Multi-Scale Aligned Distillation for Low-Resolution Detection

Lu Qi[1†], Jason Kuen[2†], Jiuxiang Gu[2], Zhe Lin[2], Yi Wang[1], Yukang Chen[1], Yanwei Li[1], Jiaya Jia[1,3]
[1]The Chinese University of Hong Kong [2]Adobe Research [3]SmartMore

Abstract

In instance-level detection tasks (e.g., object detection), reducing input resolution is an easy option to improve runtime efficiency. However, this option traditionally hurts the detection performance much. This paper focuses on boosting performance of low-resolution models by distilling knowledge from a high- or multi-resolution model. We first identify the challenge of applying knowledge distillation (KD) to teacher and student networks that act on different input resolutions. To tackle it, we explore the idea of spatially aligning feature maps between models of varying input resolutions by shifting feature pyramid position and introduce ***aligned multi-scale training*** *to train a multi-scale teacher that can distill its knowledge to a low-resolution student. Further, we propose* ***crossing feature-level fusion*** *to dynamically fuse teacher's multi-resolution features to guide the student better. On several instance-level detection tasks and datasets, the low-resolution models trained via our approach perform competitively with high-resolution models trained via conventional multi-scale training, while outperforming the latter's low-resolution models by* 2.1% *to* 3.6% *in terms of mAP. Our code is made publicly available at https://github.com/Jia-Research-Lab/MSAD.*

1. Introduction

Deep learning [29, 54, 59, 60, 18, 58, 26, 25] has enabled instance-level detection (object detection [13, 10, 49], instance segmentation [16, 36], human keypoint detection [53, 80, 57], etc.) methods to achieve previously unattainable performance. Heavy computation requirement of deep-learning-based instance-level detection models, however, remains an issue for easy adoption of these models in real-world applications [41, 52, 45, 7].

While many model compression techniques [44, 5, 32, 28] have been proposed to train compact models for accelerated inference, they mostly focus on trimming networks along depth or width [9, 82, 12, 81, 1], or adopting efficient block structure design [24, 51, 23, 79, 39, 20, 4]. Besides depth/width, another critical dimension for the compound scaling of network architectures is the input resolution [61, 31]. However, reducing input resolutions to accelerate instance-level detection is generally not regarded as a decent solution in existing work due to severe performance degradation. For example, for the recent one-stage detector FCOS [64], its mean average precision (AP) drops from 38.7 to 34.6 when the detector is naively trained on 400px images instead of the default 800px images.

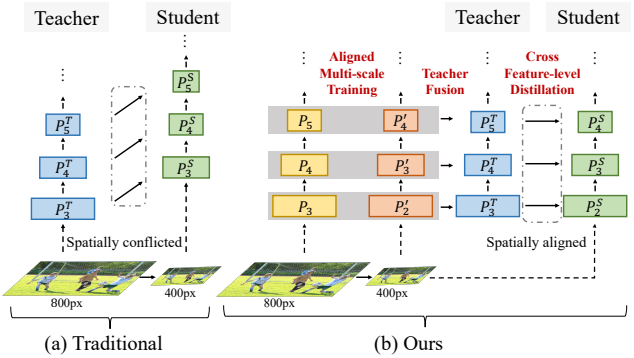


Figure 1: Conceptual comparison between (a) traditional teacher-student approach and (b) ours, in the setting of using a high-resolution teacher to guide a low-resolution student. In this setting, the traditional approach of transferring knowledge along the same feature levels fails due to spatially-conflicted feature maps. To resolve it, we introduce a multi-scale aligned distillation approach.

We are thus interested to study the fundamental problem to *upgrade performance of a low-resolution detection model up to that of its high-resolution counterpart*.

There was study to mitigate performance drop by distilling knowledge (KD) from a high-res teacher to a low-res student [31, 73]. KD methods distill knowledge from a heavier teacher network to a compact student mostly in the context of image classification [76, 14, 50], since the spatial scales of final output of the teacher and student networks are identical. In the context of instance-level detection, it is not trivial to apply KD to high-res teacher and low-res student networks because they do not share the same feature/output spatial size at the same network stages, as illus-

†Equal contribution.

trated in Fig. 1(a). Downsampling feature maps and output of the pre-trained large-resolution teacher to match those of the low-resolution student is one naive workaround. But this operation significantly corrupts predicted features and output, making them poorly reflective of the actual knowledge learned by the teacher.

We, instead, explore alignment of feature maps to resolve the output size mismatch between high-res teacher and low-res student. For the feature pyramid (FPN) structure [33] widely used in instance-level detection networks, the feature map size in last network stage is $2\times$ larger than that of current stage. Based on this observation, for the low-res student, we adopt input resolution $2\times$ smaller than the typical input used for the teacher. This provides feature-level consistency between the two input resolution models and allows their features to match spatially. As shown in Fig. 1(b), the spatial size of P_2 with low-res (downsampled by $2\times$) input shares the same spatial size as P_3 of the high-res input. This simple strategy quickly and effectively enables knowledge distillation from teacher to student.

With this novel alignment idea, we propose an aligned multi-scale training method and a crossing feature-level fusion module to train a strong teacher. Aligned multi-scale training qualifies a ready-for-distillation robust teacher that performs well across multiple input resolutions. Whereas the crossing feature-level fusion module dynamically fuses the features from multiple-res models within the same teacher network. Finally, the rich multi-scale and multi-resolution knowledge of the *multi-scale fusion teacher* is distilled to the low-res student, resulting in a high-performing low-resolution model. Fig. 1(b) provides a high-level overview of our approach.

Our main contribution is threefold.

- The alignment concept to align feature maps of models at different input resolutions.
- A framework for training a strong multi-scale and multi-resolution fusion teacher that provides more informative training signals to the low-res student that does not have access to fine visual details in high-res images.
- Extensive ablation studies and comparative experiments on different instance-level detection tasks to demonstrate the effectiveness of our methods.

2. Related Work

2.1. Instance-level Detection Tasks

Instance-level detection tasks, including object detection [13, 10, 49, 33, 34, 64, 8], instance segmentation [16, 36, 71, 3, 78, 46], and key point detection [16, 80, 57, 70], require detecting objects at the instance level. From the viewpoint of coarse-to-fine recognition, an instance can be represented by a bounding box in object detection, a pixel-wise mask in instance segmentation, and a sequence of key points.

Recently, single-shot instance-level detection methods [34, 37, 47, 48, 67, 74, 78, 46, 71, 63] have gained interest. Single-shot methods are aimed at accelerating model inference time while maintaining good detection performance, by designing new network modules and architectures. While network design is an important factor in determining runtime efficiency [44, 5, 32, 28, 9, 82, 12, 81, 1], it is not the only way. In this paper, we take a new direction to improve runtime efficiency through input resolution reduction, without structurally modifying the network significantly.

2.2. Knowledge Distillation

[22] is a seminal work in knowledge distillation, which can be used to train a compact network by distilling knowledge from a larger teacher network. Over the years, many improved KD methods have been proposed that perform distillation over spatial attention [77], intermediate features [50, 21, 62], relational representation [43, 65], improved teachers [6, 40], etc. In contrast to existing KD methods that focus on improving the performance of compact networks, the focus of this paper is on improving the low-resolution model with KD which poses unique challenges.

2.3. Improving Low-Resolution Models

Multi-scale training is commonly used in image classification [30, 42, 68, 73, 31] and object detection [17, 55, 56] to improve model robustness against input resolution variation. It is an easy approach to improve performance at multiple or even low input resolutions. Conventional multi-scale training does not involve KD and thus generally does not guarantee spatially-aligned features required for seamless KD in instance-level detection. There is work to apply KD to improve low-resolution image classification in conjunction with multi-scale training [42, 68, 73, 31]. This strategy is straightforward for image classification given that the multi-resolution models share the same output size. However, it is not the case for instance-level detection and there is difficulty to apply KD. To circumvent this difficulty, we propose an aligned multi-scale training method.

3. Methodology

Fig. 2 provides an overview of the proposed framework. The framework is divided into two stages where the first (left) trains a strong multi-resolution teacher, and the second stage (right) trains a low-resolution one with the guidance of the multi-resolution teacher. In the following, we revisit a base detection method. For convenience, we adopt object detection as the base task to demonstrate our method, and take a strong one-stage detector FCOS [64] as the base

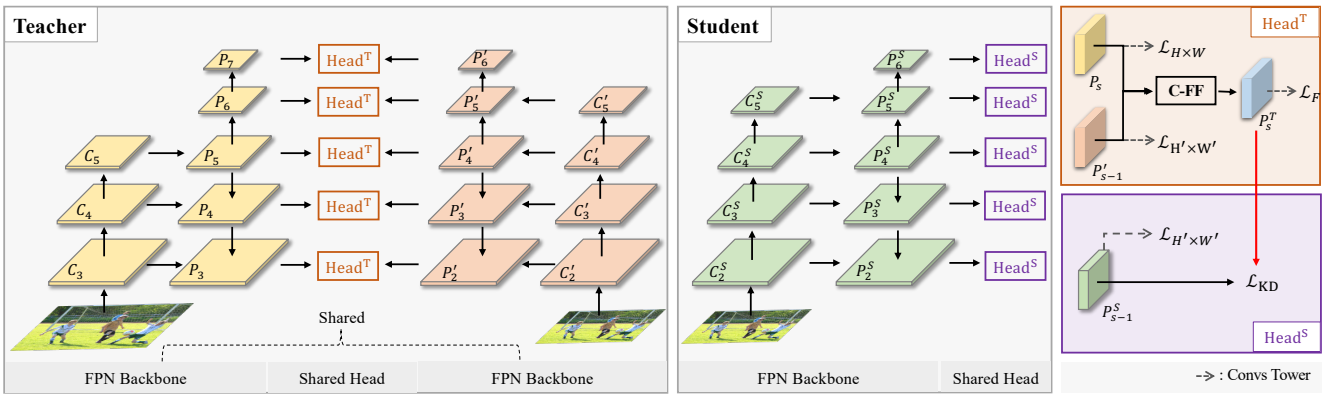


Figure 2: Overview of the proposed multi-scale aligned distillation framework. k=2 is used here for illustration. "Convs Tower" refers to the convolution blocks in the detection head [64]. In the first stage, we train a multi-scale teacher (T) that uses the same FPN [33] backbone for both high- and low-resolution input in an **aligned multi-scale training** fashion. The pyramid features from the two input resolutions are dynamically fused using the crossing feature-level fusion (C-FF) module. In the second stage, the trained multi-scale fusion teacher guides the low-resolution student (S) training via the distillation loss $\mathcal{L}_{\text{KD}}$.

detector. Our methods are applicable to other instance-level detection tasks. Then, we introduce the teacher formation process that involves our proposed aligned multi-scale training and crossing feature-level fusion. Finally, with a strong multi-resolution teacher, we propose crossing feature-level knowledge distillation to guide the training of low-resolution student effectively.

3.1. Base Detection Architecture

As shown in the left of Fig 2, our framework is based on FCOS [64], in which FPN [33] backbone and a detection head perform pixel-level box prediction and regression to achieve object detection.

FPN backbone adopts a feature pyramid scheme to compute features at multiple scales for detecting objects at different sizes. Specifically, FPN backbone extracts feature maps $P_s \in \mathbb{R}^{H^{P_s} \times W^{P_s} \times 256}$ of several resolutions at different FPN levels from the input image $I \in \mathbb{R}^{H \times W \times 3}$, where H and W denote the height and width of the image respectively. H^{P_s}, W^{P_s} refer to height and width of FPN feature maps, where $s \in \{2, 3, 4, 5, 6, 7\}$ indexes the *level* of the multi-scale feature maps generated by FPN. The FPN feature maps are spatially smaller than the input image by factors of $\{4, 8, 16, 32, 64, 128\}$.

The detection head has two output branches: classification branch and regression branch. Each has four convolutional blocks with convolutional layers and rectified linear unit (ReLU) layers. These blocks are shared among all FPN levels from P_2 to P_7. FPN features go through the detection head to perform instance-level classification and box regression. It trains with the following loss for a high-resolution input image with height H and width W:

$$\mathcal{L}_{H \times W} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{ctr}}, \tag{1}$$

where $\mathcal{L}_{\text{cls}}$ is the classification loss, $\mathcal{L}_{\text{reg}}$ is the bounding box regression loss, and $\mathcal{L}_{\text{ctr}}$ is the centerness loss.

3.2. Multi-Scale Fusion Teacher

One of the most important factors for knowledge distillation is a strong teacher. Here, we focus on feature-level knowledge distillation that distills through the FPN's pyramidal features. In this section, we propose methods to train a teacher to distill strong multi-resolution knowledge at feature level [50], to guide the training of a low-resolution student.

Note we do not consider final output-level (*e.g.*, classification, regression outputs) knowledge distillation in this paper. The output-level knowledge distillation works well for image classification tasks. But it is not straightforward to apply output-level distillation to instance-level detection due to the extreme imbalance of background and foreground classes. Further, feature-level knowledge distillation is more generally applicable than output-level distillation where the latter may involve different loss functions for various instance-level tasks.

Although the most straightforward approach is to train a single-scale high-resolution teacher to guide the student, the single-scale teacher is not aware of multi-resolution input. Thus its learned features may be poorly compatible with those of low-resolution student. To address this problem, we adopt and extend the widely-used multi-scale training strategy to train a strong multi-scale teacher, via feature pyramid alignment.

It is common knowledge that multi-scale training only improves the teacher network at the network parameter/weight level and does not explicitly incorporate multi-scale information to the features of an input image at a

given single resolution. To distill knowledge with enhanced multi-scale information to a low-resolution student, we introduce crossing feature-level fusion to dynamically fuse two-resolution features generated by the same teacher network on two input resolutions.

Aligned Multi-Scale Training. Multi-scale training perturbs the base input resolution (H, W) by rescaling it with a random scaling factor $\hat{\alpha}$ sampled from the range of $[\alpha_{\text{min}}, \alpha_{\text{max}}]$ (*e.g.*, [0.8, 1.0]) at every training iteration. It can be seen as training multiple models that act at different perturbed input resolutions and share the same network parameters/weights. Within the same network, the high-resolution models supposedly have strong knowledge that can be distilled to the low-resolution models. However, knowledge distillation is nontrivial due to the spatial size mismatch between the output and feature maps of models acting at different input resolutions.

In the FPN structure, the spatial sizes of any two adjacent pyramidal feature maps differ by $2\times$ along each spatial dimension. Motivated by this observation, we adopt two base resolutions $((H, W), (H', W'))$ respectively for high- and low-resolution models that share the same network weights, where $H' = H/k$, $W' = W/k$, and k is any *valid*[1] even number. Such a reduction factor allows us to shift the FPN's position at backbone network and obtain FPN pyramidal feature maps whose spatial sizes match those of high-resolution model.

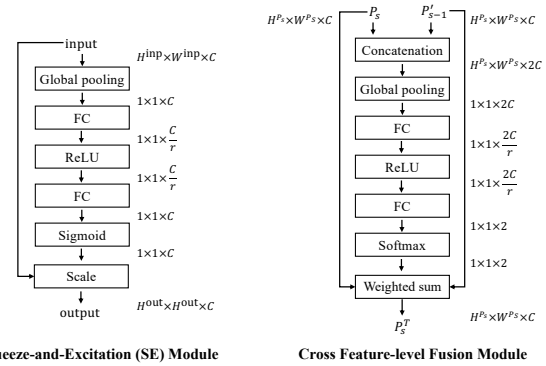
For clarity, we denote this shift offset as m, where $m = k/2$. In FCOS, a low-resolution (k=2 and m=1) model outputs pyramidal feature maps at $\{P_2', P_3', P_4', P_5', P_6'\}$ levels that have the same spatial sizes as the default $\{P_3, P_4, P_5, P_6, P_7\}$ levels of high-resolution model. During training, both high- and low-resolution models are trained simultaneously in the multi-scale training fashion with distinctly-sampled $\hat{\alpha}$.

Put differently, the low-resolution model uses earlier network blocks to generate pyramidal features in order to spatially match the pyramidal features of the high-resolution model. With this alignment approach, all models trained using the lower-resolution input (obtained by varying k) is *aligned* with the high-resolution model in terms of the pyramidal feature map sizes. This approach critically eliminates the feature-size inconsistency between models of multiple input resolutions and is beneficial for crossing-resolution knowledge distillation. The proposed aligned multi-scale training loss is defined as

$$\mathcal{L}_{\text{Align}} = \mathcal{L}_{H\times W} + \mathcal{L}_{H'\times W'}, \tag{2}$$

where $\mathcal{L}_{H\times W}$ and $\mathcal{L}_{H'\times W'}$ are the losses for default-/high-resolution and low-resolution input respectively. For simplicity, we include just one low-resolution ($H' \times W'$) model in the loss function. The aligned multi-scale training can be easily extended to include multiple low-resolution models.

[1] k preserves the expected number of FPN levels in the model.



Squeeze-and-Excitation (SE) Module **Cross Feature-level Fusion Module**

Figure 3: Comparison between Squeeze-and-Excitation [25] module and our proposed crossing feature-level fusion module. In our module, P_s and P'_{s-1} refer to the features generated by high- and low-resolution input respectively.

Cross Feature-level Fusion. Intuitively, a high-resolution model performs better for small object detection due to better preservation of fine visual details in high-resolution images. Whereas, a low-resolution model works better for large object detection, as the backbone network captures information on the larger portions of the whole image, compared to high-resolution models at the same receptive field. This intuition is verified experimentally in Table 1. Although aligned multi-scale training encourages a network to be robust against multiple input resolutions, the network runs on just one of the "seen" input resolutions during inference. Thus, its predicted features for any of the input resolutions do not incorporate the *best of both worlds* from high- and low-resolution models.

Inspired by the Squeeze-and-Excitation (SE) module [25], we propose a feature fusion module to dynamically fuse pyramidal feature maps from different resolution input in an input-dependent manner. Fig. 3 illustrates our feature fusion module. It enables the network to adjust the degrees of contributions from different resolution input, depending on the content of the input image.

For example, an image with only large objects benefits more from features of low-resolution model, and vice versa. With this fusion module, the resulting features much improve. Note that, in this paper, we only consider two-resolution input for easy demonstration of the idea. Actually, this module is readily extensible to fuse features in multiple resolutions. For each pair of feature maps, which share the same spatial sizes (*e.g.*, P_2 and P_3 from low-resolution and high-resolution input with aligned multi-scale training), the fusion scores for each of the pair are dynamically predicted and used to fuse or combine them.

Initially, the module concatenates the two feature maps

along the channel dimension and performs global average pooling to obtain 1D contextual features of

$$P_s^p = \frac{1}{H^{P_s} \times W^{P_s}} \sum_{i=1}^{H^{P_s}} \sum_{j=1}^{W^{P_s}} [P_s, P'_{s-m}](i,j), \tag{3}$$

where $s \in \{3,4,5,6,7\}$ is for FCOS, P and P' are the pyramid feature maps of default-/high-resolution and low-resolution input images respectively. H^{P_s} and W^{P_s} are the height and width of the feature map P_s. $[\cdot]$ indicates the concatenate operation for two feature maps along the channel dimension. P_s^p is then fed to a multi-layer perceptron (MLP) denoted as $\mathcal{H}$ to obtain Softmax-normalized fusion weights for the weighted sum that fuses[2] the two feature maps of

$$P_s^T = h_s\{0\} \cdot P_s + h_s\{1\} \cdot P'_{s-m}, \tag{4}$$

where $h_s = \mathcal{H}(P_s^p)$ and $\mathcal{H}$ is a MLP that comprises a FC layer with $\frac{2C}{r}$ output channels (r is the channel compression ratio), ReLU function, a FC layer with 2 output channels, and Softmax function. $\{0\}$ and $\{1\}$ are the indexing mechanisms used to obtain the fusion scores for P_s and P'_{s-1} feature maps, respectively. In contrast to SE module that employs Sigmoid function and treats the output channels independently (*i.e.*, multimodal distribution), we apply softmax normalization to explicitly encourage the "either-or" behavior (*i.e.*, unimodal distribution) through the competitions among input of different resolutions.

Subsequently, the fused P_s^T with strong multi-scale information is fed to the detection head for either training or inference. Given P_s^T, the training loss is defined as

$$\mathcal{L}_F = \lambda \cdot \mathcal{L}_{(H\&H')\times(W\&W')}, \tag{5}$$

where $(H\&H') \times (W\&W')$ indicates the use of the fused features from high- and low-resolution images. λ is the loss weight.

Training Strategies. Since the fused multi-scale features are stronger than single-scale features from either high- or low-resolution input, we take the fused model as the strong multi-scale teacher to guide the training of low-resolution student in the next subsection. To obtain the fused multi-scale teacher, we train it with a two-step strategy, where the first stage performs aligned multi-scale training, and the second stage only trains the fusion module while "freezing" the FPN and detection head. Alternatively, we can perform end-to-end training with joint aligned multi-scale training and feature fusion losses as

$$\mathcal{L}_T = \mathcal{L}_{\text{Align}} + \mathcal{L}_F. \tag{6}$$

[2]The same strategy is used to fuse two-/multi-resolution students with varying model complexity to counteract the loss of visual details needed for small object detection (see Table 7 and supplementary material).

We empirically show that the two training strategies produce similar teacher's detection performance.

3.3. Cross Feature-level Knowledge Distillation

With aligned multi-scale training and crossing feature-level fusion, we obtain a strong multi-scale fusion teacher whose multi-resolution features can be seamlessly distilled to the low-resolution student. Similar to those in previous sections, we denote high resolution and input resolution as $H \times W$ and $H' \times W'$ used by the teacher and student respectively. Knowledge is distilled from teacher's features P_s^T to student's P_{s-m}^S via L1 loss as

$$\mathcal{L}_{\text{KD}} = \tau \cdot \sum_s |P_s^T - P_{s-m}^S|, \tag{7}$$

where T and S respectively refer to teacher and student, s is the teacher's pyramidal feature level (*e.g.*, 3 to 7 for default input resolution in FCOS), m is the shift offset used to spatially align student's feature maps with the teacher's, and τ is the loss weight hyperparameter. Following conventional knowledge distillation [22, 50], the student is trained with both knowledge distillation loss and original detection loss, weighted by γ as

$$\mathcal{L}_S = \gamma \cdot \mathcal{L}_{\text{KD}} + (1-\gamma) \cdot \mathcal{L}_{H'\times W'}. \tag{8}$$

4. Experiments

Dataset & Evaluation Metrics. We compare our method with state-of-the-art approaches on the challenging COCO dataset [35]. Following common practice for COCO [16, 33, 36, 27], we use 115,000 training images and report evaluation results on the 5,000 validation images for the ablation experiments. Results on the 20,000 test-dev images are also reported for further comparison. We follow standard average precision metrics of AP (IoU range of 0.5:0.95:0.05), AP_{50} (IoU@0.5), AP_{75} (IoU@0.75), $\text{AP}_{\mathbb{S}}$ (small-sized objects), $\text{AP}_{\mathbb{M}}$ (medium-sized objects), and $\text{AP}_{\mathbb{L}}$ (large-sized objects). To avoid confusion, we specify $\mathbf{AP}^{\boldsymbol{T_1}}$ for the teacher with only multi-scale training, $\mathbf{AP}^{\boldsymbol{T_2}}$ for the full multi-scale fusion teacher, and $\mathbf{AP}^{\boldsymbol{S}}$ for the student.

Implementation Details. All ablation experiments are conducted with FCOS with ResNet-50 [19] backbone. For our final method, we perform evaluation on another popular detection method (RetinaNet [34]), and on other key instance-level tasks – instance segmentation (Mask R-CNN [16]), and keypoint detection (Mask R-CNN [16]). Following existing practice in detection frameworks [69, 2], we train either teacher or student networks using batch size 16 for 12 epochs ($90,000$ iterations or $1\times$ schedule) in ablation experiments.

In general, we use high resolution $(800, 1,333)$ and set $(400, 677)$ as our low resolution. The first and second elements are image's short and maximum of long sides. For

Training	Input	AP^{T_1}	$AP^{T_1}_{50}$	$AP^{T_1}_{75}$	$AP^{T_1}_{\mathbb{S}}$	$AP^{T_1}_{\mathbb{M}}$	$AP^{T_1}_{\mathbb{L}}$
Single-scale	H	38.6	57.6	41.8	23.0	42.4	49.9
	L	34.1	51.6	36.1	15.0	37.6	50.8
Vanilla Multi-scale	H	40.3	59.3	43.8	25.6	44.4	51.5
	L	35.9	53.2	38.1	15.8	39.4	54.0
Aligned Multi-scale (ours)	H	40.1	58.4	43.5	27.4	44.3	49.8
	L	37.8	55.7	40.6	18.9	40.5	54.4

Table 1: Ablation study on multi-scale training for teacher. In single-scale training, $\hat{\alpha}$ is fixed to 1.0. H and L in column "Input" indicates whether the inference is carried out with high- or low-resolution input.

multi-scale training, α_{min} and α_{max} are set to 0.8 and 1.0 respectively. The two resolutions are denoted as **H** and **L**.

Stochastic gradient descent (SGD) with learning rate 0.01 is used as the optimizer. We decay the learning rate with 0.1 after 8 and 11 epochs for $1\times$ training schedule and scale these epoch numbers proportionally for longer training schedules. For teacher backbone networks, we initialize them with ImageNet pretrained models. Teacher and student models share the same training setting, except that we consider only single-scale low-resolution images for the student.

By default, every low-resolution student shares the same backbone architecture and is initialized by its multi/high-resolution teacher. Note that we do not tune the hyperparameters for optimal performance. For all experiments (unless otherwise specified), we set λ, γ, and τ to 1.0, 0.2, and 3.0 respectively. The performance updates with respect to various hyperparameter values, as shown in Fig. 4.

4.1. Ablation Study

Our proposed framework consists of two main stages: multi-scale fusion teacher training and crossing feature-level knowledge distillation. We provide ablation study on them.

4.1.1 Multi-Scale Fusion Teacher

Aligned Multi-scale Training. The ablation study on multi-scale training is provided in Table 1. The teacher model trained with *single-scale training* achieves 38.6(H)/34.1(L) AP. With *vanilla multi-scale training* using input of two base resolutions, the model obtains 40.3(H)/35.9(L) AP.

The proposed aligned multi-scale training (with two base resolutions and feature map alignment) enables the teacher model to achieve better performance balance between H and L input resolutions at 40.1(L)/37.8(H) AP. In particular, it improves over *vanilla multi-scale training* by 1.9 AP on low-resolution input. The gap between H and L is smaller than that of vanilla approach, indicating that the feature map alignment is important. It ensures that the model perform robustly against large variation of input resolutions.

Cross Feature-level Fusion. Table 2(a) shows the effects of using different feature fusion strategies to fuse features of multi-resolution teacher models, compared to models without fusion. It reveals that models that employ feature fusion outperform single-resolution non-fusion ones, due to the incorporation of multi-resolution information in the fused features. Among the fusion models, our crossing feature-level fusion module that fuses features with dynamic fusion weights achieves better performance than both SC-SUM and CC that fuse features with static weights.

Table 2(b) demonstrates the effects of changing output activation function and generating channel-wise output (*i.e.,* $1\times1\times2C$) in the crossing feature-level fusion (C-FF) module. Substituting the softmax activation in C-FF with Sigmoid (SE [25] module) degrades the AP by 0.9%. Softmax encourages the module to be more decisive when selecting features from either of the input resolutions, making the features not include unimportant resolutions. Generating channel-wise output to weight the two resolution's features does not bring any improvement.

Training Strategy. Fig 4(a) shows multi-scale fusion teacher's performance using different training strategies and λ. Two-step and joint training perform similarly. The best AP is achieved at $\lambda = 0.4$.

4.1.2 Cross Feature-level Knowledge Distillation

Choice of Teacher. We analyze how different teacher variants affect low-resolution students' performance in Table 2(c). It shows that stronger teachers produce stronger students, and multi-scale fusion teachers produce the strongest students. We conjecture that multi-scale fusion teachers have greater compatibility with low-resolution students, due to the inclusion of low-resolution model's features in the feature fusion process.

Knowledge Distillation Methods. In Table 3(a), we study the effect of using different distillation methods. Feature-level distillation methods work better than traditional output-level KD [22], since intermediate features are more informative. With our alignment approach, all feature-level distillation methods perform quite similarly.

Loss Balancing Weights. Table 4(b) shows the influence of the loss balancing weights τ and γ on student's performance. γ plays an important role in student's training, by balancing the knowledge distillation and original detection losses. τ can be easily tuned once γ is fixed.

Inference at Multiple Resolutions. Table 4(c) shows the AP performance of the student trained with three base resolutions, including 200-pixel (short side) resolution with $P1$-$P5$ features via our alignment approach. Even at the small

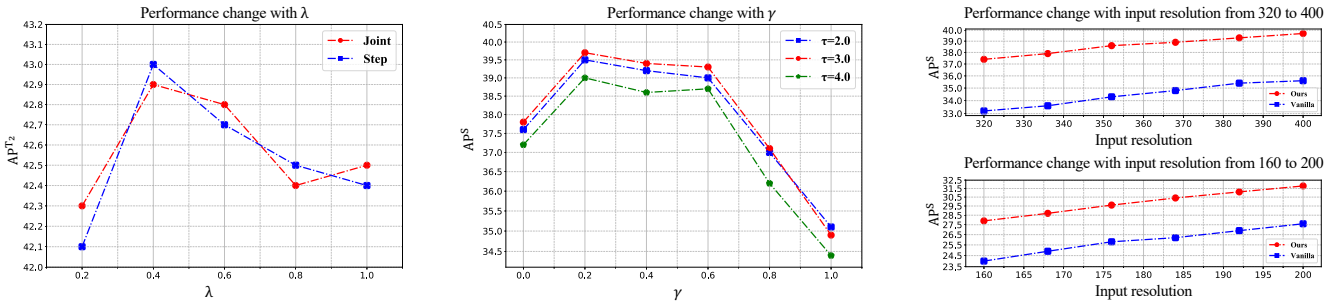


Figure 4: **(a)**: Performance change of teachers with respect to λ and training strategy; **(b)**: Performance change of teachers with respect to γ and τ, **(c)**: Performance change of students with respect to inference resolution (320 to 400 & 160 to 320).

Fusion approach	AP^{T_2}	$AP^{T_2}_{50}$	$AP^{T_2}_{75}$
No fusion (H)	40.1	58.4	43.5
No fusion (L)	37.8	55.7	40.6
SC-SUM	40.3	59.2	43.6
CC	41.4	60.3	44.7
C-FF	42.5	61.1	46.3

(a)

Design of C-FF	AP^{T_2}	$AP^{T_2}_{50}$	$AP^{T_2}_{75}$
No fusion (H)	40.1	58.4	43.5
No fusion (L)	37.8	55.7	40.6
Sigmoid [25]	41.6	59.7	45.0
Channel-wise	42.5	61.0	46.2
C-FF	42.5	61.1	46.3

(b)

Teacher	AP^{T_2}	AP^S	AP^S_{50}	AP^S_{75}
800	40.1	38.2	56.0	40.9
800&400 (CC)	41.4	38.8	56.7	41.5
800&400 (Sigmoid)	41.6	38.9	56.6	41.8
800&400 (C-FF) λ=0.4	43.0	39.9	58.2	42.4
800&400 (C-FF) λ=1.0	42.5	39.7	58.0	42.5

(c)

Table 2: Ablation studies. **(a)**: **Feature feature fusion approaches**: "SC-SUM" sums the *separately convolved* feature maps of two resolutions, "CC" applies convolution to the concatenated features of the two resolutions, and "C-FF" is our *crossing feature-level fusion*. **(b)**: **Design of crossing feature-level fusion module**: "Sigmoid" replaces the softmax with sigmoid activation as used in SE module [25]. "Channel-wise" outputs channel-wise weights for feature fusion. **(c)**: **Effectiveness of different teachers** on guiding low-resolution student: single-scale 800px (high resolution) teacher and different multi-scale (800px & 400px) fusion teachers with CC, sigmoid activation, or our crossing feature-level fusion.

Distillation	Type	Aligned	AP^S	AP^S_{50}	AP^S_{75}
KD [22]	Output	○	36.4	54.0	38.6
		✓	37.5	55.8	39.7
FGFI [66]	Feature	○	36.6	54.3	38.8
		✓	39.5	57.9	42.2
PODNet [11]		✓	38.9	57.0	41.2
SKD [38]	Attention	✓	39.7	57.9	42.5
Ours	Feature	✓	39.7	58.0	42.5

(a)

Input	Backbone	Width	AP^S	AP^S_{50}	AP^S_{75}	$AP^S_{\mathbb{S}}$	$AP^S_{\mathbb{M}}$	$AP^S_{\mathbb{L}}$	$GFLOPS_b$
L	R-50	0.25×	30.4	46.5	32.3	15.1	31.5	44.1	2.85
		0.50×	36.1	53.7	38.8	18.4	38.2	50.4	8.56
		0.75×	38.6	56.7	41.3	20.8	41.2	53.8	19.98
		1.00×	39.7	58.0	42.5	21.7	42.9	55.0	36.03
	R-101	1.00×	41.6	60.1	44.9	24.0	45.7	57.8	55.83
	X-101	1.00×	43.1	61.7	46.5	25.7	47.3	59.4	74.57
H	R-50	1.00×	42.3	61.3	45.9	28.2	45.9	53.5	142.69

(b)

Table 3: Ablation studies. **(a)**: **Distillation methods**. "Type" indicates the type of knowledge distilled and "Aligned" indicates whether the student's feature maps are aligned with the teacher's or not. All rows use the same multi-scale fusion teacher trained with C-FF. **(b)**: **Backbone architectures & network widths**. "$GFLOPS_b$" is the number of *floating point operations* of the backbone network in giga unit. We average the GFLOPS computed over the 5,000 images per resolution type. The detection head is not included as it has a fixed GFLOPS of 57.09 regardless of the backbone architecture.

200-pixel resolution, our student achieves reasonably good AP 31.8.

Different Backbones. We present the results of using different backbone architectures and network widths [75] to train teacher-student pairs using our framework in Table 3(b). Here, teacher and student share the same backbone. Our multi-scale fusion teacher can also be used to train a strong high-resolution student (last row). Remarkably, our low-resolution ResNeXt-101 [72] (second last row) student outperforms the high-resolution R-50 student, at half of the required FLOPs.

Our framework works very well with the slimmed/compact pre-trained backbones[3] [75] at multiple network widths. More results on slimmed backbones are reported in Table 5. Additionally, we report the results of using distinct teacher and student backbones in Table 6. Our approach benefits from using a stronger backbone for

[3]Slimmable training [75] is not used for detector training.

Task	Detection Method	Training	Input	AP	AP_{50}	AP_{75}	$AP_{\mathbb{S}}$	$AP_{\mathbb{M}}$	$AP_{\mathbb{L}}$
Object Detection	FCOS [64]	Vanilla	H	42.8	62.4	46.7	27.3	46.5	55.2
			L	38.2	56.3	41.0	17.8	42.2	56.7
		Ours	L	41.6 (+3.4)	59.9 (+3.6)	44.9 (+3.9)	22.8 (+5.0)	44.8 (+2.6)	57.0 (+0.3)
	RetinaNet [34]	Vanilla	H	40.7	60.7	43.4	26.8	44.2	50.3
			L	37.2	55.7	39.5	16.7	42.3	55.7
		Ours	L	40.3 (+3.1)	59.4 (+3.7)	44.2 (+4.7)	21.6 (+4.9)	43.7 (+1.4)	55.5 (-0.2)
Instance Segmentation	Mask R-CNN [16]	Vanilla	H	39.8	57.6	42.9	19.8	44.5	57.8
			L	35.2	55.8	37.8	13.8	37.6	56.8
		Ours	L	37.3 (+2.1)	58.2 (+2.4)	39.7 (+1.9)	16.5 (+2.7)	39.4 (+1.8)	57.3 (+0.5)
Keypoint Detection	Mask R-CNN [16]	Vanilla	H	66.4	87.1	72.8	63.5	72.0	73.6
			L	62.9	86.2	68.4	56.8	73.4	70.3
		Ours	L	65.0 (+2.1)	88.5 (+2.3)	69.9 (+1.5)	59.3 (+2.5)	75.3 (+1.9)	70.9 (+0.5)

Table 4: Overall performance evaluation on mainstream instance-level detection tasks with ResNet-50. "Vanilla" refers to the standard multi-scale training. "Ours" is the proposed framework with fusion teacher and crossing feature-level distillation. Here, we adopt $3\times$ training schedule to understand how our approach performs in the "push-the-envelope" regime [15].

Width	Role	Input	AP	AP_{50}	AP_{75}	$AP_{\mathbb{S}}$	$AP_{\mathbb{M}}$	$AP_{\mathbb{L}}$
$1.0\times$	T	H	39.7	57.9	43.2	27.2	44.0	49.3
		L	37.8	55.7	40.6	18.9	40.5	54.4
		H&L	42.5	61.1	46.3	28.0	45.7	55.7
	S	L	39.7	58.0	42.5	21.7	42.9	55.0
$0.5\times$	T	H	36.0	53.5	39.0	23.2	39.8	44.7
		L	33.8	50.8	36.4	16.3	35.4	49.1
		H&L	38.5	56.3	41.7	23.5	40.9	50.6
	S	L	36.1	53.7	38.8	18.4	38.2	50.4

Table 5: Performance evaluation on slimmed ResNet-50 [75] backbones. H&L is the multi-scale fusion teacher that distills knowledge to student S.

Teacher (H&L)	Student (L)	AP^S	AP^S_{50}	AP^S_{75}
R-50	R-50	39.7	58.0	42.5
R-101	R-50	40.5	58.8	43.7
	R-101	41.6	60.1	44.9

Table 6: Performance evaluation on multi-scale fusion teacher and low-resolution student models using distinct backbone architectures.

Width (H)	Width (L)	AP^S	AP^S_{50}	AP^S_{75}	$AP^S_{\mathbb{S}}$	$AP^S_{\mathbb{M}}$	$AP^S_{\mathbb{L}}$
$0.50\times$	$1.00\times$	41.4	59.9	44.5	24.6	44.5	54.6
$0.50\times$	$0.75\times$	41.1	59.4	44.0	24.5	43.6	54.5
$0.50\times$	$0.50\times$	40.1	58.5	43.2	24.5	42.7	51.8
$0.25\times$	$0.50\times$	37.6	55.5	40.6	20.8	40.0	50.9

Table 7: Performance evaluation on using dual-resolution (high/H and low/L input resolutions) slimmed backbones within multi-scale fusion student models.

the teacher, like traditional KD [22].

Multi-scale Fusion Students with Slimmed Backbones. In Table 7, we show the performance of several such backbone combinations for the multi-scale fusion students that requires less computational footprints (FLOPS) than the full-width high-resolution model. Performance on small-sized objects $AP^S_{\mathbb{S}}$ is much improved compared to the models' single- and low-resolution counterparts reported in the supplementary material.

4.2. Overall Performance Evaluation

We apply our proposed framework to several recent methods across mainstream instance-level detection tasks of object detection, instance detection, and keypoint detection. The evaluation results are reported in Table 4. We notice that the low-resolution models trained with our approach outperform the widely-adopted *vanilla multi-scale training* approach by 2.1% to 3.6%, and performs competitively with its high-resolution models.

5. Conclusion

In this paper, we have boosted the performance on low-resolution instance-level detection tasks using our proposed framework. The framework comprises aligned multi-scale training and crossed feature-level fusion for training a strong teacher that dynamically fuses features from high-resolution and low-resolution input. By aligning the feature maps of teacher and student, knowledge of the multi-scale fusion teacher is correctly and effectively distilled to the low-resolution student.

The extensive experiments demonstrate that our approach improves even strong baseline and vanilla multi-scale trained models by significant margins. Moreover, the proposed low-resolution detection approach is compatible with and complements compact networks (obtained with model compression techniques [75]) to reduce overall model complexity. We can extend our crossing feature-level fusion module to combine two lightweight models to achieve better instance-level detection performance, while maintaining low computational cost.

References

[1] Miguel A Carreira-Perpinán and Yerlan Idelbayev. "learning-compression" algorithms for neural net pruning. In *CVPR*, 2018. 1, 2

[2] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. 2019. 5

[3] Xinlei Chen, Ross Girshick, Kaiming He, and Piotr Dollár. Tensormask: A foundation for dense object segmentation. In *ICCV*, 2019. 2

[4] Yukang Chen, Tong Yang, Xiangyu Zhang, Gaofeng Meng, Xinyu Xiao, and Jian Sun. Detnas: Backbone search for object detection. In *NeurIPS*, 2019. 1

[5] Ting-Wu Chin, Ruizhou Ding, Cha Zhang, and Diana Marculescu. Towards efficient model compression via learned global ranking. In *CVPR*, 2020. 1, 2

[6] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *ICCV*, 2019. 2

[7] Ruihang Chu, Yifan Sun, Yadong Li, Zheng Liu, Chi Zhang, and Yichen Wei. Vehicle re-identification with viewpoint-aware metric learning. In *ICCV*, 2019. 1

[8] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN: object detection via region-based fully convolutional networks. In *NeurIPS*, 2016. 2

[9] Xiaohan Ding, Xiangxin Zhou, Yuchen Guo, Jungong Han, Ji Liu, et al. Global sparse momentum sgd for pruning very deep neural networks. In *NeurIPS*, 2019. 1, 2

[10] Piotr Dollár and C Lawrence Zitnick. Fast edge detection using structured forests. In *PAMI*, 2015. 1, 2

[11] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *ECCV*, 2020. 7

[12] Xitong Gao, Yiren Zhao, Łukasz Dudziak, Robert Mullins, and Cheng-zhong Xu. Dynamic channel pruning: Feature boosting and suppression. *ICLR*, 2019. 1, 2

[13] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1, 2

[14] Qiushan Guo, Xinjiang Wang, Yichao Wu, Zhipeng Yu, Ding Liang, Xiaolin Hu, and Ping Luo. Online knowledge distillation via collaborative learning. In *CVPR*, 2020. 1

[15] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *ICCV*, 2019. 8

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *ICCV*, 2017. 1, 2, 5, 8

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014. 2

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[20] Yihui He, Ji Lin, Zhijian Liu, Hanrui Wang, Li-Jia Li, and Song Han. Amc: Automl for model compression and acceleration on mobile devices. In *ECCV*, 2018. 1

[21] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In *ICCV*, 2019. 2

[22] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *arXiv*, 2015. 2, 5, 6, 7, 8

[23] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *ICCV*, 2019. 1

[24] 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. In *arXiv*, 2017. 1

[25] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 1, 4, 6, 7

[26] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 1

[27] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *CVPR*, 2019. 5

[28] Jangho Kim, SeongUk Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer. In *NeurIPS*, 2018. 1, 2

[29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 1

[30] Jason Kuen, Xiangfei Kong, Zhe Lin, Gang Wang, Jianxiong Yin, Simon See, and Yap-Peng Tan. Stochastic downsampling for cost-adjustable inference and improved regularization in convolutional networks. In *CVPR*, 2018. 2

[31] Duo Li, Anbang Yao, and Qifeng Chen. Learning to learn parameterized classification networks for scalable input images. In *ECCV*, 2020. 1, 2

[32] Yuchao Li, Shaohui Lin, Baochang Zhang, Jianzhuang Liu, David Doermann, Yongjian Wu, Feiyue Huang, and Rongrong Ji. Exploiting kernel sparsity and entropy for interpretable cnn compression. In *CVPR*, pages 2800–2809, 2019. 1, 2

[33] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 2, 3, 5

[34] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 2, 5, 8

[35] 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 *ECCV*, 2014. 5

[36] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, 2018. 1, 2, 5

[37] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 2

[38] Yifan Liu, Changyong Shu, Jingdong Wang, and Chunhua Shen. Structured knowledge distillation for dense prediction. *TPAMI*, 2020. 7

[39] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *ECCV*, 2018. 1

[40] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *AAAI*, 2020. 2

[41] Douglas Morrison, Adam W Tow, M McTaggart, R Smith, N Kelly-Boxall, S Wade-McCue, J Erskine, R Grinover, A Gurman, T Hunn, et al. Cartman: The low-cost cartesian manipulator that won the amazon robotics challenge. In *ICRA*, 2018. 1

[42] Pramod Kaushik Mudrakarta, Mark Sandler, Andrey Zhmoginov, and Andrew Howard. K for the price of 1: Parameter-efficient multi-task and transfer learning. In *ICLR*, 2019. 2

[43] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *CVPR*, 2019. 2

[44] Antonio Polino, Razvan Pascanu, and Dan Alistarh. Model compression via distillation and quantization. In *arXiv*, 2018. 1, 2

[45] Lu Qi, Li Jiang, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Amodal instance segmentation with kins dataset. In *CVPR*, 2019. 1

[46] Lu Qi, Xiangyu Zhang, Yingcong Chen, Yukang Chen, Jian Sun, and Jiaya Jia. Pointins: Point-based instance segmentation. In *arXiv*, 2020. 2

[47] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 2

[48] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. In *arXiv*, 2018. 2

[49] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 1, 2

[50] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, 2015. 1, 2, 3, 5

[51] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 1

[52] Guang Shu. Human detection, tracking and segmentation in surveillance video. 2014. 1

[53] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017. 1

[54] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1

[55] Bharat Singh and Larry S Davis. An analysis of scale invariance in object detection snip. In *CVPR*, 2018. 2

[56] Bharat Singh, Mahyar Najibi, and Larry S Davis. Sniper: Efficient multi-scale training. In *NeurIPS*, 2018. 2

[57] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019. 1, 2

[58] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *arXiv*, 2016. 1

[59] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 1

[60] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 1

[61] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 1

[62] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *ICLR*, 2020. 2

[63] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *ECCV*, 2020. 2

[64] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 2019. 1, 2, 3, 8

[65] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *ICCV*, 2019. 2

[66] Tao Wang, Li Yuan, Xiaopeng Zhang, and Jiashi Feng. Distilling object detectors with fine-grained feature imitation. In *CVPR*, 2019. 7

[67] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations. In *arXiv*, 2019. 2

[68] Yikai Wang, Fuchun Sun, Duo Li, and Anbang Yao. Resolution switchable networks for runtime efficient image recognition. 2020. 2

[69] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 5

[70] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, 2018. 2

[71] Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. Polarmask: Single shot instance segmentation with polar representation. In *CVPR*, 2020. 2

[72] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. 7

[73] Taojiannan Yang, Sijie Zhu, Chen Chen, Shen Yan, Mi Zhang, and Andrew Willis. Mutualnet: Adaptive convnet via mutual learning from network width and resolution. In *ECCV*, 2020. 1, 2

[74] Hui Ying, Zhaojin Huang, Shu Liu, Tianjia Shao, and Kun Zhou. Embedmask: Embedding coupling for one-stage instance segmentation. In *arXiv*, 2019. 2

[75] Jiahui Yu, Linjie Yang, Ning Xu, Jianchao Yang, and Thomas Huang. Slimmable neural networks. In *ICLR*, 2019. 7, 8

[76] Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *CVPR*, 2020. 1

[77] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017. 2

[78] Rufeng Zhang, Zhi Tian, Chunhua Shen, Mingyu You, and Youliang Yan. Mask encoding for single shot instance segmentation. In *CVPR*, 2020. 2

[79] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, 2018. 1

[80] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. 2019. 1, 2

[81] Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression. In *arXiv*, 2017. 1, 2

[82] Zhuangwei Zhuang, Mingkui Tan, Bohan Zhuang, Jing Liu, Yong Guo, Qingyao Wu, Junzhou Huang, and Jinhui Zhu. Discrimination-aware channel pruning for deep neural networks. In *NeurIPS*, 2018. 1, 2