


RESEARCH

Open Access



MSSD: multi-scale self-distillation for object detection

Zihao Jia^{1*} , Shengkun Sun¹, Guangcan Liu² and Bo Liu³

Abstract

Knowledge distillation techniques have been widely used in the field of deep learning, usually by extracting valid information from a neural network with a large number of parameters and a high learning capacity (the teacher model) to a neural network with a small number of parameters and a low learning capacity (the student model). However, there are inefficiencies in the transfer of knowledge between teacher and student. The student model does not fully learn all the knowledge of the teacher model. Therefore, we aim to achieve knowledge distillation of our network layer by a single model, i.e., self-distillation. We also apply the idea of self-distillation to the object detection task and propose a multi-scale self-distillation approach, where we argue that knowledge distillation of the information contained in feature maps at different scales can help the model better detect small targets. In addition, we propose a Gaussian mask based on the target region as an auxiliary detection method to improve the accuracy of target position detection in the distillation process. We then validate our approach on the KITTI dataset using a single-stage detector YOLO. The results demonstrate a 2.8% improvement in accuracy over the baseline model without the use of a teacher model.

Keywords: Knowledge distillation, Multiscale detection, Feature pyramid networks, Gaussian mask

1 Introduction

Object detection is one of the most important tasks in the field of computer vision. It has attracted increasing attention for applications in areas such as autonomous driving [1]. With the development of deep learning, object detection has also incorporated many learning methods based on convolutional neural networks (CNN) [2, 3], in which the backbone of the detector usually consists of a large number of convolutional operations to achieve better feature extraction. In previous works, in pursuit of better detection results, object detection models became increasingly large in terms of the number of parameters and computational complexity, ignoring the real-time nature of object detection and making it difficult to deploy on low-computing devices such as mobile devices. Therefore, to reduce the complexity of the model, methods such

as quantization [4–7] and pruning [8–10] can effectively reduce the size of the model and improve the speed of detection by pruning redundant connections in the network [11], although at the cost of some reduction in detection accuracy.

Knowledge distillation [12, 13] is an approach that can improve the accuracy of a model without changing the size of the network by learning the behavior of a more powerful network. When knowledge distillation was first proposed, it was used more for image classification tasks and less for object detection, mainly because the soft label output of the teacher network did not directly help the student network to further find the location of the target. Knowledge distillation was then proposed to pass the middle layer feature information from the teacher network to the student network to obtain the needed localization information for object detection. However, this learning process is inefficient, and the student network often does not learn all the knowledge of the teacher network. We therefore want to address the shortcomings of traditional distillation tech-

*Correspondence: 1299339316@qq.com

¹Nanjing University of Information Science and Technology, Nanjing, China
Full list of author information is available at the end of the article

niques by self-distillation [14, 15]. Instead of training a large teacher network, we extract valid information from the student network itself and let the student network be its own teacher, which reduces the computational cost of training.

The framework of the object detection model usually consists of a backbone network, a neck network and a detection head, where the backbone network is responsible for extracting the image feature information and the neck network usually combines different scales of feature map extraction information [16, 17], such as feature pyramid network (FPN) [18, 19] and path aggregation network (PAN) [20], to better fuse the semantic and positional information of the backbone and neck networks interactively. The final detection head is responsible for detecting the output feature map. In previous knowledge distillation methods used for object detection, the output feature maps of intermediate layers in the backbone network were typically distilled. This involved assigning a global weight to the information in the teacher network's feature map, which the student network could then learn from. However, we believe that the information in the neck network is richer. The shallow network feature maps contain information at a larger scale, which is better suited for detecting small targets, while the deeper network feature maps are at a smaller scale and are better suited for detecting large targets. We perform distillation learning for three different scales of feature maps in the neck network simultaneously, and let the output part of the network learn feature information of the corresponding scale in its own middle layer. This approach can be more conducive to detecting multi-scale targets. Our method differs from other multi-scale detection modules [21] in that it does not change the network structure and only performs distillation operations within the original network structure, and aims to explore the feature information of different degrees of deep and shallow networks.

Moreover, knowledge distillation methods commonly used in the field of object detection distill and learn the entire feature map information, which can lead to errors in the process of determining the target position, making it difficult to accurately capture the target location information. Therefore, we want to learn more effectively about the local area of the target during the distillation process. In this case, we have added a Gaussian mask for assisted detection [22, 23], whose main purpose is to distinguish the foreground from the background region by first encoding the ground truth region of the target with Gaussian values and then setting the rest of the background region to 0 for processing. Finally, we calculate the mean square error loss between this encoded region and the output feature map of the model. The Gaussian mask we generate can match output feature maps at different scales through the feature adaptation layer, enabling its application in a variety of object detectors to improve detection accuracy.

In summary, this article has the following key contributions.

- 1) Our strategy eliminates the need to train a huge teacher model. Instead, a simple model needs to be trained as its own teacher. This approach eliminates significant training time and computational costs, and facilitates real-time processing and deployment on the device side.
- 2) We propose a multi-scale distillation scheme by distilling the feature maps at different scales in the neck network to extract their effective feature information and calculate the distillation loss between them and the output feature maps. This method facilitates the detection of multi-scale targets, where the accuracy of small target detection is also improved.
- 3) In addition, in order to achieve more accurate positioning of the target during distillation learning, we first generate a Gaussian mask to distinguish the foreground and background in the image processing stage, and calculate the mask loss between the detection stage and the output result to improve the detection accuracy of the model. These methods do not change the basic structure of the model, so they do not significantly increase the number of parameters.

2 Related works

Object detection, as one of the most important tasks in computer vision, aims to find the class and position of a given target in an image. In the last few years, object detection has evolved very rapidly. There are two main categories, one of which is two-stage detectors such as Faster-RCNN [24], Mask-RCNN [25], and Cascade-RCNN [26]. This type of algorithm usually generates candidate frames first and then fine-tunes the bounding boxes. The other class is the single-stage detectors represented by YOLO [27–31], SSD [32–34], FCOS [35] and RetinaNet [36, 37]. The single-stage algorithms do not generate candidate regions, but directly classify and localize the targets. Over time, both types of algorithms have been improving their model structures in order to improve detection efficiency. Although they are now equipped with richer network structures, the computational cost and network size of these algorithms are gradually increasing, making it difficult to meet the requirements of mobile deployment. Designing a lightweight backbone network of detectors [38, 39] has therefore become a research trend to speed up detection. In addition, there are a number of studies that aim to transfer knowledge from a large detector to a simple detector, which is also a research approach to improve the performance of small detection models.

Knowledge distillation (KD) has become one of the most effective techniques for compressing large models into

smaller and faster models, and can improve its own detection accuracy by learning from the knowledge of large models compared to pruning and quantization techniques. The idea of knowledge distillation was first proposed by Bucila et al. [40] and popularized by Hinton et al. [41] to transfer knowledge from the teacher's network to the student's network through soft-labeled output. fitNets [42] showed that in addition to the loss of KD, the feature information in the middle layer of both networks could also be used to guide the student's knowledge. However, the idea of knowledge distillation was more often applied to image classification tasks at that time, and subsequently the knowledge distillation approach was also widely used in the field of object detection, where Chen et al. [43] performed knowledge distillation from three parts: the backbone network, the neck features and the detection head. FGI [44], on the other hand, instructed students by extracting fine-grained features in the foreground object region, leaving them with only ground truth neighborhoods. DeFeat [45] considered that the background region also contained useful information, so a decoupling of the foreground and background regions were used to transfer useful knowledge to the student network through the decoupling of neck features and the decoupling of classification heads, respectively.

With the rapid development of knowledge distillation, it was found that there was a limit to what a student network could learn from a teacher network, and that there were limits to the efficiency of that learning process. This was the reason why the idea of self-distillation was born. Students use only what they have learned inside the model to guide themselves without the guidance of a large teacher model. SAD [46] is a classic self-distillation framework that allows a network to use the attention map obtained from its own middle layer as its distillation target for lower layers, without proper labeling or additional supervision, that is, to perform distillation learning through top-down and layered attention maps within the network itself. DLB [47] is also a fast self-distillation framework, which mainly distills the soft targets generated from the previous iteration by half of each small batch. This method does not require additional runtime memory or modification of the model structure. As a relatively advanced self-distillation framework, LGD [48] enhances the relationship with the appearance of the target through label guidance and self-attention mechanism, thereby improving detection accuracy. FRSKD [49], on the other hand, is a self-distillation approach based on data augmentation (the network produces consistent predictions for targets of the same class of objects) and auxiliary network (using additional branches in the middle of the classifier network and guiding these branches to similar outputs through knowledge distillation), respectively. The approach uses soft labels and feature graphs for self-distillation, combining different depth

feature layers for integration and refinement to guide the feature maps at the same level. Moreover, we find that the knowledge learned from feature maps at different scales can also help the network to improve its accuracy to address the detection needs of multi-scale targets.

3 Method

3.1 Multi-scale distillation loss

In this section, we describe our proposed multi-scale distillation framework in detail. As a classic object detector, you only look once (YOLO) incorporates a multi-scale detection structure in the model. Considering the volume and computational cost of the model, we choose the YOLOv5 network as the main framework for the experiment. Its main framework is shown in Fig. 1 and contains three main components: the backbone network, the neck network and the detection head.

As CNN-based detectors continue to evolve and the need for multi-scale target detection grows, different detectors have added modules to their networks that facilitate multi-scale target detection. Figure 2 demonstrates the network structure of YOLOv5, where the neck network is a combination of FPN+PAN modules. We use the feature layers of different scales of the FPN as teachers, and the deeper PAN output part as students. The scale of the feature map has changed three times. Since the feature map with a larger scale has a smaller downsampling rate compared with the original image, and the receptive field is smaller, we can detect some objects with smaller scales, and the smaller anchor is assigned. Therefore, we detect large targets (20×20) on small feature maps, medium-sized targets (40×40) on medium-sized feature maps, and small targets (80×80) on large feature maps. We calculate the distillation losses between feature maps of the same scale. The distillation losses of three different scales are calculated separately, and in order to better improve the detection accuracy of targets at different scales, we assign different weight coefficients to the three losses.

$$L_s = \frac{\gamma}{N} \sum_{h=1}^H \sum_{w=1}^W \sum_{c=1}^C (S_{(h,w,c)}^s - T_{(h,w,c)}^s)^2, \quad (1)$$

$$L_m = \frac{1}{N} \sum_{h=1}^H \sum_{w=1}^W \sum_{c=1}^C (S_{(h,w,c)}^m - T_{(h,w,c)}^m)^2, \quad (2)$$

$$L_l = \frac{1}{N} \sum_{h=1}^H \sum_{w=1}^W \sum_{c=1}^C (S_{(h,w,c)}^l - T_{(h,w,c)}^l)^2. \quad (3)$$

The above equations show the calculation process of distillation loss for three different scales of targets, where L_s ,

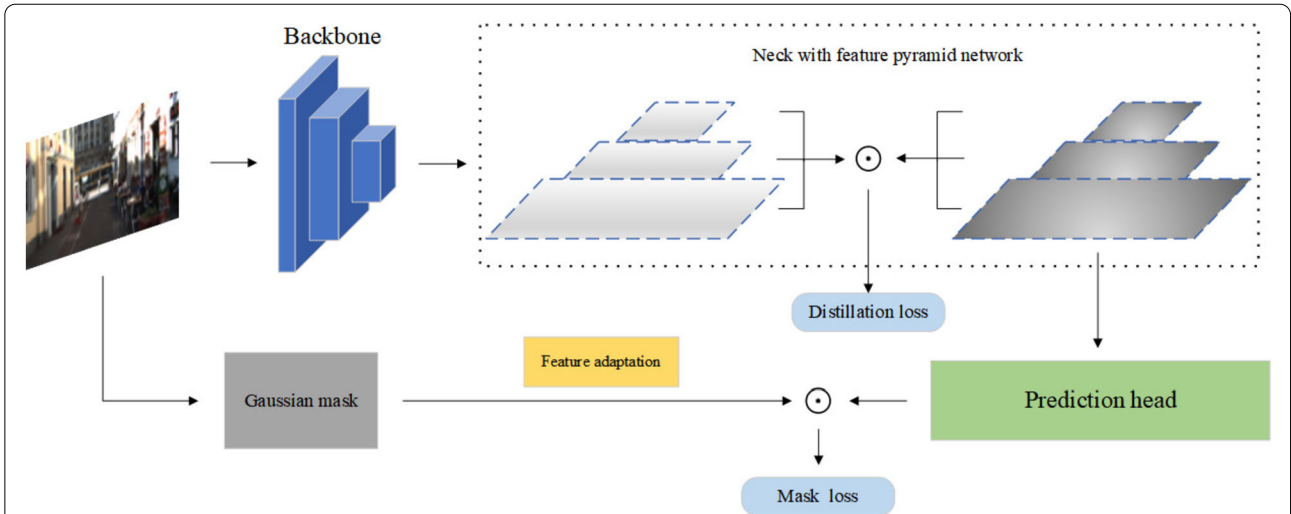


Figure 1 Overview of the proposed multi-scale self-distillation framework, which extracts feature maps at different scales from the feature pyramid network (FPN) structure of the neck network for shallow and deep layers, respectively, and distills them using sibling feature maps between feature layers of different depths. We then generate a Gaussian mask for the input image from the ground truth and calculate the mask loss between the output feature maps and the detection head. The yellow part is the feature adaptation layer of the Gaussian mask

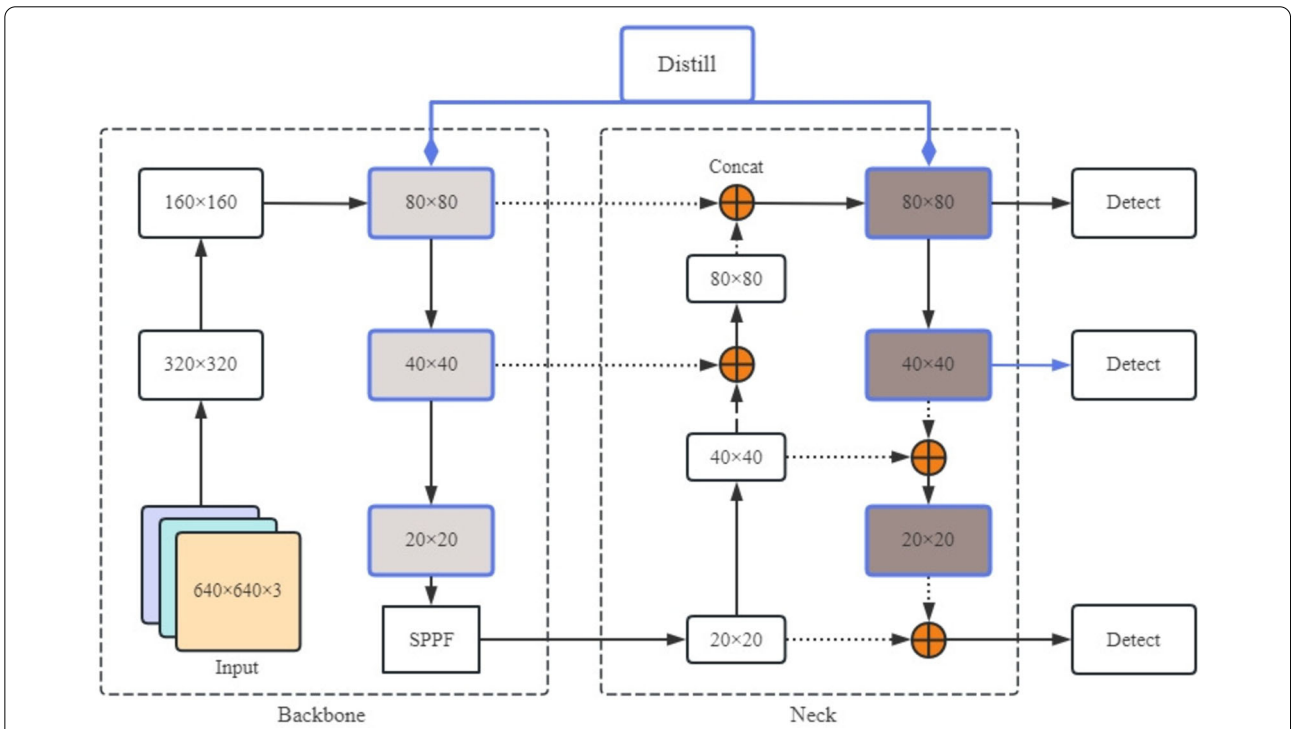


Figure 2 YOLOv5 network structure diagram and distillation layer indicator diagram, where SPPF refers to the spatial pyramid pooling fast module

L_m and L_l represent distillation loss for small, medium and large scale targets, respectively, $S_{(h,w,c)}^k$ and $T_{(h,w,c)}^k$ ($k = s, m, l$) are the feature maps of students and teachers corresponding to different size scales, and $N = HWC$ represents the total number of elements. Meanwhile, small tar-

get detection has been a major difficulty in the study. We add a weighting factor γ to the small target loss to balance the distillation loss scale and we can adjust the γ factor in the experiment. We then weight the three losses to obtain the total distillation loss $L_{distill}$ that is displayed in Eq. (4),

where k indicates the number of feature maps.

$$L_{\text{distill}} = \frac{1}{k}(L_s + L_m + L_l). \tag{4}$$

3.2 Mask assisted detection

To improve the target localization efficiency and detection accuracy, we design a mask-assisted detection method that focuses on the output feature map of the network. We find that features in the central region of the target can be better generalized to the model, so we introduce a Gaussian mask to highlight the ground pixel features of the target region and suppress the surrounding background region when the image is input to the network. Specifically, assuming that the true frame region of the target is B , the size is W and H , and the center coordinates are (x_c, y_c) . The Gaussian mask is defined as follows in Eq. (5).

$$M_{(x,y)} = e^{-\frac{(x-x_c)^2}{\sigma_x^2} - \frac{(y-y_c)^2}{\sigma_y^2}}, \quad (x, y) \in B. \tag{5}$$

The current pixel point coordinates are denoted as $(x, y) = 0$ when the pixel coordinates do not fall within the groundtruth region. Where σ_x^2 and σ_y^2 represent the decay factors for the coordinates in both directions, we set $\sigma_x^2 = \sigma_y^2$ for ease of calculation. This mask is only valid within the target truth frame and is equal to 0 in all the other regions, so we hope that the mask will help the network to focus more on the foreground region. The visual results of a Gaussian mask are displayed in Fig. 3.

Since the mask that we generate is based on the size of the input image, and the size of the feature map and the number of channels may be different after the network processing, we need to add a feature adaptation layer so that the mask corresponds to the size of the feature map and the number of channels. The structure of this feature adaptation layer is relatively simple, consisting of a convolutional transformer layer and a ReLU activation layer. The mask assisted detection flowchart is demonstrated in Fig. 4.

After passing through the feature adaptation layer, the Gaussian mask has the same size and channel as the output feature map, and then the mask loss L_{mask} between them is calculated and continuously minimized by training the network:

$$L_{\text{mask}} = \frac{1}{N} \sum_{h=1}^H \sum_{w=1}^W \sum_{c=1}^C (M_{(h,w,c)}^{\text{adap}} - F_{(h,w,c)})^2, \tag{6}$$

where $N = HWC$ is the total number of pixels, $M_{(h,w,c)}^{\text{adap}}$ is the mask adjusted by the feature adaptation layer and $F_{(h,w,c)}$ is the output feature map. Mask-assisted detection helps the output feature map to better highlight information about the target and suppress information in the background region, improving detection results.

Combining the multi-scale distillation losses L_{distill} introduced previously with the losses L_{gt} generated by the training of detector, we define the total losses L used by the algorithm in this paper as a weighted calculation of the

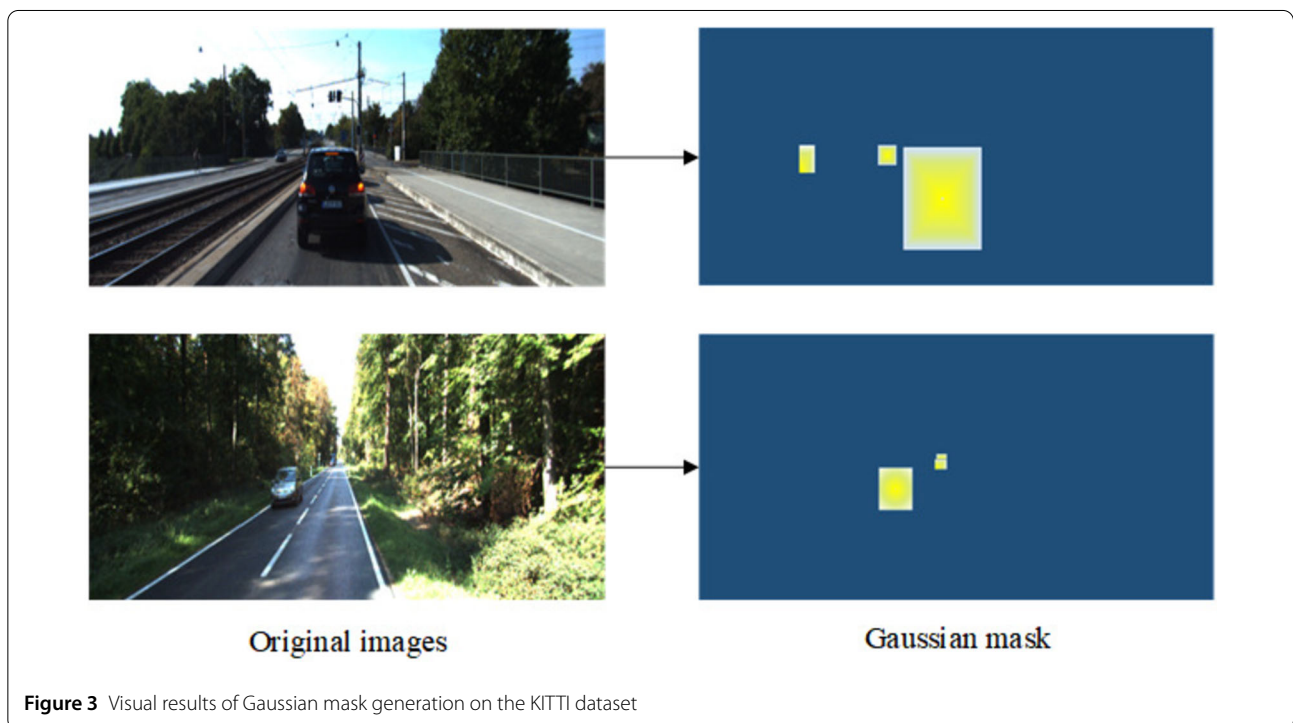
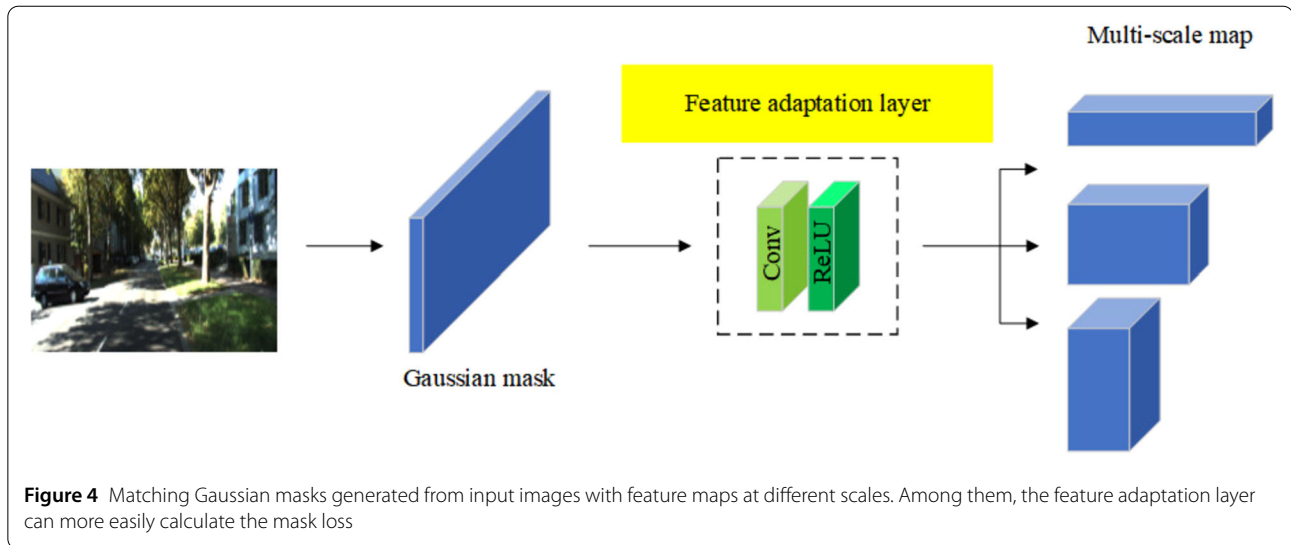


Figure 3 Visual results of Gaussian mask generation on the KITTI dataset



three losses, as shown in Eq. (7).

$$L = L_{gt} + L_{distill} + L_{mask}. \quad (7)$$

4 Experiments

4.1 Experimental setup

To validate the effectiveness of our method, we first conduct experiments on a state-of-the-art single-stage detector, YOLOv5, using the KITTI dataset. The KITTI dataset is currently the largest computer vision evaluation dataset in the world for autonomous driving scenarios, including real image data collected from urban, rural and highway scenes. Each image can contain up to 15 vehicles and 30 pedestrians, meeting the needs of multi-scale and multi-objective detection. It includes 7481 training set images and 7518 test set images and mainly detects three types of targets: vehicles, pedestrians and cyclists. To validate the generalization of our method, we also apply the improved method to different detectors for comparison experiments, where the evaluation metric is the average accuracy, i.e. ΔmAP , AP_{50} , AP_{75} , AP_s , AP_m and AP_l , with the last three evaluating the accuracy of different scales of targets.

All experiments are conducted in Windows 11, CUDA 11.2 environment, GPU configuration: NVIDIA RTX 2080ti, and PyTorch is used as the main framework. The number of training iterations for all experiments is set to 200 epochs, the processing size per batch is set to 16, the learning rate decay strategy is cosine annealing, and the initial learning rate is set to 0.01 and the cycle learning rate is 0.1.

4.2 Experimental results and analysis

YOLO, as a classic object detector, has released multiple versions in recent years. To select the most suitable model for our method, we conduct comparative experiments on

different versions of YOLO models. The experimental results are depicted in Table 1. The experimental results show that our method has improved performance on different versions of YOLO models. Among them, YOLOv3 and YOLOv4 network models have too large structures, which result in low accuracy. Although YOLOv7 has high detection accuracy, it also has high requirements for memory usage. Therefore, the experimental framework of this article chooses a more balanced YOLOv5 model.

We conduct ablation experiments using small(s) and medium(m) versions of YOLOv5, respectively, and the experimental results show that both mask assisted detection(MAD) and multi-scale self-distillation(MSSD) improve the detection of the model, with the MSSD method improving the detection of multi-scale targets in images more significantly. The results of the module ablation experiment are presented in Table 2. When the model is chosen as the version of YOLOv5s, MSSD could help small target accuracy AP_s improved by 3.9%, and the ΔmAP of the model improved by 2.8% after combining the two improved methods. The network training process before and after using MSSD is shown in Fig. 5.

It is worth noting the change in the number of model parameters and the amount of computation. Our improved method does not increase the overall computational pressure on the model, except for the feature adaptation layer, which increases the number of parameters, but the rest of the work is computed outside the network framework and does not increase the number of parameters or the amount of computation.

In addition, we have also conducted ablation experiments with our improved method on single stage detectors such as YOLOX and FCOS, adding MAD and MSSD methods to YOLOX-s and FCOS networks, respectively. The experimental results are demonstrated in Table 3. Exper-

Table 1 Comparative experiments on the parameters of different versions of the YOLO model

Model	FLOPs (G)	Params (M)	FPS (Frame/s)	Baseline (%)	Ours (%)	ΔmAP (%)
YOLOv3	193.89	61.53	54.6	32.5	33.2	+0.7
YOLOv4	119.83	52.50	55.2	33.2	35.4	+2.2
YOLOv5s	16.14	7.10	95.1	35.6	38.4	+2.8
YOLOv6s	44.12	17.19	97.0	34.9	36.7	+1.8
YOLOv7	103.50	36.49	79.5	37.3	39.2	+1.9

Table 2 Results of ablation experiments with different versions of YOLOv5 using the improved method. A tick in the box indicates that the method was used

Model	FLOPs(G)	Params(M)	MAD	MSSD	AP_{50} (%)	AP_{75} (%)	AP_s (%)	AP_m (%)	AP_l (%)	mAP (%)
YOLOv5m	48.67	21.09			78.5	44.5	21.1	43.2	61.9	42.4
	49.11	21.33	✓		79.6	44.6	21.4	44.1	62.6	43.1
	49.24	21.16		✓	80.1	44.9	22.7	44.5	62.5	43.6
	49.28	21.53	✓	✓	81.7	45.5	23.0	45.7	63.2	44.3
YOLOv5s	15.89	7.05			70.2	33.5	18.2	36.9	53.5	36.6
	16.03	7.09	✓		71.5	33.4	20.9	38.0	53.9	37.6
	16.07	7.06		✓	71.8	33.9	22.1	38.3	54.1	38.4
	16.14	7.10	✓	✓	73.2	34.6	24.4	40.2	55.8	39.4
YOLOv5n	4.19	1.77			55.1	22.1	15.2	28.2	44.1	29.4
	4.24	1.79	✓		57.1	22.6	18.7	29.1	44.6	30.8
	4.27	1.77		✓	57.6	23.1	19.1	30.6	46.9	31.1
	4.31	1.79	✓	✓	59.4	23.9	22.5	32.4	48.8	33.2

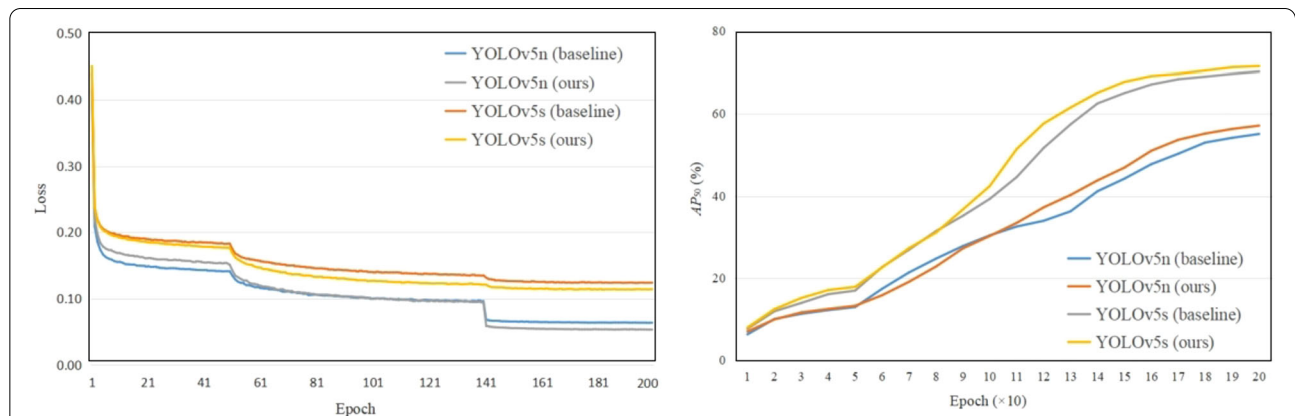


Figure 5 Comparison of training processes before and after the improvement of different versions of YOLOv5. The left sub-figure shows the convergence comparison results of the training losses, and the right sub-figure shows the comparison results of the AP_{50} training processes

Table 3 Comparison of ablation experiments using our method with single stage detectors of different frameworks(YOLOX and FCOS). A tick in the box indicates that the method was used

Model	FLOPs (G)	Params (M)	MAD	MSSD	AP_{50} (%)	AP_{75} (%)	AP_s (%)	AP_m (%)	AP_l (%)	mAP (%)
YOLOX-s	26.81	8.94			83.9	49.2	24.4	47.9	61.2	41.5
	27.21	9.74	✓		84.3	49.5	24.7	47.7	61.6	41.7
	27.69	9.35		✓	85.4	50.4	26.9	48.6	61.6	42.7
	28.53	10.12	✓	✓	86.1	51.2	27.3	50.3	62.2	43.6
FCOS	76.13	29.46			80.4	48.7	20.9	47.1	63.0	40.7
	76.92	30.76	✓		82.1	49.2	21.5	47.8	63.4	41.2
	77.39	30.13		✓	82.7	50.1	23.2	47.7	63.6	41.8
	78.98	31.88	✓	✓	83.2	50.6	23.9	48.5	64.1	42.5

imental results show that our method can effectively improve the detection accuracy of the model, with YOLOX's ΔmAP improving by 2.1% and FCOS's ΔmAP improving by 1.8%.

We also compare our method with some classical methods used for knowledge distillation on object detection, where KD, fine-grained feature imitation (FGFI) and distilling object detectors via decoupled features (DeFeat) are teacher-based methods, and we use YOLOv5s as the student model with two backbones, CSPDarkNet and ConvNext. The teacher model is set to YOLOv5m. Label-guided self-distillation (LGD) is the recently proposed teacherless distillation method. Our comparative experiment replaces the backbone of all models with CSPDarkNet. The experimental results are displayed in Table 4. It can be found that the accuracy of LGD is superior to that of FGFI and DeFeat by 0.6% and 0.2%, respectively, while the accuracy of our method is the same as that of LGD at 39.4%. When ConvNext is chosen as backbone, the detection accuracy outperforms CSPDarkNet, with our method outperforming LGD by 0.2%. However, ConvNext has a larger number of module parameters and a slower training speed, which may not be conducive to model lightweighting and real-time target detection.

Table 4 Experimental results comparing the YOLOv5s model with different knowledge distillation methods using CSPDarkNet and ConvNext as the backbone, where the evaluation metric is mAP (%)

Method	Teacher	Student backbone	
		CSPDarkNet	ConvNext
Baseline	N/A	36.6	44.1
KD	YOLOv5m	37.1	44.9
FGFI	YOLOv5m	38.8	45.5
DeFeat	YOLOv5m	39.2	45.8
LGD	N/A	39.4	45.5
Ours	N/A	39.4	45.7

Table 5 Comparison of experimental results for hyperparametric tuning experiments, where the parameters of γ are adjusted in the multi-scale self-distillation (MSSD) method

Model	γ	AP_s (%)	AP_m (%)	AP_l (%)	mAP (%)
YOLOv5s(baseline)	–	18.2	36.9	53.5	36.6
+MSSD	1.000	20.9	37.6	54.3	37.3
+MSSD	0.100	21.7	38.1	54.1	37.7
+MSSD	0.001	19.8	37.4	53.9	36.6
+MSSD	0.010	21.4	37.8	54.4	37.9
+MSSD	0.050	22.1	38.3	54.1	38.4

Table 6 Hyperparametric tuning experiments for Gaussian masks using the experimental model YOLOv5s+MAD

$\sigma_x^2 = \sigma_y^2$	0.5	1.0	2.0	4.0	8.0
mAP	36.9	37.4	37.6	37.4	36.7

The detection of multi-scale targets has been the focus of research in this field, with the detection of small targets being one of the difficult areas. We propose a multi-scale self-distillation method that helps to improve the detection of multi-scale targets. As shown in Eq. (1), we can find that decreasing γ can reduce the small target loss L_s accordingly and optimize this loss more effectively, where the γ parameter regulates the scale of small target loss. Table 5 demonstrates the results of the parameter comparison experiments. $\gamma = 1$ means that the loss weights of the three scales are equal, while γ is too small to be 0.001, which can lead to local optimization of target loss and make it difficult to train the model more effectively. The small target accuracy AP_s improves the most when γ is set to 0.05.

Our proposed mask-assisted detection method focuses on highlighting features in the foreground region by changing the discrepancy between the foreground and the background. The experimental results are shown in Table 6. The Gaussian mask is generated by Eq. (5), where the parameters σ_x^2 and σ_y^2 adjust the influence range of the mask, and we set $\sigma_x^2 = \sigma_y^2$ for convenience. If the parameter is larger, the Gaussian mask will be scattered toward the boundary of the ground truth, and if the parameter is smaller, the Gaussian mask will be more concentrated in the central region of the ground truth. When $\sigma_x^2 = \sigma_y^2 = +\infty$ is set, the Gaussian mask actually becomes a binary mask based on the ground truth. To verify the validity of the Gaussian mask, we conducted an experiment by adjusting different values of σ_x^2 and σ_y^2 . The results show that the Gaussian mask assists best when $\sigma_x^2 = \sigma_y^2 = 2$.

To verify the generalization of our method, we apply our proposed method to different object detectors to test the effectiveness. The single-stage detectors such as YOLOX, RetinaNet and FCOS contain multi-scale detection structures, so we add MSSD to these detectors. In the FPN structure of the two-stage detector, there are significant changes in the scale of the feature map and large convolutional kernels, resulting in a large overall parameter and

Table 7 Comparison of experimental results of MSSD applied to different detectors

Detector	FLOPs(G)	Params(M)	Baseline(%)	Ours(%)	ΔmAP (%)
YOLOv5s	16.14	7.10	36.6	39.4	+2.8
YOLOX-s	28.53	10.12	41.5	43.6	+2.1
FCOS	78.98	31.88	40.7	42.5	+1.8
RetinaNet	82.97	36.43	34.5	35.6	+1.1
Faster-RCNN	91.11	41.22	34.8	36.3	+1.5
Cascade-RCNN	118.86	68.96	38.7	40.5	+1.8



Figure 6 Comparison of multi-scale self-detection(MSSD) detection before and after using the KITTI dataset, with baseline detection results on the left and MSSD detection results on the right

computational complexity of the network. Therefore, it is necessary to change the original network structure in order to generate a distillation layer. Although this will improve the detection accuracy to a certain extent, it will also increase the computational cost of the network and slow the detection speed. The experimental results are depicted in Table 7, where it can be seen that the accuracy of the detectors was improved after the addition of MSSD, with the ΔmAP of RetinaNet and FCOS improving by 1.1% and 2.1%, respectively. YOLOv5s has the most obvious improvement effect, with a 2.8% increase in ΔmAP .

A comparison of our test results on the KITTI dataset is shown in Fig. 6, which focuses on the detection of car, pedestrian and cyclist targets. Generally speaking, cars occupy more anchors than pedestrians and cyclists at the same distance, so many car targets in the image are large, while many cyclist and pedestrian targets are small and medium-sized, which makes the detection process more difficult. Our method can effectively improve the detection of small and medium-sized targets, solving the problem of missed detection and false detection caused by target occlusion in the long-range view.

5 Conclusion

In this paper, we propose a novel self-distillation framework, called MSSD, which is mainly used for knowledge distillation of multi-scale targets. It targets the multi-scale detection module structures such as FPN and PAN. In the network structure, we use shallow networks as teachers and deep networks as students to extract information from feature maps of different scales, calculate corresponding multi-scale target losses, and perform distillation. Among them, the small target loss is optimized to effectively improve the detection accuracy of small targets. In addition we add a Gaussian mask based on the real frame of the target to mask assisted detection, which can suppress the background region information to highlight the feature information of the target during detection. Our approach is computationally inexpensive without the guidance of a large teacher model. Our approach demonstrates good performance compared to other methods and can be applied to different object detectors.

Funding

This work is supported by the New Generation AI Major Project of Ministry of Science and Technology of China (No. 2018AAA0102501).

Abbreviations

CNN, convolutional neural networks; FPN, feature pyramid network; KD, knowledge distillation; MAD, mask assisted detection; MSSD, multi-scale self-distillation; PAN, path aggregation network; YOLO, you only look once.

Data availability

The data that support the findings of this study are available from the corresponding author, upon reasonable request.

Declarations

Competing interests

Guangcan Liu is an Associate Editor at Visual Intelligence and was not involved in the editorial review of this article or the decision to publish it. The authors declare that they have no other competing interests.

Author contributions

All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by ZJ, SS and GL. The first draft of the manuscript was written by ZJ and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Author details

¹Nanjing University of Information Science and Technology, Nanjing, China.

²Southeast University, Nanjing, China. ³JD Finance America Corporation, Mountain View, USA.

Received: 7 June 2023 Revised: 15 February 2024

Accepted: 16 February 2024 Published online: 21 March 2024

References

1. Yurtsever, E., Lambert, J., Carballo, A., & Takeda, K. (2020). A survey of autonomous driving: common practices and emerging technologies. *IEEE Access*, 8, 58443–58469.
2. Gidaris, S., & Komodakis, N. (2015). Object detection via a multi-region and semantic segmentation-aware CNN model. In *Proceedings of the IEEE international conference on computer vision* (pp. 1134–1142). Piscataway: IEEE.
3. Du, J. (2023). Understanding of object detection based on CNN family and YOLO. Retrieved November 2, 2023, from <https://iopscience.iop.org/article/10.1088/1742-6596/1004/1/012029/pdf>.
4. Polino, A., Pascanu, R., & Alistarh, D. (2018). Model compression via distillation and quantization. [Poster presentation]. Proceedings of the 6th international conference on learning representations, Vancouver, Canada.
5. Zhou, Y., Moosavi-Dezfooli, S. M., Cheung, N. M., & Frossard, P. (2018). Adaptive quantization for deep neural network. In S. A. McIlraith & K. Q. Weinberger (Eds.), *Proceedings of the 32nd AAAI conference on artificial intelligence* (pp. 4596–4604). Palo Alto: AAAI Press.
6. Peterson, H. A., Ahumada, A. J., & Watson, A. B. (1993). Improved detection model for DCT coefficient quantization. In *Proceedings of SPIE conference on human vision, visual processing and digital display* (pp. 191–201). Bellingham: SPIE.
7. Shkolnik, M., Chmiel, B., Banner, R., Shomron, G., Nahshan, Y., Bronstein, A., et al. (2020). Robust quantization: one model to rule them all. In H. Larochelle, M. Ranzato, R. Hadsell, et al. (Eds.), *Proceedings of the 34th international conference on neural information processing systems* (pp. 1–10). Red Hook: Curran Associates.
8. Liu, J., Zhuang, B., Zhuang, Z., Guo, Y., Huang, J., Zhu, J., et al. (2022). Discrimination-aware network pruning for deep model compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8), 4035–4051.
9. Luo, J., & Wu, J. (2020). Autopruner: an end-to-end trainable filter pruning method for efficient deep model inference. *Pattern Recognition*, 107, 107461.
10. Zhang, X., He, Y., & Jian, S. (2017). Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 1398–1406). Piscataway: IEEE.
11. Srinivas, S., & Babu, R. V. (2015). Data-free parameter pruning for deep neural networks. In *Proceedings of the British machine vision conference* (pp. 1–12). Swansea: BMVA Press.
12. Gou, J., Yu, B., Maybank, S. J., & Tao, D. (2021). Knowledge distillation: a survey. *International Journal of Computer Vision*, 129(6), 1789–1819.
13. Mirzadeh, S. I., Farajtabar, M., Li, A., Levine, N., & Ghahemzadeh, H. (2020). Improved knowledge distillation via teacher assistant. In *Proceedings of the 34th AAAI conference on artificial intelligence* (pp. 5191–5198). Palo Alto: AAAI Press.
14. Zhang, L., Song, J., Gao, A., Chen, J., Bao, C., & Ma, K. (2019). Be your own teacher: improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 3712–3721). Piscataway: IEEE.

15. Allen-Zhu, Z., & Li, Y. (2023). Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. In *The 11th international conference on learning representations* (pp. 1–12). Retrieved November 2, 2023, from <https://openreview.net/pdf?id=Uuf2q9TfXGA>.
16. Qian, X., Fu, Y., Jiang, Y. G., Xiang, T., & Xue, X. (2017). Multi-scale deep learning architectures for person re-identification. In *Proceedings of the IEEE international conference on computer vision* (pp. 5409–5418). Piscataway: IEEE.
17. Neverova, N., Wolf, C., Taylor, G. W., & Taylor, F. N. (2014). Multi-scale deep learning for gesture detection. In L. Agapito, M. M. Bronstein, & C. Rother (Eds.), *Proceedings of the 13th European conference on computer vision workshops* (pp. 474–490). Cham: Springer.
18. Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 936–944). Piscataway: IEEE.
19. Gong, Y., Yu, X., Ding, Y., Peng, X., Zhao, J., & Han, Z. (2021). Effective fusion factor in FPN for tiny object detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 1159–1167). Piscataway: IEEE.
20. Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path aggregation network for instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8759–8768). Piscataway: IEEE.
21. Fan, J., Bocus, M. J., Hosking, B., Wu, R., Liu, Y., Vityazev, S., et al. (2021). Multi-scale feature fusion: learning better semantic segmentation for road pothole detection. In *Proceedings of the IEEE international conference on autonomous systems* (pp. 1–5). Piscataway: IEEE.
22. Huertas, A., & Medioni, G. G. (1986). Detection of intensity changes with subpixel accuracy using Laplacian-Gaussian masks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(5), 651–664.
23. Chen, Q., & Sang, L. (2018). Face-mask recognition for fraud prevention using Gaussian mixture model. *Journal of Visual Communication and Image Representation*, 55, 795–801.
24. Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, et al. (Eds.), *Proceedings of the 29th international conference on neural information processing systems* (pp. 91–99). Red Hook: Curran Associates.
25. He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980–2988). Piscataway: IEEE.
26. Cai, Z., & Vasconcelos, N. (2018). Cascade R-CNN: delving into high quality object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 6154–6162). Piscataway: IEEE.
27. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779–788). Piscataway: IEEE.
28. Redmon, J., & Farhadi, A. (2017). Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6517–6525). Piscataway: IEEE.
29. Redmon, J., & Farhadi, A. (2018). YoloV3: an incremental improvement. Preprint. [arXiv:1804.02767](https://arxiv.org/abs/1804.02767).
30. Bochkovskiy, A., Wang, C. Y., & Liao, H. (2020). YoloV4: optimal speed and accuracy of object detection. Preprint. [arXiv:2004.10934](https://arxiv.org/abs/2004.10934).
31. Wang, C. Y., Bochkovskiy, A., & Liao, H. (2023). YoloV7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7464–7475). Piscataway: IEEE.
32. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. E., Fu, Y. C., et al. (2016). SSD: single shot multibox detector. In B. Leibe, J. Matas, N. Sebe, et al. (Eds.), *Proceedings of the 14th European conference on computer vision* (pp. 21–37). Cham: Springer.
33. Jiang, D., Sun, B., Su, S., Zuo, Z., Wu, P., & Tan, X. (2020). FASSD: a feature fusion and spatial attention-based single shot detector for small object detection. *Electronics*, 9(9), 1536.
34. Rosas-Arias, L., Benitez-Garcia, G., Portillo-Portillo, J., Sanchez-Perez, G., & Yanai, K. (2021). Fast and accurate real-time semantic segmentation with dilated asymmetric convolutions. In *Proceedings of the 25th international conference on pattern recognition*, Piscataway: IEEE.
35. Tian, Z., Shen, C., Chen, H., & He, T. (2020). FCOS: fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, Piscataway: IEEE.
36. Wang, Y., Wang, C., Zhang, H., Dong, Y., & Wei, S. (2019). Automatic ship detection based on retinanet using multi-resolution Gaofen-3 imagery. *Remote Sensing*, 11(5), 531.
37. Ale, L., Ning, Z., & Li, L. (2018). Road damage detection using retinanet. In *Proceedings of the IEEE international conference on big data* (pp. 5197–5200). Piscataway: IEEE.
38. Sinha, D., & El-Sharkawy, M. (2019). Thin mobilenet: an enhanced mobilenet architecture. In *Proceedings of the IEEE 10th annual ubiquitous computing, electronics & mobile communication conference* (pp. 280–285). Piscataway: IEEE.
39. Biswas, A. (2019). An automatic traffic density estimation using single shot detection (SSD) and MobileNet-SSD. *Physics and Chemistry of the Earth*, 110, 176–184.
40. Bucila, C., Caruana, R., & Niculescu-Mizil, A. (2006). Model compression. In *ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 535–541). New York: ACM.
41. Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. Preprint. [arXiv:1503.02531](https://arxiv.org/abs/1503.02531).
42. Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., & Bengio, Y. (2015). Fitnets: hints for thin deep nets. [Poster presentation]. *Proceedings of the 3rd international conference on learning representations*, San Diego, USA.
43. Chen, G., Choi, W., Yu, X., Han, T., & Chandraker, M. (2017). Learning efficient object detection models with knowledge distillation. In I. Guyon, U. Von Luxburg, S. Bengio, et al. (Eds.), *Proceedings of the 31st international conference on neural information processing systems* (pp. 742–751). Red Hook: Curran Associates.
44. Wang, T., Yuan, L., Zhang, X., & Feng, J. (2019). Distilling object detectors with fine-grained feature imitation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4933–4942). Piscataway: IEEE.
45. Guo, J., Han, K., Wang, Y., Wu, H., Chen, X., Xu, C., et al. (2021). Distilling object detectors via decoupled features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2154–2164). Piscataway: IEEE.
46. Hou, Y., Ma, Z., Liu, C., & Loy, C. C. (2019). Learning lightweight lane detection CNNs by self attention distillation. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 1013–1021). Piscataway: IEEE.
47. Shen, Y., Xu, L., Yang, Y., Li, Y., & Guo, Y. (2022). Self-distillation from the last mini-batch for consistency regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11943–11952). Piscataway: IEEE.
48. Zhang, P., Kang, Z., Yang, T., Zhang, X., Zheng, N., & Sun, J. (2022). LGD: label-guided self-distillation for object detection. In *Proceedings of the 36th AAAI conference on artificial intelligence* (pp. 3309–3317). Palo Alto: AAAI Press.
49. Ji, M., Shin, S., Hwang, S., Park, G., & Moon, I. C. (2021). Refine myself by teaching myself: feature refinement via self-knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10664–10673). Piscataway: IEEE.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.