Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

ThumbDet: One thumbnail image is enough for object detection

Yongqiang Zhang[a,1,*], Yin Zhang[a,1], Rui Tian[a], Zian Zhang[a], Yancheng Bai[c], Wangmeng Zuo[b], Mingli Ding[a]

[a] School of Instrumentation Science and Engineering, Harbin Institute of Technology, Harbin 15001, China
[b] School of Computer Science and Technology, Harbin Institute of Technology, Harbin 15001, China
[c] Institute of Software, Chinese Academy of Sciences, Beijing 100053, China

ARTICLE INFO

Article history:
Received 19 July 2022
Revised 17 December 2022
Accepted 12 February 2023
Available online 14 February 2023

Keywords:
Object detection
Down-sampling network
Knowledge distillation

ABSTRACT

Computer vision fields have witnessed great success thanks to deep convolutional neural networks (CNNs). However, state-of-the-art methods often benefit from large models and datasets, which introduce heavy parameters and computational requirements. Deploying such large models in real-world applications is very difficult because of the limited computing resources. Although many researchers focus on designing efficient block structures to compress model parameters, they ignore that the role of large-scale input images is also an important factor for algorithm efficiency. Reducing input resolution is a useful method to boost runtime efficiency, however, traditional interpolation methods assume a fixed degradation criterion that greatly hurts performance. To solve the above problems, in this paper, we propose a novel framework named ThumbDet for reducing model computation while maintaining detection accuracy. In our framework, we first design an image down-sampling module to learn a small-scale image that looks realistic and contains discriminative properties. Furthermore, we propose a distillation-boost supervision strategy to maintain the detection performance of small-scaled images as the original-size inputs. Extensive experiments conducted on a standard object detection dataset MS COCO demonstrate the effectiveness of the proposed method when using very low-resolution images (*i.e.* 4× down-sampling) as inputs. In particular, ThumbDet achieves satisfactory detection performance (*i.e.* 32.3% in mAP) while drastically reducing computation and memory requirements (*i.e.* speed up of 1.26×), outperforming the traditional interpolation methods (*e.g.* bicubic) by +3.2% absolutely in terms of mAP.

© 2023 Elsevier Ltd. All rights reserved.

1. Introduction

In recent decades, deep convolutional neural networks (CNNs) have led to a series of breakthroughs in high level vision tasks, such as object detection [1–3], semantic segmentation [4,5] and human pose estimation [6,7], *etc.* . With the rapid development of hardware devices (*e.g.* GPUs), excellent performance is achieved by training deep/wide networks with large-scale images, which usually introduces high computation and memory requirements. Thus, such a deep model is difficult to apply on the computing limited hardware devices (*e.g.* mobile phones) in real-world applications

To run deep networks in real-time, some methods are proposed to accelerate and compress deep models [8,9]. Most of them focus on compressing deep models into a shallow (with fewer layers) network or designing efficient blocks to reduce network parameters, but they neglect a fact that the large-scale input image is also an important factor in memory consumption and overall complexity. In the task of classification, some works are proposed to recognize objects on thumbnail images ($\geq 4\times$ down-sampled images), and a comparable Top-1 accuracy is achieved compared with the original-size images. As reported in SimMIM [10] (in the 3rd row of Table 1), when the scale of the input image is reduced by 16 times, the Top-1 accuracy is almost unchanged (drops from 82.8% to 82.7%). However, we found that reducing input image resolution achieves poor performance in the object detection task. For example, in the recent transformer-based object detection method Deformable-DETR [11], its mean average precision (mAP) drops from 43.8% to 35.3% when the input image is down-sampled by a factor of 2, and even detection failures occur when the down-sampled factor is ≥ 4 (in the 6th row in Table 1). Actually, the reason for this phenomenon can be deduced from the formula of image degradation:

$$I_{lr} = (I_{org} * k) \downarrow_s + n \tag{1}$$

where k and n indicate the degradation kernel and noise respectively. The goal of Eq (1) is to obtain low-resolution images I_{lr} from

[*] Corresponding author.
E-mail address: zhangyongqiang@hit.edu.cn (Y. Zhang).
[1] Equal contribution.

https://doi.org/10.1016/j.patcog.2023.109424
0031-3203/© 2023 Elsevier Ltd. All rights reserved.

Table 1
Results of classification and object detection on different resolution images.

Ratio of inputs (Image size)	1/1 (192*192)	1/2 (96*96)	1/4 (48*48)	1/8 (24*24)	1/16 (12*12)	1/32 (6*6)
Top-1 acc(%)	82.8	82.8	82.7	82.8	82.7	82.3
Ratio of inputs (Image size)	1/1 (800*max=1333)	1/2 (400*max=667)	1/4 (200*max=333)	1/8 (100*max=167)	1/16 (50*max=83)	1/32 (25*max=42)
mAP(%)	43.8	35.3	0	0	0	0

original large-scale images I_{org} through down-sampling operation $\downarrow_s$. Most existing methods use an ideal down-sampling process that assumes the kernel is known and fixed, so the distribution between low-resolution images and original large-scale images is different. Therefore, it is not surprising that such low detection results are obtained on low-resolution images I_{lr} obtained by down-sampling methods. In addition, since these down-sampling images have some artifacts, they are often inadequate for machine perception [12].

To overcome the above mentioned problems, in this paper, we propose a novel object detection framework named ThumbDet from a new perspective to reduce the computation of deep networks. Our main motivation is to explore i) how to obtain a thumbnail image that removes the redundant information in the original-size image but contains key information for object detection, while making the distribution of low-resolution images is similar to the original large images; ii) how to maintain the performance of a low-resolution image as its high-resolution counterparts, making it possible to deploy deep networks on mobile devices in practical applications.

Specifically, for motivation i), inspired by some super-resolution algorithms[13–15], we design an online image down-sampling module to obtain a thumbnail image. Instead of generating low-resolution images with an ideal degradation criterion, we utilize CNNs to generate thumbnail images from original images in a supervised manner. The online image down-sampling module is reliably trained by exploring supervised image down-scaling, knowledge distillation, and object detection simultaneously, which ensures the small-sized images look realistic and contain discriminative properties, and thus making it possible to replace their original-size counterparts for reliable object detection. Finally, the generated thumbnail images replace the original-size images and are fed into the subsequent network for object detection.

For motivation ii), we propose a distillation-boost supervision strategy [16] to maintain the detection performance of low-resolution images as high-resolution counterparts. This strategy not only enables the network with low-resolution inputs to obtain some prior knowledge and similar features with the network of high-resolution images, but it can also accelerate the network with low-resolution images as inputs (*i.e.* inference network). In detail, a teacher network is trained with the input of original-size images, which is frozen and used to guide the training of the student network with small-size inputs under the supervision of logit loss and feature map distillation loss. Moreover, different from other DETR-like detectors in which the output of the decoder is sent to feed forward networks (FFNs) for direct prediction, we propose a Query Filter algorithm to remove some redundant queries before logit distillation for reducing computation and further improving detection performance.

Note that the training process of ThumbDet is separated into two stages: 1) image down-sampling stage and 2) prior information transfer stage. Moreover, it should be emphasized that the image down-sampling module is not discarded during the second stage. It is trained together with other losses of Deformable-DETR [11], which includes the classification loss and bounding box loss for the whole process of ThumbDet. These losses guide the down-sampling module to generate small-sized images that look realistic and contain discriminative properties, making it possible to replace their original-size counterparts for reliable object detection. The details of the training strategy are described in Section 3.5.

With this fascinating structure, the model is more computationally efficient in the inference stage compared to using large-scale images as inputs, because the generated thumbnail images can remove some redundant information in large-scale images. In addition, the rich knowledge of the large-scale image trained teacher is transferred to the low-resolution student, resulting in satisfactory performance on low-resolution images (the red points shown in Fig. 1).

For object detection, ThumbDet is a generic and implementation-friendly method of model compression and acceleration, that can not only address the limitations of network compression in simplicity and universality, but also achieve impressive performance on thumbnail images. Moreover, the well-trained image down-sampling module obtained by ThumbDet can be used in other high-level tasks (*e.g.* segmentation, human pose estimation, *etc.*) in practical applications. To sum up, this paper makes the following main contributions:

(1) A novel object detection framework named ThumbDet is proposed from a new perspective to reduce the computation of deep networks. Unlike traditional methods that compress the network by reducing its parameters or parameter-incurred computations, we first learn a small-scale image in a supervised manner, and then use the generated thumbnail image as inputs without changing the network structure to significantly reduce the computation and memory consumption.

(2) We design an image down-sampling module that takes full advantage of the powerful feature extraction capabilities of CNNs to generate a thumbnail image from the original image. The thumbnail image is obtained under the supervision of image down-scaling, knowledge distillation, and object detection losses, leading the small-size image looks realistic and contains discriminative properties, which make it possible to replace the original-size counterparts for reliable object detection. Experiments demonstrate that our down-sampling module is better than bicubic interpolation in object detection on small-size images.

(3) A distillation-boost supervision strategy is proposed to maintain the detection performance of thumbnail images as the original-size inputs. A teacher network is trained with the input of original-size images, which is frozen and used to guide the training of the student network with small-sized inputs under the supervision of the logit distillation loss and feature map distillation loss. Moreover, we propose a Query Filter algorithm to remove redundant queries before logit distillation for reducing computation and further improving the detection performance.

(4) ThumbDet is a generic and implementation-friendly framework that can be integrated into any object detection architecture. Extensive experiments conducted on MS COCO dataset show that our method can drastically reduce computation and memory requirements through input resolution reduction (*i.e.* 4× down-sampling). Moreover, ThumbDet achieves satisfactory detection performance (*i.e.* 32.3% in mAP) while drastically reducing computation and memory requirements (*i.e.* speed up 1.26×), outperforming the traditional interpolation methods (*e.g.* bicubic) by +3.2% absolutely in terms of mAP.

The rest of the paper is organized as follows. We review the related recent literature in Section 2. In Section 3, the detailed architecture of our network for low-resolution object detection (ThumbDet) is presented, and the loss function of the proposed down-sampling module and distillation-boost supervision strategy are described in detail. In Section 4, we first show the main results of our method with 2× and 4× down-sampling ratios. Then, we conduct ablation study analysis to validate the effectiveness of each proposed component in our pipeline, and some experiments that compared our method with previous state-of-the-arts are shown on a widely used object detection dataset (*i.e.* MS COCO). Finally, we show some visualization results on MS COCO val dataset. This is followed by the conclusions and future work in Section 5.

2. Related work

Generic Object Detection. Object detection is a fundamental task in computer vision. With the development of CNNs, object detection has entered the era of deep learning, and impressive performance has been achieved in recent years. Generally, existing deep object detection methods can be divided into three categories: two-stage [1,17–19], one-stage [20–23] and transformer-based detectors [2,11,24,25]. Two-stage object detection methods first generate thousands of candidate proposals and then perform classification and localization based on these proposals, where R-CNN [17] is the pioneering work that uses the deep neural network for object detection. R-CNN generates about 2000 proposals by using the selective search algorithm, and then uses CNNs to learn the representation of each proposal for classification and localization. Then, fast-RCNN [18] designs ROI pooling to improve training and testing speed while increasing the detection performance. Faster-RCNN [1] replaces selective search with a Region Proposal Network (RPN) to generate proposals and implements the object detection task in an end-to-end manner. In contrast, one-stage object detection methods directly predict object categories and regress object bounding boxes based on the inputted image regions or anchors, where the famous works are YOLO [20], SSD [21], RetinaNet [22], FCOS [23], *etc.* . For example, RetinaNet [22] proposes a focal loss to solve the extreme imbalance of positive and negative samples for one-stage object detection. FCOS [23] gets out of the limits of anchors and directly predicts the distance of the point from the left side, upper side, right side and lower side of the target at each position of feature maps. DSLA [3] introduces a dynamic smooth label assignment strategy based on FCOS to improve the quality of localization.

Recently, a tremendous successful model in the field of natural language processing [26] has been introduced to computer vision [27], and a transformer-based detector named DETR [2] has stepped into the spotlight. DETR is an anchor-free detector that removes hand-designed components (*e.g.* non-maximum suppression procedure or anchor generation) and directly predicts a set of objects by learnable queries with Hungarian bipartite matching. Because DETR is a Transformer encoder-decoder architecture, it still suffers from the slow convergence problem. To this end, some DETR-like methods have attempted to speed up the training process of DETR. For instance, Deformable-DETR [11] designs deformable attention modules with deformable convolution to replace the original attention modules, and these deformable attention modules only attend to a small set of key sampling points around a reference, which mitigates the slow convergence problem of DETR. DAB-DETR [24] uses anchor boxes, *i.e.* 4D vectors (*x*, *y*, *w*, *h*), as queries and updates them layer by layer to improve the cross-attention computation. DN-DETR [25] proposes a novel denoising training method by adding noised ground-truth bounding box to Transformer decoder to make bipartite matching easier.

Although great achievements have been made in object detection, the impressive performance of existing deep models is achieved based on large-scale images. Moreover, such deep models have the problem of huge computation, making it is difficult to apply them in computing limited hardware devices in real-world applications. To overcome these problems, in this paper, we focus on object detection on low-resolution images, which is a new perspective to reduce the computation of deep networks while maintaining satisfactory detection performance.

Knowledge Distillation. Knowledge distillation (KD) [16] is a model compression method that aims to guide a small student network to learn knowledge from a large teacher network. In order to make the student matching the softmax output of the teacher, temperature $\mathcal{T}$ is used to soften the output logit of the teacher network, and then KL-Divergence is used to minimize the logit between the teacher and student networks. FitNets [28] is the first to mimic the intermediate layers feature of the teacher model as hints to guide the student model. LightweightNet [29] designs an efficient block to distill the knowledge of convolutional layers on the basis of fully analyzing the network architecture.

Recently, knowledge distillation has been successfully applied to object detection model compression and incremental learning tasks, *etc.* . For the task of object detection, existing knowledge distillation methods generally fall into three categories: 1) logit distillation, 2) feature map distillation and 3) bounding-box regression distillation. For example, DKD [30] uses mathematical inference to demonstrate that the traditional logit KD loss is highly coupled with target and non-target class knowledge distillation, and then decouples it into two parts to provide a novel viewpoint of logit distillation. FGD [31] proposes focal and global distillation to force the student network to select crucial parts of features, and rebuilds the relation between foreground and background to improve the ability of feature distillation in object detection. [32] introduces a new correlation distillation loss to select specific features from three channels for optimizing the object detector regularly. LD [33] switches the bounding box representation to a probability distribution and combines it with KL-Divergence for object detection distillation.

In summary, these existing knowledge distillation methods use large-scale images as inputs to distill the knowledge from a large teacher network to a small student network, *i.e.* the architectures of teacher and student networks are different. However, different from the aforementioned methods, in this paper, we just use the knowledge distillation strategy to maintain the object detection performance of thumbnail images as high as the large-scale images. More specifically, we use the same network structure in teacher and student networks, and the knowledge from the teacher network using large-scale images as inputs is learned to guide the training of the student network with generated thumbnail images.

Object Detection in degradation Images. For object detection, directly training the detection model on low-resolution images would suffer a tremendous performance drop when compared with the performance of large-scale images. To tackle this problem, some works have confirmed that the pre-processing module is effective in improving the accuracy on degraded images [12,34–36]. For example, Wang *et. al* [34] attempt to solve the low-resolution recognition problem by using a super-resolution pre-training method. KGSNet [35] introduces a super-resolution network to generate clear images that contain rich visual details for small-scale and heavily occluded pedestrian detection. RestoreDet [12] captures the dynamics feature by learning the degradation equivalent representation to detect objects on degraded images. Similarly, [37] designs a super-resolution detection network that separates pedestrians from the blurred background to achieve higher accuracy on pedestrian detection.

Furthermore, multi-scale training strategy is also a method to improve detection performance on low-resolution inputs. SAN [38] uses meta learners to generate the convolutional weights of networks for various input scales. RS-Net [39] proposes a paral-

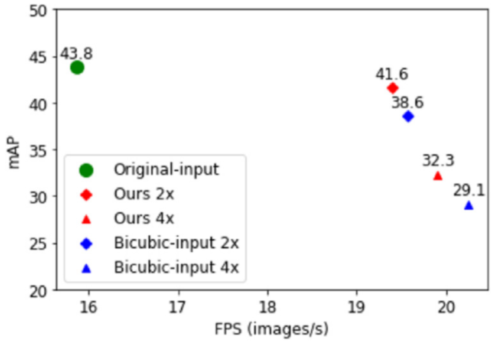

Fig. 1. The results of object detection with different inputs at 2× and 4× down-sampling rates. We show mAP and FPS results with images of original large-scale, bicubic down-sampled and our down-sampled as inputs, where the green point denotes the results of the original large-scale inputs (*i.e.* teacher model), the blue point is the results of bicubic down-sampled images, and the red color is the results of our proposed ThumbDet. Diamonds and triangles denote the 2× and 4× down-sampling rate, respectively. From Fig. 1, we can see that our model is efficient while maintaining a similar accuracy compared with original large-scale images. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

lel training framework and a multi-resolution ensemble distillation to maintain accuracy at different resolutions. Nevertheless, both of these methods are for the task of image classification.

The above methods try to restore high-quality and high-resolution images from low-resolution images (*e.g.* low-resolution, occluded, noising images) by specific algorithms or introduce efficient block structures, and then perform object detection on the restored images. In contrast, in this paper, we propose a novel approach to directly detect objects on thumbnail images, which can reduce the model computation while maintaining the detection accuracy.

3. Proposed method

In this section, we describe the details of the proposed thumbnail image detection network ThumbDet. First, we introduce the pipeline and the overview of the proposed novel framework, as shown in Fig. 2, and then we give a brief review of Deformable-DETR. In addition, an image down-sampling module is proposed for learning a thumbnail image that looks realistic and contains discriminative properties. Finally, a distillation-boost supervision strategy is used to maintain the detection performance of thumbnail images as the original-size inputs.

3.1. Overview of ThumbDet

To pursue higher detection accuracy, current works usually expand the depth of the network or use large-scale images to train a deep model, which is difficult applied to real-world applications. Moreover, reducing the inputted image resolution greatly damages the detection performance, as shown in Table 1. Inspired by thumbnail image classification methods [40], we propose a novel framework named ThumbDet that aims to solve the above problems, *i.e.* large memory costs and degraded detection performance when using low-resolution images as inputs.

In this paper, we choose Deformable-DETR [11] as our baseline, and a novel framework named ThumbDet is proposed to detect objects on the learned thumbnail images, which can greatly reduce model computation while maintaining the detection performance. The architecture of our method is shown in Fig. 2, and there are two components in the proposed ThumbDet. The first component is an image down-sampling module, which generates thumbnail images by CNNs, thus containing discriminative properties corresponding to the original large-scale images. The second component is a distillation-boost supervision strategy, in which the teacher network is well trained from original large-scale images and the student network architecture is exactly the same as the teacher except a thumbnail image is used as the input. Moreover, we propose a Query Filter algorithm to remove some redundant queries before logit distillation for reducing computation and further improving the detection performance. The overall loss functions include $\mathcal{L}_{mm}$ loss that supervises thumbnail images to preserve discriminative properties and look realistic with the original ones, $\mathcal{L}_{fm}$ and $\mathcal{L}_{cls}$ KD loss force the student network to mimic the feature map and classification logit of the teacher, and $\mathcal{L}_{det}$ loss denotes the original detection loss function.

3.2. Review of deformable-DETR

3.2.1. Deformable convolution

Traditional 2D convolution is limited by the fixed geometric structures of CNN modules, which makes it difficult to adaptively adjust the shape of objects. Deformable convolution network [41] adds learnable offsets to each point of the receptive field in the standard convolution. After offsets are learned via additional convolutional layers, the receptive field is no longer a fixed square, which can match the actual shape of objects, thus solving the problem of weak adaptability to deformation.

3.2.2. Multi-scale Deformable Attention Module

The main contribution of Deformable-DETR is that multi-scale deformable attention ($\mathcal{MDA}$) modules are used to replace multi-head attention ($\mathcal{MA}$) modules for accelerating convergence of models. Specifically, the multi-head attention module in Transformer [26] is calculated by :

$$\mathcal{MA} = \sum_{m=1}^{M} W_m \left[\sum_{k \in \Omega_k} A_{mqk} \cdot W'_m x_k \right] \tag{2}$$

where x is the input feature and W'_m is the learnable weight, $A_{mqk} \cdot W'_m x_k$ means self-attention. m and W_m denote the m_{th} head in multi-head attention and the transformation matrix. And the multi-scale deformable attention module in Deformable-DETR is calculated by :

$$\mathcal{MDA} = \sum_{m=1}^{M} W_m \left[\sum_{k=1}^{K} A_{mqk} \cdot W'_m x(p_q + \Delta p_{mqk}) \right] \tag{3}$$

where p_q denotes a random point in 2D space and Δp_{mqk} denotes the sampling offset of the k_{th} sampling point in the m_{th} attention head. The difference between these two formulas lies in two aspects: 1) in the $\mathcal{MA}$ module, $k \in \Omega_k$ means that all keys need to be computed, but in the $\mathcal{MDA}$ module $k \in [1, K](K \ll HW)$ means just need to consider a small number of keys; 2) in the $\mathcal{MA}$ module, input feature x is fixed, but in the $\mathcal{MDA}$ module, p_q and sampling offset Δp_{mqk} are added to obtain a new point of the feature map inspired by deformable convolution [41]. The above mentioned two changes could greatly mitigate the issue of slow convergence.

3.2.3. Deformable-DETR

Deformable-DETR [11] is an efficient and fast-converging end-to-end object detector. The whole detection process of Deformable-DETR is similar to DETR [2]: A CNN backbone extracts feature maps of an input image, and then the feature maps with positional encoding are sent to the Transformer encoder-decoder architecture for transforming them to be the features of a set of object queries. Finally, the prediction is computed by a feed forward networks (FFNs). Deformable-DETR replaces all multi-head

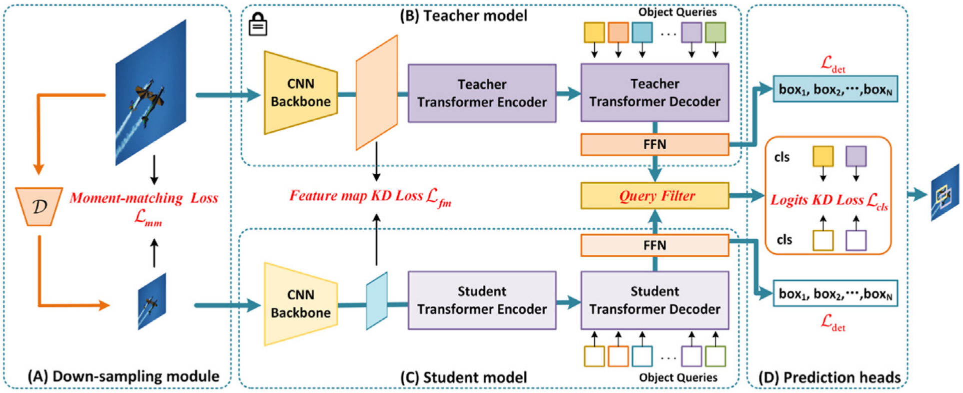

Fig. 2. Architecture of our ThumbDet. We propose a novel framework for conducting object detection on thumbnail images, where a down-sampling module and a distillation-boost supervision strategy are used to simultaneously accelerate the inference procedure and maintain detection performance. The inputs of the proposed ThumbDet are HR images, where HR denotes original large-scale images, and the detection procedure (*i.e.* inference process) is conducted on thumbnail low-resolution images (*i.e.* LR images) generated by a down-sampling module. To further improve the detection performance on LR images, we propose a distillation-boost supervision strategy in the ThumbDet framework. **(A)** Down-sampling module $\mathcal{D}$ is designed to learn a thumbnail image (4× down-sampling) from an original large-scale image by optimizing a Moment-matching loss $\mathcal{L}_{mm}$, which ensures the learned small-sized image looks realistic and contains discriminative properties. **(B)** The teacher model is a well-trained Deformable-DETR network with original HR images, and the weights are frozen during the whole training process. In the distillation-boost supervision strategy, large-scale images and thumbnail images are fed into ResNet50 to extract feature maps respectively, and we use Feature map KD loss $\mathcal{L}_{fm}$ to minimize the distance between teacher and student feature maps. Moreover, different from other DETR-like detectors, in which the output of the decoder is sent to feed forward networks (FFNs) for direct prediction, we propose a Query Filter algorithm to remove redundant queries before logit distillation for reducing computation and improving performance. **(C)** The student model is the final inference model, which is the same as the teacher model except the input is thumbnail images. **(D)** Prediction heads contain a series of boxes and ground truth predict logit. We use logit distillation $\mathcal{L}_{cls}$ to transfer knowledge from teacher to student via soft labels. ThumbDet is also optimized by the object detection loss $\mathcal{L}_{det}$ which can refer to the paper [11] for more details. All components in ThumbDet are trained in an end-to-end manner.

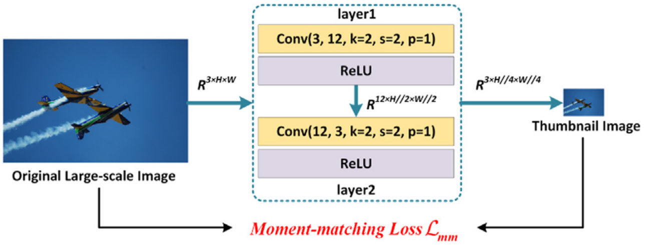
layer1
Conv(3, 12, k=2, s=2, p=1)
ReLU
$R^{3\times H\times W}$
$R^{12\times H//2\times W//2}$
$R^{3\times H//4\times W//4}$
Conv(12, 3, k=2, s=2, p=1)
ReLU
layer2
Original Large-scale Image
Thumbnail Image
Moment-matching Loss $\mathcal{L}_{mm}$

Fig. 3. The structure of the down-sampling module. The first layer is a 3 × 3 convolutional operation with stride 2. It maps the original large-scale images to 12-channel features for learning some hidden information. The second layer is similar to the first layer except channels are set to 3 for reconstructing an RGB image. Note that each convolutional layer is followed by a rectified linear unit (ReLU). The original large-scale image is eventually turned into a 4× down-sampling thumbnail image under the supervision of moment-matching loss $\mathcal{L}_{mm}$.

attention modules in the Transformer encoder with multi-scale deformable attention modules to accelerate model convergence. Moreover, Deformable-DETR solves the problems of slow convergence and poor performance on small objects in DETR, to this end, we choose Deformable-DETR as our baseline detector in this paper.

3.3. Image Down-sampling module

Given a large-scale image, in the image down-sampling module, our goal is to generate a thumbnail image (*e.g.* 4× down-sampling) whose distribution is as close as possible to the distribution of the original large-scale image. In principle, the architecture of a single convolutional layer is sufficient for down-sampling described by Eq (1). However, we experimentally found that such an architecture does not get satisfactory performance (see Section 4.3.2 for analysis). The reason may be that the thumbnail image itself has little information, and the feature extraction capability of a single convolutional layer is too weak.

To obtain a better down-sampling result while not introducing too much computation, as shown in Fig. 3, we design a down-sampling module that only contains two convolutional layers, and each layer is followed by a rectified linear unit (ReLU). In addition, the pooling operation is discarded to preserve more discriminative properties in our down-sampling module. Specifically, the first layer is a 3 × 3 convolutional operation with stride 2, and it maps the large-scale image to 12-channel features for learning some hidden information. The second layer is similar to the first layer except channels are set to 3 for reconstructing an RGB thumbnail image. The process of generating a thumbnail image can be formulated as follows:

$$y = \mathcal{D}(x; \delta) \tag{4}$$

where $\mathcal{D}$ denotes the down-sampling module, x and y denote original large-scale images and thumbnail images respectively, and δ is the parameters of the image down-sampling module.

As stated above, our motivation of designing the down-sampling module is not only just for reducing the size of images, but also we hope that thumbnail images should follow the same distribution as the original large-scale images. Furthermore, the generated thumbnail image should be machine vision oriented and human visually pleasant. That is to say, the distribution of pixel values in thumbnail images follows the same distribution with the original images, and the information in each color channel keep unchanged or aligned. Toward this end, a simple way to enforce this purpose by minimizing the pixel-wise MSE loss, but it makes thumbnail images look unpleasant and have more artifacts. Inspired by [40], we adopt the moment-matching loss to optimize our down-sampling module, and the moment-matching loss $\mathcal{L}_{mm}$ can be defined as follows:

$$\mathcal{L}_{mm} = \frac{1}{3}\sum_{i=1}^{3} \left[\|(\mu(x_i) - \mu(y_i)\|_2^2 + \lambda\|(\sigma(x_i) - \sigma(y_i)\|_2^2 \right] \tag{5}$$

where x and y denote original large-scale and thumbnail images, λ is a hyper-parameter to balance these two components, and $\mu(\cdot)$ and $\sigma(\cdot)$ compute the first and the second moment in each color channel. This moment-matching loss can supervise the mean and variance (*i.e.* roughly mimicking the distribution) of thumbnail images to be as close as possible to the mean and variance of the original images.

We need to emphasize that this down-sampling module is not trained independently with merely the moment-matching loss

$\mathcal{L}_{mm}$, but is instead plugged into the whole framework of ThumbDet. That is to say, the down-sampling module is trained together with other losses of Deformable-DETR, which includes classification loss and bounding box loss. These losses guide the down-sampling module to generate small-sized images that look realistic and contain discriminative properties, making it possible to replace their original-size counterparts for reliable object detection. Note that the down-sampling module trained in a supervised manner in this paper can not only be used for generating small-sized images for object detection, but also can be used in other high-level related tasks (*e.g.* key-point estimation, segmentation, *etc.*) as well.

3.4. Distillation-boost Supervision Strategy

After using the down-sampling module, we can obtain a thumbnail image from a large-scale image, and the thumbnail image follows the style of the original image and looks realistic under the supervision of $\mathcal{L}_{mm}$ loss. Although the thumbnail image can replace a small-sized image by interpolation methods and feed it into the encoder-decoder for subsequent object detection tasks, 4× down-sampling operation still loses much high-frequency detail information for accurate detection. To further maintain the detection performance of thumbnail images as the original-size inputs, we propose a distillation-boost supervision strategy to transfer key knowledge information from original images to thumbnail images via soft label supervision and feature map alignment. Specifically, we use logit distillation $\mathcal{L}_{cls}$ and feature map distillation $\mathcal{L}_{fm}$ on the classification head and the output features of the ResNet50 backbone for knowledge distillation in our proposed strategy.

a) Logit distillation. The concept of logit distillation was first proposed by Hinton *et. al* [16]. Following [16], the hyper-parameter temperature $\mathcal{T}$ is introduced to soften the labels in our distillation-boost supervision strategy. To shorten the distance between a teacher prediction p^t and a student prediction p^s, KL-Divergence is usually used and it can be formulated as:

$$\mathrm{KL}(p^t \| p^s) = \sum_{i=1}^{C} \quad p_i^t \log(\frac{p_i^t}{p_i^s}) \tag{6}$$

where C denotes the classes in the dataset. In the logit distillation loss $\mathcal{L}_{cls}$, the KL-Divergence between the classification outputs of the teacher and student networks can be calculated by:

$$\mathcal{L}_{cls} = \mathrm{KL}(\log\mathcal{S}(\frac{p^s}{\mathcal{T}}), \mathcal{S}(\frac{p^t}{\mathcal{T}})) \tag{7}$$

where $\mathcal{S}(\cdot)$ denotes a softmax function, p^s and p^t denote the classification probability of the student and teacher model, and $\mathcal{T}$ denotes the hyper-parameter to soften the classification probability.

However, DETR-like methods predict a set of objects by learnable queries directly, and DETR-like logit distillation is different from CNN-based methods. Specifically, the dimension of the CNN-based logit is $p \in R^{B\times C}$, while the dimension of the DETR-like logit is $p^{'} \in R^{B\times N\times C}$, where B, N, C denote the batch size, the number of queries and the number of classes respectively. Usually, the number of ground-truth objects $M \ll N$, which means most of queries are matched as the background by Hungarian bipartite matching. If these useless queries participate in the calculation in the logit distillation, it will consume considerable computation. To this end, we propose a Query Filter algorithm before logit distillation to eliminate background and no-matching classes for reducing computation and further improving performance in our method.

As shown in Fig. 4, N learnable queries (*e.g.* N is set to 300 in Deformable-DETR) are transformed into an output embedding by the teacher and student Transformer decoders respectively, and then they are passed through a feed forward network to compute the final prediction. As stated above, using all queries for distillation will increase the computation and disturb the logit distillation. In order to only distill the reliable logit of objects, we propose a Query Filter algorithm (illustrated in the yellow area of Fig. 4) to remove redundant queries, and the detailed procedure is summarized in Algorithm 1. As shown in Algorithm 1, in Step 1, for all queries in teacher prediction p^t, we use a descending sort function to rank them in descending order according to their classification probability. Then, the classification probability greater than a preset threshold (*i.e.* 0.5) is set to True to obtain a binary *keep* matrix. In Step 2, we use this binary *keep* matrix in teacher and student predictions to remove the redundant queries. Finally, the remaining p^t_{cls} and p^s_{cls} are reserved as the final queries for calculating the logit distillation loss.

Algorithm 1 Query Filter Algorithm. Selecting the prediction outputs when its class probability is greater than *threshold*. p^t and p^s denote teacher and student prediction respectively. *threshold* is set to 0.5, and Descending_Sort is a descending sort function.

```
Require: $p^t \in R^{B\times N\times C}$, $p^s \in R^{B\times N\times C}$, $threshold$
  Step 1: obtain a binary keep matrix by threshold
  $p^t_{sort}$ = Descending_Sort($p^t$), $p^t_{sort} \in R^{B\times N\times C}$
  $keep = p^t_{sort}[:, :, 0] > thresh$, $keep \in R^{B\times N}$ type is bool
  Step 2: remove redundant queries
  $p^t_{cls} = p^t \cap keep$
  $p^s_{cls} = p^s \cap keep$
Ensure: $p^t_{cls}$, $p^s_{cls}$
```

b) Feature map distillation. Distilling feature maps from intermediate layers has been demonstrated to be effective in other tasks [28,42], in this paper, we also use feature map distillation to further improve the detection performance on thumbnail images. As shown in Fig. 2, we have the same backbone architecture (*e.g.* ResNet50) in both teacher and student models, but the input of the student model is the thumbnail image, which is 4× smaller than the large-scale image in the teacher model. Thus, the dimension of feature maps between teacher and student models is different. To calculate the feature map distillation loss, generally an operation f (*e.g.* de-convolution, interpolation) is used to reshape the feature map of the student model to the same size as the feature of the teacher model. In this paper, we choose the linear interpolation method to align the dimensions of teacher and student feature maps.

Specifically, we employ ResNet50 as our backbone, and only *conv*3_*x*, *conv*4_*x*, *conv*5_*x* are used for feature map distillation and other layers are frozen. Here, the feature map distillation $\mathcal{L}_{fm}$ can be calculated by:

$$\mathcal{L}_{fm} = \frac{1}{LCHW}\sum_{l=1}^{L}\sum_{k=1}^{C}\sum_{i=1}^{H}\sum_{j=1}^{W}(F^T_{l,k,i,j} - f(F^S_{l,k,i,j}))^2 \tag{8}$$

where F^T and F^S denote the output feature maps from ResNet50 in the teacher and student models respectively. L denotes convolutional layers of ResNet50, C, H, W represent the channel, height and width of the feature map, and f denotes the feature alignment operation.

Finally, we optimize the total loss of the distillation-boost supervision strategy including logit distillation loss $\mathcal{L}_{cls}$ and feature map distillation loss $\mathcal{L}_{fm}$, and the overall distillation loss $\mathcal{L}_{\mathcal{KD}}$ is defined as:

$$\mathcal{L}_{KD} = \alpha\mathcal{L}_{cls} + \beta\mathcal{L}_{fm} \tag{9}$$

where α and β are hyper-parameters to balance the different loss terms.

To sum up, we train ThumbDet with the total loss as follows:

$$\mathcal{L}_{total} = \gamma\mathcal{L}_{mm} + \mathcal{L}_{KD} + \mathcal{L}_{det} \tag{10}$$

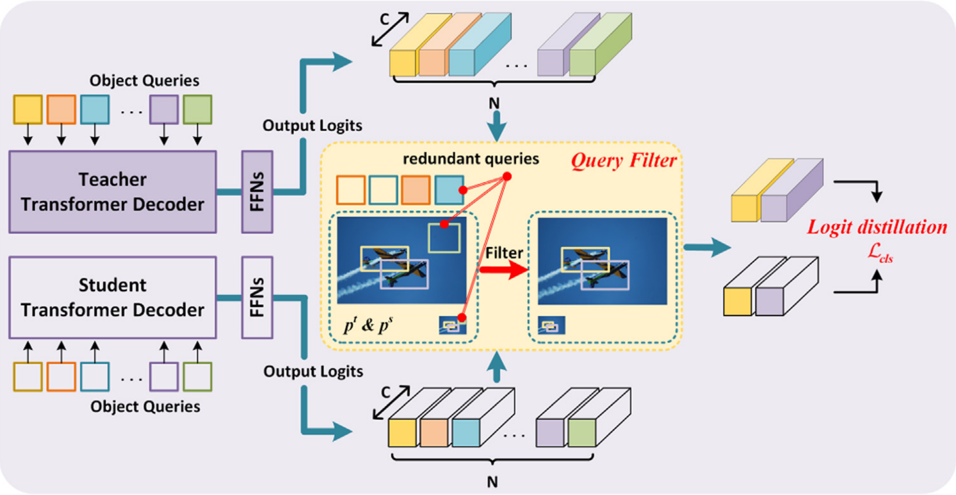

Fig. 4. The structure of the logit distillation. The decoders of teacher and student networks are the standard Transformer structure, and FFNs contain three linear layers with the ReLU activation function. The output logit of FFNs is $p \in R^{B\times N\times C}$, *e.g.* training Deformable-DETR detector on MS COCO dataset with batch size 2, $p \in R^{2\times 300\times 91}$. p^s and p^t denote the classification probability of the student and teacher model respectively. The yellow box in this figure is a Query Filter algorithm to remove some redundant queries before logit distillation. Finally, only reliable object queries are distilled by the logit distillation loss $\mathcal{L}_{cls}$. It should be noted that for clarity, the bounding box prediction is omitted after FFNs. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

where γ is the weight of the moment-matching loss and $\mathcal{L}_{det}$ denotes the original object detection loss function in Deformable-DETR.

3.5. Training strategy

For simple implementation and training stability, we divide the whole training process of ThumbDet into two stages, and the training strategy of ThumbDet is summarized in Algorithm 2. In Algorithm 2, W_t is the well-trained parameters of the teacher model. W_m, W_s, W_d denote the trainable parameters of down-sampling module, student model and object detection respectively. All the trainable parameters are initialized by random values in our method. In stage one, only the down-sampling module is trained. We perform supervision pre-training by minimizing moment-matching loss $\mathcal{L}_{mm}$ and obtain the parameters $W_m^{'}$. Stage two is the prior information transform stage, where we introduce a distillation-boost supervision strategy by minimizing the KD loss $\mathcal{L}_{KD}$ and original object detection loss $\mathcal{L}_{det}$ in Deformable-DETR to train a detector with W_s and W_d. Note that in this stage, the down-sampling module is also trained by the parameters $W_m^{'}$ from stage one. Eventually, ThumbDet obtains well-trained parameters W_m^{*}, W_s^{*} and W_d^{*} to detect objects on thumbnails images.

Algorithm 2 Training Strategy of ThumbDet. W_t is the well-trained parameters of the teacher model. W_m, W_s, W_d denote trainable parameters of the down-sampling module, student model and object detection respectively. All the trainable parameters are initialized by random values.

```
Require: $W_t$, $W_m$, $W_s$, $W_d$, $\mathcal{L}_{mm}$, $\mathcal{L}_{KD}$, $\mathcal{L}_{det}$
    Stage One: image down-sampling stage
    $W_m^{'} \leftarrow \arg\min_{W_m} \mathcal{L}_{mm}(W_m)$
    Stage Two: prior information transfer stage
    $W_m^{*}, W_s^{*}, W_d^{*} \leftarrow \arg\min_{W_m^{'}, W_s, W_t, W_d} \left[\mathcal{L}_{mm}(W_m^{'}) + \mathcal{L}_{KD}(W_s, W_t) + \mathcal{L}_{det}(W_d)\right]$
Ensure: $W_m^{*}, W_s^{*}, W_d^{*}$
```

4. Experiments

In this section, we conduct experiments to validate our proposed framework on MS COCO dataset [43]. First, we give a brief introduction of the used dataset and implementation details. Then, we show the main results of our method with 2× and 4× down-sampling rates. Moreover, some ablation studies are conducted to verify the effectiveness of each component in our ThumbDet pipeline. Finally, we compare our proposed method with some SOTA methods and show some qualitative results to further validate our proposed method.

4.1. Dataset and implementation details

MS COCO dataset. MS COCO [43] is a popular and widely used dataset in the task of object detection, which includes 80 categories taken in natural settings from daily life. There are 80k/40k/5k images selected randomly for training, validation, and testing respectively in this dataset. Following previous works [11,43], we use the union of 80k training images and a subset of 35k validation images (*i.e.* trainval 135k) to train our model, and use the remaining 5k validation images (*i.e.* minival) to evaluate our proposed method. The performance of all experiments follows the standard COCO-style precision metrics, *i.e.* mAP(IoU range of 0.5:0.95:0.05), AP$_{50}$(IoU@0.5), AP$_{75}$(IoU@0.75), AP$_S$(small-sized object), AP$_M$(medium-sized object) and AP$_L$(large-sized object) are reported in our paper.

Implementation details. We use Deformable-DETR with ResNet50 [44] as our baseline detector. AdamW is used as the optimizer with a base learning rate of 2×10^{-4}, and we decay it by a weight of 0.1 every 40 epochs. The size of the original large-scale image is (800, 1333), which denotes the short and maximum long side of the images. In our proposed method, the teacher is a well-trained model by using original large-scale images, and the student network shares the same training setting with the teacher except it uses 4× or 2× down-sampled input images. In the down-sampling stage, the λ in Eq (5) is set to 0.1 and the weight of moment-matching loss γ is set to 1. In the distillation-boost strategy, the temperature $\mathcal{T}$ is set to 1, and the weight of logit distillation α and feature map distillation β in Eq

Table 2
Main Results of our proposed method with ResNet50 backbone on COCO val dataset. Deformable-DETR is our baseline detector, which is also our teacher model trained on the original large-scale images. **Bicubic** denotes the low-resolution image is generated by the traditional bicubic interpolation method. **Ours** denotes the proposed framework with a down-sampling module and a distillation-boost supervision strategy. **Dow. rate** denotes the down-sampling rate. **FPS** (images/s, lower is better) and **GFLOPs** (larger is better) are computed on the first 100 images of COCO val dataset with an RTX3090 GPU.

Method	Dow. rate	mAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Params	FPS	↑ rate	GFLOPs	↓ rate
Deformable-DETR	-	43.8	62.6	47.7	26.4	47.1	58.0	40M	15.86	-	172.9	-
Bicubic	4×	29.1	45.4	30.4	9.0	29.5	48.0	40M	20.25	1.28×	13.25	13.05×
Ours	4×	32.3 (+3.2)	49.6 (+4.2)	33.9 (+3.5)	10.7 (+1.7)	33.7 (+4.2)	53.5 (+5.5)	40M	19.91	1.26×	13.36	12.94×
Deformable-DETR	-	43.8	62.6	47.7	26.4	47.1	58.0	40M	15.86	-	172.9	-
Bicubic	2×	38.6	56.7	41.6	19.0	41.7	56.6	40M	19.57	1.23×	44.82	3.86×
Ours	2×	41.6 (+3.0)	60.5 (+3.8)	44.8 (+3.2)	21.0 (+2.0)	45.4 (+3.7)	59.6 (+3.0)	40M	19.40	1.22×	45.72	3.78×

(9) are set to 0.5 and 0.1 respectively. Other parameters are the same as Deformable-DETR. As stated in the training strategy, in the first 5 epochs, only the down-sampling module is applied to generate thumbnail images, and then it is trained with a detector by knowledge distillation in the following training process. Our method is implemented in PyTorch on 8 RTX6000 GPUs.

4.2. Main results

In order to verify the effectiveness of our ThumbDet for object detection on low-resolution images, we perform object detection on 2× or 4× down-sampled images generated by our image down-sampling module. A comparison is performed between our proposed method, the baseline (*i.e.* detecting on the original large-scale images), and the baseline with bicubic down-sampling method, and the results on COCO val set are shown in Table 2. In our experiment, since both the teacher network (*i.e.* frozen network) and the student network (*i.e.* inference network) are Deformable-DETR with ResNet50 backbone, the parameters of all models are 40M.

For 4× down-sampling rate, compared to the bicubic down-sampling method, ThumbDet has an obvious advantage in performance owing to its down-sampling module and distillation-boost strategy. As shown in Table 2, ThumbDet brings a 3.2% mAP improvement (32.3% *vs.* 29.1%), where the most notable improvements are on large-size objects (from 48.0% to 53.5%, +5.5%). Moreover, ThumbDet achieves a large margin improvement (from 29.5% to 33.7%, +4.2%) on middle-size objects, and an impressive result of 10.7% is obtained on small-sized objects, surpassing the bicubic down-sampling method by 1.7% absolutely. The above comparison clearly demonstrates the effectiveness of our proposed method on very low-resolution object detection.

Moreover, to verify the efficiency of ThumbDet, we compute the FPS (images/s) and GFLOPs on the first 100 images of COCO val dataset with an RTX3090 GPU. From the last third and forth columns of Table 2, we can see that the FPS of ThumbDet is 19.91 images/s, which is 1.26× faster than the baseline method (15.86 images/s) trained with the original large-scale images. The main reason is that ThumbDet conducts convolutional computation on the thumbnail image which only contains discriminative properties, whereas the baseline detector (*i.e.* Deformable-DETR) has to compute extra redundant information in the large-scale images. Meanwhile, as shown in the last two columns of Table 2, we can see that GFLOPs, a metric of algorithm complexity, has a sharp drop (from 172.9 to 13.36) when comparing our method with the baseline [11], which means the complexity of our ThumbDet is reduced by a factor of 12.94×. Although the inference speed and algorithm complexity of our ThumbDet are comparable to the bicubic down-sampling method (19.91 images/s *vs.* 20.25 images/s, 13.36 GFLOPs *vs.* 13.25 GFLOPs, where the slight increments come from two convolutional layers in our down-sampling module), our method has a better detection performance than the bicubic down-sampling method. In summary, our proposed method not only reduces the model computation but also maintains a satisfactory detection performance.

For 2× down-sampling rate, following [40], the structure of the 2× down-sampling module also contains 2 convolutional layers, and each convolutional layer has a 5 × 5 convolution kernel followed by a ReLU operation. The numbers of convolution kernel in the first layer and the second layer are 12 and 3 respectively. As shown in Table 2, ThumbDet outperforms the traditional bicubic method by 3.0% in terms of mAP (from 38.6% to 41.6%). Furthermore, our method has approximately the same inference speed and algorithm complexity as the bicubic method (19.40 images/s *vs.* 19.57 images/s, 45.72 GFLOPs *vs.* 44.82 GFLOPs). Moreover, compared to the baseline detector (*i.e.* Deformable-DETR) trained with the original large-scale images, although we use 2× down-sampled images as inputs, we can still achieve competitive results (41.6% *vs.* 43.8%). To sum up, ThumbDet achieves a 1.22× improvement in inference speed and a 3.78× reduction in algorithm complexity while only dropping 2.2% in mAP performance compared to the baseline detector with the original-size image, which further verifies the effectiveness of our ThumbDet for object detection on low-resolution images.

4.3. Ablation studies

In order to explore the effectiveness of each component in our ThumbDet, we ablate each component individually and report the detection performance on MS COCO [43] validation set. In this section, for easy implementation and fair comparison, we use a training schedule of 50 epochs with a learning rate dropped by a factor of 10 after 40 epochs in all experiments.

4.3.1. Effectiveness of each component

In this paper, we propose a down-sampling module and a distillation-boost supervision strategy to accelerate network training and maintain detection performance. Here, we conduct some experiments to investigate the effects of each component.

Down-sampling module. To verify the effectiveness of our down-sampling module, we conduct an ablation experiment by replacing the bicubic down-sampling method with our down-sampling module, and then perform object detection on the generated low-resolution images. As shown in Table 3, the first row denotes that the bicubic down-sampling method is used to obtain low-resolution images for object detection, and the second row denotes using thumbnail images learned by our down-sampling module as the input of the detection network. From Table 3, we can see that the down-sampling module gains 1.6% mAP improvement (from 29.1% to 30.7%) compared to the bicubic down-sampling

Table 3
Ablation study of each component in ThumbDet. $\mathcal{L}_{mm}$, $\mathcal{L}_{cls}$ and $\mathcal{L}_{fm}$ denote the loss of down-sampling module, logit distillation and feature map distillation respectively. The first row is the detection performance using the bicubic method to obtain low-resolution images. The second row uses thumbnail images generated by our down-sampling module as inputs for object detection. The third row only uses logit distillation, and the last row further includes feature map distillation for object detection on Thumbnail images.

$\mathcal{L}_{mm}$	$\mathcal{L}_{cls}$	$\mathcal{L}_{fm}$	mAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
-	-	-	29.1	45.4	30.4	9.0	29.5	48.0
✓			30.7	47.4	32.2	10.4	31.4	50.7
✓	✓		30.9	47.6	32.6	9.9	32.0	51.8
✓	✓	✓	31.2	48.4	32.6	9.8	32.0	52.5

Table 4
Comparison of different Layers in the Down-sampling Module. In order to avoid the influence of other factors, here, a distillation-boost strategy is not used in this experiment. $\mathcal{N}$ denotes the number of layers. The single convolutional layer in the first row is 4 strides with a 5 × 5 convolution kernel. The second row is the setting of our down-sampling module and the details are described in Section 3.3. The third row is the 5 convolutional layers with kernels set to 7 × 7, 5 × 5, 3 × 3, 1 × 1 and 1 × 1 respectively. FPS (images/s) is computed on the first 100 images of COCO val dataset with an RTX3090 GPU.

$\mathcal{N}$	mAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	FPS
1	10.9	16.6	11.5	1.7	9.4	21.4	20.26
2	**30.7**	47.4	32.2	10.4	31.4	50.7	19.95
5	26.7	42.6	27.7	8.4	26.7	44.1	12.61

method. The reason is that our down-sampling module is trained in a supervised manner, and the generated low-resolution image contains discriminative properties, which are machine vision oriented and good for object detection.

Distillation-boost supervision strategy. To further verify the effectiveness of distillation-boost supervision strategy in maintaining the detection performance of low-resolution images, we conduct an ablation study with/without the distillation-boost supervision strategy. As shown in Table 3, the last two rows are the performance of adding the distillation-boost supervision strategy on the basis of the down-sampling module. Here, we explore logit distillation and feature map distillation separately. Specifically, in the third row in Table 3, compared to only using the down-sampling module, logit distillation brings +0.2% mAP improvement (from 30.7% to 30.9%), where it achieves impressive performance on middle and large-size objects, *i.e.* AP_M and AP_L improved by 0.6% and 1.1% in mAP respectively.

Moreover, when using both feature map distillation and logit distillation, we found that the result can be further improved. As shown in the last row of Table 3, after adding feature map distillation, the detection performance increases from 30.9% to 31.2% (+0.3% in mAP). Based on this foundation, we can conclude that knowledge distillation at the feature-map level is more efficient on maintaining high-frequency detailed information than logit distillation.

4.3.2. Analysis number of layers in the down-sampling module

Intuitively, in Eq (1), a simple single convolutional layer is sufficient for generating 4× down-sampling thumbnail images. However, we experimentally found that such a structure does not get satisfactory performance. We present the results of using different layers in the down-sampling module, as shown in Table 4, where the single convolutional layer in the first row is 4 strides with a 5 × 5 convolution kernel, the second row is the setting of our down-sampling module with 2 convolutional layers described in Section 3.3, and the third row is 5 convolutional layers with kernels set to 7 × 7, 5 × 5, 3 × 3, 1 × 1, and 1 × 1 respectively. As shown in Table 4, from the first row, we find that a simple single convolutional layer gets poor performance, *i.e.* only 10.9% in mAP. The reason is that the feature extraction capability of a single convolution layer is too weak, which can not learn the key information for object detection on thumbnail images. However, the two layers down-sampling module (in the second row) achieves a huge improvement, and it gets 30.7% in mAP surpassing the single layer by 19.8% absolutely. Meanwhile, the inference speed is comparable to the single convolutional layer (19.95 images/s *vs.* 20.26 images/s). The last row in Table 4 shows the performance of a 5 convolutional layers down-sampling module, and we can see that its mAP (26.7%) is lower than two layers down-sampling module (30.7%), while it decreases the value of FPS from 19.95 images/s to 12.61 images/s. In order to balance the detection accuracy and inference speed, we choose two layers in all our experiments.

We further analyze and discuss the influence of different convolution layers in the down-sampling module by visualizing the down-sampled results as shown in Fig. 5, *i.e.* we visualize the images including original large-scale images, thumbnail images from the single convolutional layer down-sampling module, 2 convolutional layers down-sampling module and 5 convolutional layers down-sampling module. From Fig. 5, we can see that the difference between (a) original large-scale images and (b) thumbnail images from the single layer down-sampling module is huge, *i.e.* the color of (b) is mainly green and black, which cannot reflect the pixel-level information of the original large-scale image and is not good for machine vision (10.9% in mAP). For the thumbnail images from the 2 layers down-sampling module (c), they have a similar distribution in each color channel to the original large-scale images, which further proves that the moment-matching loss (in Section 3.3) can supervise the mean and variance of thumbnail images as close as possible to the mean and variance of original images. Comparing the thumbnail images from the 2 layers down-sampling module (c) with the thumbnail images from the 5 layers down-sampling module (d), the pixel distribution of thumbnail images from 5 layers is unfriendly, *i.e.* missing some semantic information for subsequent object detection tasks. We think this is the reason why when using 2 convolutional layers, the performance of the down-sampling module tends to be better than when using 5 convolutional layers (30.7% *vs.* 26.7% in mAP).

4.3.3. Effectiveness of two stage strategy

In our method, we design a two-stage strategy to train our model. In the first stage, we only use the down-sampling module to obtain a thumbnail image for fast convergence. In the second stage, we perform knowledge distillation and object detection simultaneously. In this subsection, to validate the effectiveness of two stage strategy, we conduct some ablation studies by using different training strategies. As shown in Table 5, the first row is the traditional bicubic method to generate low-resolution images, *i.e.* no down-sampling module is applied during the whole training process. The second row is the one-stage strategy, which means the down-sampling and detection operations are trained simultaneously in the training process. We can see that the one-stage strategy is worse than the traditional bicubic method (22.2% *vs.* 29.1% in mAP). Actually, this is the main reason why we design a two-stage training strategy, *i.e.* separate the training of the down-sampling module and other components in different stages.

In the initial two-stage strategy (Two-Stage w/o Dow.), we train a down-sampling module to obtain thumbnail images separately in the first stage, then the down-sampling module is frozen and the learned thumbnail images are fed into the second stage to perform object detection. The result is shown in the third row of Table 5, surprisingly, the detection result (*i.e.* 18.1% in mAP) is even worse than the one-stage strategy (*i.e.* 22.2% in mAP). Therefore,

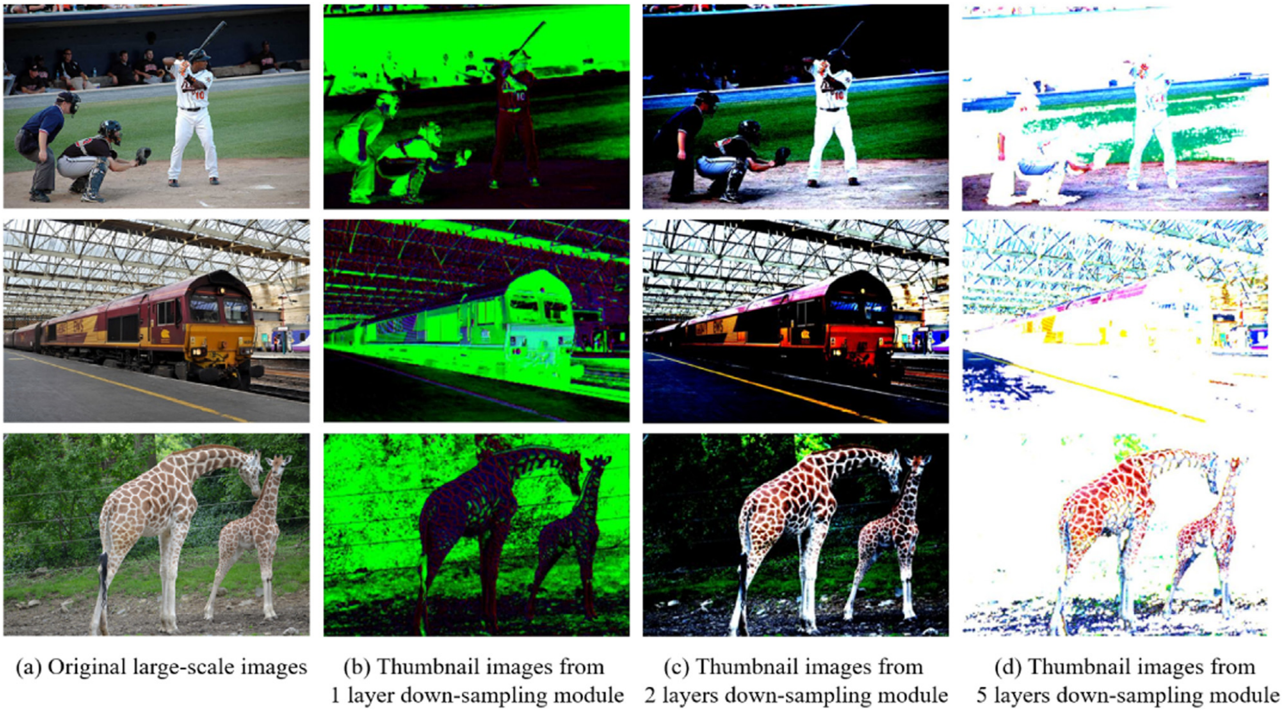
(a) Original large-scale images
(b) Thumbnail images from 1 layer down-sampling module
(c) Thumbnail images from 2 layers down-sampling module
(d) Thumbnail images from 5 layers down-sampling module

Fig. 5. Visualization of Thumbnail images from the down-sampling module by using different convolutional layers on MS COCO val dataset. The first column (a) shows the original large-scale images. The last three columns (b), (c) and (d) are thumbnail images from the 1 layer down-sampling module, 2 layers down-sampling module and 5 layers down-sampling module respectively. Note that thumbnail images of the last three columns are zoomed in four times for looking clear and best seen on the computer.

Table 5
Effectiveness of the Two Stage Strategy. Bicubic denotes the traditional bicubic method is used to generate low-resolution images, *i.e.* no down-sampling module is applied during the whole training process. **One-Stage** denotes the down-sampling module and detection operation are trained simultaneously in the whole training process. **Two-Stage w/o Dow.** denotes that we train a down-sampling module to obtain thumbnail images in the first stage separately, while this module is forbidden to use in the second stage for object detection. Here, **Dow.** denotes the down-sampling module. **Two-Stage w/ Dow.** is our final used training strategy, where the down-sampling module is used in both of the first and second stages.

Strategy	mAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Bicubic	29.1	45.4	30.4	9.0	29.5	48.0
One-Stage	22.2	36.0	22.8	6.3	21.2	37.0
Two-Stage w/o Dow.	18.1	30.4	18.7	5.6	17.7	30.7
Two-Stage w/ Dow.	**30.7**	47.4	32.2	10.4	31.4	50.7

Table 6
Results of different Temperatures $\mathcal{T}$ in logit distillation. Some cases of $\mathcal{T} < 1$ and $\mathcal{T} > 1$ are used to study the influence of temperature $\mathcal{T}$.

$\mathcal{T}$	mAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
0.5	30.5	47.0	32.1	10.1	31.2	51.2
1	**31.2**	48.4	32.6	9.8	32.0	52.5
1.5	27.0	42.5	28.2	8.1	21.1	45.9
2	27.1	42.3	28.1	8.5	27.1	45.5

we have a question whether the down-sampling module is necessary in the second stage. To this end, we further use the down-sampling module in both the first and second stages (Two-Stage w/ Dow.), as shown in the last row of Table 5, and our two-stage strategy achieves the best performance (*i.e.* 30.7% mAP) on thumbnail images, outperforming other training strategies by a large margin. The above comparison demonstrates the effectiveness of our proposed two-stage training stage, *i.e.* training the down-sampling module for 5 epochs independently in the first stage and then training all components in the second stage simultaneously.

4.3.4. Analysis of temperature $\mathcal{T}$ in logit distillation

In Eq (7), we use hyper-parameter $\mathcal{T}$ to soften the classification probability in logit distillation, here we conduct some experiments on $\mathcal{T} < 1$ and $\mathcal{T} > 1$ to study the influence of temperature $\mathcal{T}$ on COCO validation set. As shown in Table 6, when $\mathcal{T} = 1$, we get the best performance of 31.2% in mAP, surpassing $\mathcal{T} = 0.5$ by 0.7% absolutely (from 30.5% to 31.2%). However, as the temperature $\mathcal{T}$ increases, the performance of the student network get worse. For example, compared with $\mathcal{T} = 1$, the performance drops by 4.1% in mAP when $\mathcal{T} = 2$ (from 31.2% to 27.1%). The above experimental results indicate that the student model is sensitive to the hyper-parameters $\mathcal{T}$, and we set $\mathcal{T} = 1$ by default in this paper.

4.4. Comparison with state-of-the-art methods

To demonstrate the effectiveness of our method on low-resolution object detection, we compare our proposed ThumbDet with state-of-the-art low-resolution detectors RestoreDet [12] and MSAD [45]. First, we compare our method with RestoreDet, which captures the dynamic feature by learning the degradation equivalent representation for detecting objects on low-resolution. The main purpose of our paper is to detect objects on very low-resolution (*i.e.* 4× down-sampling). However, to verify the effectiveness of our ThumbDet, similar to [12], we also report our performance with a 2× down-sampling rate. The structure of the 2× down-sampling module is described in Section 4.2, *i.e.* contains 2 convolutional layers, and each convolutional layer with a 5×5 kernel followed by a ReLU operation. The numbers of convolution kernels in the first layer and the second layer are 12 and 3 respectively. The comparison results are shown in Table 7, where the results of RestoreDet are directly taken from the published paper.

From Table 7, we can observe that the proposed ThumbDet surpasses all settings of RestoreDet (*i.e.* both 4× and 2× down-

Table 7
Comparison of our proposed ThumbDet with state-of-the-art low resolution detectors. The detection results are reported on MS COCO [43] val dataset with 4× and 2× down-sampling rates (**Dow. rate**). **Df-DETR** denotes Deformable-DETR [11]. $\mathcal{A}\downarrow$ (%) is the Accuracy Drop Rate (lower is better) which is calculated by Eq (11), and Δ (larger is better) denotes the gap between Accuracy Drop Rate $\mathcal{A}\downarrow$ of the traditional bicuibic method, our proposed method and RestoreDet. **FPS** (images/s, lower is better) and **GFLOPs** (larger is better) are computed on the first 100 images of COCO val dataset with an RTX3090 GPU.

Method	Dow. rate	Detector	Degradition	mAP	AP_S	AP_M	AP_L	$\mathcal{A}\downarrow$ (%)	Δ(%)	FPS ↑	GFLOPs ↓
RestoreDet[12]	4×	CenterNet	Bicubic	12.9	0.8	8.7	34.4	57.0	4.7↑	58.7	3.47
RestoreDet[12]	4×		Learnable	14.3 (+1.4)	1.5	12.6	34.9	52.3		50.5	3.69
RestoreDet Teacher[12]	-		-	30.0	10.6	33.2	47.2	-		35.5	48.67
ThumbDet (Ours)	4×	Df-DETR	Bicubic	29.1	9.0	29.5	48.0	33.7	7.4 ↑	20.25	13.25
ThumbDet (Ours)	4×		Learnable	32.3 (+3.2)	10.7	33.7	53.5	26.3		19.91	13.36
ThumbDet Teacher	-		-	43.8	26.4	47.1	58.0	-		18.56	172.9
RestoreDet[12]	2×	CenterNet	Bicubic	19.5	5.3	20.1	33.3	35.0	6.7 ↑	49.2	12.65
RestoreDet[12]	2×		Learnable	21.5 (+2.0)	3.2	20.8	47.3	28.3		46.4	13.7
RestoreDet Teacher[12]	-		-	30.0	10.6	33.2	47.2	-		35.5	48.67
ThumbDet (Ours)	2×	Df-DETR	Bicubic	38.6	19.0	41.7	56.6	11.9	6.9 ↑	19.57	44.82
ThumbDet (Ours)	2×		Learnable	41.6 (+3.0)	21.0	45.4	59.6	5.0		19.40	45.72
ThumbDet Teacher	-		-	43.8	26.4	47.1	58.0	-		18.56	172.9

Table 8
Comparison of our proposed ThumbDet with the 2× down-scaled SOTA object detection method MSAD [45]. **Dow. rate** denotes the down-sampling rate. Note that for the metric of **Params** and **GFLOPs**, lower is better, and for **FPS**, larger is better.

Method	Dow. rate	mAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Params ↓	FPS ↑	GFLOPs ↓
MSAD [45]	2×	41.6	59.9	44.9	22.8	44.8	57.0	68.6M	14.16	57.09
ThumbDet (Ours)	2×	41.6	60.5	44.8	21.0	45.4	59.6	40M	19.40	45.72

sampling rates) on the COCO val dataset. To be specific, for 4× down-sampling, we can see that RestoreDet achieves 14.3% detection performance, it improves 1.4% in mAP (12.9% mAP *vs.* 14.3% mAP) compared to using traditional bicubic method to obtain low-resolution images in the top of the Table 7. However, our ThumbDet obtains 32.3% in mAP, surpassing RestoreDet by a large margin, *i.e.* 18% absolutely in mAP (from 14.3% to 32.3%). Moreover, we achieve a 3.2% mAP improvement (from 29.1% to 32.3%) when compared with the bicubic method. For 2× down-sampling, compared to the bicubic method, RestoreDet improves 2.0% in mAP (19.5% mAP *vs.* 21.5% mAP), while our ThumbDet bring 3.0% mAP improvement (from 38.6% to 41.6% in mAP). The above experiments clearly demonstrate the effectiveness of our proposed object detection method on very low-resolution images.

Moreover, considering that RestoreDet uses CenterNet [46] as the baseline detector, for fair comparison, we define $\mathcal{A}\downarrow$ (%) as the Accuracy Drop Rate (lower is better) to compare our method with RestoreDet, which can be calculated by:

$$\mathcal{A}\downarrow=\frac{\mathcal{A}_{teacher}-\mathcal{A}_{student}}{\mathcal{A}_{teacher}} \tag{11}$$

where $\mathcal{A}_{teacher}$ denotes the performance of the teacher model, and $\mathcal{A}_{student}$ denotes the performance of the corresponding student model.

As shown in Table 7, for 4× down-sampling, the Accuracy Drop Rate gap Δ (larger is better) between the traditional bicubic method and RestoreDet is +4.7% (from 57.0% to 52.3%), but for our proposed ThumbDet it is +7.4% (from 33.7% to 26.3%), which means our method outperforms RestoreDet by 2.7%. For 2× down-sampling, the Accuracy Drop Rate $\mathcal{A}\downarrow$ (lower is better) of ThumbDet is only 5.0% which is much lower than RestoreDet (28.3%), which also clearly demonstrates the effectiveness of our method on low-resolution object detection.

We also use FPS and GFLOPs in Table 7 to compare the computational time and complexity of our proposed method with other low-resolution detectors. For both 2× and 4× down-sampling rates, as shown in Table 7, ThumbDet is not better than RestoreDet [12] on FPS and GFLOPs metrics, but we surpass RestoreDet [12] on detection performance by a large margin, *i.e.* +18.0% in mAP and +20.1% in mAP for 4× and 2× down-sampling rates respectively. The reason is that RestoreDet [12] uses ResNet18 backbone and is a CNN-based anchor free object detection network, which are different from our method (ResNet50 as the backbone and Deformabe-DETR as the baseline detector), thus the metrics of FPS and GFLOPs may not truly reflect the inference speed and complexity of these two models. To prove our viewpoint, we perform an additional experiment by replacing ResNet18 in RestoreDet [12] with Swin-Transformer, and the inference speed and complexity of RestoreDet are decreased in both 4× and 2× down-sampling rates. Specifically, for 4× down-sampling rate, FPS (larger is better) decreases from 50.5 images/s to 33.4 images/s, and GFLOPs (lower is better) increases from 3.69 to 8.53. For 2× down-sampling rate, FPS decreases from 46.4 images/s to 31.5 images/s, and GFLOPs increases from 13.70 to 31.7. By replacing ResNet18 in RestoreDet with the complex Swin-Transformer, the gap of FPS and GFLOPs between ThumbDet and RestoreDet is further narrowed, and we believe applying our ThumbDet to some light backbones and faster detectors can also get higher FPS and lower GFLOPs. Note that under the same conditions, as shown in Table 2, our proposed method has a huge improvement on FPS and GFLOPs, which undoubtedly proves the correctness of our motivation, *i.e.* reducing the computation of deep networks by obtaining a thumbnail image.

To further validate the effectiveness of our ThumbDet, we also compare our proposed method with MSAD [45], which is a SOTA object detection method for 2× down-scaled images. As shown in Table 8, our method achieves a comparable result in overall mAP and AP_{75} with MSAD (41.6% *vs.* 41.6%, 44.8% *vs.* 44.9% respectively), while it obtains a +0.6% improvement in AP_{50} (from 59.9% to 60.5%). Moreover, compared to MSAD, ThumbDet achieves a large margin improvement (from 57.0% to 59.6%, +2.6% in mAP) on the large-scale image set, and an improvement of 0.6% in mAP on the middle-scale image set (from 44.8% to 45.4% in mAP). However, our method does not perform well on the small-scale set (21.0% (ours) *vs.* 22.8% (MSAD)), and the main reason is that our method is a DETR-like architecture, which has a weak detection ability on the small-scale image set. As mentioned earlier, our ThumbDet focuses on detecting objects on very low-resolution images (*i.e.* 4× down-

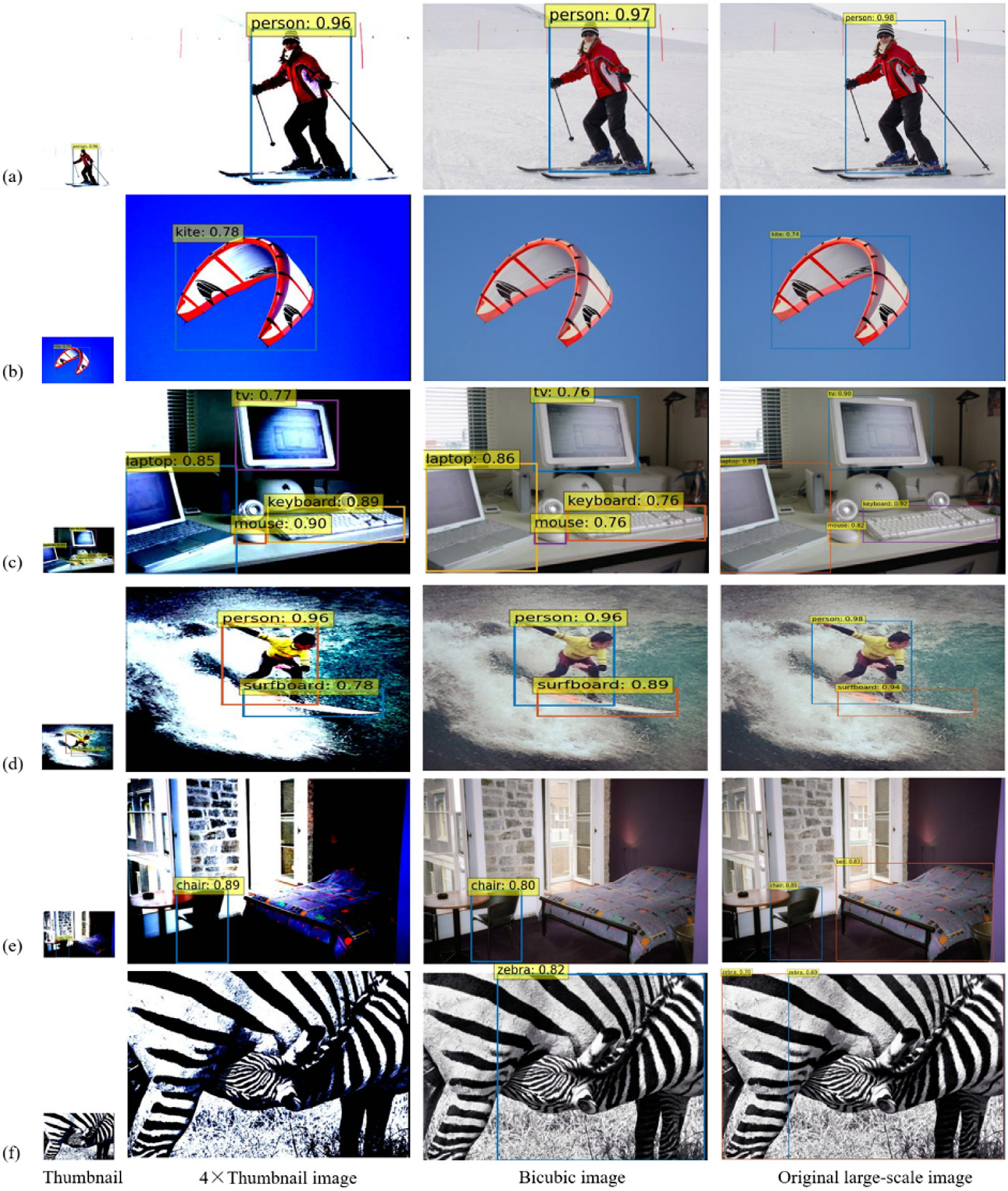

Fig. 6. Examples of object detection results on MS COCO val dataset. The four columns are the thumbnail images, object detection results on thumbnail images and bicubic images that are zoomed in four times for clarity, and Deformable-DETR object detection results on original large-scale images, respectively. For each row of pictures, (a)/(b) are the detected large-scale objects, and (c)/(d) are some middle and small cases detected by different methods. We also show the missed cases in picture (e), and even failed examples in picture (f). Best seen on computer, in color and zoomed in.

sampling), however MSAD is a well-designed method for object detection only in 2× down-sampling rate. Even under this unfair condition, our proposed ThumbDet can still achieve comparable or better performance, which demonstrates the robustness and generalization of our ThumbDet for low-resolution object detection.

Furthermore, we also provide a more accurate comparison of the computational time and complexity of ThumbDet with MSAD [45], *i.e.* Params (parameters), FPS and GFLOPs. As shown in Table 8, we can get the following conclusion. (1) For the metric of Params and GFLOPs (both of them, lower is better), which can reflect the size of the model and the complexity of the algorithm, our method has lower parameters (40M *vs.* 68.6M) and GFLOPs (45.72 *vs.* 57.09). It means that ThumbDet is lightweight and easy to deploy in real-world applications with limited computing resources. (2) We can see that the FPS (larger is better) of ThumbDet is 19.40 images/s, which is 1.37× faster than MSAD (14.16 images/s). These conclusions further demonstrate the efficiency of our ThumbDet in low-resolution object detection.

4.5. Visualization results of ThumbDet

We show some visualization results on MS COCO val dataset [43] in Fig. 6. As shown in Fig. 6, the first column shows thumbnail images generated from our method. In order to look clarity, we zoom in them to be as large as the original large-scale image as shown in the second column. The third column shows the results of the bicubic method and zoomed in as original large-scale images. The last column shows the detection results on original large-scale images by Deformable-DETR.

From pictures (a)-(d), we can see that our ThumbDet achieves competitive results with Deformable-DETR, and all objects can be detected accurately, while the bicubic method fails in some cases. From these visualization results, we can conclude that although the thumbnail images are different from original images in colors and lightness, we think these thumbnail images are more machine vision oriented rather than human visual effect and they already contain discriminative properties for easier object detection. Moreover, we also show some failure cases, where missing detection is the main problem of low-resolution object detection. For example, in picture (e), although ThumbDet detects the presence of a chair, it misses the bed target object. The situation becomes even worse when the object and background are very similar, as shown in picture (f), and our method fails to detect the presence of zebras. These failed results indicate that more progress is needed to further improve the performance on low-resolution images.

5. Conclusion and future work

In this paper, we propose a novel framework, ThumbDet, to boost the performance of object detection on very low-resolution images. To overcome the problem that images small-scaled by traditional interpolation methods hurt the detection performance dramatically, a down-sampling module is proposed to learn thumbnail images under the supervision of image down-scaling, knowledge distillation, and object detection losses, and then the obtained thumbnail images look realistic and contain discriminative properties are used to replace their original-size counterparts for reliable object detection. To further maintain the detection performance of low-resolution images as the original-size inputs, a distillation-boost strategy is introduced in our framework, where logit distillation and feature map distillation are used to transfer the knowledge from the teacher network to the student network. Based on the proposed method, we can achieve satisfactory detection performance while drastically reducing computation and memory requirements when using very low-resolution images (*i.e.* 4× down-sampling) as inputs. However, we still cannot solve some extremely hard thumbnail images, *e.g.* some objects are very similar to the background. We plan to use contrastive learning on thumbnail images and original images to learn robust features for detecting those failure cases in the future.

However, object detection on low-resolution images is still a challenging task, especially for the problem of missing detection. In the future, we plan to design more effective methods to learn the stronger feature maps of small-scale images for achieving satisfactory performance while maintaining a low computational cost. We will also extend our method to low-quality (*e.g.* dark or blur environment) object detection.

Declaration of Competing Interest

We would like to note that in the manuscript entitled ǣThumbDet: One Thumbnail Image is Enough for Object Detectiong, no conflict of interest exits in the submission of this manuscript, and manuscript is approved by all authors for publication.

Data availability

No data was used for the research described in the article.

Acknowledgment

This work is supported by the China Postdoctoral Science Foundation (Grant No. 259822), the National Postdoctoral program for Innovative Talents (Grant No. BX20200108), the National Science Foundation of China (Grant No. 62206077, No. 61976070), and the Science Foundation of Heilongjiang Province (LH2021F024).

References

[1] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, Adv Neural Inf Process Syst 28 (2015).

[2] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: European Conference on Computer Vision, Springer, 2020, pp. 213–229.

[3] H. Su, Y. He, R. Jiang, J. Zhang, W. Zou, B. Fan, DSLA: dynamic smooth label assignment for efficient anchor-free object detection, Pattern Recognit (2022) 108868.

[4] R. Karthik, R. Menaka, M. Hariharan, D. Won, Contour-enhanced attention CNN for ct-based covid-19 segmentation, Pattern Recognit 125 (2022) 108538.

[5] H. Basak, R. Kundu, R. Sarkar, Mfsnet: a multi focus segmentation network for skin lesion segmentation, Pattern Recognit 128 (2022) 108673.

[6] C. Wang, F. Zhang, X. Zhu, S.S. Ge, Low-resolution human pose estimation, Pattern Recognit 126 (2022) 108579.

[7] J. Mei, X. Jiang, H. Ding, Spatial feature mapping for 6dof object pose estimation, Pattern Recognit (2022) 108835.

[8] E.L. Denton, W. Zaremba, J. Bruna, Y. LeCun, R. Fergus, Exploiting linear structure within convolutional networks for efficient evaluation, Adv Neural Inf Process Syst 27 (2014).

[9] X. Zhang, J. Zou, K. He, J. Sun, Accelerating very deep convolutional networks for classification and detection, IEEE Trans Pattern Anal Mach Intell 38 (10) (2015) 1943–1955.

[10] Z. Xie, Z. Zhang, Y. Cao, Y. Lin, J. Bao, Z. Yao, Q. Dai, H. Hu, Simmim: A simple framework for masked image modeling, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2022, pp. 9653–9663.

[11] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, J. Dai, Deformable DETR: deformable transformers for end-to-end object detection, International Conference on Learning Representations (2020).

[12] Z. Cui, Y. Zhu, L. Gu, G.-J. Qi, X. Li, P. Gao, Z. Zhang, T. Harada, RestoreDet: degradation equivariant representation for object detection in low resolution images, arXiv preprint arXiv:2201.02314 (2022).

[13] R. Zhou, S. Susstrunk, Kernel modeling super-resolution on real low-resolution images, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 2433–2443.

[14] X. Ji, Y. Cao, Y. Tai, C. Wang, J. Li, F. Huang, Real-world super-resolution via kernel estimation and noise injection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 466–467.

[15] S. Bell-Kligler, A. Shocher, M. Irani, Blind super-resolution kernel estimation using an internal-gan, Adv Neural Inf Process Syst 32 (2019).

[16] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network (2015), Neural Information Processing Systems 2 (2015).

[17] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.

[18] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.

[19] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE Trans Pattern Anal Mach Intell 37 (9) (2015) 1904–1916.

[20] J. Redmon, A. Farhadi, Yolo9000: better, faster, stronger, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7263–7271.

[21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, Ssd: Single shot multibox detector, in: European Conference on Computer Vision, Springer, 2016, pp. 21–37.

[22] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2980–2988.
[23] Z. Tian, C. Shen, H. Chen, T. He, FCOS: a simple and strong anchor-free object detector, IEEE Trans Pattern Anal Mach Intell (2020).
[24] S. Liu, F. Li, H. Zhang, X. Yang, X. Qi, H. Su, J. Zhu, L. Zhang, DAB-DETR: dynamic anchor boxes are better queries for detr, International Conference on Learning Representations (2022).
[25] F. Li, H. Zhang, S. Liu, J. Guo, L.M. Ni, L. Zhang, DN-DETR: Accelerate detr training by introducing query denoising, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2022, pp. 13619–13627.
[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv Neural Inf Process Syst 30 (2017).
[27] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: transformers for image recognition at scale, International Conference on Learning Representations (2020).
[28] A. Romero, N. Ballas, S.E. Kahou, A. Chassang, C. Gatta, Y. Bengio, Fitnets: hints for thin deep nets, International Conference on Learning Representations (2014).
[29] T.-B. Xu, P. Yang, X.-Y. Zhang, C.-L. Liu, Lightweightnet: toward fast and lightweight convolutional neural networks via architecture distillation, Pattern Recognit (2019) 272–284.
[30] B. Zhao, Q. Cui, R. Song, Y. Qiu, J. Liang, Decoupled knowledge distillation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2022, pp. 11953–11962.
[31] Z. Yang, Z. Li, X. Jiang, Y. Gong, Z. Yuan, D. Zhao, C. Yuan, Focal and global knowledge distillation for detectors, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2022, pp. 4643–4652.
[32] D. Yang, Y. Zhou, A. Zhang, X. Sun, D. Wu, W. Wang, Q. Ye, Multi-view correlation distillation for incremental object detection, Pattern Recognit (2022) 108863.
[33] Z. Zheng, R. Ye, P. Wang, D. Ren, W. Zuo, Q. Hou, M.-M. Cheng, Localization distillation for dense object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2022, pp. 9407–9416.
[34] Z. Wang, S. Chang, Y. Yang, D. Liu, T.S. Huang, Studying very low resolution recognition using deep networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4792–4800.
[35] Y. Zhang, Y. Bai, M. Ding, S. Xu, B. Ghanem, KGSNET: key-point-guided super-resolution network for pedestrian detection in the wild, IEEE Trans Neural Netw Learn Syst 32 (5) (2020) 2251–2265.
[36] D. Liu, B. Wen, X. Liu, Z. Wang, T.S. Huang, When image denoising meets high-level vision tasks: a deep learning approach, Arxiv:1706.04284 (2017).
[37] Y. Jin, Y. Zhang, Y. Cen, Y. Li, V. Mladenovic, V. Voronin, Pedestrian detection with super-resolution reconstruction for low-quality image, Pattern Recognit 115 (2021) 107846.
[38] D. Li, A. Yao, Q. Chen, Learning to learn parameterized classification networks for scalable input images, in: European Conference on Computer Vision, Springer, 2020, pp. 19–35.
[39] Y. Wang, F. Sun, D. Li, A. Yao, Resolution switchable networks for runtime efficient image recognition, in: European Conference on Computer Vision, Springer, 2020, pp. 533–549.
[40] C. Zhao, B. Ghanem, Thumbnet: One thumbnail image contains all you need for recognition, in: Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 1506–1514.
[41] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, Deformable convolutional networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 764–773.
[42] B. Heo, J. Kim, S. Yun, H. Park, N. Kwak, J.Y. Choi, A comprehensive overhaul of feature distillation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1921–1930.
[43] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: Common objects in context, in: European Conference on Computer Vision, Springer, 2014, pp. 740–755.
[44] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[45] L. Qi, J. Kuen, J. Gu, Z. Lin, Y. Wang, Y. Chen, Y. Li, J. Jia, Multi-scale aligned distillation for low-resolution detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2021, pp. 14443–14453.
[46] X. Zhou, D. Wang, P. Krähenbühl, Objects as points, Arxiv:1904.07850(2019).

Yongqiang Zhang received the M.S. and Ph.D. degrees in instrument science and technology from Harbin Institute of Technology (HIT), Harbin, China, in 2015 and 2020, respectively. He worked at King Abdullah University of Science and Technology (KAUST) as a visiting student from 2017 to 2018. Currently, he is an assistant professor in the School of Instrumentation Science and Engineering at Harbin Institute of Technology. His research areas are computer vision, pattern recognition, machine learning, and deep learning. His research interests mainly include face detection, weakly/fully supervised object detection, activity detection, image and video understanding in the real-world.

Yin Zhang received the MS degree in instrument science and technology from Harbin Institute of Technology (HIT), Harbin, China, in 2021. He is currently a PhD student from the Harbin Institute of Technology (HIT). His research areas are computer vision, pattern recognition, machine learning, and deep learning. His research interests mainly include object detection, knowledge distillation, image super-resolution.

Rui Tian received the bachelor's degree from Harbin University of Science and Technology in 2021. He is studying for a master's degree from Harbin Institute of Technology. His research areas are computer vision, pattern recognition, deep learning and weakly/fully supervised object detection.

Zian Zhang received the B.S. and M.S. degrees in instrument science and technology from Harbin Institute of Technology (HIT), Harbin, China, in 2018 and 2020, respectively. Currently, he is a doctoral student in the School of Instrumentation Science and Engineering at Harbin Institute of Technology. His research areas are computer vision, pattern recognition, machine learning, and deep learning. His research interests mainly include human pose estimation, object detection, action recognition, gait recognition, image and video understanding in the monitoring system.

Yancheng Bai is an associate professor in Pattern Recognition and Intelligent System at Institute of Software, Chinese Academy of Sciences, Beijing, China. He received the MS degree from Sichuan university, Sichuan, China, in 2009. He received the PhD degree from Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2014. He worked as a post doctoral at King Abdullah University of Science and Technology (KAUST) from 2016 to 2018. His research interests include computer vision, pattern recognition, artificial intelligence, machine learning, and deep learning.

Wangmeng Zuo received the Ph.D. degree in computer application technology from the Harbin Institute of Technology, Harbin, China, in 2007. From 2004 to 2006, he was a Research Assistant with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. From 2009 to 2010, he was a Visiting Professor with Microsoft Research Asia. He is currently a Professor with the School of Computer Science and Technology, Harbin Institute of Technology. He has published over 100 articles in toptier academic journals and conferences. His current research interests include image enhancement and restoration, image/video generation, and image classification. He has served as an Associate Editor for IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, and a Senior Editor of Journal of Electronic Imaging.

Mingli Ding received the B.S., M.S. and Ph.D. degrees in instrument science and technology from Harbin Institute of Technology (HIT), Harbin, China, in 1996, 1997 and 2001, respectively. He worked as a visiting scholar in France from 2009 to 2010. Currently, he is a professor in the School of Instrumentation Science and Engineering at Harbin Institute of Technology. Prof. Ding's research interests are intelligence tests and information processing, automation test technology, computer vision, and machine learning. He has published over 40 papers in peer-reviewed journals and conferences.