Revisiting Class Imbalance for End-to-end Semi-Supervised Object Detection Supplementary Material

Purbayan Kar, Vishal Chudasama, Naoyuki Onoe, Pankaj Wasnik*
Media Analysis Group, Sony Research India, Bangalore, India
{purbayan.kar, vishal.chudasama1, naoyuki.onoe, pankaj.wasnik}@sony.com

Section 1 of this supplemental material provides the details of data augmentation process, and Section 2 describes details of additional ablation studies. The convergence analysis of our proposed network is discussed in Section 3, followed by an additional analysis of results in Section 4.

1. Data augmentation

The proposed Efficient teaching network follows the data augmentation guidelines given by STAC [5] and FixMatch [4]. Here, we generate weak and strong augmented samples that have been used to train teacher and student networks. To synthesize weak augmented samples, we exert a random resize process followed by a random horizontal flip operation. While for the generation of strong augmented samples, we add complex augmentation operations like auto contrast, random equalize, color jitter, random translate, random rotate, random shear and random erase. The summary of the step-wise process of producing the weak and strong augmented samples is described in Table 1.

2. Ablation analysis

In the main manuscript (i.e., subsection 4.5), we presented an ablation study conducted on the MS-COCO based 10% partially labeled data setting. In this section, we provided the additional ablation analysis for the 1% and 5% partially labeled data settings of MS-COCO dataset.

2.1. Effect of loss functions:

To inspect the significance of the introduced novel losses (i.e., Equation no. 10 and 11 from main manuscript), the proposed network is trained without the introduced losses (Case I), with only background similarity loss (Case II), and with only foreground-background dissimilarity loss (Case III). The corresponding results for 1% and 5% partially labeled data settings are shown in Table 2. It can be seen here that both introduced losses aid the proposed network in acquiring better mAP measures. Furthermore, the proposed network employs the supervised losses on teacher and student network outcomes. Similar characteristics can also be seen in Table 2, where the supervised losses from both teacher and student networks can assist the proposed network in obtaining improved mAP results.

Table 1. The summary of the used data augmentation process.

Steps.	Process	Parameter	Details
Weak Augmentation			
1.	RandomResize	—	Resize the input image to the given size
2.	RandomFlip	flip ratio = 0.5	Flip the given image randomly with a given probability.
Strong Augmentation			
1.	RandomResize	—	Resize the input image to the given size
2.	RandomFlip	flip ratio = 0.5	Flip the given image randomly with a given probability.
3. (Any one of these)	AutoContrast	prob. = 0.25	Autocontrast the pixels of the given image randomly with a given probability.
	RandEqualize	prob. = 0.25	Equalize the histogram of the given image randomly with a given probability.
	RandSolarize	prob. = 0.25	Solarize image randomly with given probability by inverting all pixel values above a threshold.
	Color Jitter	brightness, contrast, hue, saturation = 0.4, 0.4, 0.1, 0.4	Randomly change brightness, contrast, saturation, and hue of an image. Brightness, contrast and saturation factor is chosen from [0.6, 1.4] while the hue factor is chosen from [-0.1, 0.1].
	RandContrast	prob. = 0.25	Randomly choose the contrast of the given image with a given probability.
	RandBrightness	prob. = 0.25	Randomly choose brightness of given image with given probability.
	RandSharpness	prob. = 0.25	Adjust the sharpness of the image randomly with a given probability.
	RandPosterize	prob. = 0.25	Posterize the image randomly with a given probability by reducing the number of bits for each color channel.
4. (Any one of these)	RandTranslate	scale = (-0.1, 0.1)	Random translate operation of image keeping center invariant.
	RandRotate	scale = $(-30^{o}, 30^{o})$	Rotate the image by angle
	RandShear	scale = $(-30^{o}, 30^{o})$	Random shear operation of image keeping center invariant.
5.	RandErase	size = [0, 0.2]	Randomly selects rectangle region in image and erases its pixels.

*Pankaj Wasnik is the corresponding author.

Table 2. Ablation analysis on different losses.

Network	mAP		mAP@50		mAP@75	
	1%	5%	1%	5%	1%	5%
Effectiveness of the introduced losses for classification task						
Case 1: $L_{fg}^{cls} + L_{bg}^{cls}$	25.7	31.7	44.1	51.5	27.4	33.9
Case 2: $L_{fg}^{cls} + L_{bg}^{cls} + L_{bg}^{sim}$	26.0	32.0	44.4	51.9	27.7	34.3
Case 3: $L_{fg}^{cls} + L_{bg}^{cls} + L_{fg-bg}^{dissim}$	26.0	31.9	44.3	51.8	27.6	34.2
Proposed: $L_{fg}^{cls} + L_{bg}^{cls} + L_{bg}^{sim} + L_{fg-bg}^{dissim}$	**26.3**	**32.2**	**44.6**	**52.1**	**28.0**	**34.6**
Effectiveness of supervised losses						
Proposed: (Both supervised losses)	**26.3**	**32.2**	**44.6**	**52.1**	**28.0**	**34.6**
- w/o teacher model based supervised loss	26.0	31.8	44.2	51.6	27.6	34.2
- w/o student model based supervised loss	25.7	31.5	43.9	51.2	27.4	34.0
- w/o both supervised losses	1.1	1.5	1.5	2.0	1.4	1.7

Table 3. Ablation analysis to check importance of label generator module.

Network	mAP		mAP@50		mAP@75	
	1%	5%	1%	5%	1%	5%
Proposed (Both label generators)	**26.3**	**32.3**	**44.6**	**52.1**	**28.0**	**34.6**
- w/o label generator (weak augmented data)	23.9	31.1	42.1	49.7	25.9	32.5
- w/o label generator (strong augmented data)	25.1	31.3	43.8	51.0	27.1	33.3

Table 4. Ablation analysis of introduced DEMA update mechanism.

Network	mAP		mAP@50		mAP@75	
	1%	5%	1%	5%	1%	5%
Deep copy	20.1	26.1	38.3	46.0	20.7	26.9
EMA	25.7	31.2	43.7	51.3	27.3	33.6
DEMA	**26.3**	**32.3**	**44.6**	**52.1**	**28.0**	**34.6**

2.2. Importance of label generator module:

Few ablation experiments have been conducted to assess the importance of two label generators associated with weak and strong augmented samples. The corresponding results are presented in Table 3, where it can be seen that the proposed network with both label generators outperforms the individual label generator settings.

2.3. Effect of introduced update mechanism:

We ablate the proposed network to validate the introduced Double Exponential Moving Average (DEMA) update mechanism. For that, the proposed network is also trained using the Exponential Moving Average (EMA) as well as with deep copy configuration (i.e., the weights of teacher network are copied from the student network). The corresponding results are shown in Table 4, where it can be observed that the DEMA mechanism obtains +0.8% and 1.1% higher mAP measures on 1% and 5% labeled data settings, respectively that proves its efficacy over the EMA update mechanism.

Table 5. Ablation analysis on hyper-parameters.

(a) Effect of different γ values

γ	mAP		mAP@50		mAP@75	
	1%	5%	1%	5%	1%	5%
0.03	26.0	31.8	44.3	51.7	27.4	34.0
0.04	26.2	32.1	44.5	51.9	27.6	34.2
0.05	**26.3**	**32.3**	**44.6**	**52.1**	**28.0**	**34.6**
0.06	26.1	31.9	44.4	51.8	27.5	34.1

(b) Effect of different N_{jitter} values

N_{jitter}	mAP		mAP@50		mAP@75	
	1%	5%	1%	5%	1%	5%
5	26.0	32.0	44.4	51.8	27.5	34.1
10	**26.3**	**32.3**	**44.6**	**52.1**	**28.0**	**34.6**
15	26.2	32.2	44.6	52.0	27.7	34.4
20	26.1	32.1	44.5	51.9	27.7	34.3

2.4. Tuning of parameters:

For tuning, we study the proposed network's two parameters called γ and N_{jitter}. Table 5 shows the impact of different values of these parameters and the best values are highlighted in bold font. One can observe that the $\gamma = 0.05$ perceives highest mAP. Similarly, $N_{jitter} = 10$ gives best mAP results than other N_{jitter} values. Hence, we choose these parameter setting in our experiments.

2.5. Importance of Jitter Bagging module:

The ablation analysis of the Jitter Bagging module is presented in Table 6, where we can see that the proposed Jitter Bagging module achieves the highest performance and shows +1.2% absolute improvement in mAP measure over

Table 6. Ablation analysis to check importance of proposed Jitter Bagging module.

Network	mAP		mAP@50		mAP@75	
	1%	5%	1%	5%	1%	5%
Proposed	**26.3**	**32.3**	**44.6**	**52.1**	**28.0**	**34.6**
- w/o Jitter Bagging	25.1	31.1	44.0	52.5	26.7	33.2
- with Box Jittering [6]	25.7	31.7	44.2	52.6	27.1	33.7

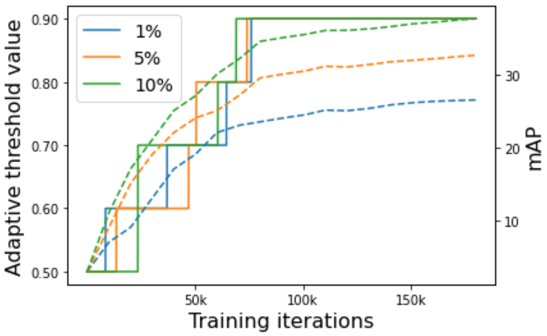


Figure 1. Analysis of adaptive threshold value and corresponding mAP measures during training iterations. Here, dashed line indicates the mAP measures while the straight line indicate the threshold value.

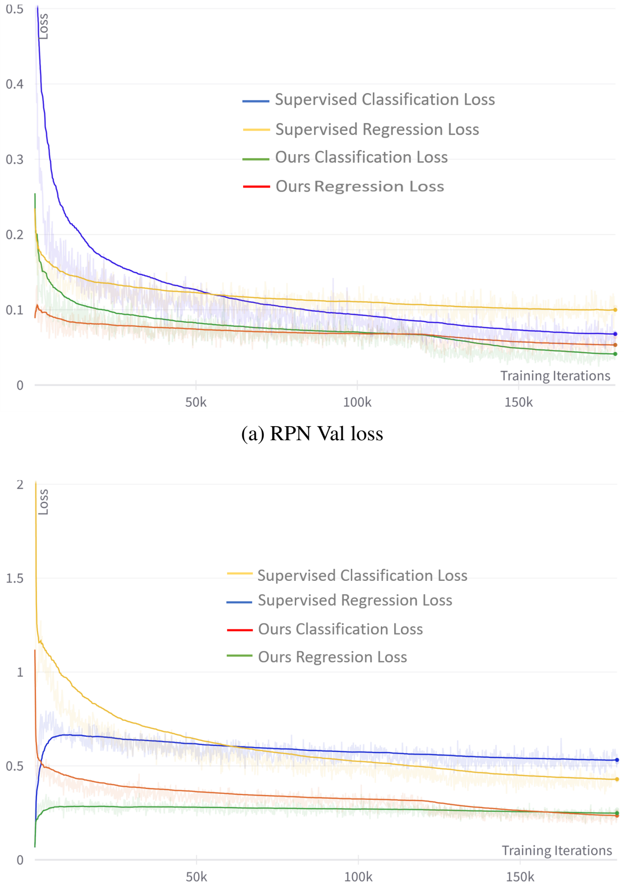
Table 7. Effectiveness of adaptive threshold value over different threshold measures on partially labeled data settings.

Proposed Network	mAP		mAP@50		mAP@75	
	1%	5%	1%	5%	1%	5%
with static 0.7 threshold	22.6	28.3	40.1	48.7	24.1	30.4
with static 0.8 threshold	25.7	31.3	43.1	50.5	26.6	32.4
with static 0.9 threshold	26.0	32.0	43.9	51.4	27.3	33.2
with dynamic thresholding [1]	26.1	31.8	44.0	51.6	27.5	33.8
with continuous form-based threshold	25.5	31.4	43.2	51.0	26.9	33.4
with proposed adaptive threshold	**26.3**	**32.2**	**44.6**	**52.1**	**28.0**	**34.6**

without Jitter Bagging module in 1% and 5% partially labeled data setting. Interestingly, when we employ the Box Jittering [6] in our network, we observe that the proposed Jitter Bagging still obtains +0.4% higher mAP than the Box Jittering on 1% and 5% labeled data setting.

2.6. Effectiveness of adaptive threshold filter:

Our proposed network introduces an adaptive threshold filter that adjusts the threshold value based on generated background/foreground bounding boxes. We have observed the effect of threshold values during the training process. Figure 1 shows this analysis for partially labeled data settings (i.e., 1%, 5% and 10% labeled data). In addition, the corresponding effect on the mAP value is also depicted in Figure 1 (i.e., highlighted with dashed lines for 1%, 5% and 10%). However, to observe the performance over the static threshold value i.e., 0.9 as used in the Soft Teacher model [6], few experiments have been performed where the proposed network with different static threshold values are trained. Table 7 presents the corresponding results where we can see that in case of mAP, the proposed adaptive threshold filter performs marginally better than the static threshold = 0.9. However, in the case of mAP@50 and mAP@75 it achieves better results with a good margin. Also, the threshold mechanism differs from that of introduced by Li *et al.* [1]. To check its effectiveness over the threshold module proposed by Li *et al.* [1], we employed their thresholding module in our framework. The corresponding results are added in Table 7, which is marginally inferior to the proposed thresholding module. In our adaptive mechanism, we have used discrete thresholding to reduce fluctuations in the threshold value, Further, we trained a variant of our model with a continuous form of threshold and found lower performance than proposed discrete form.



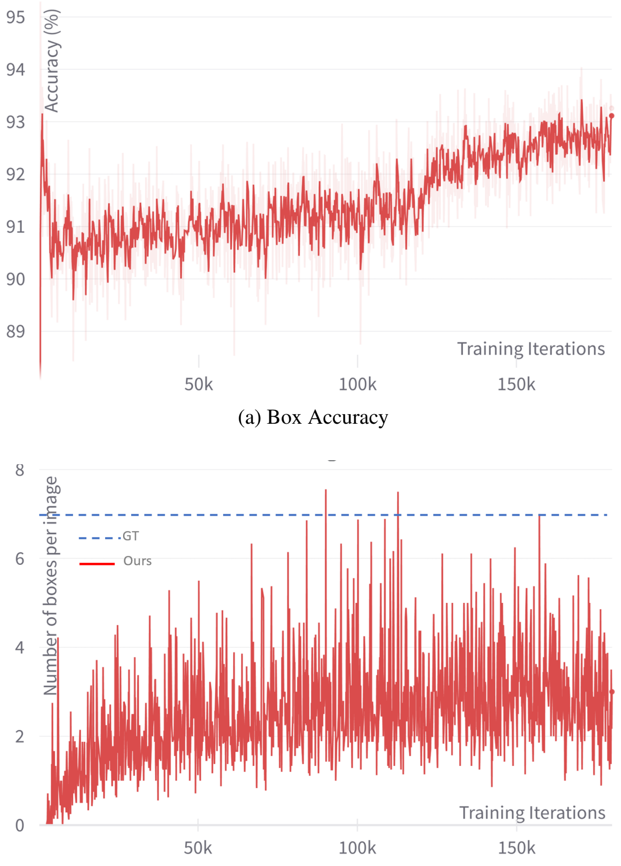
(a) RPN Val loss

(b) RoIhead Val loss

Figure 2. Comparison of validation loss plots for 10% labeled training data.

3. Convergence Analysis

In this section, we empirically analyze the convergence of our proposed efficient teaching network. Here we ob-



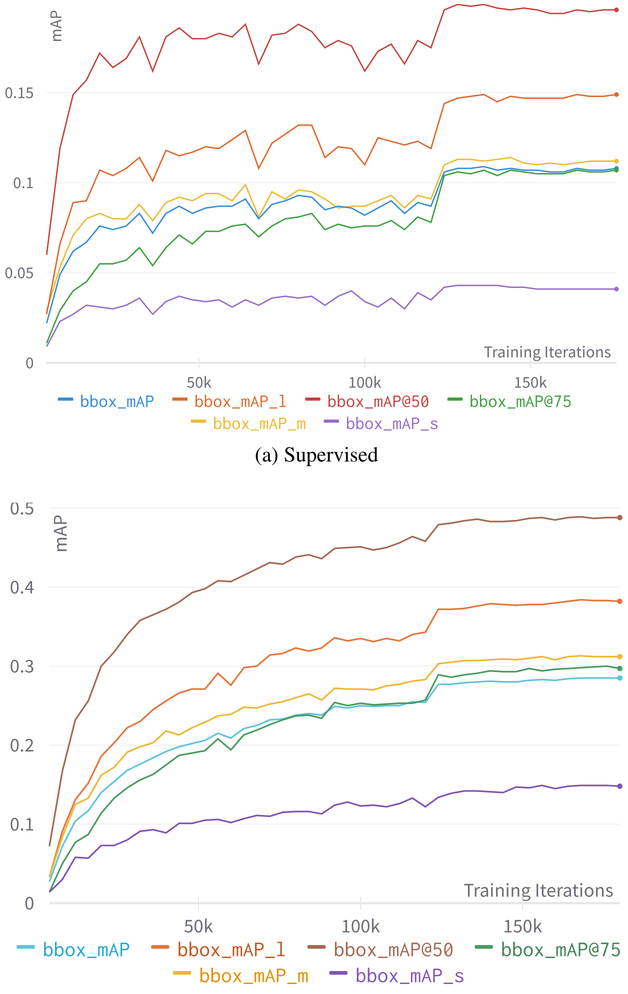
(a) Box Accuracy

(b) Number of Pseudo boxes

Figure 3. Improvement analysis of Pseudo-label in terms of (a) accuracy, and (b) number of bounding boxes.

serve the classification and regression loss values of supervised and our proposed networks when 10% labeled data is used during training. We have used the Faster R-CNN [3] as our default detector, where the Region Proposal Network (RPN) generates the proposals to predict the possibilities of an object to either be in the foreground or background. Using these data, the Region over Interest pooling coupled with a classifier and regressor head (RoIhead) network classifies what is in the proposals and determines the bounding box size. Both the networks are trained using the classification and regression loss functions and the corresponding loss values are noted. The changes in the loss values obtained from the MS-COCO validation dataset [2] are shown as plot graphs in Figure 2. Here, one can see that the proposed network achieves better convergence than the supervised network for all protocols.

We have also observed improvement in pseudo-label box accuracy and the number of pseudo boxes during the training process. Figure 3 shows these improvements for the case of 10% labeled data. Here, we measure the accuracy by comparing the generated pseudo-boxes with ground-truth boxes and the corresponding analysis is presented in Figure 3(a). We also observe the improvement in several pseudo-boxes during the training process, which is illustrated in Figure 3(b). Here, GT indicates the average number of bounding boxes in the ground-truth labels (i.e., seven bounding boxes per image).



(a) Supervised

(b) Ours

Figure 4. Comparison of validation mAP plots for 1% labeled training data.

Table 8. Analysis of different ResNet backbones for fully labeled data.

Network	Supervised		Our	
	ResNet50	ResNet101	ResNet50	ResNet101
mAP@50	57.6	64.8	65.2	68.0
mAP@75	40.4	47.8	48.1	51.5
mAP	37.9	42.7	44.0	46.8

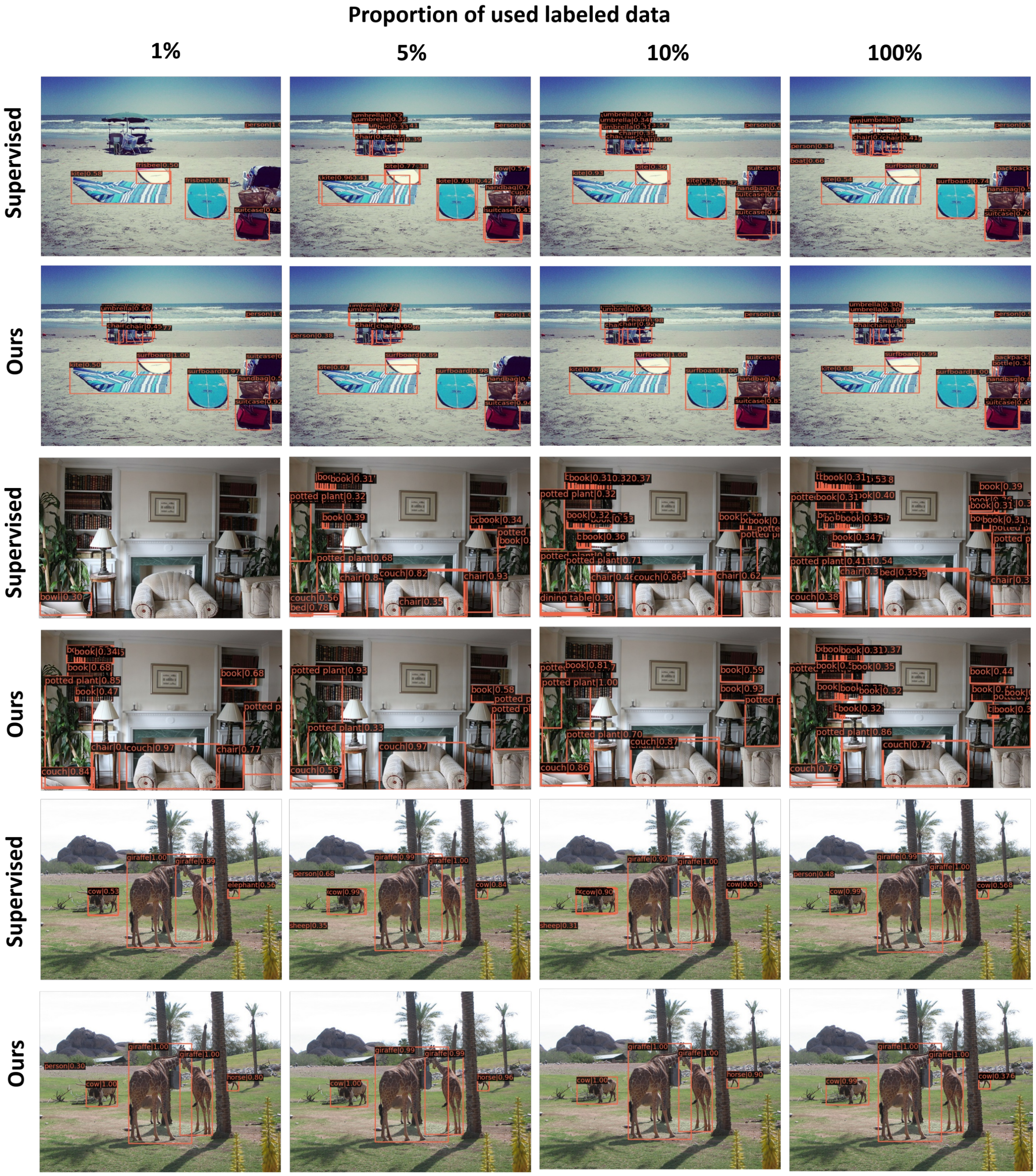


Figure 5. Visual comparison with supervised results.

4. Result analysis

In this section, we provide the results of the proposed network along with its supervised network on MS-COCO dataset based partially labeled and fully labeled data settings. Figure 4 shows the changes in mAP measures obtained from validation dataset during the training of the supervised and proposed networks. For a fair comparison with existing methods, we use ResNet50 as a backbone in the Faster-RCNN detector. Therefore, we have also monitored the effect of more complex networks like ResNet101 in the supervised baseline and proposed frameworks. The corresponding results are illustrated in Table 8 where it can be noticed that the ResNet101-based networks acquire higher mAP values than that of ResNet50-based networks.

In addition to inspection of the main manuscript, we

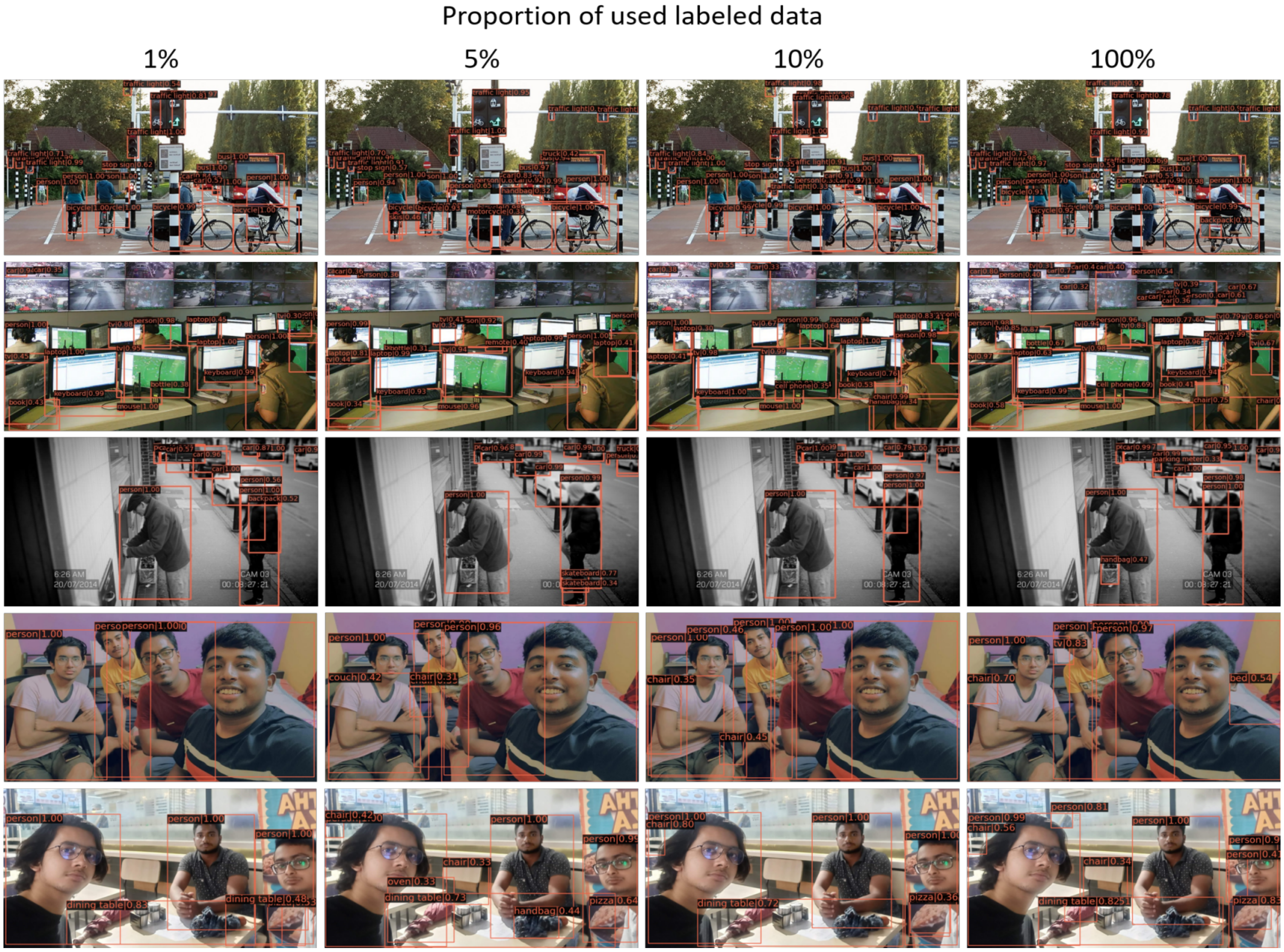


Figure 6. Visual analysis on in-the-wild images: 1^{st} and 2^{nd} row images are taken from online, 3^{rd} row image represents footage taken from CCTV camera and last two row images are captured using mobile camera.

have appended additional qualitative comparisons in this section. Here, we compare the outcomes of the supervised and our proposed networks for a different proportion of labeled data. This can be visualized in Figure 5. Here, it is analyzed that the proposed network detects the objects with a better confidence score than the supervised framework. We further analyzed the behavior of the proposed network with unseen data by performing object detection on a few in-the-wild images. The corresponding qualitative results are presented in Figure 6.

References

[1] Hengduo Li, Zuxuan Wu, Abhinav Shrivastava, and Larry S Davis. Rethinking pseudo labels for semi-supervised object detection. *arXiv preprint arXiv:2106.00168*, 2021.

[2] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[3] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[4] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33:596–608, 2020.

[5] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. *arXiv preprint arXiv:2005.04757*, 2020.

[6] Mengde Xu, Zheng Zhang, Han Hu, Jianfeng Wang, Lijuan Wang, Fangyun Wei, Xiang Bai, and Zicheng Liu. End-to-end semi-supervised object detection with soft teacher. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3060–3069, 2021.