Instance-Conditional Knowledge Distillation for Object Detection

Anonymous Author(s)
Affiliation
Address
email

Abstract

Despite the success of Knowledge Distillation (KD) on image classification, it is still challenging to apply KD on object detection owing to the unevenly distributed instance-related information. In this paper, we propose an instance-conditional knowledge distillation given the instance information, aided by the correlation between the teacher representations and human observed instances. Specifically, we measure their correlation towards each object by instance-aware attention mask, representing the selection of the effective regions for distillation. The decomposition and attention are accomplished by an instance-conditional decoding procedure with respect to each object, guided by a localization-sensitive auxiliary task. Extensive experiments demonstrate the efficacy of our method: we observe impressive improvements under various settings. Notably, we boost RetinaNet with ResNet-50 backbone from 37.4 to 40.7 mAP ($+3.3$) under $1\times$ schedule, that even surpasses the teacher (40.4 mAP) with ResNet-101 backbone under $3\times$ schedule.

1 Introduction

In recent years, deep learning applications blossom with the breakthrough of Deep Neural Networks (DNNs) [14, 21, 18]. In pursuit of high performance, advanced DNNs usually involve tons of neural blocks with millions of parameters, which are computation and memory-consuming. Yet, this might prevent the practical downstream applications like object detection from being deployed to low-power devices which are resource-limited. This demands quite a trade-off between the efficacy and the efficiency. To resolve this, plenty of techniques have been proposed, like network pruning [12, 24, 15], quantization [7, 19, 32], mobile architecture design [35, 36] and knowledge distillation [16, 34, 40] (KD). Among them, KD is one of the most popular choices, since it can boost a target network without introducing extra inference-time burden. KD is popularized by Hinton *et al.* [16], where knowledge from a strong pretrained teacher network is transferred to a small target network called student. Many works have been proposed later to improve the classification results of the student network. The distilling knowledge forms could be divided into three types, *i.e.*logit [16], intermediate representations [46, 34] and relations [29]. By replacing the teacher knowledge with the student knowledge itself, there develops the self-distillation methods [9, 48].

However, the knowledge distillation methods for image classification are sub-optimal in the detection scenarios and obtain degraded performance [25, 47]. This could be attributed to two reasons:

(1) Other than category classification, the detection task also requires to localize the object locations which is also challenging. (2) Normally, images under classification scenes are often object-centric that each image is entirely filled with a single object. In contrast, in object detection, there could be multiple objects distributed unevenly inside an image.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

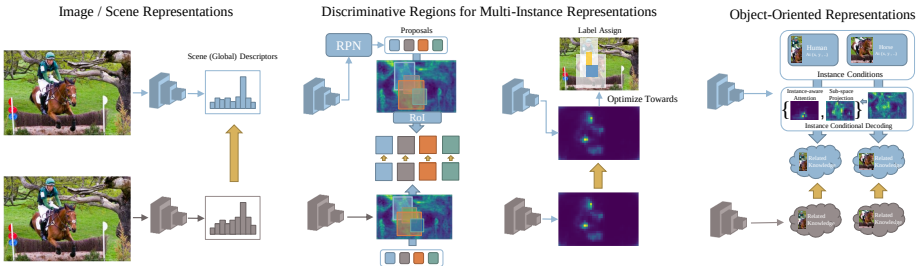


Figure 1: Comparison with different types of knowledge for distillation. Most KD algorithms distill image-level representations. Previous studies for object detection distill discriminate regions by sampling proposals or center areas. We propose to learn instance-aware attention to find and distill information that correlate with each instance.

Previous KD methods for object detection normally involve distilling the intermediate feature representations. Owing to the localization demands required by detection, it is important to determine where to transfer between student and teacher detector. According to the distilling region selection paradigm, existent works could be divided into two types: (i) heuristics-based and (ii) attention-based.

(i) In [25, 42], foreground proposal regions predicted by RPN [33] towards the ground-truths or nearby regions are proposed for knowledge transfer. Guo *et al.* [11] further consider separate distillation with regard to foreground and background region. Dai *et al.* [8] select the regions with the most differed predicted confidence between teacher and student networks. Above methods either neglect the informative context regions [25, 42] or involve meticulous decision rules and complicated label assignment details [11, 8].

(ii) There are few works investigating attention-based region selection. Recently, Zhang *et al.* [47] build the spatial-channel-wise attention mask upon the intermediate representations to guide the distillation. However, the visually discriminative regions inferred from such attention might not be related to the meaningful distilling regions.

To model the attention mechanism that facilitates detection distillation more explicitly, we propose **I**nstance-**C**onditional knowledge **D**istillation (ICD). It aims to transfer representations that are related to ground-truth human observed instances. We deem such representations as instance-conditional knowledge. Fig. 1 show the comparison of ours to the former's. Notably, former studies either focus on the aimless global context or restricted foreground regions, while ours is more target-oriented and flexible. In fact, we design a conditional decoding procedure to locate related regions given the instance information. The knowledge retrieval implemented via the transformer decoder will learn to infer the beneficial distilling regions represented by an instance-aware attention mask, inspired by the transformers [4, 40] and conditional modeling [38, 43]. It is optimized by an auxiliary task on identification and localization, which enforces the decoder to find related knowledge. In the end, we perform distillation weighted by the instance-aware mask. In short, our contribution is three folds:

- To our knowledge, it is the first trial to explore target-oriented knowledge for finding effective distillation regions.
- We introduce a novel conditional modeling paradigm to model instance-wise correlations. We decompose the teaching representations in key-query manner and learn fine-grain knowledge.
- We perform comprehensive experiments on challenging benchmark MS-COCO. Results demonstrate impressive improvements over various detectors with at most 4 AP gain, including recent detectors for instance segmentation [38, 43, 13]. Under some cases, students with $1\times$ schedule are able to outperform their teachers with larger backbones trained $3\times$ longer.

2 Related Works

2.1 Object Detection

Object detection has been developed rapidly. Modern detectors are roughly divided into two-stage or one-stage detectors. Two-stage detectors usually adopt Region Proposal Network (RPN) to generate a rough prediction, the typical example is the widely-known detector Faster R-CNN [33]. On the

contrary, one-stage detectors directly predict on the feature map, which are usually faster, they include RetinaNet [27], FCOS [39]. Besides of this rough division, many extensions are introduced in recent years, *e.g.*, extension to instance segmentation [38, 43, 13], anchor-free models [39, 22] and end-to-end detection [4, 41, 17]. Among these works, some detectors propose to model instances by conditional computation on queries [4] or intermediate predictions [38], they are further discussed in Sec. 2.3.

2.2 Knowledge Distillation

Knowledge distillation involves a strong teacher teaching a weaker student network. Many methods have been proposed especially for image classification [40, 34]. However, it is non-trivial to apply knowledge distillation to object detection. Former studies in detection distillation focus on the foreground-background unbalancing. It could ultimately be ascribed to the distilling region selection problem for which current methods could mainly be divided into two kinds: (i) heuristics-based and (ii) attention-based.

(i) Li *et al.* [25] and Wang *et al.* [42] propose to distilling upon the foreground regions predicted by RPN [33] or nearby regions determined by the GT-anchor matching, respectively. Guo *et al.* [11] further considers separate distillation with regard to foreground and background region. Dai *et al.* [8] define GI scores to denote the controversial regions between teacher and student networks for . However, these methods either overlook the usage of context information [25, 42] or involve meticulous decision rules and ticklish label assignment across multi-scale feature levels [11, 8].

(ii) There are few works investigating attention-based region selection. Recently, Zhang *et al.* [47] build the spatial-channel-wise attention mask upon the intermediate representations to guide the distillation. However, the visually discriminative regions inferred from such attention might not be related to the meaningful distilling regions.

To overcome the above limitations, we discover relations between human observed instances and various visual concepts learned by the network and focus on these related concepts.

2.3 Conditional Computation

Conditional computation is wildly adopted to inference on a given condition, *e.g.*, GAN [10] and VAE [20] reconstruct images conditioned on signals or priors, [45, 6] compute conditional convolution weights. This study mostly focuses on how to identify visual contents given an instance as condition, which is usually formulated as query an instance on the image, *e.g.*, in visual question answer [1, 2] and image-text matching [23], the query is controlled by natural language. Besides query by language, which is not economic or necessary for object detection. query based detection methods are proposed in recent years. For example, DETR [4] queries on fixed embeddings and updates them iteratively, Chen *et al.* [5] encodes points are quires to facilitate weakly-supervised learning following similar settings. These works are built upon global receptive fields, that cover all visual contents, yet these methods usually rely on cascaded decoders that are costly for training. From another perspective, CondInst [38] and SOLOv2 [43] adopt a cheaper solution, they build queries based on network predictions and achieves great performance on instance segmentation. Our work also adopts a query-based approach to locating knowledge among all visual contents, we build our query on annotated instances different from former studies.

3 Method

3.1 Overview

As discussed in former studies [11, 47], knowledge in detection scenario distributes unevenly. To facilitate effective distillation, we propose to transfer instance correlated information which we deem as instance-wise conditional knowledge between student and teacher network, termed $\kappa_i^{\mathcal{S}}$ and $\kappa_i^{\mathcal{T}}$:

$$\mathcal{L}_{distill} = \sum_{i=1}^{N} \mathcal{L}_{dist}(\kappa_i^{\mathcal{S}}, \kappa_i^{\mathcal{T}}) \tag{1}$$

Specifically, the instance-conditional knowledge towards teacher representation $\mathcal{T}$ is calculated as $\kappa_i^{\mathcal{T}} = \mathcal{G}(\mathcal{T}, y_i)$ (likewise for $\kappa_i^{\mathcal{S}}$ towards student representation $\mathcal{S}$). $\mathcal{G}$ stands for the instance-conditional decoding procedure, which will be optimized via an auxiliary loss illustrated in Sec. 3.3. The overall framework is shown in Fig. 2

In the following sections, we will introduce the instance-conditional knowledge (Sec. 3.2), describe the auxiliary task design (Sec. 3.3), and discuss the knowledge transfer (Sec. 3.4).

3.2 Instance-conditional Knowledge

In this section, we elaborate the instance-conditional decoding procedure $\mathcal{G}$, which computes instance-conditional knowledge $\kappa_i^{\mathcal{T}}$ from (a) *unconditional knowledge* given (b) *instance information*.

(a) The unconditional knowledge $\mathcal{T}$, symbolizes all available information from the teacher detector. Since modern detectors commonly involve a feature pyramid network (FPN) [26] to extract rich multi-scale representations, we adopt FPN representations as $\mathcal{T} = \{X_p \in \mathbb{R}^{D \times H_p \times W_p}\}_{p \in \mathcal{P}}$, where $\mathcal{P}$ signifies the spatial resolutions while D is the channel number. By concatenating representations at different scales along the spatial dimension, we obtain $\mathrm{A}^{\mathcal{T}} \in \mathbb{R}^{L \times D}$, where $L = \sum_{p \in \mathcal{P}} H_p W_p$ is the sum of total pixel number across scales.

(b) The instance information, originally describing human-observed objects, is denoted by $\mathcal{Y} = \{\mathrm{y}_i\}_{i=1}^{N}$, where N is the object number and $\mathrm{y}_i = (c_i, \mathbf{b}_i)$ is the annotation for the i-th instance, including category c_i and box location $\mathbf{b}_i = (x_i, y_i, w_i, h_i)$ which specifies the box centre and size information.

To produce a learnable embedding for each instance, the annotation is mapped to a *query* feature vector $\mathbf{q}_i$ in the hidden space that will serve as instance condition and helps collect desired knowledge:

$$\mathbf{q}_i = \mathcal{F}_q(\mathcal{E}(\mathrm{y}_i)), \quad \mathbf{q}_i \in \mathbb{R}^D \tag{2}$$

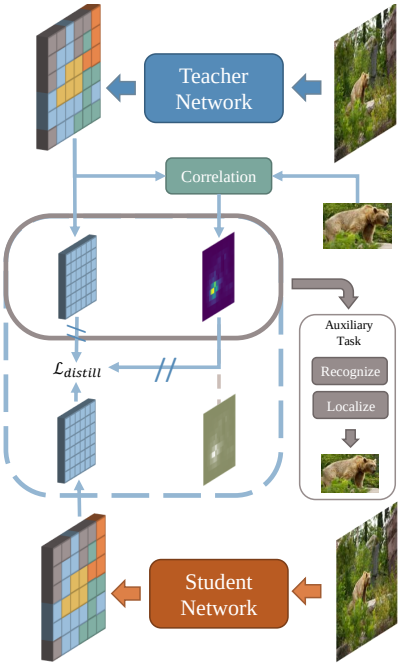


Figure 2: Overall framework.

where $\mathcal{E}(\cdot)$ is an instance encoding function and $\mathcal{F}_q$ is a Multi-Layer Perception network (MLP).

We retrieve knowledge from $\mathcal{T}$ given $\mathbf{q}_i$ by measuring the correlated responses in spatial-channel extent. This is formulated by multi-head attention [40] with M concurrent heads in a key-query-value manner. Different from the query feature $\mathbf{q}_i$, each head $j, (1 \leq j \leq M)$ corresponds to two linear layers ($\mathcal{F}_j^k, \mathcal{F}_j^q, \mathcal{F}_j^v$) *w.r.t.* the key, query and value construction.

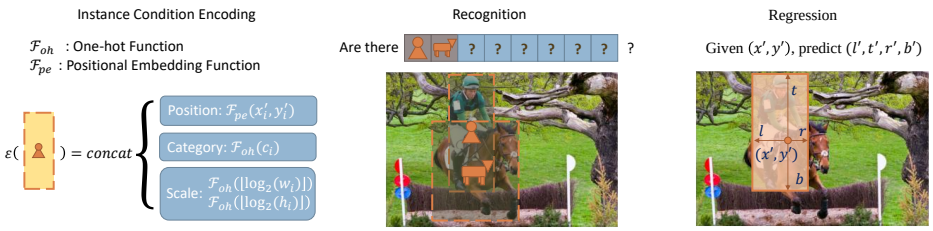
The head-wise key feature $\mathrm{K}_j^{\mathcal{T}}$ is computed by considering the appearance knowledge $\mathrm{A}^{\mathcal{T}}$ and the positional embedding [40, 4] $\mathrm{P} \in \mathbb{R}^{L \times D}$ as shown in Eq. 3. $\mathcal{F}_{pe}$ denotes linear projection over position embeddings. The head-wise value feature $\mathrm{V}_j^{\mathcal{T}}$ and $\mathbf{q}_{ij}$ is obtained by the dimension-reduced projection $\mathcal{F}_j^v$ on $\mathrm{A}^{\mathcal{T}}$ and $\mathcal{F}_j^q$ over $\mathbf{q}_i$ respectively, with output dimension $d = D/M$, as shown in Eq. 4. At last, an instance-aware attention mask $\mathbf{m}_{ij}$ of the i-th instance by the j-th head is obtained by normalized matrix multiplication between $\mathrm{K}_j^{\mathcal{T}}$ and $\mathbf{q}_{ij}$:

$$\mathrm{K}_j^{\mathcal{T}} = \mathcal{F}_j^k(\mathrm{A}^{\mathcal{T}} + \mathcal{F}_{pe}(\mathrm{P})), \quad \mathrm{K}_j^{\mathcal{T}} \in \mathbb{R}^{L \times d} \tag{3}$$

$$\mathrm{V}_j^{\mathcal{T}} = \mathcal{F}_j^v(\mathrm{A}^{\mathcal{T}}), \quad \mathrm{V}_j^{\mathcal{T}} \in \mathbb{R}^{L \times d} \tag{4}$$

$$\mathbf{q}_{ij} = \mathcal{F}_j^q(\mathbf{q}_i), \quad \mathbf{q}_{ij} \in \mathbb{R}^d \tag{5}$$

$$\mathbf{m}_{ij} = softmax(\frac{\mathrm{K}_j^{\mathcal{T}} \mathbf{q}_{ij}}{\sqrt{d}}), \quad \mathbf{m}_{ij} \in \mathbb{R}^L \tag{6}$$



(a) (b) (c)

Figure 3: Illustration for our auxiliary task. (a) We encode potential objects to vectors. (b) The decoder learns to identify each instance. (c) The decoder learns to regress boundaries, similar to [39].

Intuitively, the querying along the spatial dimension of the key features and the dimension-reduced projection onto the value features enable the spatial-channel correlation. We collect $\kappa_i^{\mathcal{T}} = \{(\mathbf{m}_{ij}, \mathrm{V}_j^{\mathcal{T}})\}_{j=1}^{M}$ as the instance-conditional knowledge towards $\mathcal{T}$.

3.3 Auxiliary Task

In this section, we introduce the auxiliary task to optimize the decoding procedure $\mathcal{G}$. It involves identifying and localizing the objects based on the instance-level information aggregated in a spatial-channel perspective. This could be obtained by aggregating the instance-conditional knowledge and the query by the function $\mathcal{F}_{agg}$, which includes linear mapping over concated features, residual connections, and a feed-forward network as proposed in [40, 4]:

$$\mathbf{g}_i^{\mathcal{T}} = \mathcal{F}_{agg}(\kappa_i^{\mathcal{T}}, \mathbf{q}_i), \quad \mathbf{g}_i^{\mathcal{T}} \in \mathbb{R}^D \tag{7}$$

To enable the instance-level aggregated information $\mathbf{g}_i^{\mathcal{T}}$ in retaining sufficient instance cues, one could optimize it with the instance information below:

$$\mathcal{L}_{aux} = \mathcal{L}_{dist}(\mathbf{g}_i^{\mathcal{T}}, \mathcal{H}(\mathrm{y}_i)) \tag{8}$$

where $\mathcal{H}$ is a label encoding function of the instance information. However, direct optimization by Eq. 8 might lead to trivial solution, since y_i is accessible for both $\mathbf{g}_i^{\mathcal{T}}$ (through $\mathbf{q}_i$, see Eq. 5) and $\mathcal{H}(\mathrm{y}_i)$. It is plausible for the learnable mappings to hack on the cost function, ignoring the teacher representation $\mathcal{T}$. To resolve this, we propose to drop information while encoding by $\mathcal{E}(\cdot)$, to enforce the aggregation function $\mathcal{F}_{agg}$ to excavate hints from $\mathcal{T}$ as much as possible.

We achieve the information dropping by replacing the annotation with coarser version. Exactly, we disturb the box centre and acquire rough categorical scale to represent the instance information used by $\mathcal{G}$. The disturbed box centre (x_i', y_i') is obtained by random jittering shown below:

$$\begin{cases} x_i' = x_i + \phi_x w_i, \\ y_i' = y_i + \phi_y h_i, \end{cases} \tag{9}$$

where (w_i, h_i) is the width and height of the bounding box and ϕ_x, ϕ_y are observations of a random jittering variable $\Phi \sim U[-a, a]$, where we set a=-0.3 for this uniform distribution empirically. The categorical box scale labels are obtained by rounding the box sizes in the logarithmic space, *i.e.*, $\lfloor log_2(w_i) \rfloor$, $\lfloor log_2(h_i) \rfloor$. As a result, we collect coarse information and obtain instance encoding through $\mathcal{E}(\cdot)$ as depicted in Fig. 3a. where c_i is the category, $\mathcal{F}_{oh}$ is the one hot vectorization, $concat$ is the concatenation operator and $\mathcal{F}_{pe}$ is the position embedding function.

Subsequently, the computed query feature $\mathbf{q}_i = \mathcal{F}_q(\mathcal{E}(\mathrm{y}_i))$ and aggregated representation $\mathbf{g}_i^{\mathcal{T}}$ are optimized by the auxiliary task. We introduces two MLPs denoted by $\mathcal{F}_{obj}$ and $\mathcal{F}_{reg}$ respectively to predict the identification and localization results. Notably for the identification task, we generate fake instances with categories and boxes according to the dataset distribution priori, in contrast to the existent real instances. The localization loss for negative examples whose are ignored following common practice. In short, we use binary cross entropy loss (BCE) to penalize the real-fake identification confidence and $L1$ distance to penalize the regressed boundaries:

$$\mathcal{L}_{aux} = \mathcal{L}_{BCE}(\mathcal{F}_{obj}(\mathbf{g}_i^{\mathcal{T}}), \delta_{obj}(\mathrm{y}_i)) + \mathcal{L}_1(\mathcal{F}_{reg}(\mathbf{g}_i^{\mathcal{T}}), \delta_{reg}(\mathrm{y}_i)) \tag{10}$$

where $\delta_{obj}(\cdot)$ is an indicator yielding 1 if y_i is real and 0 otherwise. δ_{reg} is the preparing function for relative coordinate targets [4], following [39]. Please refer to Appendix A for more details.

3.4 Instance Conditional Distillation

Recalling to knowledge retrieval process described in Sec. 3.2, we obtain the projected value feature $\mathrm{V}_j^{\mathcal{S}} \in \mathbb{R}^{L\times d}$ of the student representations analogous to Eq. 4. By adopting the instance-aware attention mask reasoned by unconditional teacher knowledge conditioned on instance information, the proposed instance-conditioned knowledge transfer is formulated as:

$$\mathcal{L}_{distill} = \frac{1}{MN_r}\sum_{j=1}^{M}\sum_{i=1}^{N}\delta_{obj}(\mathrm{y}_i)\cdot < \mathbf{m}_{ij}, \mathcal{L}_{MSE}(\mathrm{V}_j^{\mathcal{S}}, \mathrm{V}_j^{\mathcal{T}}) > \tag{11}$$

where $N_r = \sum_{i=1}^{N}\delta_{obj}(\mathrm{y}_i)$, $(N_r \leq N)$ is the real instances number. $\mathcal{L}_{MSE}(\mathrm{V}_j^{\mathcal{S}}, \mathrm{V}_j^{\mathcal{T}}) \in \mathbb{R}^L$ is the pixel-wise mean-square error along the hidden dimension[1]. $< \cdot, \cdot >$ is the Direc notation for inner product. The overall loss is summarized below where λ is the coefficient for distillation loss.

$$\mathcal{L}_{total} = \mathcal{L}_{det} + \mathcal{L}_{aux} + \lambda\mathcal{L}_{distill} \tag{12}$$

Worth for noticing, the gradients *w.r.t.*, $\mathcal{L}_{det}$ update the student network only. The gradients *w.r.t.* $\mathcal{L}_{distill}$ back-propagate to the student network (from representations $\mathcal{S}$) and the projection layer $\mathcal{F}_j^{\mathcal{S}}$ that creates $\mathrm{V}_j^{\mathcal{S}}$, leaving $\mathbf{m}_{ij}$ and $\mathrm{V}_j^{\mathcal{T}}$ detached. The gradients of $\mathcal{L}_{aux}$ update only the instance-conditional decoding function $\mathcal{G}$.

4 Experiments

4.1 Experiment Settings

We conduct our experiments on Pytorch [31] with the widely used Detectron2 library [44] and AdelaiDet library [2] [37]. All experiments are running with 16 batch size on eight 2080ti GPUs. We adopt the $1\times$ scheduler, which denotes 9k iterations of training, following the standard protocols in Detectron2 unless otherwise specified. We use scale jitters and random flip as data augmentation.

For distillation, the hyper-parameter λ is set to 8 for one-stage detectors and 3 for two-stage detectors respectively. To optimize the transformer decoder, we adopt AdamW optimizer [30] for the decoder and MLPs following common settings for transformer [40, 4], the corresponding hyper-parameters follows [4], where the initial learning rate and weight decay are set to 1e-4. We adopt the 256 hidden dimension for our decoder and all MLPs, the decoder has 8 heads in parallel. The projection layer $\mathcal{F}_q$ is a 3 layer MLP, and F_{reg} and F_{obj} share another 3 layer MLP. In addition, we notice some newly initialized modules share the same size between teacher and student, *e.g.*, the detection head, FPN. We find inheriting parameters of these modules from the teacher will lead to faster convergence.

The experiments is conducted on a large scale object detection benchmark **MS-COCO** [3][28] which contains 80 object classes. We train our model on MS-COCO 2017 `trainval115k` subset and validate on `minival` subset. Following common protocols, we report mean Average Precision (AP) as an evaluation metric, together with Average Precision under different thresholds and scales, *i.e.*, AP_{50}, AP_{75}, AP_{S}, AP_{M}, AP_{L}.

4.2 Main Results

Compare with state-of-the-art methods. We compare our method (ICD) with previous state-of-the-arts (SOTAs), including a classic method FitNet [34], two typical detection KD methods [25, 42], and a very recent work with strong performance from Wang *et al.* [42]. The comparison is conducted on two classic detectors: Faster R-CNN [33] and RetinaNet [27], note that methods from Li *et al.* [25]

[1] We introduce an extra parameter-free LayerNorm [3] in $\mathcal{L}_{MSE}(.)$ over value features to stabilize training.

[2] All libraries are open-sourced and public available, please refer to citations for more details.

[3] MS-COCO is publicly available, the annotations are licensed under a `Creative Commons Attribution 4.0 License` and the use of the images follows `Flickr Terms of Use`, read [28] for more details.

Table 1: Comparison with previous methods on challenging benchmark MS-COCO. The method proposed by Li *et al.* [25] do not support RetinaNet. † denotes using inherit strategy.

Method	Faster R-CNN [33]				RetinaNet [27]			
	AP	AP_S	AP_M	AP_L	AP	AP_S	AP_M	AP_L
Teacher w. ResNet-101 (3×)	42.0	25.2	45.6	54.6	40.4	24.0	44.3	52.2
Student w. ResNet-50 (1×)	37.9	22.4	41.1	49.1	37.4	23.1	41.6	48.3
+ FitNet [34]	39.3 (+1.4)	22.7	42.3	51.7	38.2 (+0.8)	21.8	42.6	48.8
+ Li *et al.* [25]	39.5 (+1.5)	23.3	43.0	51.4	-	-	-	-
+ Wang *et al.* [42]	39.2 (+1.3)	23.2	42.8	50.4	38.4 (+1.0)	23.3	42.6	49.1
+ Zhang *et al.* [47]	40.0 (+2.1)	23.2	43.3	52.5	39.3 (+1.9)	23.4	43.6	50.6
+ Ours	40.4 (+2.5)	23.4	44.0	52.0	39.9 (+2.5)	25.0	43.9	51.0
+ Ours †	40.9 (+3.0)	24.5	44.2	53.5	40.7 (+3.3)	24.2	45.0	52.7

Table 2: Experiments on more detectors with ICD. Type denotes the AP score is evaluated on bounding box (`BBox`) or instance mask (`Mask`). † denotes using inherit strategy.

Detector	Setting	Type	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
FCOS [39]	Teacher (3×)	`BBox`	42.6	61.6	45.8	26.2	46.3	53.8
	Student (1×)		39.4	58.2	42.4	24.2	43.4	49.4
	+ Ours		41.7(+2.3)	60.3	45.4	26.9	45.9	52.6
	+ Ours †		42.9(+3.5)	61.6	46.6	27.8	46.8	54.6
Mask R-CNN [13]	Teacher (3×)	`BBox`	42.9	63.3	46.8	26.4	46.6	56.1
	Student (1×)		38.6	59.5	42.1	22.5	42.0	49.9
	+ Ours		40.4 (+1.8)	60.9	44.2	24.4	43.7	52.0
	+ Ours †		41.2 (+2.6)	62.0	45.0	25.1	44.5	53.6
	Teacher (3×)	`Mask`	38.6	60.4	41.3	19.5	41.3	55.3
	Student (1×)		35.2	56.3	37.5	17.2	37.7	50.3
	+ Ours		36.7 (+1.5)	58.0	39.2	18.4	38.9	52.5
	+ Ours †		37.4 (+2.2)	58.7	40.1	19.1	39.8	53.7
SOLOv2 [43]	Teacher (3×)	`Mask`	39.0	59.4	41.9	16.2	43.1	58.2
	Student		34.6	54.7	36.9	13.2	37.9	53.3
	+ Ours		37.2 (+2.6)	57.6	39.8	14.8	40.7	57.0
	+ Ours †		38.5 (+3.9)	59.0	41.2	15.9	42.3	58.9
CondInst [38]	Teacher (3×)	`BBox`	43.3	62.2	47.0	26.6	46.8	55.9
	Student (1×)		39.7	58.8	43.1	23.9	43.3	50.1
	+ Ours		42.4 (+2.7)	61.5	46.1	25.3	46.0	54.3
	+ Ours †		43.7 (+4.0)	62.9	47.2	27.1	47.3	56.6
	Teacher (3×)	`Mask`	38.6	59.9	41.3	19.2	41.7	56.1
	Student (1×)		35.7	56.7	37.7	16.8	39.1	50.3
	+ Ours		37.8 (+2.1)	59.1	40.4	17.5	41.4	54.7
	+ Ours †		39.1 (+3.4)	60.5	42.0	19.1	42.6	57.0

rely on Region Proposal Network (RPN) that only supports Faster R-CNN. We adopt detectron2 official models with ResNet-101 trained on $3\times$ scheduler as teacher networks, the student is trained on $1\times$ scheduler with ResNet-50 backbone following the common protocols.

As shown in Table 1, ICD leads to about 2.5 AP and 3.0 AP improvement on the student for plain training and inherit strategy respectively. Especially for RetinaNet, the proposed method even outperforms a strong teacher trained on $3\times$ scheduler. Compare with previous SOTAs, the proposed method leads to a considerable margin for about 0.5 AP without the inherit strategy.

Results on other settings. We further evaluate ICD under various detectors, *e.g.*, a commonly used anchor-free detector FCOS [39], and three detectors that have been extended to instance segmentation: Mask R-CNN [13], CondInst [38] and SOLOv2 [43]. We adopt ResNet-101 and $3\times$ scheduler for all teachers and ResNet-50 with $1\times$ scheduler for all students following the above setting. As shown in Table 2, we observe a consistent improvement for both object detection and instance segmentation. There are at most around 4.0 AP improvement on CondInst [38] on object detection and SOLOv2 [43] on instance segmentation. Moreover, students with weaker backbone (ResNet-50 *v.s.* ResNet-101) and less training images ($1\times$ *v.s.* $3\times$) even outperform (CondInst, FCOS) or perform on par (SOLOv2) with teachers.

We also conduct experiments on other datasets and backbones, *etc.*. For example, our method achieves 5.2 AP gain on RetinaNet under mobile backbone MobileNet V2 [35] (26.4 $\rightarrow$ 31.6), and 3.1 AP gain on EfficientNet-B0 [36] (34.9 $\rightarrow$ 38.0). Due to the page limit, please refer to Appendix B.

4.3 Ablation Studies

To verify the design options and the effectiveness of each component, we conduct ablation studies with the classic RetinaNet detector on MS-COCO following the above settings.

Design of the auxiliary task. To better understand the role of our auxiliary task, we conduct experiments to evaluate the contribution of each sub-task. Specifically, our auxiliary task is composed of an identification task with binary cross-entropy loss and a localization task with regression loss, the localization task is further augmented with a hint on bounding box scales. As shown in Table 4, the identification task itself leads to 2.2 AP gain compare with the baseline, this high-light the importance of knowledge on object perception. The regression task itself leads to 1.8 AP gain, and the scale information boosts it for extra 0.2 AP gain. Combine two of them, we achieve a 2.5 overall improvement, which indicates the fusion of two knowledge brings extra benefits.

Table 3: Comparison with different auxiliary task designs.

Idf.	Reg.	w. Scale	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
			37.4	56.7	40.3	23.1	41.6	48.3
✓			39.6	59.2	42.8	23.4	44.0	50.4
	✓		39.2	58.6	42.4	23.1	43.5	50.3
	✓	✓	39.4	58.8	42.4	23.2	43.8	50.5
✓	✓	✓	39.9	59.4	43.1	25.0	43.9	51.0

Impact of the instance-aware attention. To verify the effectiveness of instance-aware attention learned by conditional computation, we directly replace it with different variations: the fine-grained mask in [42], pixel-wise attention activation [47, 40] and foreground mask analog to [11], lastly, we also compare it with no attention. The result shows our instance-aware mask leads to about 0.9 AP gain over the baseline and 0.4 AP gain compare with the best replacement.

Table 4: Comparison with different types of attention.

Attention Type	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
No Attention	39.0	58.4	42.1	23.5	43.2	49.9
Foreground Mask	39.4	58.9	42.4	23.5	43.6	50.0
Fine-grained Mask [42]	39.5	59.0	42.4	23.4	43.8	50.2
Attention Activation [40]	39.3	58.6	42.3	22.6	43.6	50.1
Instance-conditional Attention	39.9	59.4	43.1	25.0	43.9	51.0

Design of the decoder. We mainly verify two properties of the decoder design, as shown in Table 5. First, we find a proper number of heads is important for the best performance. The head number balances the dimension for each sub-space and the number of spaces. The best number lies around 8, which is consistent with former studies [40, 4]. Second, we also evaluate the effectiveness of cascaded decoders, as it is widely adopted in former studies [40, 4, 5]. However, we do not find significant differences between different cascade levels, it might because we use a fixed teacher to provide knowledge, thus limited the learning ability.

Table 5: Ablation on the number of heads and cascade levels.

(a) Number of heads.

Heads	AP	AP_S	AP_M	AP_L
1	39.4	23.7	43.8	50.5
4	39.7	24.0	43.9	51.2
8	39.9	25.0	43.9	51.0
16	39.7	23.6	44.2	50.5

(b) Number of cascade levels.

Levels	AP	AP_S	AP_M	AP_L
1	39.9	25.0	43.9	51.0
2	39.9	24.1	44.0	51.4
4	39.8	24.3	44.0	50.9

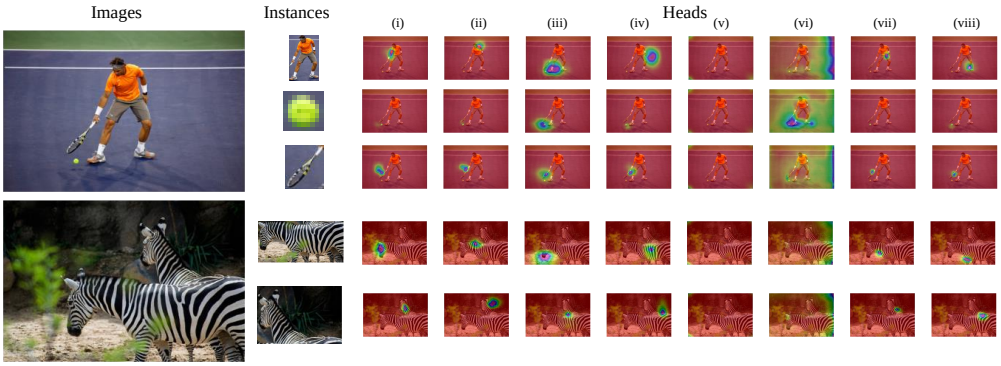


Figure 4: Visualization of our learned instance-aware attention over each head.

5 Discussion

Qualitative analysis. We present a visualization of our learned instance-aware attention in Fig. 4. We observe multiple heads attend on instance boundaries, this observation might suggest knowledge for regression locate around object boundaries, instead of averaged on the foreground as assumed in former studies. The head (iii) and (iv) are smoother around key regions, which may relate to context. The (vi) attends to large areas, which may also relate to context. The head (v) mainly attends on image boundaries, which might be a fail case.

Time consumption. We benchmark the training speed for our distillation method. As shown in Fig. 5, training the decoder introduces negligible cost. Specifically, we benchmark on $1\times$ scheduler on RetinaNet [27] with eight 2080ti, following the same configuration in Section 4.2. The major time consumption is spent on training the teacher ($3\times$), which takes about 33 hours. Training (forward and backward) the student takes about 8 hours, while training decoder and distillation only take 1.3 hours.

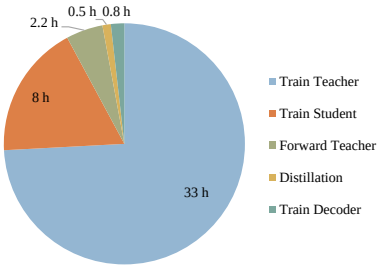


Figure 5: Time consumption.

Real-world impact. The proposed method provides a convenient approach to enhance a detector under a certain setting, results in a small model performs on par with a larger one. On the positive side, it allows users to replace a big model with a smaller one, which reduces energy consumption. On the potential negative side, the trained student is learned from a teacher that could be costly, and the student might inherit biases from the teacher which could be hard to identify.

6 Conclusion

We propose a novel object-oriented solution for knowledge distillation. The proposed **I**nstance-**C**onditional knowledge **D**istillation (ICD) method utilizes instance-feature cross attention to select and locate knowledge that correlates with human observed instances. The attention is formulated scaled dot-product attention, which utilizes encoded instance as query, and teacher's representation as keys. To teach the decoder how to find knowledge, we design an auxiliary task that relies on recognition and localization knowledge. The proposed method consistently improves various detectors, leading to impressive performance gain, some student networks even surpass their teachers.

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018.

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: visual question answering. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, 2015.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*. Springer, 2020.

[5] Liangyu Chen, Tong Yang, Xiangyu Zhang, Wei Zhang, and Jian Sun. Points as queries: Weakly semi-supervised object detection by points. *arXiv preprint arXiv:2104.07434*, 2021.

[6] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 2020.

[7] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*, 2016.

[8] Xing Dai, Zeren Jiang, Zhao Wu, Yiping Bao, Zhicheng Wang, Si Liu, and Erjin Zhou. General instance distillation for object detection. *arXiv preprint arXiv:2103.02340*, 2021.

[9] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International Conference on Machine Learning*, pages 1607–1616. PMLR, 2018.

[10] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[11] Jianyuan Guo, Kai Han, Yunhe Wang, Han Wu, Xinghao Chen, Chunjing Xu, and Chang Xu. Distilling object detectors via decoupled features. *arXiv preprint arXiv:2103.14475*, 2021.

[12] Song Han, Jeff Pool, John Tran, and William J Dally. Learning both weights and connections for efficient neural networks. In *NeurIPS*, 2015.

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016.

[15] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1389–1397, 2017.

[16] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[17] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018.

[18] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017.

[19] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *The Journal of Machine Learning Research*, 18(1):6869–6898, 2017.

[20] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, 2012.

[22] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, 2018.

[23] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[24] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. In *ICLR*, 2016.

[25] Quanquan Li, Shengying Jin, and Junjie Yan. Mimicking very efficient network for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017.

[26] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017.

[27] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017.

[28] 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*. Springer, 2014.

[29] Yufan Liu, Jiajiong Cao, Bing Li, Chunfeng Yuan, Weiming Hu, Yangxi Li, and Yunqiang Duan. Knowledge distillation via instance relationship graph. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019.

[30] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019.

[32] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European conference on computer vision*, pages 525–542. Springer, 2016.

[33] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, 2015.

[34] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[35] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018.

[36] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, Proceedings of Machine Learning Research, 2019.

[37] Zhi Tian, Hao Chen, Xinlong Wang, Yuliang Liu, and Chunhua Shen. AdelaiDet: A toolbox for instance-level recognition tasks. `https://git.io/adelaidet`, 2019.

[38] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *Proc. Eur. Conf. Computer Vision (ECCV)*, 2020.

[39] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: fully convolutional one-stage object detection. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, 2019.

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 2017.

[41] Jianfeng Wang, Lin Song, Zeming Li, Hongbin Sun, Jian Sun, and Nanning Zheng. End-to-end object detection with fully convolutional network. *arXiv preprint arXiv:2012.03544*, 2020.

[42] Tao Wang, Li Yuan, Xiaopeng Zhang, and Jiashi Feng. Distilling object detectors with fine-grained feature imitation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019.

[43] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. SOLOv2: Dynamic and fast instance segmentation. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[44] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[45] Brandon Yang, Gabriel Bender, Quoc V. Le, and Jiquan Ngiam. Condconv: Conditionally parameterized convolutions for efficient inference. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019.

[46] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[47] Linfeng Zhang and Kaisheng Ma. Improve object detection with feature-based knowledge distillation: Towards accurate and efficient detectors. In *International Conference on Learning Representations*, 2021.

[48] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3713–3722, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5, 'Real-world impact'.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 5, 'Real-world impact'.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We include detail instructions, the baseline codebase and data is public available.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4.2
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 5, 'Time consumption'.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cite [44, 31, 37].
   (b) Did you mention the license of the assets? [Yes] See in Footnote 2 and Footnote 3
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] See in Footnote 3 and Footnote 2.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]