Supplementary for
CanKD: Cross-Attention-based Non-local operation for Feature-based Knowledge Distillation

Shizhe Sun, Wataru Ohyama
Tokyo Denki University
Tokyo, Japan
w.ohyama@mail.dendai.ac.jp

1. Details about training strategy

In this paper, we verify the superior performance of CanKD by conducting experiments with various model architectures on both object detection and semantic segmentation tasks. We train all models on NVIDIA RTX 6000 Ada 48GB with 2 GPUs. In the object detection task, following the guidelines provided by the mmdetection[3] documentation and adhering to the linear scaling rule[7], we set the learning rate to 0.005. Meanwhile, following the mmdetection 2× training strategy, we employ a multi-step learning rate decay at epochs 16 and 22 with a decay factor of 0.1. In the semantic segmentation task, following the mmsegmentation[4] official training strategy of an 80K schedule, we set the learning rate to 0.01 with a weight decay of 0.0005 and use a polynomial function to decay our learning rate. The number of parameters and flops in the segmentation task results are all calculated by *thop*.

2. Details about affinity function

In this section, we provide a detailed analysis of two alternative affinity functions that deviate from our original method: the Gaussian method[1, 17] and the Embedding Gaussian method[18].

2.1. Gaussian affinity function

From the perspective of early non-local methods, directly processing two feature maps with a Gaussian function represented a straightforward choice. We omit two 1×1 modules from θ and ϕ, and directly compute the affinity between the student's feature map and the teacher's feature map. The Gaussian affinity function is determined by:

$$\xi(\boldsymbol{x}_i, \boldsymbol{y}_j) = \exp(\boldsymbol{x}_i^\top \boldsymbol{y}_j). \tag{1}$$

where, $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ represent the feature vectors at the position i in the student feature map and the position j in the teacher feature map, respectively. The normalization factor for the Gaussian affinity function is $C = \sum_j \xi(\boldsymbol{x}_i, \boldsymbol{y}_j)$. In terms of implementation, this operation can be easily carried out by applying a *Softmax* layer.

2.2. Embedded Gaussian affinity function

Analogous to self-attention in Transformers, the Embedded Gaussian method introduces an embedding space. Specifically, the student and teacher feature maps are projected into the embedding space via two 1×1 convolutional layers, denoted as θ and ϕ. The affinity is then calculated using a Gaussian function, as shown in equation (2).

$$\xi(\boldsymbol{x}_i, \boldsymbol{y}_j) = \exp\left(\theta(\boldsymbol{x}_i)^\top \phi(\boldsymbol{y}_j)\right). \tag{2}$$

The normalization factor for the embedded Gaussian affinity function is also $C = \sum_j \xi(\boldsymbol{x}_i, \boldsymbol{y}_j)$.

3. Natural corrupted augmentation analysis

Following the [13], we evaluate the student RetinaNet-R50 detector, which CanKD trained in the COCO-C dataset. The COCO-C dataset is derived from the COCO validation dataset by applying four types of corruption, i.e., transformations—noise, blurring, weather, and digital corruption, to evaluate the model's robustness. Each corruption category consists of multiple corruption methods, each with six levels of severity. The test results are summarized in Table 1. The results show that CanKD exhibits substantially higher robustness than the other methods. Specifically, it achieves a 2.2 improvement in mPC and a 2.7 improvement in rPC compared to the benchmark.

4. Analysis about maxpooling layer

In this section, we examine whether different scales of the teacher feature maps after max pooling influence the performance of the CanKD. We experimented with max pooling at 4×4 and 8×8 scales. It is important to note that while reducing the pooling scale can facilitate the student

Table 1. **Result of robust object detection via CanKD on COCO-C dataset.**

Method	$AP_{\texttt{clean}}$	mPC	rPC
RetinaNet-R50	37.4	18.3	48.9
FGD	39.6	20.3	51.3
DiffKD	39.7	20.3	51.1
CanKD	**39.8**	**20.5**	**51.6**

Table 2. **Ablation study on maxpooling scales.** We use RepPoints-X101[19] as teacher and RepPoints-R50 as student.

Scaled	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
2×2	**42.4**	**62.9**	**45.6**	**24.1**	46.5	**56.4**
4×4	42.0	62.3	45.5	24.0	46.2	55.0
8×8	42.1	62.3	45.6	23.9	**46.7**	55.7

Table 3. **Ablation study on residual connection.** We use RepPoints-X101[19] as teacher and RepPoints-R50 as student.

Scaled	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
W/O residual connection	41.4	61.8	44.6	24.4	45.6	54.3
W/ residual connection	**42.4**	**62.9**	**45.6**	**24.1**	**46.5**	**56.4**

model's more comprehensive acquisition of the teacher feature maps, it simultaneously increases the memory footprint on the hardware. The experimental results are presented in Table.2.

From the results, we observe that employing a large-scale max pooling operation eliminates certain crucial information within the teacher feature maps. This reduction hampers the student feature maps from fully assimilating the teacher's knowledge. Meanwhile, we suggest that employing a max pooling layer to select critical information from the teacher feature maps may not be the optimal choice. A more refined strategy could better preserve essential information while reducing the spatial resolution of the teacher feature maps.

5. Analysis about residual connection

In this section, we examine the role of the residual connection within the Can module, which is in function 5 in the original paper. We believe that directly comparing the feature map produced by the attention component of the Can module with the teacher's original feature map would cause issues because we do not apply any additional operations to the teacher's feature map. Therefore, we retain the residual connection so that the student's original feature map remains involved. The ablation study is in table.3.

According to the experimental results, CanKD with a residual connection outperforms CanKD without a residual connection by a significant margin. Therefore, we conclude that the residual connection is indispensable in CanKD.

6. Confusion matrix

Here, we present the confusion matrices for the teacher models, the student models, and our proposed CanKD method. Here, Figure 1 is from the FasterRCNN-R50 [14] student model, and Figure 2 is from the FasterRCNN-R50 model distilled with CanKD.

7. Detail for training

By analyzing the output logs from each model, we generated the changes in mAP for each model throughout the training process. The figure is shown in Figure.3.

8. Visualization

In Figure 5, we present a series of visualization figures generated from the student model, CanKD, and the teacher model on the COCO val 2017 dataset[12] to demonstrate the performance and effectiveness of our method in object detection. Meanwhile, in Figure 4, we also present a series of visualization figures generated from the student model, CanKD, and the teacher model on the Cityscapes validation dataset[5]. In Figure 6, we present the heatmaps from different FPN layers for the student model, the distilled student model, and the teacher model. Compared to the original student model, these examples demonstrate that our method has better performance and effectiveness.

9. Experiments on classification task.

To complement the evaluation of CanKD on fundamental tasks, following the official schedule, we conducted experiments on the ImageNet-1K [6] dataset with two teacher–student model pairs, ResNet-34→ResNet-18, ResNet-50→MobileNetv1 in 100e. We choose the last layer output in backbone as our distillation position. The results are shown in table.4. The balanced weight μ in CanKD are all set to 5. However, CanKD is specifically designed for dense prediction tasks, where pixel-level feature alignment is critical. In contrast, classification tasks do not involve pixel-level alignment; therefore, the use of cross-attention may be unnecessary compared to traditional logit distillation methods that already achieve strong results. Meanwhile, CanKD outperforms w/o student, KD [9], and AT [23]. This demonstrates that the performance gains of CanKD in dense prediction do not come at the expense of classification performance. The classification heat map images are shown in Figure 7 which generated by *Grad-CAM* [15].

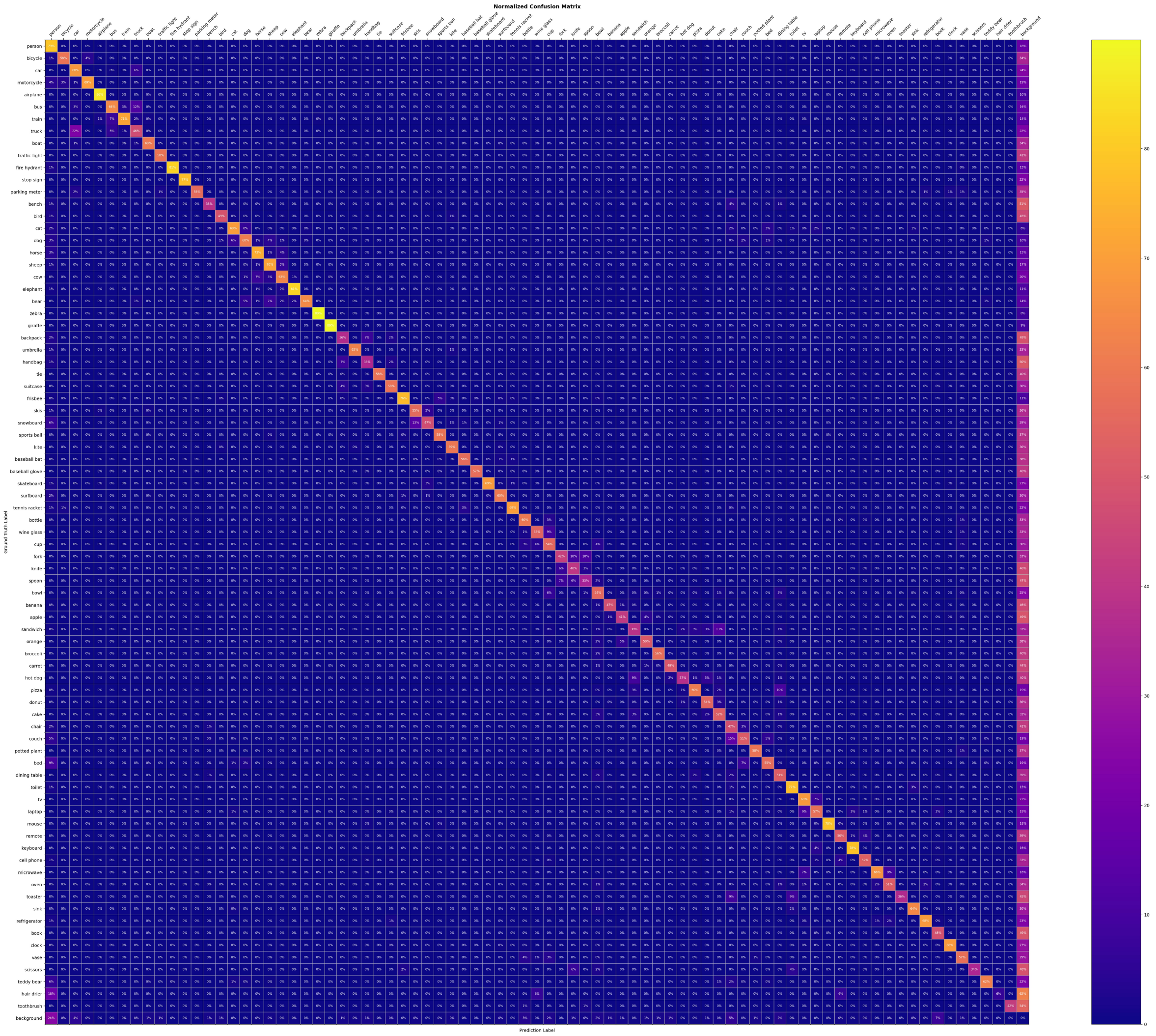

Figure 1. **Confusion matrix from FasterRCNN-R50.**

Table 4. **Classification task on ImageNet-1K.** We use ResNet-34, ResNet-50 [8] as teachers and Resnet-18, MobileNet-v1 [10] students. All baselines results are from [21].

Method	$Top1$	$Top5$	Method	$Top1$	$Top5$
T:ResNet-34	73.62	91.59	T:ResNet-50	76.55	93.06
T:ResNet-18	69.90	89.43	S:MobileNetv1	69.21	89.02
KD [9]	70.68	**90.16**	KD [9]	70.68	**90.30**
AT [23]	70.59	89.73	AT [23]	70.72	90.03
CanKD	**70.73**	89.81	CanKD	**70.89**	89.90

10. Analysis about parameter and computational complexity

In this section, we analyze the parameter count and computational complexity of CanKD. We compare its parameters and FLOPs with several strong feature distillation methods in table.5, and our results show that CanKD has fewer parameters and lower complexity than existing attention-based and mask-based distillation methods. Moreover, since CanKD is applied only during training, it does not affect the parameter count or complexity of the student model at inference time, further demonstrating that CanKD is a lightweight yet effective attention-based distillation approach.

11. Limitation of CanKD

Attention-based distillation methods share a common drawback: when the teacher and student feature maps differ in spatial resolution (H×W), performance degradation occurs. We demonstrate this phenomenon using a simple teacher–student pair in Table.6. Furthermore, from the dis-

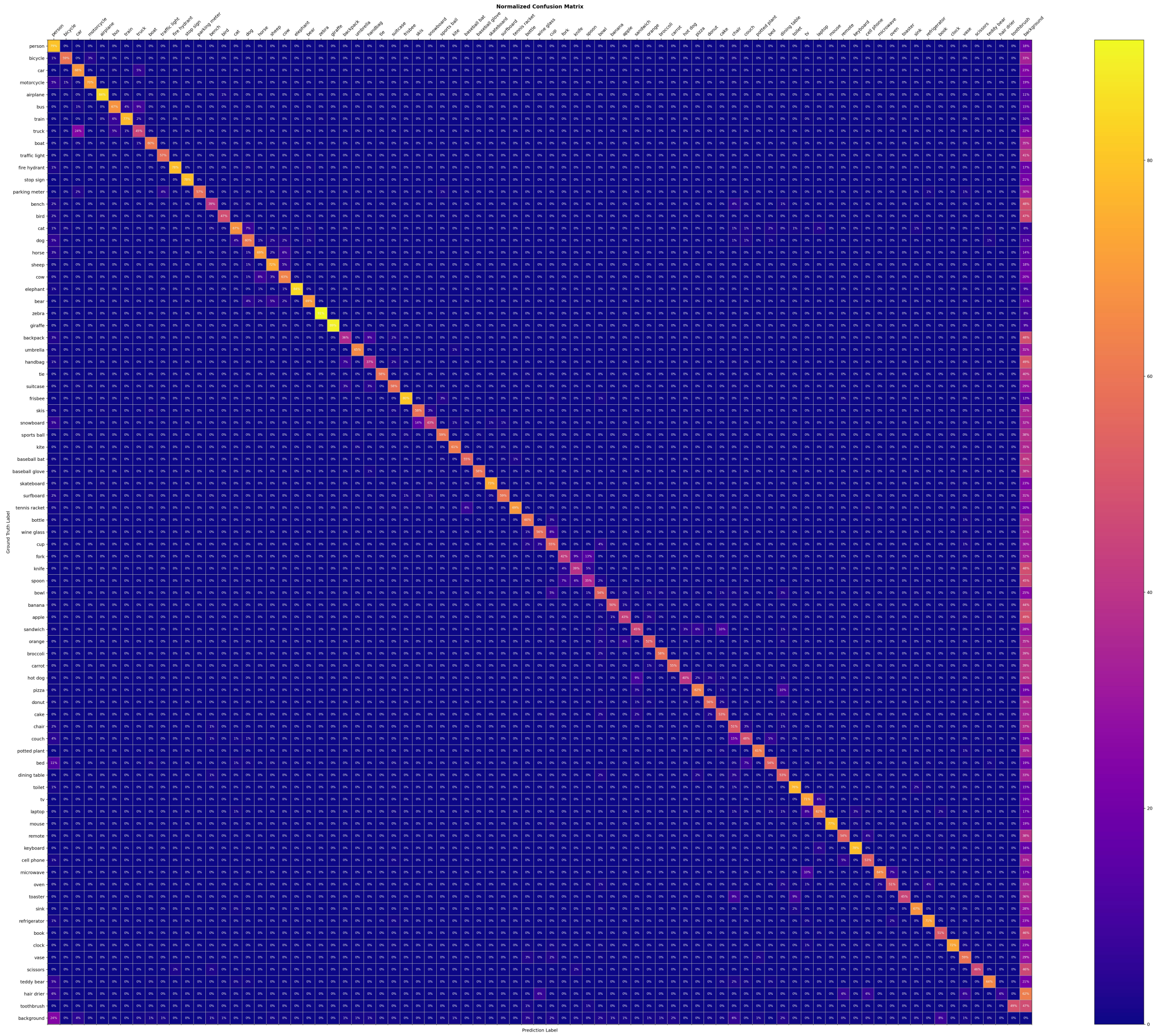

Figure 2. **Confusion matrix from FasterRCNN-R50 distilled with CanKD.**

Table 5. **Analysis on parameter and FLOPs.** We use *thop* library to calculate params and FLOPs in same student and teacher's input with 2×256×64×64. Because FGD [20] including bounding box distillation, it's hard to clarify the complexity.

Method	KD method	FLOPs(G)	Params(M)
FKD [24]	Attention	3.23	0.46
MGD [21]	Mask	9.66	1.18
FGD [20]	Attention	-	0.14
CanKD	Attention	**1.09**	**0.13**

Table 6. The experiment result with mismatch H×W

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
T:MaskRCNN-SwinS	48.2	69.8	52.8	32.1	51.8	62.7
S:Retina-R50	37.4	56.7	39.6	20.0	40.7	49.7
PKD [2]	41.5	60.6	44.1	22.9	45.2	56.4
DetKDS [11]	41.4	60.8	44.4	23.4	45.1	55.5
CanKD	40.5	59.0	43.3	22.5	44.7	54.2

tillation results of Dino-Swin-L and Dino-Swin-B, CanKD shows only marginal improvement for this pair, which previous research attribute to the substantial architectural gap between Transformer-based backbones, where a simple CNN-based approach cannot be effectively applied [22]. The observation that CWD [16] performs even worse than the baseline further corroborates this finding.

References

[1] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *2005 IEEE computer society conference on computer vision and pattern recognition*

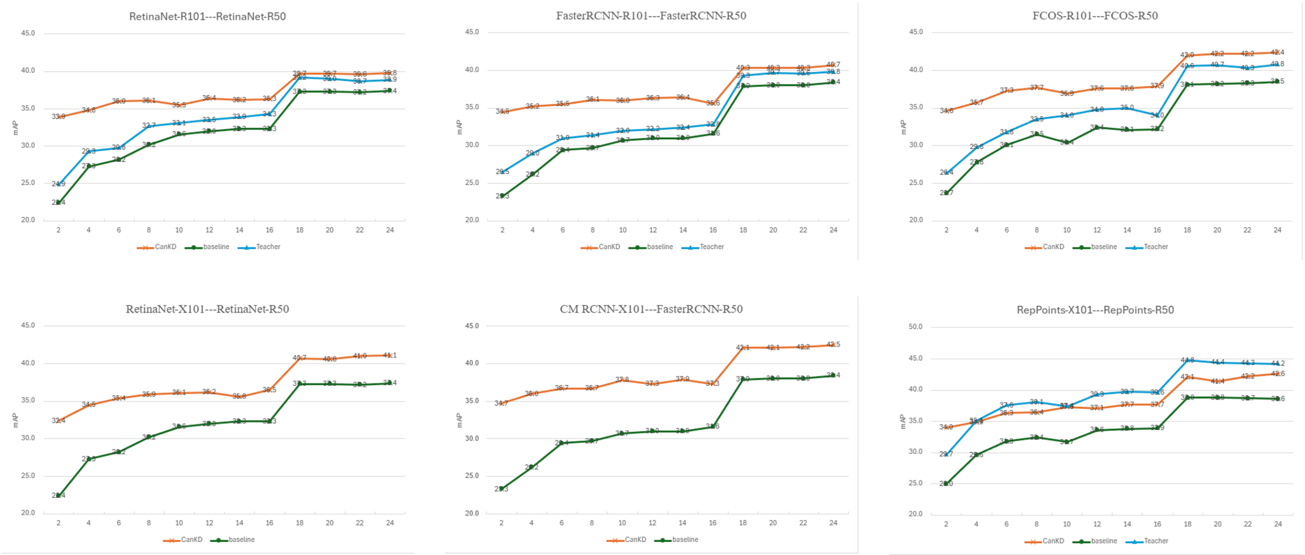

Figure 3. **Line chart about all student models, distilled student models, and teacher models.**

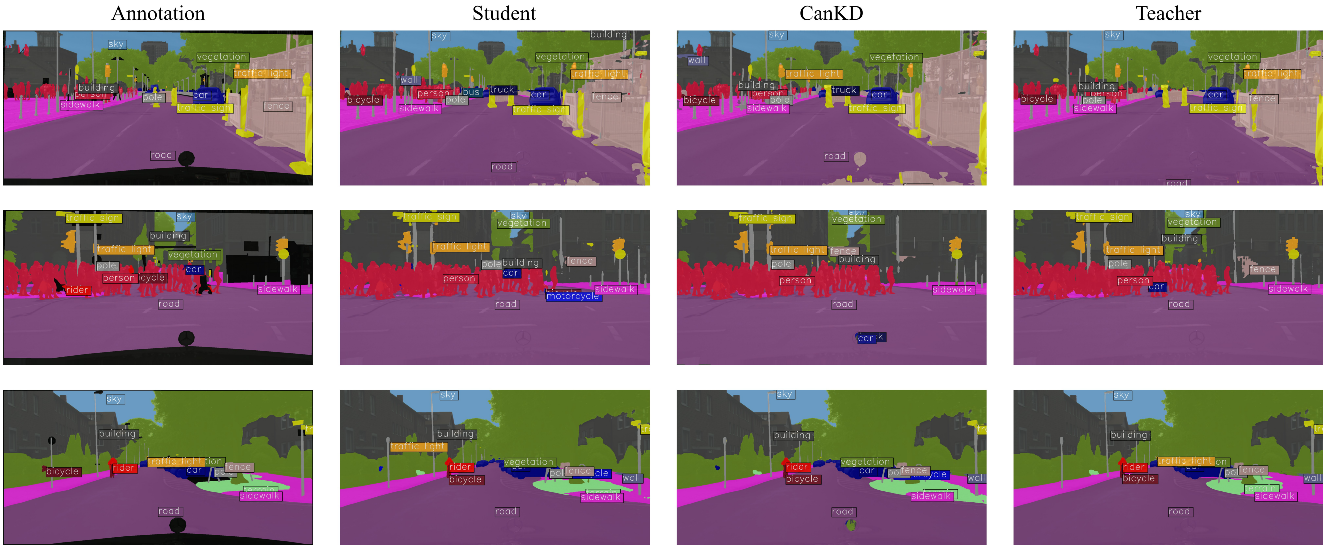

Figure 4. **Additional sampling from PSPNet-R18[25], ditilled PSPNet-R18 with CanKD and PSPNet-R101.** All of these figures are selected from Cityscapes val dataset.

(CVPR'05), pages 60–65. Ieee, 2005. 1

[2] Weihan Cao, Yifan Zhang, Jianfei Gao, Anda Cheng, Ke Cheng, and Jian Cheng. Pkd: General distillation framework for object detectors via pearson correlation coefficient. In *2022 Advances in Neural Information Processing Systems*, pages 15394–15406, 2022. 4

[3] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 1

[4] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 1

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 2

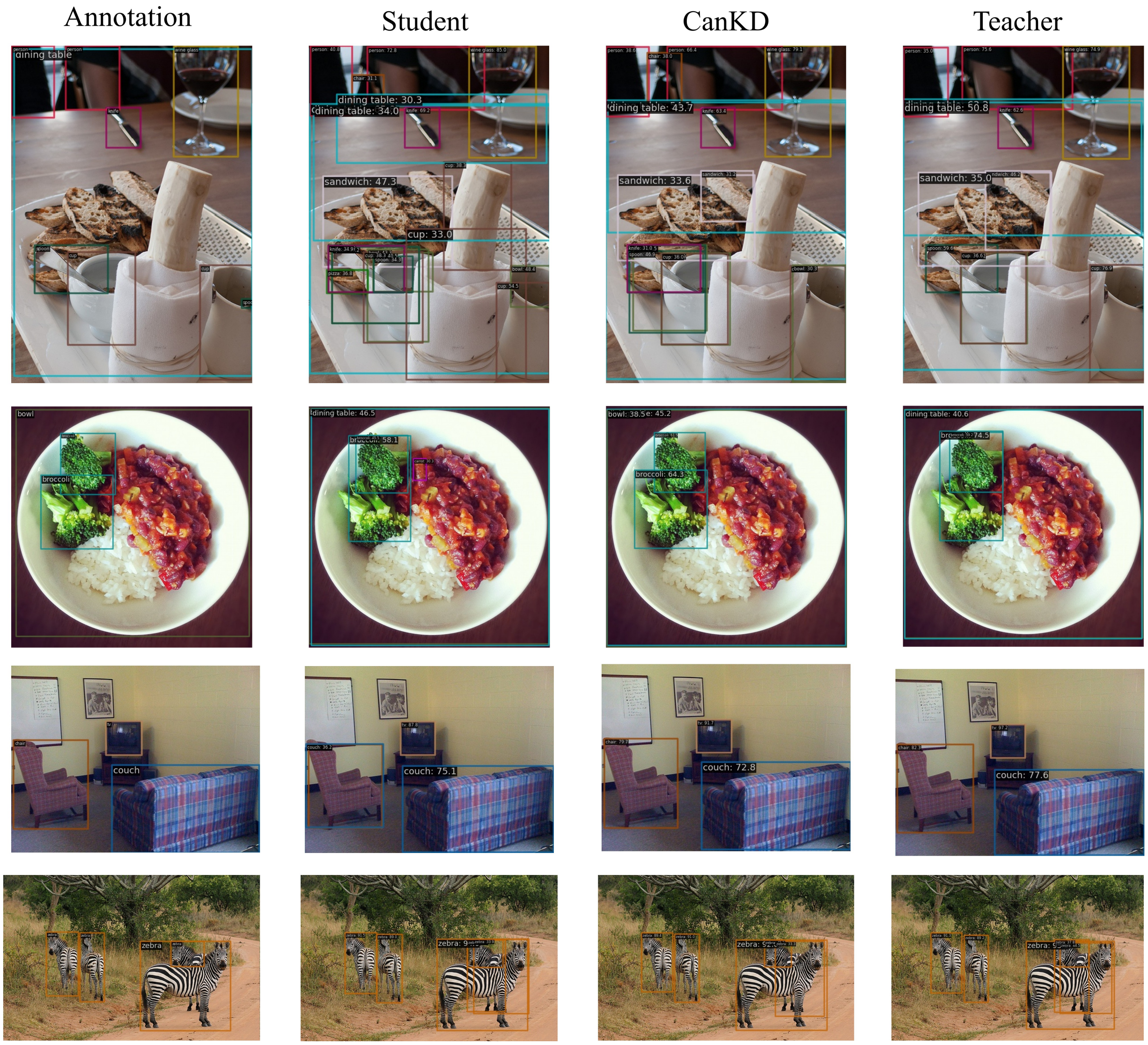

Figure 5. **Additional sampling from RepPoints-R50, ditilled RepPoints-R50 with CanKD and RepPoints-X101.** All of these figures are selected from COCO val 2017 dataset.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2

[7] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour, 2018. 1

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[9] Geoffrey Hinton. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2, 3

[10] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 3

[11] Lujun Li, Yufan Bao, Peijie Dong, Chuanguang Yang, Anggeng Li, Wenhan Luo, Qifeng Liu, Wei Xue, and Yike Guo. Detkds: Knowledge distillation search for object detectors. In *2024 International Conference on Machine Learning*, 2024. 4

[12] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 2

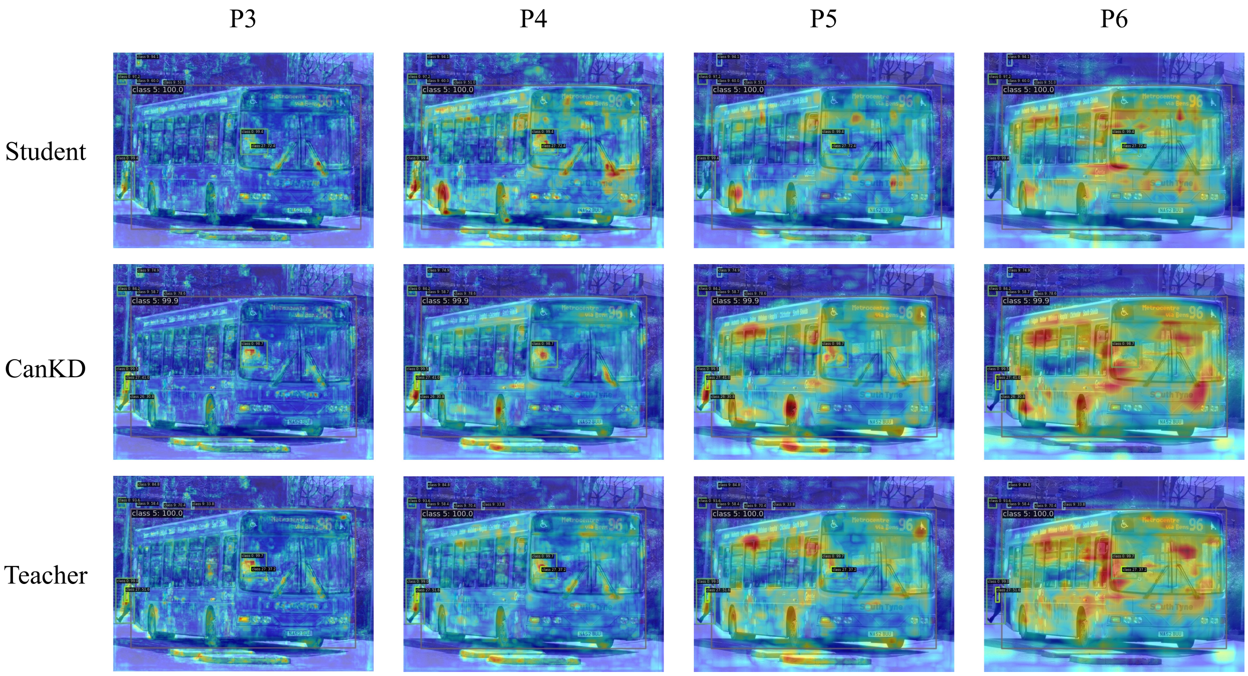

Figure 6. **Heatmaps from FasterRCNN-R50, ditilled FasterRCNN-R50 with CanKD and FasterRCNN-R101.** These figures are generated from P3 to P5 in FPN layers

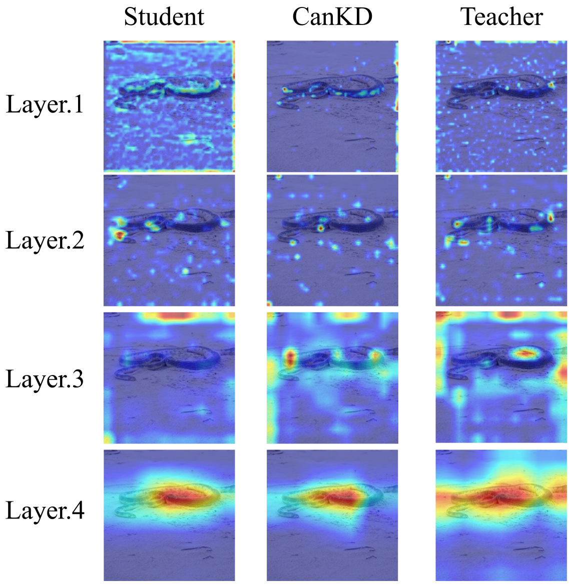

Figure 7. **Heatmaps from ResNet-18, ditilled ResNet-18 with CanKD and ResNet-32.** These figures are generated from Layer.1 to Layer.4 in backbone.

[13] Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S. Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv:1907.07484*, 2019. 1

[14] Shaoqing Ren. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015. 2

[15] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 2

[16] Changyong Shu, Yifan Liu, Jianfei Gao, Zheng Yan, and Chunhua Shen. Channel-wise knowledge distillation for dense prediction. In *2021 IEEE/CVF International Conference on Computer Vision*, pages 5311–5320, 2021. 4

[17] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*, pages 839–846. IEEE, 1998. 1

[18] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 1

[19] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9657–9666, 2019. 2

[20] Zhendong Yang, Zhe Li, Xiaohu Jiang, Yuan Gong, Zehuan Yuan, Danpei Zhao, and Chun Yuan. Focal and global knowledge distillation for detectors. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4643–4652, 2022. 4

[21] Zhendong Yang, Zhe Li, Mingqi Shao, Dachuan Shi, Zehuan

Yuan, and Chun Yuan. Masked generative distillation. In *2022 European Conference on Computer Vision*, pages 53–69, 2022. 3, 4

[22] Zhendong Yang, Zhe Li, Ailing Zeng, Zexian Li, Chun Yuan, and Yu Li. Vitkd: Feature-based knowledge distillation for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1379–1388, 2024. 4

[23] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016. 2, 3

[24] Linfeng Zhang and Kaisheng Ma. Improve object detection with feature-based knowledge distillation: Towards accurate and efficient detectors. In *2020 International Conference on Learning Representations*, 2020. 4

[25] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 5