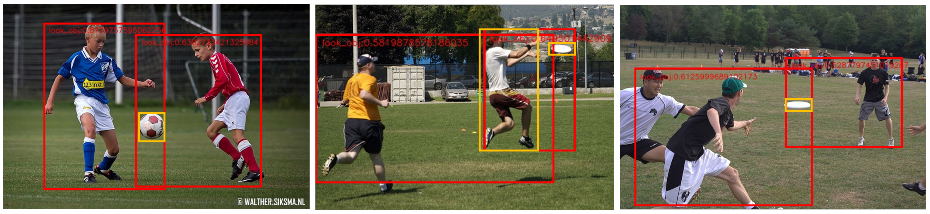

Fig. 8. Union-level detection working in complex scenes where more than one ground-truth union regions involves the same target object or unions that overlap over each other. The target object is in yellow.

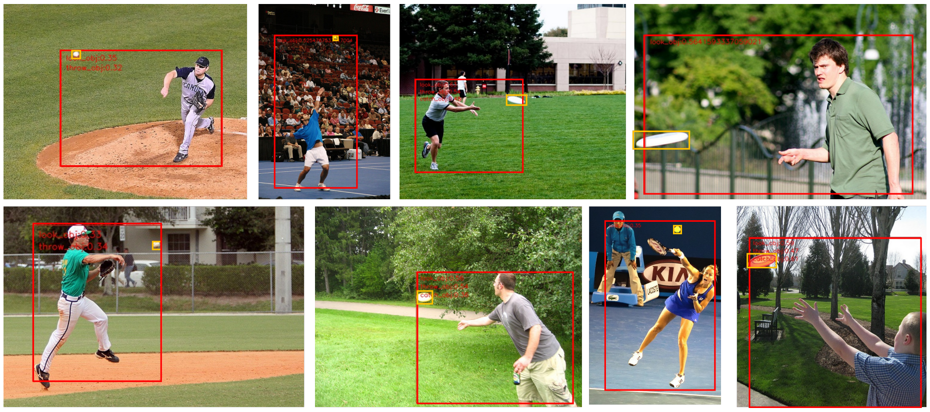

Fig. 9. Union-level detection successfully capturing small and remote target objects. Predicted union-level region with the correct interaction type in red, and target object in yellow.

References

1. Bansal, A., Rambhatla, S.S., Shrivastava, A., Chellappa, R.: Detecting human-object interactions via functional generalization. In: AAAI. pp. 10460–10469 (2020)
2. Chao, Y.W., Wang, Z., He, Y., Wang, J., Deng, J.: Hico: A benchmark for recognizing human-object interactions in images. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1017–1025 (2015)
3. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 764–773 (2017)
4. Gao, C., Zou, Y., Huang, J.B.: ican: Instance-centric attention network for human-object interaction detection. arXiv preprint arXiv:1808.10437 (2018)
5. Girshick, R., Radosavovic, I., Gkioxari, G., Dollár, P., He, K.: Detectron. https://github.com/facebookresearch/detectron (2018)
6. Gkioxari, G., Girshick, R., Dollár, P., He, K.: Detecting and recognizing human-object interactions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8359–8367 (2018)
7. Gupta, S., Malik, J.: Visual semantic role labeling. arXiv preprint arXiv:1505.04474 (2015)
8. Kolesnikov, A., Kuznetsova, A., Lampert, C., Ferrari, V.: Detecting visual relationships using box attention. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 0–0 (2019)
9. Li, Y.L., Zhou, S., Huang, X., Xu, L., Ma, Z., Fang, H.S., Wang, Y., Lu, C.: Transferable interactiveness knowledge for human-object interaction detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3585–3594 (2019)
10. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
11. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
12. Qi, S., Wang, W., Jia, B., Shen, J., Zhu, S.C.: Learning human-object interactions by graph parsing neural networks. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 401–417 (2018)
13. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
14. Wang, T., Anwer, R.M., Khan, M.H., Khan, F.S., Pang, Y., Shao, L., Laaksonen, J.: Deep contextual attention for human-object interaction detection. arXiv preprint arXiv:1910.07721 (2019)