Supplemental Materials for "SIOD: Single Instance Annotated Per Category Per Image for Object Detection "

Hanjun Li[1]*, Xingjia Pan[2], Ke Yan [2]†, Fan Tang [3], Wei-Shi Zheng[1,4,5] †
[1]School of Computer Science and Engineering, Sun Yat-sen University
[2]Youtu Lab, Tencent [3]Jilin University [4]Peng Cheng Laboratory
[5]Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education
lihj85@mail2.sysu.edu.cn, {xjia.pan,tfan.108}@gmail.com,kerwinyan@tencent.com, wszheng@ieee.org

1. Meaning of Score-aware Detection Evaluation Protocol

Detector	Task	AP(%)
CenterNet-Res18	FSOD	28.1
	SIOD	25.1
CenterNet-Res101	FSOD	34.2
	SIOD	27.8

Table 1. The changes of AP from FSOD to SIOD task with CenterNet framework. The performance is evaluated on COCO2017-Val.

In this section, we dive into analyzing the defect of COCO style evaluation protocol when it is applied to SIOD task. We first evaluate the performance of detector trained on FSOD and SIOD task, respectively. As shown in Table 1, it seems that the detector still performs well on SIOD task, although only 40% instance annotations are preserved compared with FSOD task. Actually, the discriminative ability of two detectors (*e.g*. CenterNet-Res18 trained on FSOD task or SIOD task) is still significantly different. We first visualize the detected bounding boxes with score threshold 0.3, as shown in Fig. 1 column (a) and column (d). Few objects are detected when the detector is trained on SIOD task. As we decrease the score threshold, an increasing number of boxes are shown(*e.g*. SIOD(base)@$\mathbb{S}_1$ and SIOD(base)@$\mathbb{S}_2$). Obviously, SIOD(base) can achieve comparable performance with FSOD regardless of the score(confidence). Since official COCO evaluation protocol determines a true match without considering the predicted scores, a large number of detected bounding boxes with low scores are recalled (similar to Fig. 1 SIOD(base)@$\mathbb{S}_1$). In this way, it results in illusory advances on SIOD task. In order to distinguish the ability of scoring between two different detectors, we propose a Score-aware Detection Evaluation Protocol, which introduces a score constraint to the match rule of official COCO evaluation protocol. In this way, we can measure the performance of different detectors across different score thresholds. Undoubtedly, a perfect detector is expected to detect objects with high scores. The proposed evaluation protocol exactly is capable to measure such ability.

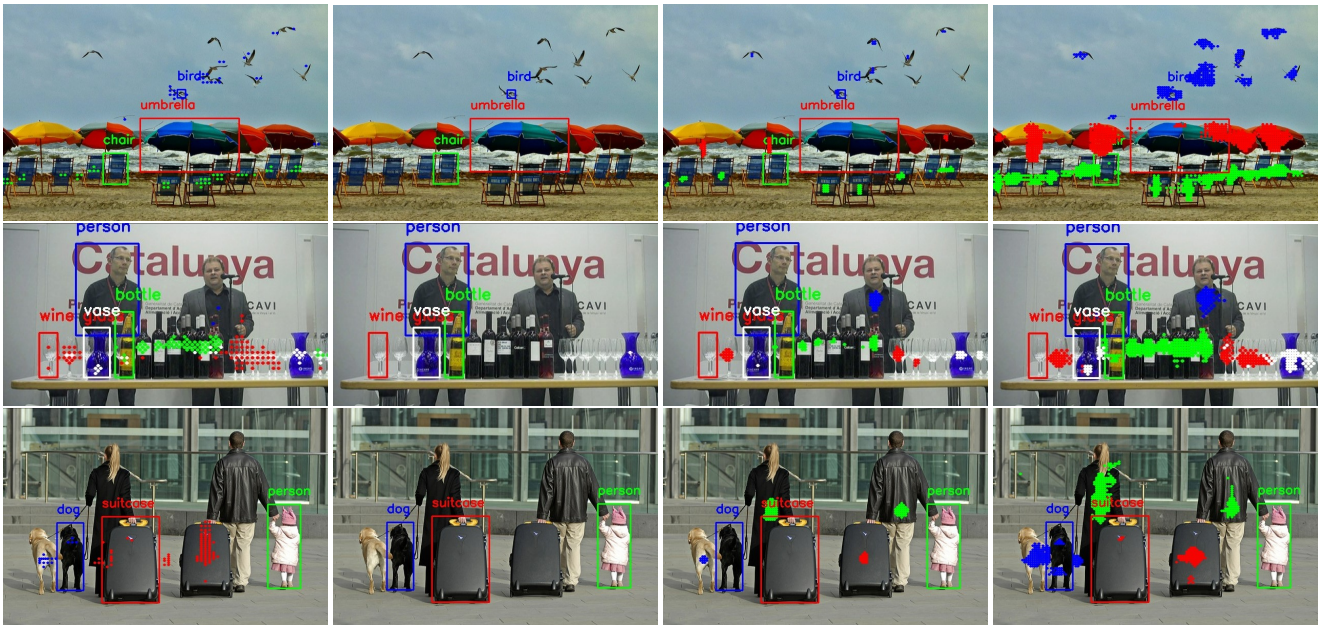
*Work partially done during the Youtu Lab internship
†Corresponding author

2. Visualization for SPLG and PGCL

In this section, we try to visualize the pseudo labels generated by the proposed Similarity-based Pseudo Label Generating module (SPLG). Note that all of positions with target values less than 1.0 are treated as penalty-reduced backgrounds as shown in main manuscript Eq.(5). We therefore visualize those high-quality positions which have large similarity with reference instances. As shown in Fig. 2 SPLG@$\mathbb{S}_8$, a large number of instances are assigned pseudo class labels correctly and some instances (*e.g*. umbrellas and birds) are ignored. However, none of instances have similarity with reference instances larger than 0.9 (SPLG@$\mathbb{S}_9$) . As for Pixel-level Group Contrastive Learning (PGCL), we select top-m positions as positive samples according to self-predicted scores. As shown in Fig. 2 PGCL, most of positions located at the center of unlabeled instances are selected and some instances are not selected due to the limited sampling. PGCL tends to minimize the distance between positive pairs and push away the negative pairs in embedding space, which undoubtedly facilitates mining more unlabeled instances in SPLG module. After integrated with PGCL, high-quality pseudo labels are generated with SPLG module, as shown in Fig. 2 SPLG_PGCL@$\mathbb{S}_9$. As an increasing number of unlabeled instances are mined for training, the performance of the detector is improved naturally.

(a) FSOD@$\mathbb{S}_3$ (b) SIOD(base)@$\mathbb{S}_1$ (c) SIOD(base)@$\mathbb{S}_2$ (d) SIOD(base)@$\mathbb{S}_3$

Figure 1. Visualization of FSOD and SIOD(base) with CenterNet-Res18 across different score thresholds. Note that SIOD(base) denotes directly training the detector on SIOD task and $\mathbb{S}_i$ denotes the score threshold is $i/10$.



(a) SPLG@$\mathbb{S}_8$ (b) SPLG@$\mathbb{S}_9$ (c) PGCL (d) SPLG_PGCL@$\mathbb{S}_9$

Figure 2. Visualization of pseudo labels generated by SPLG(column(a),(b) and (d)) and top-m positions selected by PGCL. Note that S_i denotes the score threshold is $i/10$. All images are selected from the Keep1-COCO2017-Train and the preserved instances are drawn with the bounding boxes. The color of each dot denotes its according pseudo category label.

3. Visualization for Faster-RCNN and FCOS

Both Faster-RCNN and FCOS are confronted with large performance degradation when applying them to SIOD task, since most of unlabeled instances are treated as backgrounds mistakenly. After equipped with the proposed DMiner, they achieve better performance as reported in main manuscript. In this section, we visualize the de-

(a) FSOD (b) SIOD(base) (c) SIOD(DMiner)

Figure 3. Visualization of Faster-RCNN-Res50-C4 on different tasks with score threshold 0.5. Note that SIOD(base) denotes directly training the detector on SIOD task and SIOD(DMiner) denotes that the detector is equipped with DMiner.

Task	#instances	instances/image	instances/image/category	time(seconds)
FSOD	36419	7.28	0.091(0.058)	81.32
SIOD	14674	2.93	0.037(0.004)	38.14

Table 2. Comparison of annotated cost between FSOD and SIOD task with 5000 images randomly selected from COCO2017-Train. Note that the #instance denotes the total number of instances to be annotated. "instances/image/category" denotes that the average number of instances for each category per image and (*) is according variance. The time cost is the average annotating time of single images.

tected results for clear comparison. As shown in Fig. 3, SIOD(base) fails to detect those small objects (*e.g.* books, pedestrians) while SIOD(DMiner) locates them successfully. As for FCOS, the detector equipped with DMiner also achieves obvious advance compared with SIOD(base) as shown in Fig. 4.

4. Comparison of Annotated Cost

Although about 60% instance annotations are reduced under the SIOD setup compared with FSOD on COCO2017, it is still unable to directly reflect the difficulty of annotating instances between SIOD and FSOD task. We therefore conduct a practical annotating experiment to obtain real statistics of annotated cost. We first randomly select 5000 images from COCO2017-Train. The detailed information is reported in Table 3. Then six professional female annotators are divided into two groups. One is asked to annotate all instances for FSOD task and another is asked to annotate one instance for each existing category in each image for SIOD task. Note that the average age of them is about 23. Additionally, they annotate the whole samples independently and we finally compute the average annotating time among the group for each task. As shown in Table 2,

(a) FSOD (b) SIOD(base) (c) SIOD(DMiner)

Figure 4. Visualization of FCOS-RCNN-Res50-FPN on different tasks with score threshold 0.5. Note that SIOD(base) denotes directly training the detector on SIOD task and SIOD(DMiner) denotes that the detector is equipped with DMiner.

only about 40% (14674/36419) instances are needed to be annotated in 5000 sampled images for SIOD task, which is consistent with the whole dataset. More specifically, it reduces about 53.1% annotating time per image under the SIOD setup compared with FSOD, which demonstrates that SIOD setup has large potential to practically reduce the annotated cost for object detection.

References

category	#instances	sample_ratio	keep_ratio	category	#instances	sample_ratio	keep_ratio
person	257253	0.042	0.252	bicycle	7056	0.040	0.474
car	43533	0.041	0.289	motorcycle	8654	0.034	0.498
airplane	5129	0.045	0.586	bus	6061	0.043	0.681
train	4570	0.040	0.800	truck	9970	0.046	0.598
boat	10576	0.045	0.300	traffic light	12842	0.044	0.330
fire hydrant	1865	0.041	0.908	stop sign	1983	0.037	0.890
parking meter	1283	0.018	0.826	bench	9820	0.041	0.585
bird	10542	0.045	0.255	cat	4766	0.040	0.837
dog	5500	0.052	0.704	horse	6567	0.049	0.443
sheep	9223	0.048	0.186	cow	8014	0.053	0.222
elephant	5484	0.035	0.424	bear	1294	0.035	0.733
zebra	5269	0.042	0.315	giraffe	5128	0.036	0.505
backpack	8714	0.040	0.609	umbrella	11265	0.045	0.340
handbag	12342	0.042	0.554	tie	6448	0.037	0.637
suitcase	6112	0.040	0.393	frisbee	2681	0.035	0.926
skis	6623	0.038	0.472	snowboard	2681	0.037	0.626
sports ball	6299	0.043	0.725	kite	8802	0.042	0.287
baseball bat	3273	0.043	0.810	baseball glove	3747	0.043	0.679
skateboard	5536	0.053	0.577	surfboard	6095	0.038	0.611
tennis racket	4807	0.040	0.782	bottle	24070	0.043	0.354
wine glass	7839	0.044	0.314	cup	20574	0.046	0.429
fork	5474	0.045	0.587	knife	7760	0.049	0.524
spoon	6159	0.045	0.564	bowl	14323	0.044	0.524
banana	9195	0.049	0.245	apple	5776	0.035	0.325
sandwich	4356	0.045	0.526	orange	6302	0.034	0.292
broccoli	7261	0.041	0.271	carrot	7758	0.039	0.237
hot dog	2884	0.037	0.444	pizza	5807	0.041	0.540
donut	7005	0.047	0.212	cake	6296	0.030	0.516
chair	38073	0.050	0.300	couch	5779	0.042	0.736
potted plant	8631	0.053	0.479	bed	4192	0.036	0.854
dining table	15695	0.046	0.719	toilet	4149	0.041	0.859
tv	5803	0.044	0.795	laptop	4960	0.040	0.719
mouse	2261	0.046	0.790	remote	5700	0.045	0.521
keyboard	2854	0.052	0.728	cell phone	6422	0.045	0.691
microwave	1672	0.035	0.966	oven	3334	0.046	0.890
toaster	225	0.040	0.889	sink	5609	0.044	0.829
refrigerator	2634	0.045	0.915	book	24077	0.041	0.227
clock	6320	0.048	0.721	vase	6577	0.038	0.553
scissors	1464	0.045	0.576	teddy bear	4729	0.032	0.523
hair drier	198	0.061	1.000	toothbrush	1945	0.047	0.418

Table 3. The detailed information of 5000 sampled images. #instances is the total number of instances for each category in COCO2017-Train. The sample_ratio denotes the proportion of instances w.r.t #instances in 5000 sampled images and the average of sample_ratio is 0.04, which is nearly same as the sampling ratio(5000/117316). The keep_ratio denotes the proportion of instances to be annotated in 5000 sampled images under the SIOD setup and the average of keep_ratio is 0.57.