sensors

MDPI

Article

Multilayer Semantic Features Adaptive Distillation for Object Detectors

Zhenchang Zhang [1,2,*], Jinqiang Liu [2], Yuping Chen [3], Wang Mei [1], Fuzhong Huang [1] and Lei Chen [1]

[1] Key Laboratory of Smart Agriculture and Forestry, College of Computer and Information Sciences, Fujian Agriculture and Forestry University, Fuzhou 350002, China
[2] College of Mechanical and Electrical Engineering, Fujian Agriculture and Forestry University, Fuzhou 350002, China
[3] Key Laboratory of Marine Biotechnology of Fujian Province, Institute of Oceanology, Fujian Agriculture and Forestry University, Fuzhou 350002, China
* Correspondence: stdin@fafu.edu.cn

Abstract: Knowledge distillation (KD) is a well-established technique for compressing neural networks and has gained increasing attention in object detection tasks. However, typical object detection distillation methods use fixed-level semantic features for distillation, which might not be best for all training stages and samples. In this paper, a multilayer semantic feature adaptive distillation (MSFAD) method is proposed that uses a routing network composed of a teacher and a student detector, along with an agent network for decision making. Specifically, the inputs to the proxy network consist of the features output by the neck structures of the teacher and student detectors, and the output is a decision on which features to choose for distillation. The MSFAD method improves the distillation training process by enabling the student detector to automatically select valuable semantic-level features from the teacher detector. Experimental results demonstrated that the proposed method increased the mAP_{50} of YOLOv5s by 3.4% and the $mAP_{50–90}$ by 3.3%. Additionally, YOLOv5n with only 1.9 M parameters achieved detection performance comparable to that of YOLOv5s.

Keywords: multilayer semantic feature; knowledge distillation; object detection; adaptive distillation

Citation: Zhang, Z.; Liu, J.; Chen, Y.; Mei, W.; Huang, F.; Chen, L. Multilayer Semantic Features Adaptive Distillation for Object Detectors. *Sensors* **2023**, *23*, 7613. https://doi.org/10.3390/s23177613

Academic Editors: Maozhen Li, Zhengwen Huang, Yang Liu and Mukhtaj Khan

Received: 8 August 2023
Revised: 25 August 2023
Accepted: 30 August 2023
Published: 2 September 2023

1. Introduction

In recent years, deep neural networks have been widely adopted in various fields [1–5], with increasingly complex model structures designed to achieve higher performance. However, these models require substantial computing resources and have very low inference speeds. Knowledge distillation (KD) [6] has been proposed to solve those problems. KD is a highly effective neural network compression method that transfers the dark knowledge contained in a bulky teacher model to a compact student model, enabling the latter to achieve advanced performance. Relative to other compression methods [7–10], KD minimizes the loss in performance caused by compression and requires no special hardware or software support.

After substantial progress in recent years, KD methods for image classification tasks have matured [11–15]. However, object detection tasks require consideration of both classification and localization, and there is an imbalance between foreground and background issues [16]. Hence, important challenges persist in using KD for object detection tasks. Therefore, several recent studies have focused on adapting KD methods for object detection tasks [17–21]. As shown in Figure 1a,b, those studies can be divided into two primary categories:

(1) Distilling only the specific semantic-level features of the detector [18,19,21] (Figure 1a). For example, [18] used the region proposal network structure of the student detector to select positive regions from the fixed-level features, and [21] distilled the foreground and background of the intermediate layer features separately.

(2) Distilling the specific semantic-level features of the detector and the logit output by the detector head [17,20] (Figure 1b). For example, in [20], semantic features were distilled independently from the backbone network, classification head, and regression head.

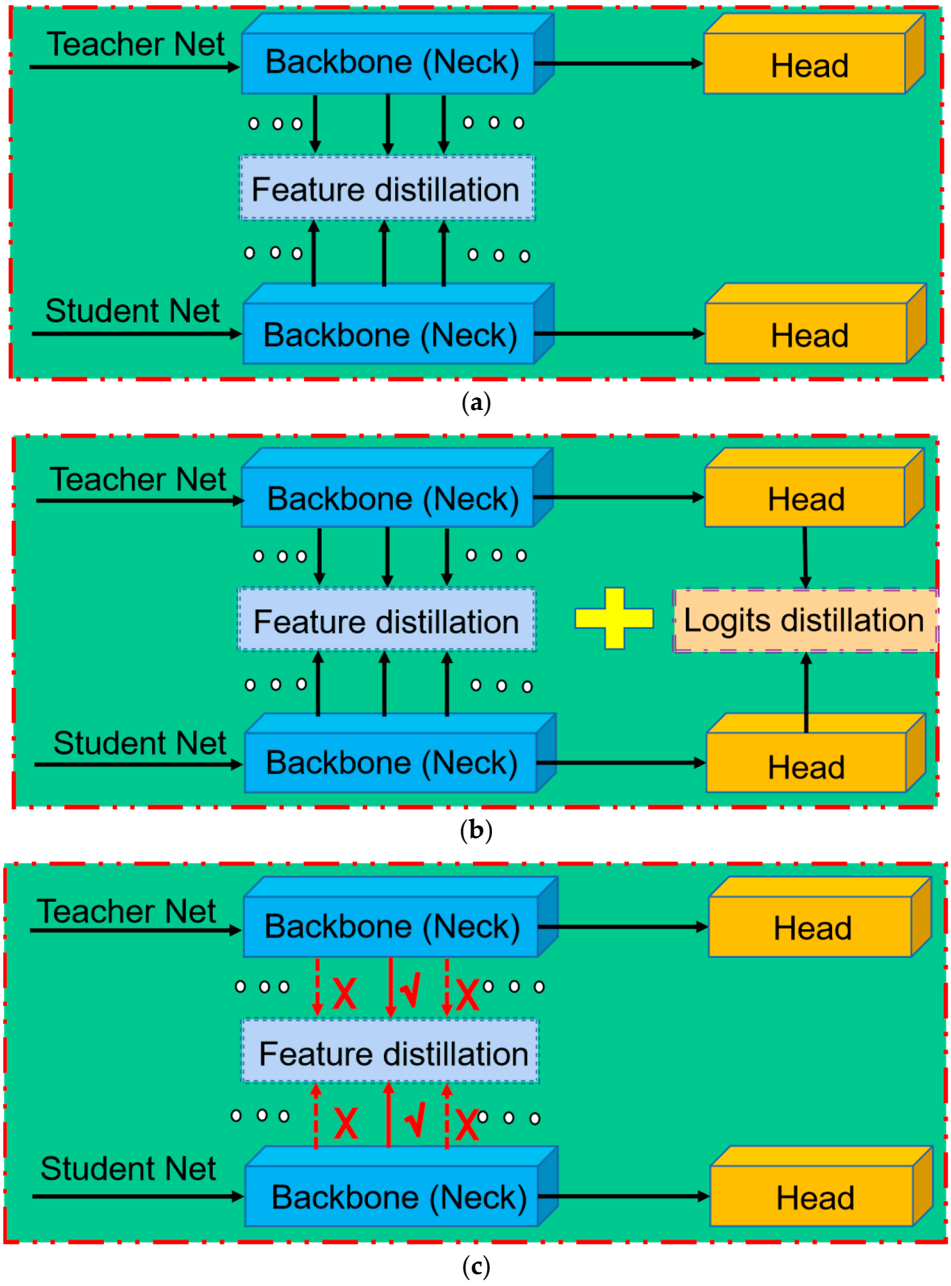


Figure 1. Overview of existing object detection distillation methods. (**a**) Conventional approach, where the distillation framework focuses solely on fixed semantic-level features of detectors. (**b**) An extension of (**a**) that incorporates logit distillation while keeping the semantic feature level fixed. (**c**) Proposed multilayer semantic feature adaptive distillation method (MSFAD), which uses adaptive semantic-level features for distillation. The red dotted lines indicate that features at that semantic level are currently not used for distillation in the current training stage and samples.

Those methods all used a fixed semantic level of features for distillation, which did not change during training.

However, ref. [22] noted that when a classifier is distilled, the distillation point's location should be adjusted in accordance with different training stages and samples.

Therefore, ref. [22] proposed a spot-adaptive distillation method, which has been shown to improve the performance of distillation methods for classification tasks.

Inspired by [22], in the present study, where to distill was also considered when distilling detectors, which has been disregarded by most current object detection distillation methods. This paper proposes a multilayer semantic feature adaptive distillation (MSFAD) method for object detectors to address the problems above. As shown in Figure 1c, the proposed method diverges from the object detection distillation approaches shown in Figure 1a,b. It empowers the student detector to autonomously discern and incorporate valuable semantic-level features from the teacher detector, depending on the training stage and samples. Specifically, the MSFAD approach uses a routing network for teacher and student detectors and an agent network for decision making. The proxy network takes the features output by the neck structures of the teacher and student detectors as input and determines whether to use the current semantic-level features for distillation.

There are two differences between our approach and that of [22]. First, all features directly fed into the detector head are adopted as input to the proxy network rather than only the last layer features of the teacher and student models. This decision was made since mainstream detectors [3,4,23–28] usually use multiscale feature fusion [29], which inputs features of various scales into the detection head to detect objects of varying sizes. Thus, the input features of the detector head often come from multiple semantic levels. In this study, it was found that using all features input into the detector head in the proxy network resulted in better decisions for final detection. Second, to ensure method generality, only the semantic features of the middle levels were distilled since different detectors have varying head structures. In summary, the contributions of this paper are:

(1) A novel MSFAD method is proposed for object detectors that addresses the problem in current object detection distillation methods of the mismatch between the semantic level of distilled features and the training stage and samples.
(2) The selection of various semantic-level features for distillation at different training stages is described, and the important effect of semantic-level selection during distillation training is highlighted.
(3) The experiments described show that the MSFAD method improved the mAP_{50} and $mAP_{50–90}$ of YOLOv5s by 3.4% and 3.3%, respectively. Moreover, it is demonstrated that MSFAD achieved detection performance similar to YOLOv5s for YOLOv5n with only 1.9 M parameters. Relative to the latest YOLOv7-tiny of the same magnitude, the YOLOv5s model distilled by our method achieved higher mAP_{50} and $mAP_{50–90}$ by 2.2% and 1.9%, respectively.

The paper is organized as follows: Section 2 reviews related work on object detection algorithms and KD algorithms. Section 3 presents the proposed semantic-level adaptive distillation algorithm. Section 4 details the experimental process and presents experimental results. Finally, Section 5 provides a conclusion.

2. Related Work

2.1. Object Detection

Object detection is a fundamental task in computer vision widely applied in various scenarios [30–34]. Currently, deep learning methods are the mainstream approach for object detection. Object detection methods can be categorized into three groups: (1) Two-stage object detection [3,4,28], (2) single-stage object detection based on anchor boxes [23–27,35], and (3) single-stage object detection that is anchor free [36,37], following various detection principles.

Two-stage object detection algorithms, such as the region-based convolutional neural network (R-CNN) family [3,4,28], first extract object areas and then classify the extracted object areas. However, the main disadvantage of that method is its slow inference speed, which limits its practical application.

In contrast, the "you only look once" (YOLO) single-stage detection algorithm, proposed by Redmon et al. [23] in 2016, directly outputs the position information and category

of the prediction frame at the output layer, which greatly improves the model's inference speed. Various improvements on the basis of YOLOv1 have been made [24–27,35]. Consequently, the YOLO family has become the preferred algorithm for various detection tasks. However, most object detection distillation methods adopt Faster-RCNN as the benchmark model, which is not proven to be suitable for YOLO. Therefore, this study uses YOLOv5 as the benchmark model for object detection distillation methods.

Two researchers have used anchor-free methods to complete target detection tasks. Duan et al. [36] modeled object detection as a center-point detection problem, combining prediction of center points and bounding boxes to achieve efficient and accurate object detection. Tian et al. [37] achieved the efficiency and accuracy of one-stage object detection by the use of innovative methods, such as a complete convolutional structure, center-width height representation, and adaptive receptive fields.

2.2. Knowledge Distillation

Knowledge distillation is a compression technique that does not modify the network architecture. The fundamental concept behind the approach is to transfer the "dark" knowledge from a larger teacher model to a smaller student model to attain similar performance. Hinton et al. [6] first introduced minimizing the KL divergence between teacher and student probability outputs to improve student performance. Later work [11] indicated that using both semantic features from middle layers and logits for distillation could lead to greater improvements. Later studies [12–15] markedly improved the student classifier's performance. Today, KD has emerged as a well-established compression method for classification tasks.

However, classification and localization problems must be considered for object detection tasks since they often lead to a marked imbalance between foreground and background objects [16]. Therefore, directly applying classification distillation methods to detection tasks is not ideal. Several studies [17–21,38] have recently attempted to apply KD to object detection tasks, improving student detectors' performance. For instance, Chen et al. [17] used hint learning [11] to distill the semantic features of the intermediate layers of a detector by designing distillation weights to suppress the background. Li et al. [18] used the region proposal network structure of the student detector to extract semantic features from the middle layer and then distilled the extracted positive feedback regions. Similarly, Sun et al. [20] distilled semantic features from the backbone network, classification head, and regression head separately, and Dai et al. [38] introduced a relation-based distillation method to simultaneously distill semantic features and detection head logits of the middle layer of the teacher detector. Additionally, attention mechanisms have been used in various fields [39–41], and Yang et al. [21] incorporated the attention mechanism to enable the student detector to focus on useful local pixels while introducing global distillation to compensate for the lack of pixel relationships.

In conclusion, the object detection distillation methods mentioned above can be categorized into two types: (1) Those that distill intermediate layer features based solely on semantics [18,19,21], and (2) those that distill intermediate layer features and detection head logits simultaneously [17,20,38]. Once the semantic level of the features used for distillation is established in those methods, it remains unchanged throughout the distillation training process. Those approaches prioritize what to distill over where to distill.

3. Method

The proposed MSFAD method (Figure 2), which involves two forward propagation processes, is introduced in this section. The first process (Figure 2a) is the distillation feedforward, focused primarily on calculating the detection and distillation losses of the student detector. The distillation loss calculation is constrained by the output of the proxy network. $P_T = 1$ indicates distillation using features from the current semantic level, whereas $P_T = 0$ implies no distillation. Due to the interdependencies between the neck structure output and the proxy network's decisions, training of both networks simultaneously can lead to training failure. To address that problem, the second process, routing feedforward,

is used (Figure 2b). The routing path is determined by the proxy network's decision during the initial feedforward. If $P_T = 1$, the teacher detector's features are used as input for the next routing network layer. If $P_S = 1$, the student detector's features are used instead. The input features of the subsequent layer are aligned in channel dimensions through a 1×1 convolution operation. Since the teacher detector has already completed training before distillation training, the output of the teacher detector serves as the final output of the routing network. The routing loss is calculated by comparing that output with the ground truth value.

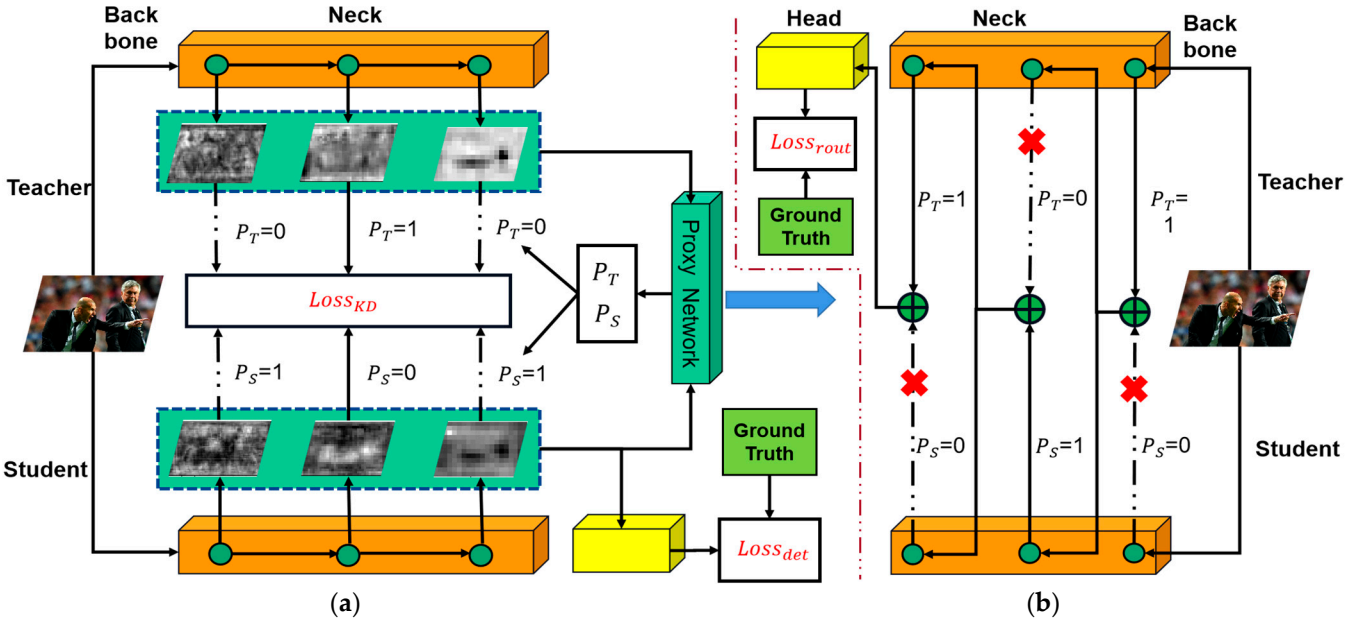


Figure 2. Overall framework of proposed multilayer semantic feature adaptive distillation. The training process of MSFAD comprises two forward processes. (**a**) Shows the distillation feedforward process. It computes the detection loss of the student detector $Loss_{det}$ and the distillation loss of the feature $Loss_{KD}$ and acquires the decision of the proxy network $P = (P_T, P_S)$. (**b**) Shows the routing feedforward process, which establishes the feedforward path by leveraging the decision made by the proxy network. It then computes the routing loss $Loss_{rout}$ using the output from the teacher detector's head and the ground truth. The dashed arrow indicates that the semantic features of the layer are not selected.

3.1. Distillation Feedforward Process

To clarify the feedforward process of the teacher and student detectors in Figure 2, we used YOLOv5 as a representative case. The feedforward process of object detection is shown in Figure 3. Assuming the input of the model is $F_{input}^{B\times C\times H\times W}$, the input data are first passed through the teacher and student detectors to complete one forward propagation. Through that forward propagation, the neck output features $F_T^{1\times i}$ and $F_S^{1\times i}$ of the teacher and student detectors can be obtained, where i is the number of features fed to the detection head for final detection in the neck output features of the detector. Equations (1) and (2) represent the forward propagation process:

$$F_T^{1\times i} = f_{te}\left(F_{input}^{B\times C\times H\times W}, \theta_{te}\right) \tag{1}$$

$$F_S^{1\times i} = f_{se}\left(F_{input}^{B\times C\times H\times W}, \theta_{se}\right), \tag{2}$$

where f_{te} and f_{se} are the feature encoding of the input data by the teacher and student detectors, respectively, and θ_{te} and θ_{se} are the model parameters corresponding to the relevant structure of the detector.

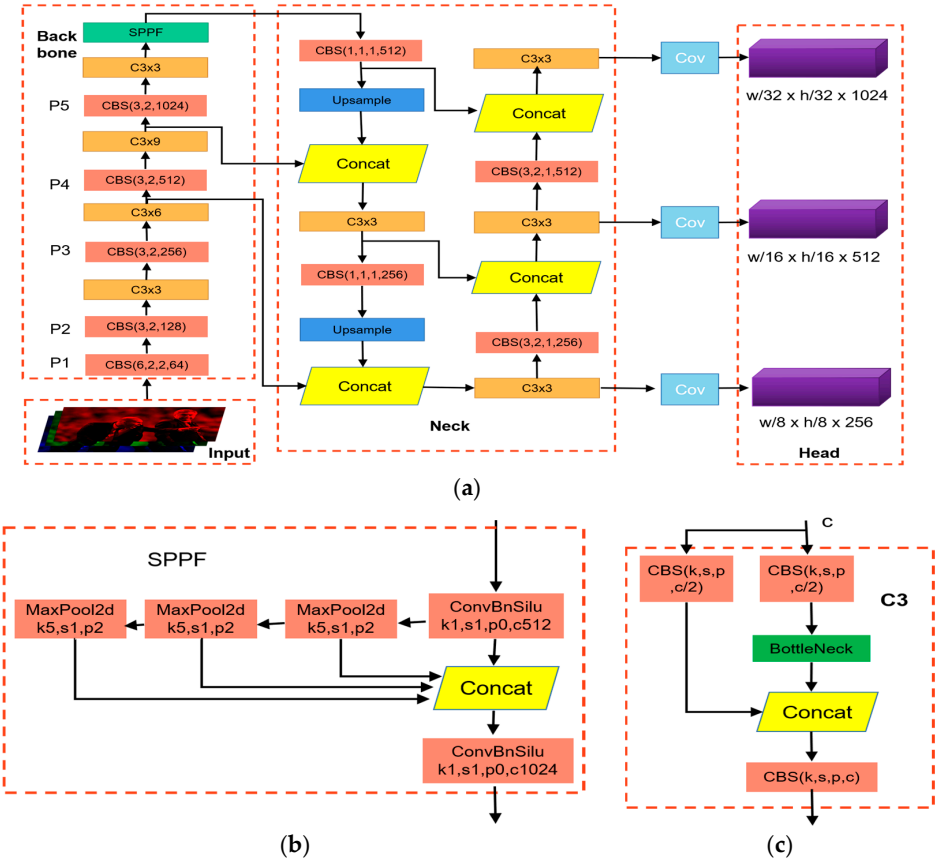


Figure 3. Network structure of YOLOv5. (**a**) Overall model framework with four components: Input, Backbone, Neck, and Head. Here, CBS is convolution, batch normalization, and SiLU activation, and C3 × *n* denotes *n* C3 layers. The image is initially fed into the Backbone structure for feature extraction. Subsequently, the Neck structure further integrates the extracted features, which are then input into the Head structure for final predictions. (**b**) Structural diagram of spatial pyramid pooling module. (**c**) C3 structure.

$F_T^{1\times i}$ and $F_S^{1\times i}$ are then fed into the proxy network. The final output of the proxy network is a feature vector of dimension 2 × k, represented as $P = [P_T^j, P_S^j]$ $(0 \le j \le k)$. Here, P_T^j and P_S^j are the probabilities of data passing through the teacher and student detectors, respectively, with values ranging from 0 to 1. $P_T^j = 0$ indicates that data do not pass through the teacher detector, and the semantic features of that level are not distilled. Moreover, k is the number of semantic levels used for feature distillation. The decision process is described in Equation (3):

$$P = \text{Gumbel\_Softmax}\left(f_c\left(F_T^{1\times i}, F_S^{1\times i}, \theta_p\right)\right), \tag{3}$$

where the function f_c is a fully connected operation and θ_p is the model parameter of the fully connected layer. "Gumbel softmax" is a method proposed in [42]. This method makes the sampling computation differentiable, allowing gradients to backpropagate to the proxy network during the backward propagation.

During the first forward process, $F_S^{1\times i}$ is also fed into the head of the student detector for detection. The detection loss L_{det} can be obtained using

$$L_{det} = f_{det}\left(f_h\left(F_S^{1\times i}, \theta_h\right), gt\right), \tag{4}$$

where the function f_h is the function of processing input features in the detection head, θ_h is the corresponding model parameter, gt is the ground truth, and f_{det} is the loss function

of the student detector, which comprises three components: localization loss, classification loss, and object confidence loss.

Finally, the decision P obtained by Equation (3) is used to distill the semantic features of the *j*th level of the teacher detector for which $P_T^j \neq 0$. The distillation loss can be calculated as

$$L_{KD}^j = \mathrm{fgd}(F_T^j, F_s^j)\ (P_T^j \neq 0), \tag{5}$$

where fgd is the detector distillation method proposed by [21]. This method was used to calculate the feature distillation loss. Specifically, Equation (5) can be further expressed as

$$L_{KD}^j = \alpha L_{fg}(F_T^j, F_s^j) + \beta L_{bg}(F_T^j, F_s^j) + \gamma L_{at} + \lambda L_{global}(F_T^j, F_s^j)\ (P_T^j \neq 0), \tag{6}$$

where L_{fg} is the foreground distillation loss function for the feature maps, L_{bg} is the background distillation loss function, L_{at} is the attention loss function which enables the student detector to mimic the spatial and channel attention masks of the teacher detector, and L_{global} is the global distillation loss. The hyperparameters α, β, γ, and λ are used to balance the weights of each loss function.

Since L_{KD}^j is calculated from features with $P_T^j \neq 0$, the semantic level of the features used for distillation can be adaptively changed based on the decisions of the proxy network.

3.2. Routing Feedforward Process

Through the first forward progress, the detection loss L_{det} and the L_{KD}^j of the student detector are computed. The P output by the policy network was also obtained. However, the output of the proxy network depends on the output features of the neck structure of the student detector, which are constrained by the output of the policy network through L_{KD}^j. Therefore, training both simultaneously is not suitable. The second round of forward propagation is used to address that problem.

At the preselected distillation feature level, the model determines the path through which the data flow based on P. When $P_T^j = 0$, the data flow through the student detector, whereas when $P_T^j = 1$, the data flow through the teacher detector. In contrast to [22], logits are not used for distillation, but the detection head of the teacher detector is used to output the final detection results of the routing network. The routing loss L_{rout} can be calculated as

$$L_{rout} = f_{det\_t}(H_{rt}, gt), \tag{7}$$

where f_{det_t} is the detection loss function of the teacher detector, and H_{rt} is the output of the head of the teacher detector in the routing feedforward process.

3.3. Overall Loss

The overall loss function is

$$L = L_{det} + \sum_{j=1}^{k} L_{KD}^j + L_{rout} \tag{8}$$

Minimizing L_{det} and L_{KD} can make the student detector achieve higher detection performance, and minimizing L_{rout} can improve the proxy network decision making.

4. Experiments

4.1. Dataset

All experiments in this work were carried out on the Pascal visual object classes (VOC) dataset [43], which comprised 20 object categories. The training and validation sets of VOC2007 and VOC2012, totaling 16,551 images, were used as the training data for our experiments. For evaluation purposes, the test set of VOC2007, which included 4952 images,

was used as the validation data. mAP_{50} and $mAP_{50–90}$ were used as the evaluation metrics to assess the detection performance.

4.2. Experimental Details

The experiments in this work were carried out using the PyTorch 1.11.1 deep learning framework, with training performed on a device equipped with an Intel Xeon Platinum 8352 V CPU (Intel, Santa Clara, CA, USA) and 2 Nvidia A40 48 G GPUs (Nvidia, Santa Clara, CA, USA). The operating system was Ubuntu 20.04.

The distillation process is shown in Figure 4. The first step in distillation training was training a teacher model. This model was then used to direct the training procedure of the student model. In this study, YOLOv5 was used as the benchmark model. Three sets of experiments were carried out: benchmark experiments, distillation experiments of semantic features at different levels, and validation experiments of MSFAD. YOLOv5l, YOLOv5s, and YOLOv5n were first trained, with YOLOv5l serving as the teacher detector and YOLOv5s and YOLOv5n as the benchmark detectors. An exploratory experiment was then carried out to study the relation between the student detector's performance and the distilled features' semantic levels. Finally, the performance of the proposed MSFAD was verified. Table 1 provides detailed parameters for all the experiments, allowing readers to refer to and reproduce the experiments.

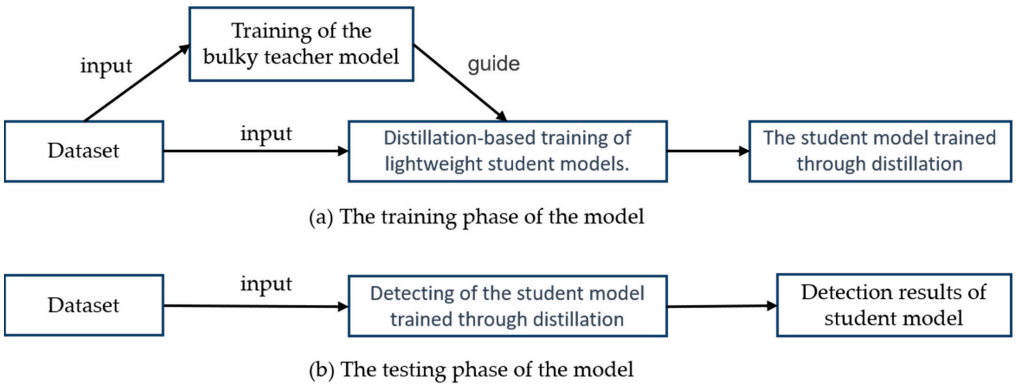


Figure 4. Distillation procedures. (**a**) Distillation training phase. During that phase, the teacher model was trained, and then the student model was trained, guided by the teacher model. (**b**) Testing phase. Here, the trained student model performed image detection and inference on the input image, with no involvement of the teacher model.

Table 1. Details of experiments.

Name	Epoch	lr	Weight_Decay	Momentum	Batch	Img_Size	T	α	β	γ	λ
YOLOv5l	300	0.01	0.0005	0.937	16	640	–	–	–	–	–
YOLOv5s	400	0.01	0.0005	0.937	16	640	–	–	–	–	–
YOLOv5n	400	0.01	0.0005	0.937	16	640	–	–	–	–	–
FDG [a]-YOLOv5s_b	400	0.01	0.0005	0.937	16	640	0.5	1×10^{-3}	5×10^{-4}	5×10^{-4}	5×10^{-6}
FGD [a]-YOLOv5s_n	400	0.01	0.0005	0.937	16	640	0.5	1×10^{-3}	5×10^{-4}	5×10^{-4}	5×10^{-6}
FGD [a]-YOLOv5s_bn	400	0.01	0.0005	0.937	16	640	0.5	1×10^{-3}	5×10^{-4}	5×10^{-4}	5×10^{-6}
FGD [a]-YOLOv5n	400	0.01	0.0005	0.937	16	640	0.5	1×10^{-3}	5×10^{-4}	5×10^{-4}	5×10^{-6}
MSFAD-YOLOv5s	400	0.01	0.0005	0.937	16	640	0.5	1×10^{-3}	5×10^{-4}	5×10^{-4}	5×10^{-6}
MSFAD-YOLOv5n	400	0.01	0.0005	0.937	16	640	0.5	1×10^{-3}	5×10^{-4}	5×10^{-4}	5×10^{-6}

[a] Focal and global knowledge distillation proposed in [21].

4.3. Comparison of Experimental Results

To evaluate the efficacy of the proposed method, a comparative analysis was carried out with widely adopted object detection distillation methods. Table 2 shows that the results demonstrate marked improvements achieved by the method. It yielded a 3.4% and 3.3% increase in mAP_{50} and a 3.3% and 2.2% enhancement in $mAP_{50–90}$ for the student detector. Those advancements exceeded the performance gains observed in other prevalent distillation methods for student detectors. It was found that before distillation, the selected student detector's performance was lower than that of the benchmark detector used for comparison. However, after using the MSFAD distillation, the detection accuracy of YOLOv5s surpassed that of all benchmark student detectors and most teacher detectors. Relative to the distillation results obtained using various benchmark methods, the MSFAD distillation approach produced a higher-performing YOLOv5s detector with substantially fewer parameters than the compared detectors. This outcome highlighted the substantial performance gains the MSFAD method brought to the student detector.

Table 2. Comparison of various distillation methods on the visual object classes dataset. The results of the benchmark methods used for comparison are from [38].

Method	Params (M)	mAP_{50} (%)		$mAP_{50–90}$ (%)	
Faster R-CNN-Res101 (teacher)	232	82.8	Improvement	56.3	Improvement
Faster R-CNN-Res50 (student)	159	82.2		54.2	
+Mimicking [18]	159	82.3	+0.1	55.5	+1.3
+Fine-grained [19]	159	82.2	=	55.4	+1.2
+Fitnet [11]	159	82.2	=	55.1	+0.9
+GID [38]	159	82.6	+0.4	56.5	+2.3
RetinaNet-Res101 (teacher)	217	81.9	Improvement	57.3	Improvement
RetinaNet-Res50 (student)	72.7	80.9		55.4	
+Fine-grained [19]	72.7	81.5	+0.6	56.6	+1.2
+Fitnet [11]	72.7	81.4	+0.5	55.8	+0.4
+GID [38]	72.7	82.0	+1.1	57.9	+1.3
FCOS-Res101 (teacher)	196	81.6	Improvement	58.4	Improvement
FCOS-Res50 (student)	123	80.2		56.1	
+Fitnet [11]	123	80.3	+0.1	57.0	+0.9
+GID [38]	123	81.3	+1.1	58.4	+2.3
YOLOv5l (teacher)	46.5	84.6	Improvement	63.1	Improvement
YOLOv5s (student)	7.2	79.1		54.0	
+MSFAD (ours)	7.2	82.5	+3.4	57.3	+3.3
YOLOv5l (teacher)	46.5	84.6	Improvement	63.1	Improvement
YOLOv5n (student)	1.9	73.1		46.6	
+MSFAD (ours)	1.9	76.4	+3.3	48.8	+2.2

Table 3 compares the MSFAD-distilled model with other lightweight YOLO models. The results show that MSFAD-YOLOv5s outperformed YOLOv3-tiny by 23.3% in mAP_{50} and reduced the model parameters by 78.4%. Similarly, compared to YOLOv3-tiny, MSFAD-YOLOv5n achieved a 17.2% improvement in mAP_{50} with only 5.7% of the model parameters. It also improved the mAP_{50} by 4.7% while reducing the model parameters by 68.1%. MSFAD-YOLOv5n achieved the same detection accuracy as YOLOv4-tiny with only 8% of its parameters. Additionally, MSFAD-YOLOv5s outperformed YOLOv4-S by 1.7% in mAP_{50} while reducing the model parameters by 56.4%. Finally, MSFAD-YOLOv5s outperformed the latest YOLO model, YOLOv7-tiny, by 2.2% in mAP_{50} and 1.9% in $mAP_{50–90}$, with a negligible increase in model parameters.

Table 3. Performance comparison of YOLO lightweight detection algorithm.

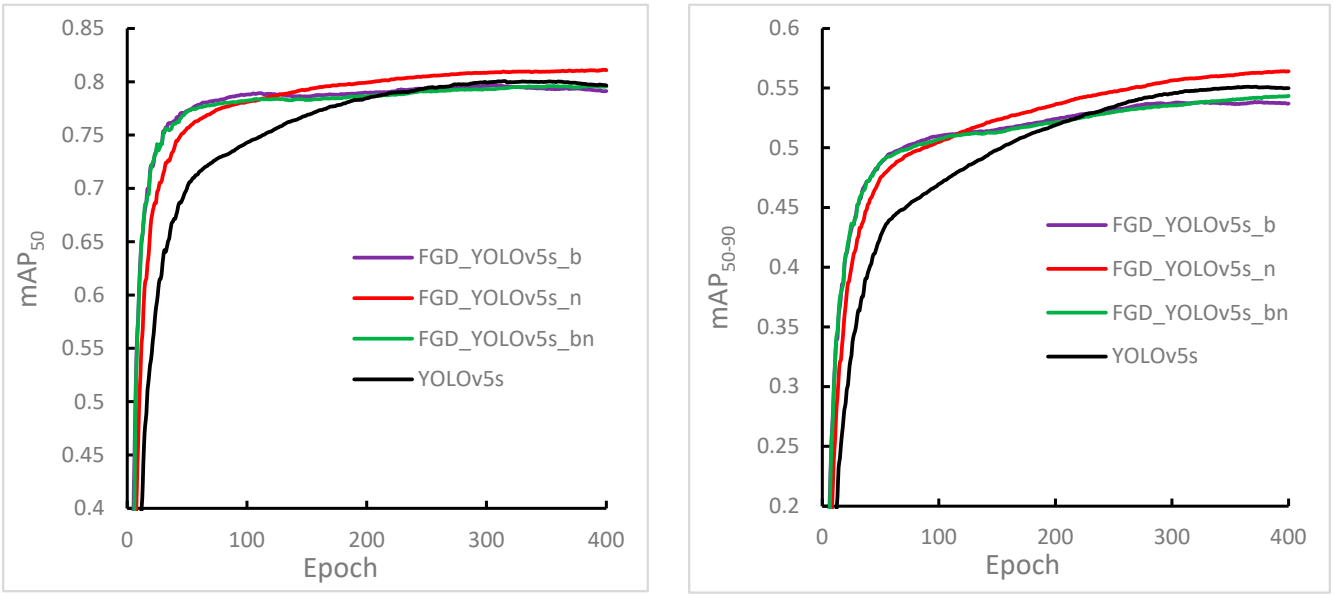
Model	P (%)	R (%)	mAP_{50} (%)	$mAP_{50–90}$ (%)	Params (M)
YOLOv3-tiny	61.5	55.1	59.2	–	17.5
YOLOv4-tiny	79.3	76.0	77.8	–	22.6
YOLOv4-S	78.9	80.1	80.8	–	16.5
YOLOv7-tiny	79.2	76.7	80.3	55.4	6.2
MSFAD-YOLOv5s	80.5	79.8	82.5	57.3	7.2
MSFAD-YOLOv5n	74.5	73.1	76.4	48.8	1.9

In summary, the proposed MSFAD method can substantially improve the detection accuracy of student detectors without adding model parameters.

4.4. Distillation of Semantic Features of Different Levels

To investigate how the semantic level of the distilled features affected the performance of the student detector, three exploratory experiments were carried out using the focal and global knowledge distillation (FGD) method proposed in [21]. YOLOv5s was selected as the student detector, and YOLOv5l as the teacher detector. The experimental conditions were as follows: (1) Distillation of features output only by the neck structure, (2) distillation of features output only by the backbone structure, and (3) distillation of the features output by both the backbone and the neck. Figure 3 shows the experimental results.

In comparing Figure 5a,b, consistent trends were found in the experimental results of mAP_{50} and $mAP_{50–90}$, indicating that the experiments accurately identified the existing problems. Additionally, the comparison between FGD-YOLOv5s_n and YOLOv5s showed that the FGD distillation method was effective for the YOLO detector, with substantial improvements in the detection accuracy and convergence speed of the student detector.



(**a**) (**b**)

Figure 5. Experimental results of distilling features of various semantic levels of YOLOv5 using the focus group discussion method. (**a**) Experimental result of mAP_{50}. (**b**) Experimental result of $mAP_{50–90}$. Among them, YOLOv5s, FGD-YOLOv5s_b, FGD-YOLOv5s_n, and FGD-YOLOv5s_bn indicate that no distillation was carried out, and the features of the 4th, 6th, and 9th semantic level output by the backbone structure were selected for distillation. The features of the 17th, 20th, and 23rd semantic level output by the neck structure were distilled, and the features of the above six semantic levels were selected for distillation.

Analyzing the results of FGD-YOLOv5s_b and FGD-YOLOv5s_n showed that selecting the output features of the backbone structure for distillation in the early training stage helped the student detector converge faster. This outcome indicates that the knowledge contained in the teacher detector's shallow semantic features was more valuable in the initial training stage. As the knowledge accumulated, the value of the shallow semantic features gradually diminished, and the deep semantic features fused by the neck structure became more valuable for the student detector. Therefore, using deeper semantic-level features for distillation in the later training stage led to higher detection accuracy of the student detector. The findings suggest that the performance of the student detector was highly dependent on the semantic level of the distilled features.

Finally, comparing the results of FGD-YOLOv5s_b, FGD-YOLOv5s_bn, and YOLOv5s showed that selecting inappropriate semantic-level features for distillation at different training stages can have adverse effects. Specifically, both FGD-YOLOv5s_b and FGD-YOLOv5s_bn achieved less effective results than those of YOLOv5s without distillation in the later stages of distillation training.

4.5. Visual Analysis of Feature Maps

To determine the effectiveness of the MSFAD method, the feature maps of each model were visualized before and after distillation for the feature maps output by the backbone structure and neck structure. Figure 6 shows the visualization results, where in each group of six images, the first three feature maps were output by the backbone and the second three by the model head. The results indicate that the distillation process markedly improved the feature maps of the model. Specifically, the model head feature maps b-1, b-2, d-1, and d-2 after distillation had more precise features and a better suppression of background noise than the backbone feature maps a-1, a-2, c-1, and c-2 before distillation. Furthermore, b-3 and d-3 after distillation extracted more abstract semantic features than a-3 and c-3, demonstrating that the backbone had more robust feature extraction capability after distillation. Similarly, compared to the feature maps a-4, a-5, c-4, and c-5 output by the head before distillation, the feature maps b-4, b-5, d-4, and d-5 after distillation had cleaner backgrounds, enabling a clearer representation of the posture of the two people. Moreover, b-6 and d-6 after distillation displayed the position information of the two people more clearly and accurately than did a-6 and c-6, with better foreground and background separation.

Visualizing feature maps can gain deeper insights into how the distilled student detector achieved high performance. Compared to feature maps output by the student detector before distillation, the proposed MSFAD method effectively reduced background noise interference. Additionally, the MSFAD method enhanced the student detector's feature extraction ability.

To verify the actual performance of the student detector distilled using the MSFAD method, the detection performances of the models were compared before and after distillation. Heat maps of the detected regions visualized with the use of gradient-weighted class activation mapping (Grad-CAM) [44] are shown in Figure 7. The images in the left column demonstrate the inference results of each model, showing that MSFAD-YOLOv5s distilled by the MSFAD method showed substantially higher detection accuracy than that of the detector before distillation, with the detection accuracy of the person on the left in the images even exceeding that of the teacher detector. The images in the right column display the target regions to which the detector paid attention, demonstrating that the regions of interest of MSFAD-YOLOv5s after distillation were highly similar to those of the teacher network. Relative to the detector before distillation, the target regions to which MSFAD-YOLOv5s paid attention are more precise in position and darker in color. Those results indicate that the MSFAD method can help the student detector accurately identify the features of target categories, leading to higher detection accuracy.

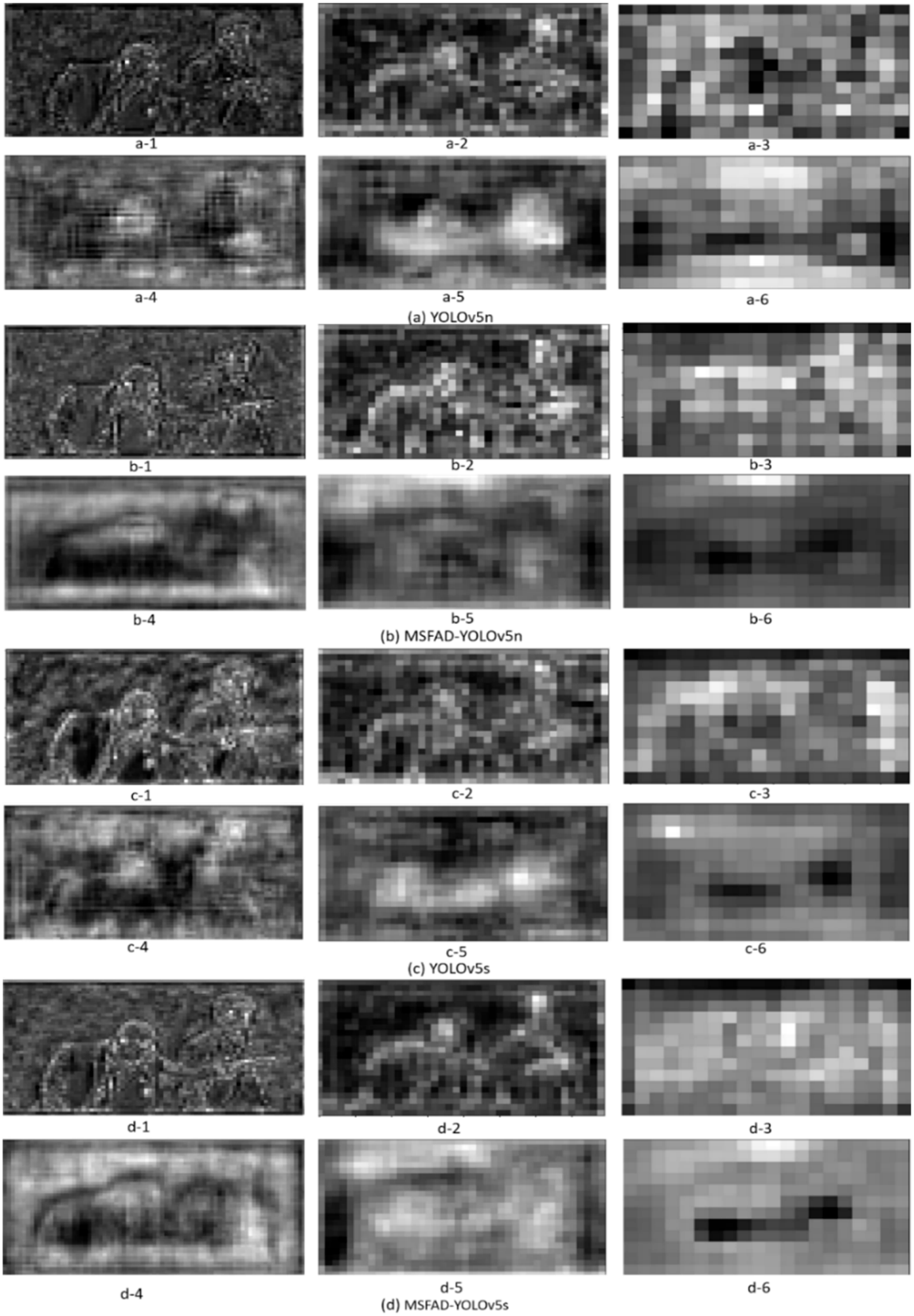


Figure 6. Features of student detector output before and after distillation. In each group of six maps, the first three were output by the backbone and the second three by the model head. (**a**,**c**) Feature maps of the detector's output before distillation. (**b**,**d**) Feature maps after MSFAD distillation.

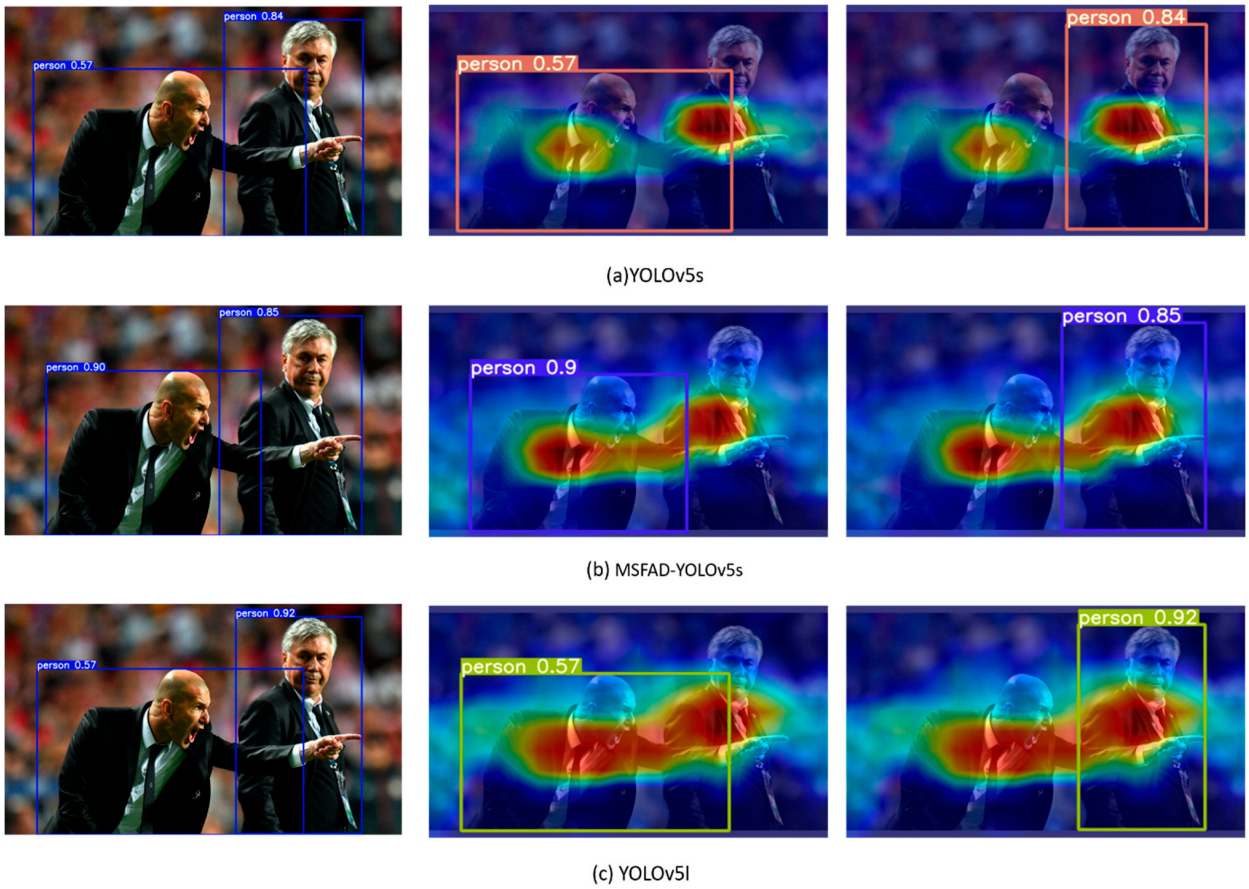


Figure 7. Detection results and corresponding feature heat maps of each model. In each image pair, the left side shows the inference result of the model, and the middle and right side shows the feature area that the inference result focused on for the detection category. The color intensities of each region indicate the contribution of those regions to the detection category, with darker regions indicating greater contributions.

4.6. Ablation Study

4.6.1. Study of Different Distillation Points

In this section, we name various distillation points to demonstrate the effectiveness of our approach. Specifically, we used a state-of-the-art FGD distillation method for comparison. The experimental results are summarized in Table 4.

The results in Table 4 show that regardless of the feature selected for distillation, the proposed MSFAD method consistently outperformed the FGD method in enhancing the performance of student detectors.

Comparing the results of FGD-YOLOv5s-b, MSFAD-YOLOv5s-b, FGD-YOLOv5n-b, and MSFAD-YOLOv5n-b to YOLOv5s and YOLOv5n, we observed that using the FGD method to distill Backbone output features resulted in a degradation in the performance of the student detector compared with the predistillation stage. This indicates that the FGD method was ineffective in mitigating the negative effect of Backbone output features on the student detector during the later stages of model training. In contrast, the MSFAD method timely mitigated that negative effect based on varying training stages and samples, ensuring that the post-distillation performance remained stable.

Finally, comparing the results of FGD-YOLOv5n-n, MSFAD-YOLOv5n-n, FGD-YOLOv5s-n, and MSFAD-YOLOv5s-n showed that Neck output features provided more valuable guidance for student detectors compared with Backbone features. The MSFAD technique effectively harnessed the latent knowledge within the teacher detectors to guide the learning process of student detectors, thereby facilitating higher performance.

Table 4. Experimental results of using global knowledge distillation (FGD) and multilayer semantic feature adaptive distillation (MSFAD) methods to distill the features of different distillation points. FGD-YOLOv5s-n and MSFAD-YOLOv5s-n denote the distillation of Neck output feature maps of YOLOv5s using FGD and MSFAD, respectively. FGD-YOLOv5s-b and MSFAD-YOLOv5s-b denote the distillation of Backbone output feature maps of YOLOv5s using FGD and MSFAD, respectively. FGD-YOLOv5n-b and MSFAD-YOLOv5n-b denote the distillation of Backbone output feature maps of YOLOv5n using FGD and MSFAD, respectively.

Model	P (%)	R (%)	mAP_{50} (%)	$mAP_{50–90}$ (%)	Params (M)
YOLOv5l (teacher)	84.6	78.8	84.6	63.1	46.5
YOLOv5s (student)	80.4	73.1	79.1	54.0	7.2
YOLOv5n (student)	73.2	69.6	73.1	46.6	1.9
FGD-YOLOv5n-n	74.1	72.8	75.0	47.6	1.9
MSFAD-YOLOv5n-n	74.5	73.1	76.4	48.8	1.9
Improvement	+0.4	+0.3	+1.4	+1.2	–
FGD-YOLOv5s-n	79.5	78.9	81.1	56.6	7.2
MSFAD-YOLOv5s-n	80.5	79.8	82.5	57.3	7.2
Improvement	+1.0	+0.9	+1.4	+0.7	–
FGD-YOLOv5s-b	78.8	76.1	78.2	53.1	7.2
MSFAD-YOLOv5s-b	79.7	75.9	79.0	54.1	7.2
Improvement	+0.9	−0.2	+0.8	+1.0	–
FGD-YOLOv5n-b	72.8	68.7	72.5	45.8	1.9
MSFAD-YOLOv5n-b	73.7	70.2	73.3	46.5	1.9
Improvement	+0.9	+1.6	+0.8	+0.7	–

4.6.2. Stability Analysis of Student Models before and after Distillation

This section describes the investigation of the student detector's stability before and after distillation.

Figure 8 shows the loss curves of the model before and after distillation on the training and validation sets. Comparing Figure 8a,b, it can be concluded that the student detector after MSFAD distillation did not have overfitting. On the contrary, the loss of the student detector after distillation on the validation set was less than that of the model before distillation, indicating that the model after distillation performed better on the validation set.

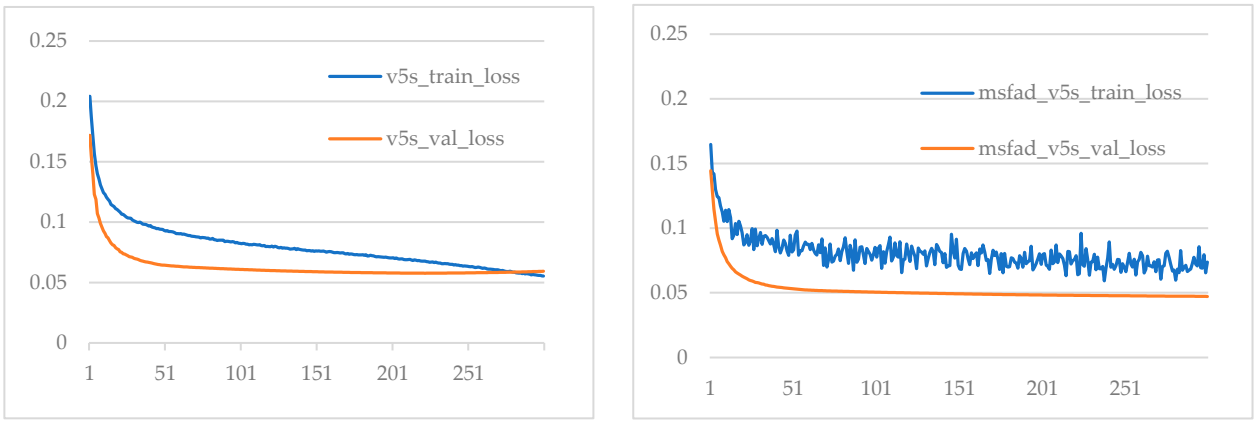
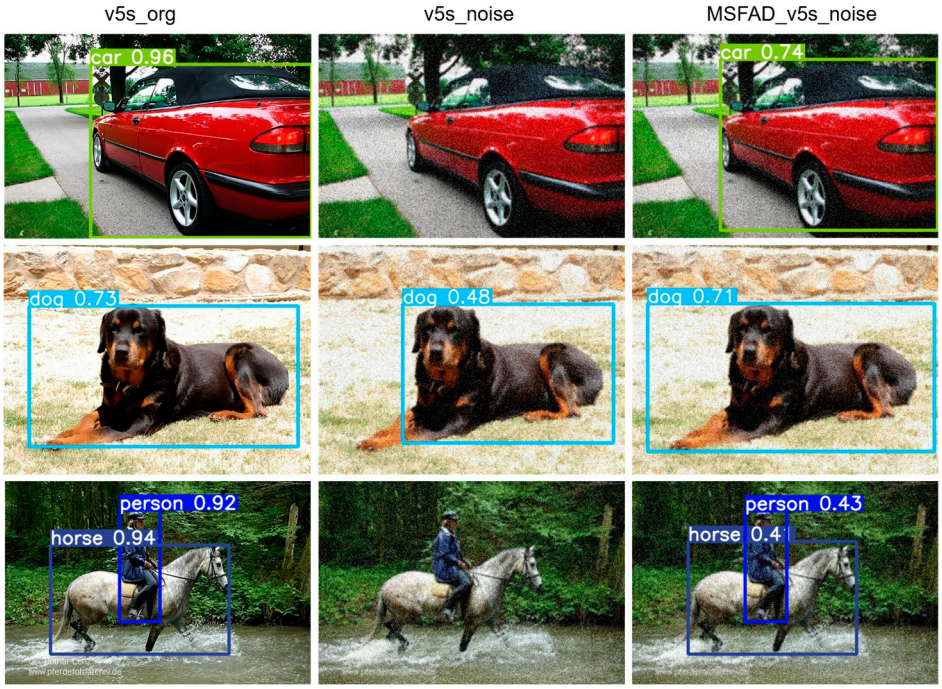


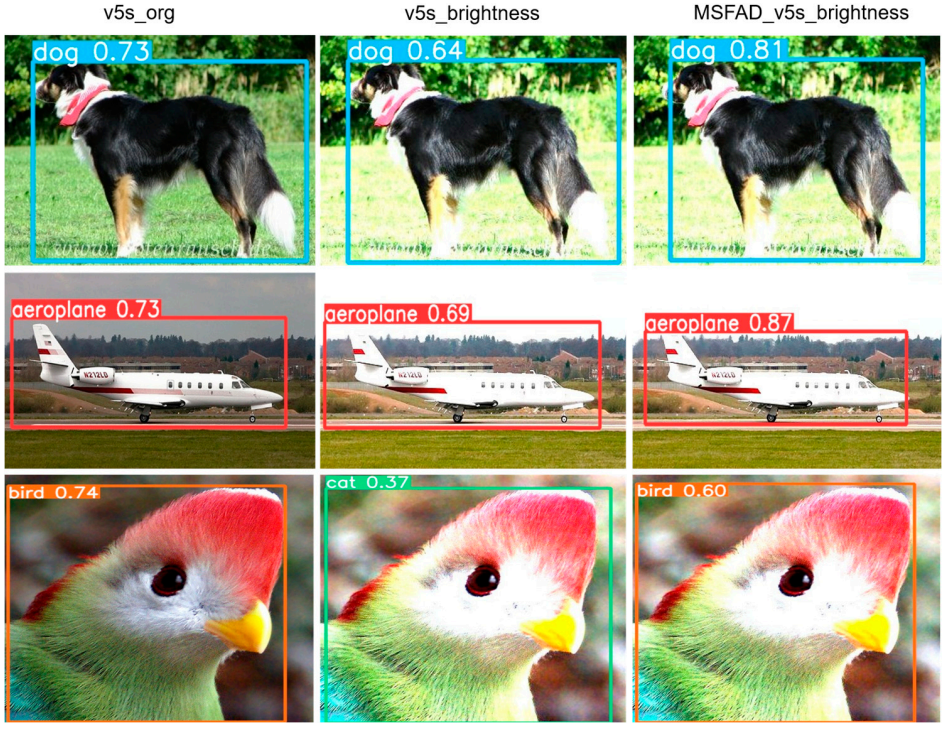
Figure 8. Comparison of the loss between the training and validation sets of the proposed model before and after distillation. (**a**) Loss of the model before distillation. (**b**) Loss of the model after distillation.

Figure 9 shows a stability analysis of the student detector before and after distillation. We examined its performance under three perturbation methods: introducing noise to the image, altering image brightness, and deforming the image. To highlight the improved

stability resulting from distillation, we initially used the non-distilled model to detect normal images as a baseline for comparison. Then, we used the non-distilled model to detect noisy images to assess its resilience to interference. Finally, we applied the MSFAD-distilled model to detect perturbed images, confirming the enhanced stability achieved through the MSFAD method. We selected images with diverse backgrounds and targets to ensure a comprehensive evaluation of our experiments.

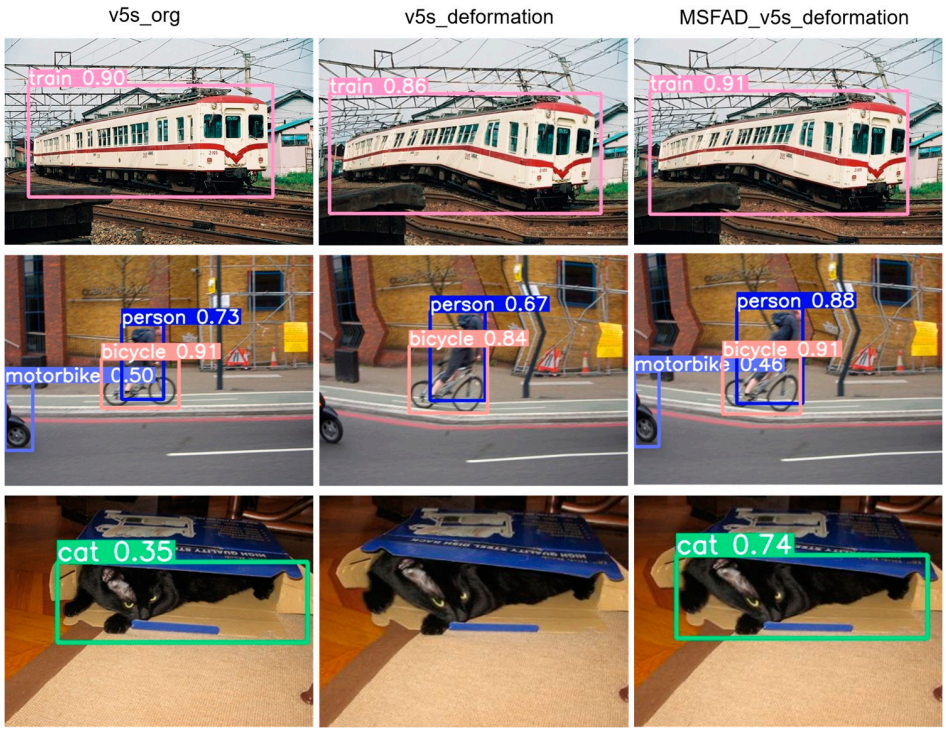


(**a**) Noise



(**b**) Brightness

Figure 9. *Cont.*



(**c**) Deformation

Figure 9. Stability analysis results for the proposed model before and after distillation. Left columns show results of detecting normal images using models without distillation. Middle columns show results of using non-distilled models to detect images with disturbances. Right columns show results of detecting perturbed images using the MSFAD-distilled model. (**a**) Model's detection performance under the influence of added noise. (**b**) Model's response to changes in image brightness. (**c**) Model's behavior when confronted with image deformation.

Figure 9a shows the marked decline in detection performance of the non-distilled model when exposed to noise disturbances, to the extent that it failed to detect targets within the image. In contrast, the model distilled with MSFAD effectively mitigated the effect of noise, resulting in a marked improvement in model stability against noise interference.

Figure 9b highlights the discernible reduction in detection accuracy of the nondistilled model when the image's brightness was adjusted. This decline became evident when a bird was inaccurately identified as a cat. Conversely, the distilled model adeptly mitigated the unfavorable effects of brightness modifications. Remarkably, the detection accuracy of specific targets exceeded that achieved during normal image detection. This pronounced improvement shows the increased stability of student detectors when confronted with fluctuations in brightness.

Figure 9c clearly shows that image deformation markedly undermined the detection accuracy of the non-distilled model, occasionally causing missed detections. In contrast, the MSFAD-distilled model effectively handled image deformation, resulting in substantially enhanced detection accuracy relative to that of the non-distilled model in normal image detection. This observation underscored the distilled model's resilience to image deformation.

In conclusion, the student detector, derived from the MSFAD distillation procedure, had substantial resilience against interference. This capability improved the model's ability to handle complex environmental scenarios.

5. Conclusions

This study demonstrated the negative effect on the student detector's performance of selecting inappropriate semantic-level features for distillation during various training stages. To address this problem, the MSFAD method is proposed, which includes a routing network for the teacher and student detectors and a proxy network for decision making. The method enables the student detector to automatically select appropriate semantic levels to learn from based on the current training stage and training samples. The effectiveness of the proposed method was validated on the YOLOv5 model. The experimental results found substantial performance gains for the student detector over state-of-the-art FGD. The proposed method increased the mAP_{50} of YOLOv5s by 3.4% and the $mAP_{50–90}$ by 3.3%. Moreover, YOLOv5n, with only 1.9 M parameters, achieved detection performance comparable to that of YOLOv5s. Compared to feature maps output by the student detector before distillation, the proposed MSFAD method reduced background noise interference and enhanced the student detector's feature extraction ability.

Our method outperformed mainstream object detection distillation algorithms and delivered substantial performance enhancements to student detectors. However, training the model through distillation demanded substantial graphics memory allocation. For instance, when configuring the batch size to eight and the input image dimensions to 640 × 640, a considerable 38 G of graphics memory became indispensable, and the entire training process consumed approximately 1 week. Future work will address the problem of high memory requirements in feature-based distillation methods during the distillation training process. This is a critical problem that limits the advancement of KD techniques.

Author Contributions: Conceptualization, Z.Z.; methodology, Z.Z. and J.L.; software, J.L.; validation, Z.Z. and J.L.; formal analysis, W.M. and Y.C.; investigation, W.M. and Y.C.; resources, Z.Z.; data curation, J.L.; writing—original draft preparation, J.L.; writing—review and editing, J.L. and Y.C.; visualization, F.H. and L.C.; supervision, Z.Z.; project administration, Z.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Fujian Provincial Marine Economy Development Special Fund Project (grant number FJHJF-L-2022-14).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Two publicly available datasets (Pascal VOC 2007, Pascal VOC 2012) were used to illustrate and evaluate the proposed method. Our code is available at https://github.com/ljq6688/msfad/tree/master (accessed on 24 August 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
2. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
3. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
4. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv* **2015**, arXiv:1506.01497. [CrossRef] [PubMed]
5. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
6. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531.
7. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
8. Wu, J.; Leng, C.; Wang, Y.; Hu, Q.; Cheng, J. Quantized convolutional neural networks for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4820–4828.
9. Wen, W.; Wu, C.; Wang, Y.; Chen, Y.; Li, H. Learning structured sparsity in deep neural networks. *arXiv* **2016**, arXiv:1608.03665.

10. Han, S.; Mao, H.; Dally, W.J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv* **2015**, arXiv:1510.00149.
11. Romero, A.; Ballas, N.; Kahou, S.E.; Chassang, A.; Gatta, C.; Bengio, Y. Fitnets: Hints for thin deep nets. *arXiv* **2014**, arXiv:1412.6550.
12. Heo, B.; Kim, J.; Yun, S.; Park, H.; Kwak, N.; Choi, J.Y. A comprehensive overhaul of feature distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1921–1930.
13. Tung, F.; Mori, G. Similarity-preserving knowledge distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1365–1374.
14. Yim, J.; Joo, D.; Bae, J.; Kim, J. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4133–4141.
15. Komodakis, N.; Zagoruyko, S. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In Proceedings of the ICLR, Toulon, France, 24–26 April 2017.
16. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
17. Chen, G.; Choi, W.; Yu, X.; Han, T.; Chandraker, M. Learning efficient object detection models with knowledge distillation. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Volume 30.
18. Li, Q.; Jin, S.; Yan, J. Mimicking very efficient network for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6356–6364.
19. Wang, T.; Yuan, L.; Zhang, X.; Feng, J. Distilling object detectors with fine-grained feature imitation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–17 June 2019; pp. 4933–4942.
20. Sun, R.; Tang, F.; Zhang, X.; Xiong, H.; Tian, Q. Distilling object detectors with task adaptive regularization. *arXiv* **2020**, arXiv:2006.13108.
21. Yang, Z.; Li, Z.; Jiang, X.; Gong, Y.; Yuan, Z.; Zhao, D.; Yuan, C. Focal and global knowledge distillation for detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 4643–4652.
22. Song, J.; Chen, Y.; Ye, J.; Song, M. Spot-adaptive knowledge distillation. *IEEE Trans. Image Process.* **2022**, *31*, 3359–3370. [CrossRef] [PubMed]
23. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
24. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
25. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
26. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
27. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
28. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
29. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
30. Zhang, Y.; Wang, C.; Wang, X.; Zeng, W.; Liu, W. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *Int. J. Comput. Vis.* **2021**, *129*, 3069–3087. [CrossRef]
31. Li, B.; Ouyang, W.; Sheng, L.; Zeng, X.; Wang, X. Gs3d: An efficient 3d object detection framework for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–17 June 2019; pp. 1019–1028.
32. Feng, D.; Haase-Schütz, C.; Rosenbaum, L.; Hertlein, H.; Glaeser, C.; Timm, F.; Wiesbeck, W.; Dietmayer, K. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 1341–1360. [CrossRef]
33. Jaeger, P.F.; Kohl, S.A.; Bickelhaupt, S.; Isensee, F.; Kuder, T.A.; Schlemmer, H.-P.; Maier-Hein, K.H. Retina U-Net: Embarrassingly simple exploitation of segmentation supervision for medical object detection. In Proceedings of the Machine Learning for Health Workshop, Virtual, 11 December 2020; pp. 171–183.
34. Li, Z.; Dong, M.; Wen, S.; Hu, X.; Zhou, P.; Zeng, Z. CLU-CNNs: Object detection for medical images. *Neurocomputing* **2019**, *350*, 53–59. [CrossRef]
35. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
36. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. Centernet: Keypoint triplets for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6569–6578.

37. Tian, Z.; Shen, C.; Chen, H.; He, T. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9627–9636.
38. Dai, X.; Jiang, Z.; Wu, Z.; Bao, Y.; Wang, Z.; Liu, S.; Zhou, E. General instance distillation for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 7842–7851.
39. Weng, W.; Li, T.; Liao, J.-C.; Guo, F.; Chen, F.; Wei, B.-W. Similarity-based Attention Embedding Approach for Attributed Graph Clustering. *J. Netw. Intell.* **2022**, *7*, 848–861.
40. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
41. Fu, J.; Zheng, H.; Mei, T. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4438–4446.
42. Jang, E.; Gu, S.; Poole, B. Categorical reparameterization with gumbel-softmax. *arXiv* **2016**, arXiv:1611.01144.
43. Everingham, M.; Van Gool, L.; Williams, C.K.L.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2009**, *88*, 303–338. [CrossRef]
44. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.