Fully Supervised and Guided Distillation for One-Stage Detectors

Deyu Wang[1], Dongchao Wen[1,*][0000−0001−7311−1842], Junjie Liu[1], Wei Tao[1],
Tse-Wei Chen[2], Kinya Osa[2], and Masami Kato[2]

[1] Canon Information Technology (Beijing) Co., LTD, China
{wangdeyu, wendongchao, liujunjie, taowei}@canon-ib.com.cn
[2] Device Technology Development Headquarters, Canon Inc., Japan
twchen@ieee.org

Abstract. Model distillation has been extended from image classification to object detection. However, existing approaches are difficult to focus on both object regions and false detection regions of student networks to effectively distill the feature representation from teacher networks. To address it, we propose a fully supervised and guided distillation algorithm for one-stage detectors, where an excitation and suppression loss is designed to make a student network mimic the feature representation of a teacher network in the object regions and its own high-response regions in the background, so as to excite the feature expression of object regions and adaptively suppress the feature expression of high-response regions that may cause false detections. Besides, a process-guided learning strategy is proposed to train the teacher along with the student and transfer knowledge throughout the training process. Extensive experiments on Pascal VOC and COCO benchmarks demonstrate the following advantages of our algorithm, including the effectiveness for improving recall and reducing false detections, the robustness on common one-stage detector heads and the superiority compared with state-of-the-art methods.

1 Introduction

With the rapid development of deep learning, there are an increasing number of practical applications with deep neural networks applied to intelligent devices, such as face recognition in mobile phones, human body detection in smart cameras, and pathogenic cell analysis in medical microscopes, etc. Since these applications are extremely demanding in terms of accuracy, speed and memory, some efficient network architectures are proposed, such as MobileNet [1], MobileNetV2 [2], ShuffleNet [3], ShuffleNetV2 [4] and IGCV2 [5]. In addition, some existing methods [6–8] mainly focus on physically pruning networks to reduce redundant weights of larger models and obtain thinner and shallower models. However, these methods only consider the effectiveness and compactness of network structures, but ignore to simulate the network potential on the premise

*Dongchao Wen is corresponding author.

of keeping structure unchanged. So, in order to achieve this, our intuition is to make compact student networks learn from larger teacher networks, because the teacher has more robust feature representation and can mine deeper information, which are valuable knowledge for guiding student networks.

An effective algorithm to induce the training of a student network by transferring experience from a teacher network is knowledge distillation [9]. Nowadays, there have been many well-designed distillation works, including the works for classification [10–13] and the works for detection [14–17]. However, existing methods decide what knowledge should be transferred mostly based on human experience or teacher's attention, but neglect to consider what knowledge the student wants to receive. As in the detection distillation methods, feature representation mimicry is generally based on full features or teacher's attention, which will undoubtedly bring two problems: (1) The former will introduce unnecessary computation and a large amount of noise from unimportant area. (2) The latter will ignore valuable knowledge in the background, because the teacher's attention tends to the foreground. Besides, most of works heavily rely on trained teacher models, which ignore the knowledge in the teacher's training process.

To tackle the mentioned limitations, we propose a fully supervised and guided distillation algorithm for one-stage detectors, which consists of three parts: (a) Inter-layer representation supervision. (b) Training process supervision. (c) Network output supervision. For the inter-layer supervision, we find that the regions where objects are detected show higher response in the feature maps and the high-response feature regions in the background are more likely to cause false detections. Therefore, in addition to mimicking the feature representation in the object regions, we also consider the importance of student's high-response feature regions and merge them into the mimicry regions to distill representation more effectively. For the training process supervision, the teacher and the student are initialized with ImageNet pre-trained models and trained together for detection task. The knowledge in the teacher's training process is continuously transferred to help the student find a better local minimum. For the output supervision, we use a multi-task loss to jointly perform the network training and mimicry learning. The contributions in this paper are summarized as follows:

- We present a novel fully supervised and guided distillation algorithm for one-stage detectors which achieves comprehensive coverage of distillation in the inter-layer representation, training process and network output.
- We design an excitation and suppression loss which innovatively considers from the perspective of the student about the importance of its high-response feature regions, so that it makes the student focus on mimicking the feature representation not only in the object regions but also in such high-response regions to improve recall and reduce false detections.
- We propose a process-guided learning strategy, where the teacher is trained along with the student and continuously transfers knowledge throughout the training process to help the student find a better local minimum.
- We verify our algorithm on the representative network structures by using public benchmarks and achieve compelling results.

The rest of paper is organized as follows. The related work is first given in Section 2. And then we elaborate and analyze our algorithm in Section 3. Next experiments are shown in Section 4. Lastly we conclude the paper in Section 5.

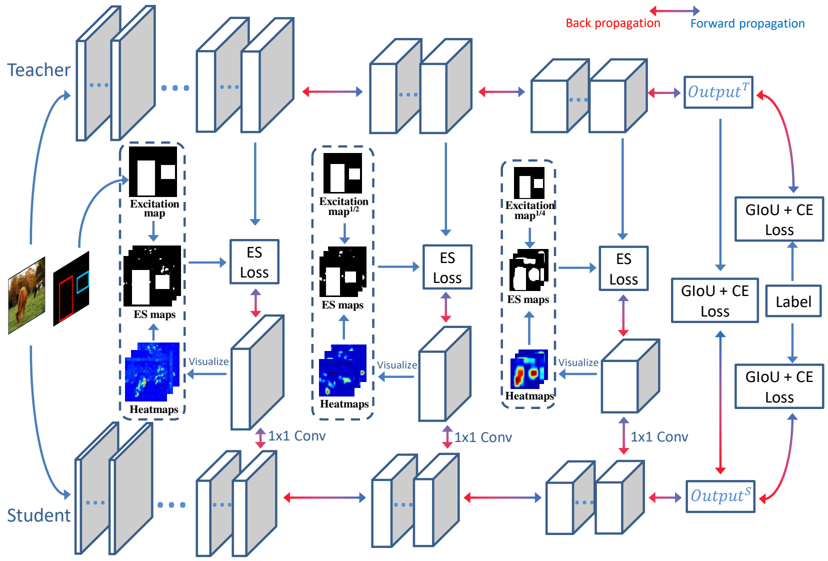


Fig. 1. Overall architecture of our fully supervised and guided distillation algorithm. The proposed architecture consists of three parts: inter-layer representation supervision, training process supervision and network output supervision. The ES loss is proposed for inter-layer representation supervision, which combines the object regions (excitation map) with the high-response feature regions (suppression map) of student network to generate ES maps, and using such maps as masks to make student network focus on mimicking the representation of teacher network in these regions. As for training process supervision, the teacher network receives labels to train along with the student network and transfer knowledge throughout the training phase. We employ a multi-task loss for network training and output distillation.

2 Related work

Knowledge distillation Inspired by pioneering works [9, 18], there are various related approaches proposed [19–23, 17]. For the distillation approaches focused on the task of classification, FitNets [24] devises a hint-based training approach to transfer representation of intermediate layers from a teacher network to a student network. Deep mutual learning proposed in [25] allows two networks to learn the output distribution from each other so as to improve performance together. Similarity-preserving knowledge distillation [26] uses the pairwise activation similarities within each input mini-batch to supervise the training of a student network with a trained teacher network. Flow-based method [27] generates distilled knowledge by determining the knowledge as the flow of the solving procedure calculated with the proposed FSP matrix. Attention transfer [28] generates activation-based and gradient-based spatial attention maps to transfer knowledge from a teacher network to a student network. VID [29] proposes a principled framework through maximizing mutual information between two networks based on the variational information maximization technique. A two structured knowledge distillation [30] is presented to enforce consistency of features and output between a student network and a teacher network.

A few recent works explore distillation for object detection. The fine-grained feature imitation method proposed in [14] makes a student network pay more attention to the feature learning on the near object anchor locations. [15] uses full feature imitation strategy for distilling representation, but we find this way brings degraded performance due to the introduction of a large amount of noise from unimportant regions. A mimic framework [31] is proposed to transfer knowledge based on the region of proposals, which is not applicable for one-stage detector. The distillation in [32] is used for multi-level features and pyramid ROI Aligned features. The latter serves for two-stage detectors, while for one-stage detectors, the method degenerates into full feature imitation of intermediate layers. An objectness scaled distillation [16] is proposed to make a student network focus on learning high score objects and ignore noisy candidates with low scores.
Object detection Deep learning has been widely used in object detection. There are two types of detectors: one-stage detectors [33–37] and two-stage detectors [38–42]. One-stage detectors are designed to be satisfied with the requirement of real-time detection and can be easily deployed into applications. With the development of one-stage detectors, many compact one-stage detectors are proposed such as ThunderNet [43] and PeleeNet [44], which are faster and require fewer resources. Two-stage detectors care more about the detection accuracy. Following the R-CNN [45, 46, 42] series, there are many efficient two-stage detectors proposed such as light-Head R-CNN [47] and Libra R-CNN [48], which further improve detection accuracy as well as speed up network inference.

3 Method

Figure 1 illustrates the overall framework of fully supervised and guided distillation algorithm (FSGD) which has three parts: (a) Inter-layer representation supervision. (b) Training process supervision. (c) Network output supervision. As for inter-layer representation supervision, we comprehensively consider the importance of feature expression in the object regions and the high-response regions of student network to propose an excitation and suppression loss function. In the training process supervision, a process-guided learning strategy is proposed to make a teacher network train along with a student network and continuously transfer knowledge. For network output supervision, we use a multi-task loss to optimize networks as well as transfer knowledge. In what follows, we elaborate these parts one by one.

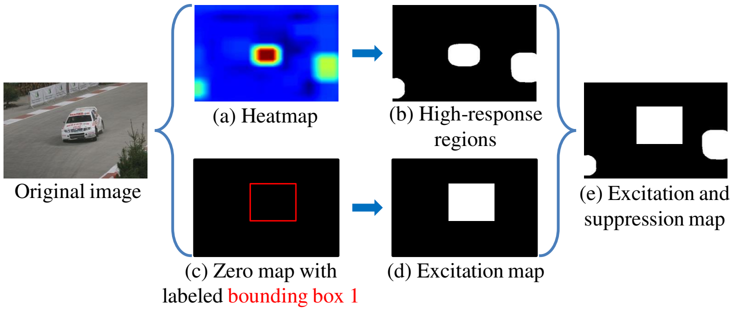

Original image
(a) Heatmap
(b) High-response regions
(c) Zero map with labeled bounding box 1
(d) Excitation map
(e) Excitation and suppression map

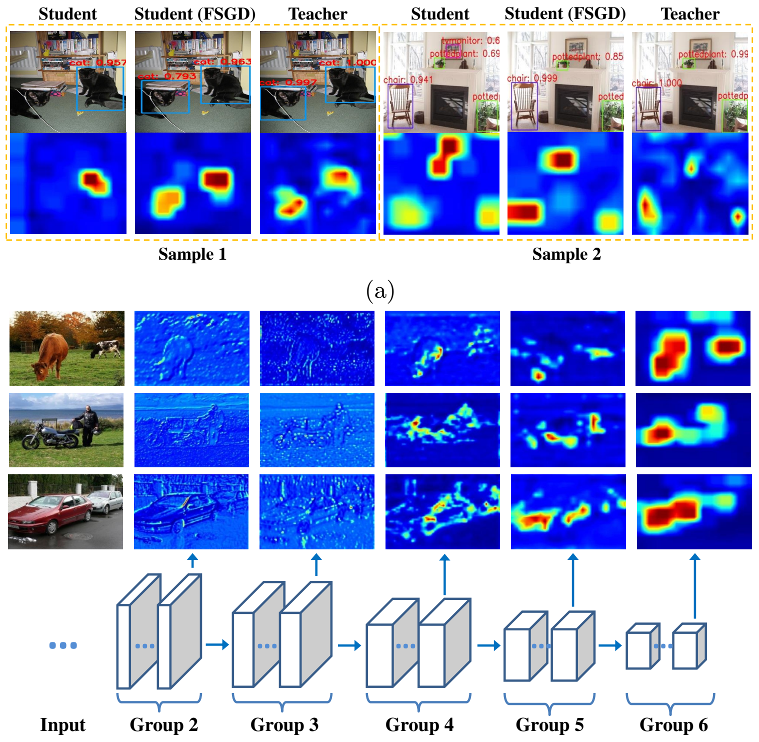
Fig. 2. Generation steps of an excitation and suppression map. (a) Visualization of the feature map from a student network. (b) High-response regions are generated based on (a) with its values greater than a threshold (Defined in Section 3.1 ES Loss). (c) The bounding box 1 obtained from ground truth is drawn in a zero map. (d) Generating an excitation map based on the bounding box region of (c). (e) Combining (b) and (d) by union operation to obtain an excitation and suppression map.

3.1 Inter-layer representation supervision

As mentioned in [28], not all the information in the feature maps of a teacher network is important for guiding a student network, and the knowledge transferred from valuable regions is more beneficial than from the overall feature maps. Inspired by it, we propose a novel excitation and suppression loss function to make a student focus on mimicking the feature representation in the object regions and its own high-response regions from a teacher so as to improve recall and reduce false detections.

Generation of excitation and suppression map (ES map) By visualizing the feature maps of detectors, there are usually many high-response regions at the location of objects. Thus, the features in the object regions are quite important for object detection. In order to help the student network mimic the feature representation in the object regions, we directly regard the bounding box regions of ground truth as object regions and add them into a zero map (an image whose all pixel values are zero) to get an excitation map as shown in Figure 2(c) and Figure 2(d). Then, this map is used as a mask to make the student network learn the feature representation in the object regions so as to excite feature expression in such regions. As shown in sample 1 and sample 2 of Figure 3(a), this way helps the student network reduce missed detections to improve recall, and promotes the precision of detected objects.

Additionally, although small networks often have the same representation capacity as large networks according to the observation [49, 18], it is hard to achieve the same representation level as large networks due to difficulty of optimization [49], so that the student may have more false detections in the complex background. Interestingly, like object regions, there are also many high-response feature regions at the location of false detections. That is, in these regions, the



(a)

(b)

Fig. 3. (a) Qualitative demonstration on the gain from our inter-layer representation supervision. The top row shows the detection results of two samples and the bottom row visualizes the corresponding feature maps at the end of the student backbone. (b) Visualization of feature maps at the end of each supervised groups for some samples. We can see that group 2 and group 3 still express some low-level features, but there are more semantic information in the feature maps of group 4, group 5 and group 6.

student's feature values are large, but for the teacher, since it has less false detection, most of feature values are small. Therefore, we can use the teacher's features as targets to guide the student to suppress its large feature values, so as to alleviate false detection. Specifically, for each supervised channel, we further merge the high-response feature regions of the student into the excitation map to generate an excitation and suppression map as illustrated in Figure 2(e). Then, the ES maps generated from all channels are used as masks to make the student focus on exciting feature expression in the object regions to improve recall and suppressing feature expression in the high-response regions to reduce false detections as shown in sample 2 of Figure 3(a).

Excitation and suppression loss (ES Loss) To ensure the same size of feature maps, the supervised layers of the student and teacher networks should be from the groups with the same scale. Besides, 1×1 convolution layers are introduced into the student network for addressing inconsistent number of channels. Then, we define s as the aligned feature maps of the student network and t as the corresponding feature maps of the teacher network, the ES loss function is defined as:

$$L_{ES} = \frac{1}{N_E + N_S} \sum_{i=1}^{W} \sum_{j=1}^{H} \sum_{c=1}^{C} (I_E \cup I_S^c)(s_{ijc} - t_{ijc})^2, \tag{1}$$

where I_E is the excitation mask generated based on ground truth, and I_S^c is the suppression mask of the cth channel in the feature maps, which is generated by using:

$$I_S^c = \cup_{x=1}^{W} \cup_{y=1}^{H} I(s_c, \alpha, x, y)(\neg I_E) \tag{2}$$

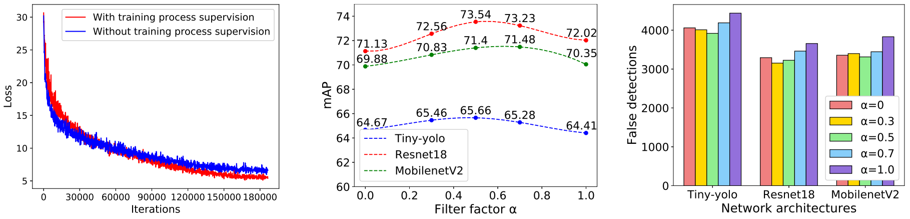
with an indicator function $I(s_c, \alpha, x, y) = \begin{cases} 1 & s_{xyc} > \alpha \times \max(s_c) \\ 0 & s_{xyc} \leq \alpha \times \max(s_c) \end{cases}$. Here W, H and C denote the width, height and channel of feature maps respectively. N_E is the number of excitation points in the excitation mask and N_S is the number of suppression points in the suppression mask. It is noted that α is a filter factor to control the generation of suppression regions. When $\alpha = 1$, only object regions are kept while all background regions are also included when $\alpha = 0$, and more details about the impact of α can be found in Section 4.3.

3.2 Training process supervision

Existing distillation methods for object detection are almost result-guided, which means that the knowledge is transferred from trained teacher models. However, we find that, for detection task, training a teacher network along with a student network and continuously transferring knowledge can help the student network converge better as shown in Figure 4. So, we use this training strategy in our algorithm and refer to it as process-guided learning.

Compared with the distillation methods based on trained teacher models, the process-guided learning is more effective for following reasons: (1) Compared with large networks, small networks are hard to train and find the right parameters that realize the desired function due to difficulty of optimization [49]. However, the process-guided learning can continuously transfer knowledge in the optimization process of teacher network, which can be regard as a constraint to guide the training of student network and make it converge better. (2) Because of the difficulty of optimization, the student network may fall into a suboptimal solution if directly regarding the features and output of a trained teacher model as the targets. Furthermore, compared with training the teacher model firstly and then distilling the student, this synchronous training strategy is time-saving.

Note that, as shown in Figure 4(a), in the early stage of training, the loss value with training process supervision is higher and more unstable, but the opposite result is obtained in the late stage. Our analysis is that the knowledge obtained from the teacher is continuously changing for each iteration, and the degree of the



(a) (b) (c)

Fig. 4. (a) Loss analysis with/without training process supervision for Tiny-YOLO [36] with the guidance of Darknet53 [37] on Pascal VOC07 dataset. (b) and (c) demonstrate the impact of filter factor α: (b) Accuracy comparison for different α. (c) Comparison of total number of false detections by using different α when $P_{class} > 0.5$.

change is relatively large at the early stage to lead the higher and unstable loss value. With the improvement of the teacher, the student receives more accurate targets so that the loss will be lower and more stable. Besides, we try to initialize models by random normal initializer in addition to ImageNet pre-training, and it presents the similar training situation as above. However, whichever initialization is used, the early instability is not too severe to cause divergence. Also, we find these ways are different in convergence speed (150 epochs by pre-training and 360 epochs by random initializer). Namely, initializing with ImageNet pre-training can speed up the convergence of our models.

3.3 Network output supervision

Since object detection requires localization in addition to classification, for these two types of tasks, we utilize following objective functions for training.

Probabilistic objective function In the detectors, foreground judgment and classification belong to probabilistic tasks. So for this type of task, we both use cross entropy loss function. Given N samples $X = \{x_i\}_{i=1}^{N}$ from M classes, the objective function is defined as follows:

$$L_{CE} = -\sum_{i=1}^{N}\sum_{m=1}^{M} I(y_i, m)log(p^m(x_i)), \tag{3}$$

where $p^m(x_i)$ is the probability output of mth category of ith sample, and y_i is the label of sample i and I is an indicator function defined as $I(y_i, m) = \begin{cases} 1 & y_i = m \\ 0 & y_i \neq m \end{cases}$. For transferring knowledge from a teacher network to a student network, the objective function is modified as:

$$L(p_t||p_s) = -\sum_{i=1}^{N}\sum_{m=1}^{M} p_t^m(x_i)log(p_s^m(x_i)), \tag{4}$$

where $p_t^m(x_i)$ and $p_s^m(x_i)$ are probability outputs of the teacher network and student network respectively.

As for probabilistic task, the advantages of using the same loss function are that gradient balance can be guaranteed without biasing towards one or some tasks, and no additional balancing factors are required to adjust different probabilistic loss functions.

Regression objective function For object localization, we use GIoU loss [50] as our objective function which is defined as:

$$GIoU = \frac{A \cap B}{A \cup B} - \frac{C - (A \cup B)}{C}, \tag{5}$$

where A and B represent the regions of two bounding boxes, and C is the smallest rectangle region enclosing both A and B. The GIoU loss is defined as:

$$L_{GIoU} = 1 - GIoU. \tag{6}$$

Compared with traditional L2 loss function, the reason we use GIoU loss function is that, in the early stage of training, due to the large gap between the result of the teacher network and ground truth, the training of the student network will be unstable and often lead to divergence by using L2 loss function in the experiments. However, according to the property of GIoU loss function, the student network tries to regress a bounding box to cover the bounding boxes obtained from the teacher network and ground truth for each object, and gradually narrows the regressed box with the improvement of the teacher network. Finally, the student network can locate objects accurately. Therefore, GIoU loss function is more suitable for our algorithm and experiments also confirm this. In a nutshell, the optimization process of the student network with GIoU loss function is from coarse to fine.

3.4 Overall objective function

For the teacher network, it only uses ground truth for training, the overall loss function L_t is defined as:

$$L_t = L_{CE_1} + L_{GIoU_1}. \tag{7}$$

For the student network, the overall loss function L_s is defined as:

$$L_s = L_{ES} + L_{CE_2} + L(p_t||p_s) + L_{GIoU_2} + L_{GIoU_t}, \tag{8}$$

In L_{GIoU_1} and L_{GIoU_2}, the bounding box A is from ground truth and the bounding box B is from network prediction, but in L_{GIoU_t}, the bounding box A is from the prediction of teacher network and the bounding box B is from student network. The targets of L_{CE_1} and L_{CE_2} both are from ground truth, and the target of $L(p_t||p_s)$ is from the output of teacher network.

In this way, the student network learns to correctly predict the label of training samples as well as to match the output and specified features of the teacher network. At the same time, the teacher network also learns to correctly predict the label of training samples.

4 Experiments

In this section, we will evaluate our algorithm on Pascal VOC [51] and MS COCO [52] benchmarks. Firstly, we describe the experimental settings. Then, we compare the performance by introducing inter-layer representation supervision into different layers and discuss the impact of filter factor α with different values. After that, ablation experiments will be presented to explain the effects of different supervision. Lastly, we compare our algorithm with state-of-the-art methods and further verify it on some common one-stage detector heads.

4.1 Implementation details

Backbone networks In our experiments, Tiny-YOLO [36], Resnet18 [53] and MobileNetV2 [2] are used as backbones of student networks for the following reasons: (1) Tiny-YOLO consists of continuous convolution and pooling operation, which is a typical network structure like VGG [54]. (2) Resnet18 can be used to represent the networks with Residual blocks [53] to verify our algorithm. (3) MobileNetV2 is composed of Depthwise Separable Convolution [1] which is a common building block used in many efficient networks. Therefore, these three networks contain the common structures used in most existing networks, which can be utilized to reflect the generality of our algorithm.

For another, Darknet53 [37] and Resnet50 [53] are used as backbones of teacher networks, and the reasons are that: (1) Since the teacher network and the student network are trained together in our algorithm, the time of each iteration depends on the teacher network and it will be longer when using a giant backbone such as Resnet101 [53]. So the use of mentioned backbones can save training time. (2) Based on the teacher networks with these backbones, the training of our algorithm can be easily set up and perform well in the single GPU (TITAN Xp), which is resource-saving. (3) The experiments verify that the teacher networks with above backbones can still significantly improve student networks and make them achieve competitive performance. Therefore, we use the aforementioned backbones into teacher networks.

Training setup We implement all networks and training procedures in TensorFlow and use the standard AdamOptimizer with default setting for training. The cosine schedule is used to adjust learning rate from 10^{-4} to 10^{-6} and the initial moving average decay is set to 0.9995. We only use single GPU (TITAN Xp) for each experiment and we use 4×8 batch size (4 sub-batch size and 8 subdivisions, the same strategy as mentioned in [37]) to train networks for 150 epochs. All teacher and student models in experiments are initialized with ImageNet pre-trained models. A single scale detection layer [37] is used as our main detector head and we also evaluate our algorithm by using other common detector heads such as SSD [35], DSSD [55] and RetinaNet [56]. Normal data augmentation methods are applied during training, such as random horizontal flipping, random cropping and random translating.

4.2 Multi-layer supervision

The inter-layer representation supervision can be used to guide the feature representation of any layer in the student network. However, by considering the feature similarity in the adjacent layers or the same groups, we only introduce inter-layer supervision at the end of each group to avoid redundant mimicry. To verify which groups should be supervised for optimal performance, we conduct comparison experiments by using Tiny-YOLO [36] with the guidance of Darknet53 [37] on Pascal VOC dataset and the results are reported in Table 1.

Table 1. Comparison for different supervised groups on Pascal VOC07 dataset by using Tiny-YOLO with the guidance of Darknet53.

Group(Backbone)	mAP(%)
Last five groups	64.49
Last four groups	64.32
Last three groups	**65.66**
Last two groups	65.36
Last group	64.89

Generally, the first group of network is mainly responsible for low-level feature extraction and the features in deep layers are rich in semantic information, so we introduce the supervision into the end of each group from deep to shallow except the first group. As reported in Table 1, we can observe that the optimal result is obtained when introducing the supervision into the end of last three groups. To better understand why this case is the best, we visualize the feature maps of some samples in Figure 3(b) and we find that group 2 and group 3 still express some low-level features. If we generate suppression regions for these groups by using the method mentioned in Section 3, the student network will learn a lot of low-level information in the useless regions of background. In contrast, as shown in Figure 3(b), there are more semantic information in the feature maps of group 4, group 5 and group 6, and high-response regions are basically concentrated in the object regions, so the features in these regions are exactly what we want the student network to mimic. In the following experiments, we introduce inter-layer representation supervision to the last three groups of backbone networks.

4.3 Filter factor α for ES map

In Section 3, we use a filter factor α to control the generation of suppression regions. To better determine the value of α, we conduct a set of experiments on Pascal VOC dataset [51] with $\alpha = 0$, $\alpha = 0.3$, $\alpha = 0.5$, $\alpha = 0.7$ and $\alpha = 1$ respectively and the results are shown in Figure 4(b) and Figure 4(c).

When $\alpha = 0$, the student network focuses on mimicking the feature representation from overall features of the teacher network, but the accuracy has dropped as shown in Figure 4(b). By observing the average pixel loss in the feature maps, the result with $\alpha = 0$ has a bigger loss value than others. Through analysis, we find that the full feature mimicry introduces a great deal of information from unimportant regions, which leads to performance degradation. In addition, when $\alpha = 1$, the student only focuses on the feature mimicry in the object regions. That is, the feature expression of false detections in the background cannot be suppressed. To verify this point, we simply count the total number of false detections with a classification score (P_{class}) greater than 0.5 for different α values and the results are shown in Figure 4(c). The number of false detections with $\alpha = 1$ is more than others, which verifies the above point. More detection analysis can be found in supplementary material. From the results, $\alpha = 0.5$ offers the best performance, so a constant $\alpha = 0.5$ is used in all experiments.

4.4 Ablation study

To further verify the effect of each component in our algorithm, we conduct ablation experiments by using Tiny-YOLO distilled with Darknet53 on VOC dataset. The results of different supervision combinations are shown in Table 2.

Table 2. Ablation experimental results for evaluating the effect of different combinations of the three supervisions on Pascal VOC07 dataset.

Network	Network output supervision	Training process supervision	Inter-layer supervision	mAP(%)
Darknet53 (teacher)	-	-	-	76.96
Tiny-YOLO (student)	-	-	-	57.10
	√	-	-	58.01
	√	√	-	59.72
	-	-	√	62.08
	√	-	√	63.37
	-	√	√	63.84
	√	√	√	**65.66**

From Table 2, there is 0.91% improvement by using output supervision, which shows that conventional output distillation is not very effective for object detection. When we only use the inter-layer representation supervision, there is a significant improvement in performance, which indicates the feature representation distillation is more important for detection and also verifies the effectiveness of proposed ES loss. Besides, the training process supervision gives a further improvement for both output supervision and inter-layer supervision, and the reason we analyzed is that the dynamically evolved features and output from the teacher carry with the experience of step-by-step learning so as to promote the training process of the student. After introducing all supervision, our algorithm significantly boosts 8.56% mAP compared to the non-distilled student.

4.5 Experiment results

As reported in Table 3, we compare our algorithm with Hints [24], FSP [27], objectness scaled distillation (OSD) [16], similarity-preserving distillation method (SP) [26] and distillation with fine-grained feature imitation (FFI) [14] on Pascal VOC benchmark. Overall, FSGD consistently outperforms the state-of-the-art methods. Especially for Tiny-YOLO, it achieves compelling 3.4% absolute improvement over the best competitor FFI. Note that Resnet18 is further boosted up to 73.54% by using FSGD, which has compelling 4.35% gains compared with original model. Besides, we find OSD method that only distills network output rarely promote the student networks, which also denotes that the distillation of intermediate representation is more important for object detection as mentioned in Section 4.4. Detailed analysis of class-wise performance for student networks can be found in the supplementary file.

Table 3. Experimental comparison of different distillation algorithms on Pascal VOC07 dataset (mAP, %).

Teacher	Darknet53	Darknet53	Darknet53	Resnet50	Resnet50	Resnet50
Student	Tiny-YOLO	Resnet18	MobileNetV2	Tiny-YOLO	Resnet18	MobileNetV2
Teacher	76.96	76.96	76.96	74.87	74.87	74.87
Student	57.10	69.19	68.59	57.10	69.19	68.59
Hints [24]	61.68	71.12	69.76	59.43	69.88	69.31
FSP [27]	61.23	71.32	69.44	59.17	69.23	68.79
OSD [16]	60.60	69.32	68.11	58.63	68.76	67.67
SP [26]	62.14	72.25	69.81	59.30	70.05	69.06
FFI [14]	62.26	71.83	70.34	59.21	70.25	69.15
FSGD(ours)	**65.66**	**73.54**	**71.40**	**61.28**	**71.01**	**70.11**

To further verify the effectiveness of proposed algorithm, we present experimental results on the challenging COCO benchmark. As shown in Table 4, our algorithm significantly improves original student networks. Tiny-YOLO, MobileNetV2 and Resnet18 get respectively 3.36%, 2.43% and 4.83% boost of AP_{50} compared with non-distilled counterpart. And there are still obviously 1.57%, 1.87% and 3.49% absolute gains in AP. Noted that FSGD improves AR for each student model, which demonstrates that our algorithm can help improve recall as discussed in Section 3.1.

Besides, we use some common one-stage detector heads (SSD [35], DSSD [55], RetinaNet [56]) to verify the robustness of FSGD. Shown in Table 5, lightweight version of detector head are used into student networks. Similar to SSDLite [2], all the regular convolutions are replaced with separable convolutions (depthwise followed by 1x1 projection) in the prediction layers of DSSD and RetinaNet. We call them DSSDLite and RetinaLite. RetinaNet uses a 600 pixel train and

Table 4. Performance verification of proposed FSGD algorithm on COCO dataset by using different teacher networks to distill different student networks.

Student	Teacher53	AP_{50}(%)	AP(%)	AR(%)
Tiny-YOLO	-	23.72	10.46	11.97
	Resnet50	26.02 (+2.30)	11.47 (+1.01)	12.81 (+0.84)
	Darknet53	27.08 (+3.36)	12.03 (+1.57)	13.38 (+1.41)
MobileNetV2	-	29.31	13.46	14.23
	Resnet50	30.46 (+1.15)	14.44 (+0.98)	15.01 (+0.78)
	Darknet53	31.74 (+2.43)	15.33 (+1.87)	15.50 (+1.27)
Resnet18	-	30.55	14.42	15.08
	Resnet50	33.77 (+3.22)	16.84 (+2.42)	17.11 (+2.03)
	Darknet53	35.38 (+4.83)	17.91 (+3.49)	17.63 (+2.55)

test image scale. Experiments show that FSGD still can help to improve the performance of student networks with such detector heads.

Table 5. Robustness verification of FSGD algorithm on Pascal VOC07 by using SSD, DSSD and RetinaNet detector heads (mAP, %).

Teacher	Student	Non-distilled Student	FSGD
Resnet50 + SSD300 [35]	Resnet18 + SSDLite	73.62	76.83 (+3.21)
	MobileNetV2 + SSDLite	73.24	75.67 (+2.43)
Resnet50 + DSSD321 [55]	Resnet18 + DSSDLite	74.53	77.28 (+2.75)
	MobileNetV2 + DSSDLite	73.85	75.76 (+1.91)
Resnet50 + RetinaNet [56]	Resnet18 + RetinaLite	75.88	78.69 (+2.81)
	MobileNetV2 + RetinaLite	75.56	77.24 (+1.68)

5 Conclusions

In this work, a novel fully supervised and guided distillation algorithm is proposed to comprehensively transfer knowledge from inter-layer feature representation, training process and network output. Besides, we design an excitation and suppression loss to make the student network focus on mimicking valuable feature representation to improve recall and reduce false detections. Then, a process-guided learning strategy is proposed for transferring the knowledge in the training process of teacher network to help the student network find a better local minimum. Extensive experiments demonstrate the effectiveness and robustness of our algorithm on the representative network architectures.

References

1. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
2. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.: Mobilenetv2: Inverted residuals and linear bottlenecks. arXiv preprint arXiv:1801.04381 (2018)
3. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018) 6848–6856
4. Ma, N., Zhang, X., Zheng, H., Sun, J.: Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: European Conference on Computer Vision (ECCV). (2018)
5. Xie, G., Wang, J., Zhang, T., Lai, J., Hong, R., Qi, G.: Igcv2: Interleaved structured sparse convolutional neural networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018)
6. He, Y., Liu, P., Wang, Z., Yang, Y.: Pruning filter via geometric median for deep convolutional neural networks acceleration. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019)
7. Lin, S., Ji, R., Yan, C., Zhang, B., Cao, L., Ye, Q., Huang, F., Doermann, D.: Towards optimal structured cnn pruning via generative adversarial learning. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
8. Frankle, J., Carbin, M.: The lottery ticket hypothesis: Finding sparse, trainable neural networks. International Conference on Learning Representations (ICLR) (2019)
9. Geoffrey, H., Oriol, V., Jeff, D.: Distilling the knowledge in a neural network. Neural Information Processing Systems (NIPS) (2015)
10. Mirzadeh, S., Farajtabar, M., Li, A., Ghasemzadeh, H.: Improved knowledge distillation via teacher assistant. The AAAI Conference on Artificial Intelligence (AAAI) (2020)
11. Liu, Y., Sheng, L., Shao, J., Yan, J., Xiang, S., Pan, C.: Multi-label image classification via knowledge distillation from weakly-supervised detection. ACM Multimedia (2018) 700–708
12. Park, W., Kim, D., Lu, Y., Cho, M.: Relational knowledge distillation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019)
13. Byeongho, H., Minsik, L., Sangdoo, Y., Jin, Young, C.: Knowledge distillation with adversarial samples supporting decision boundary. In: The AAAI Conference on Artificial Intelligence (AAAI). (2019)
14. Wang, T., Yuan, L., Zhang, X., Feng, J.: Distilling object detectors with fine-grained feature imitation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019) 4933–4942
15. Chen, G., Choi, W., Yu, X., Han, T., Chandraker, M.: Learning efficient object detection models with knowledge distillation. In: Advances in Neural Information Processing Systems 30. (2017) 742–751
16. Rakesh, M., Cemalettin, O.: Object detection at 200 frames per second. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018)
17. Wei, Y., Pan, X., Qin, H., Ouyang, W., Yan, J.: Quantization mimic: Towards very tiny cnn for object detection. In: The European Conference on Computer Vision (ECCV). (2018)
18. Cristian, B., Rich, C., Alexandru, N.M.: Model compression. In KDD (2006)

19. Junjie, L., Dongchao, W., Hongxing, G., Wei, T., Tse-Wei, C., Kinya, O., Masami, K.: Knowledge representing: Efficient, sparse representation of prior knowledge for knowledge distillation. In: The IEEE Conference on Computer Vision and Pattern Recognition Workshops. (2019)
20. Yu, L., Yazici, V.O., Liu, X., Weijer, J.v.d., Cheng, Y., Ramisa, A.: Learning metrics from teachers: Compact networks for image embedding. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019)
21. He, T., Shen, C., Tian, Z., Gong, D., Sun, C., Yan, Y.: Knowledge adaptation for efficient semantic segmentation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019)
22. Chen, L., Chunyan, Y., Lvcai, C.: A new knowledge distillation for incremental object detection. In: International Joint Conference on Neural Networks (IJCNN). (2019)
23. Yousong, Z., Chaoyang, Z., Chenxia, H.: Mask guided knowledge distillation for single shot detector. In: International Conference on Multimedia and Expo (ICME). (2019)
24. Romero, A., Ballas, N., Kahou, S.E., Chassang: Fitnets: Hints for thin deep nets. In: In Proceedings of International Conference on Learning Representations. (2015)
25. Zhang, Y., Xiang, T., Hospedales, T.M., Lu, H.: Deep mutual learning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018)
26. Tung, F., Mori, G.: Similarity-preserving knowledge distillation. International Conference on Computer Vision (ICCV) (2019)
27. Yim, J., Joo, D., Bae, J., Kim, J.: A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017) 7130–7138
28. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. International Conference on Learning Representations (ICLR) (2017)
29. Ahn, S., Hu, S.X., Damianou, A., Lawrence, N.D., Dai, Z.: Variational information distillation for knowledge transfer. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019)
30. Liu, Y., Chen, K., Liu, C., Qin, Z., Luo, Z., Wang, J.: Structured knowledge distillation for semantic segmentation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019)
31. Li, Q., Jin, S., Yan, J.: Mimicking very efficient network for object detection. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017)
32. Rui, C., Haizhou, A., Chong, S.: Learning lightweight pedestrian detector with hierarchical knowledge distillation. In: 2019 IEEE International Conference on Image Processing (ICIP). (2019)
33. Zhao, Q., Sheng, T., Wang, Y., Tang, Z., Chen, Y., Cai, L., Ling, H.: M2det: A single-shot object detector based on multi-level feature pyramid network. In: The AAAI Conference on Artificial Intelligence (AAAI). (2019)
34. Law, H., Deng, J.: Cornernet: Detecting objects as paired keypoints. In: The European Conference on Computer Vision (ECCV). (2018)
35. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., Berg, A.C.: Ssd: Single shot multibox detector. In: The European Conference on Computer Vision (ECCV). (2016)
36. Redmon, J., Farhadi, A.: Yolo9000: Better, faster, stronger. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017)

37. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
38. Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018)
39. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask r-cnn. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017)
40. Deng, J., Pan, Y., Yao, T., Zhou, W., Li, H., Mei, T.: Relation distillation networks for video object detection. In: The IEEE International Conference on Computer Vision (ICCV). (2019)
41. Jifeng, D., Yi, L., Kaiming, H., Jian, S.: R-FCN: Object detection via region-based fully convolutional networks. In: Advances in Neural Information Processing Systems (NIPS). (2016)
42. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems (NIPS). (2015)
43. Qin, Z., Li, Z., Zhang, Z., Bao, Y., Yu, G., Peng, Y., Sun, J.: Thundernet: Towards real-time generic object detection on mobile devices. In: The IEEE International Conference on Computer Vision (ICCV). (2019)
44. Wang, R.J., Li, X., Ling, C.X.: Pelee: A real-time object detection system on mobile devices. In: Advances in Neural Information Processing Systems (NIPS). (2018) 1967–1976
45. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2014)
46. Girshick, R.: Fast r-cnn. In: International Conference on Computer Vision (ICCV). (2015)
47. Li, Z., Peng, C., Yu, G., Zhang, X., Deng, Y., Sun, J.: Light-head r-cnn: In defense of two-stage object detector. arXiv preprint arXiv:1711.07264 (2017)
48. Pang, J., Chen, K., Shi, J., Feng, H., Ouyang, W., Lin, D.: Libra r-cnn: Towards balanced learning for object detection. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019)
49. Ba, L.J., Caruana, R.: Do deep nets really need to be deep. In: Advances in Neural Information Processing Systems (NIPS). (2013)
50. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: A metric and a loss for bounding box regression. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019)
51. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision **88** (2010) 303–338
52. Lin, T., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollar, P.: Microsoft coco: Common objects in context. In European Conference on Computer Vision (ECCV) (2014)
53. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016)
54. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. ICLR (2015)
55. Fu, C.Y., Liu, W., Ranga, A., Tyagi, A., Berg, A.C.: DSSD: Deconvolutional single shot detector. In: arXiv preprint arXiv:1701.06659. (2017)

56. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollar, P.: Focal loss for dense object detection. In: The IEEE International Conference on Computer Vision (ICCV). (2017)