Pixel Distillation: Cost-flexible Distillation across Image Sizes and Heterogeneous Networks

Guangyu Guo, Dingwen Zhang, *Member, IEEE* Longfei Han, Nian Liu, Ming-Ming Cheng, *Senior Member, IEEE* and Junwei Han, *Fellow, IEEE*

Abstract—Previous knowledge distillation (KD) methods mostly focus on compressing network architectures, which is not thorough enough in deployment as some costs like transmission bandwidth and imaging equipment are related to the image size. Therefore, we propose Pixel Distillation that extends knowledge distillation into the input level while simultaneously breaking architecture constraints. Such a scheme can achieve flexible cost control for deployment, as it allows the system to adjust both network architecture and image quality according to the overall requirement of resources. Specifically, we first propose an input spatial representation distillation (ISRD) mechanism to transfer spatial knowledge from large images to student's input module, which can facilitate stable knowledge transfer between CNN and ViT. Then, a Teacher-Assistant-Student (TAS) framework is further established to disentangle pixel distillation into the model compression stage and input compression stage, which significantly reduces the overall complexity of pixel distillation and the difficulty of distilling intermediate knowledge. Finally, we adapt pixel distillation to object detection via an aligned feature for preservation (AFP) strategy for TAS, which aligns output dimensions of detectors at each stage by manipulating features and anchors of the assistant. Comprehensive experiments on image classification and object detection demonstrate the effectiveness of our method. Code is available at https://github.com/gyguo/PixelDistillation.

Index Terms—Knowledge distillation, pixel distillation, cost-flexible, image size, teacher-assistant-student.

1 Introduction

Recently, great success has been made in the computer vision community based on the rapid development of CNNs [1], [2], [3], [4], [5], ViTs [6], [7], [8] and foundation models [9], [10], [11]. While these models have been able to achieve very promising performance on high-performance computing devices, it is hard to equip them on edge devices like smartphones, embedded devices, small-size UAVs, *etc*. This is because these approaches are usually designed with complex network architectures and large-scale network parameters, while some edge devices require lower transmission bandwidth and computing resources.

To deal with this situation, KD techniques that aim at using smaller network architectures received great attention in the past few years—usually with fewer network layers or smaller channel dimensions, thus reducing the requirement in computation. However, besides the internal network architecture, the external factor, *i.e.*, the input size, reminds us that the existing research is not sufficient. As we illustrated in Fig. 1a, besides the computational complexity and running memory, the input size also matters for the costs of transmission and imaging equipment. For example, if the width and height of an image become K times smaller, the network would only require approximately $1/K^2$ of the original computational complexity and running memory. Moreover, for the cases where the computation is completed on the remote server, the transmission cost will also be

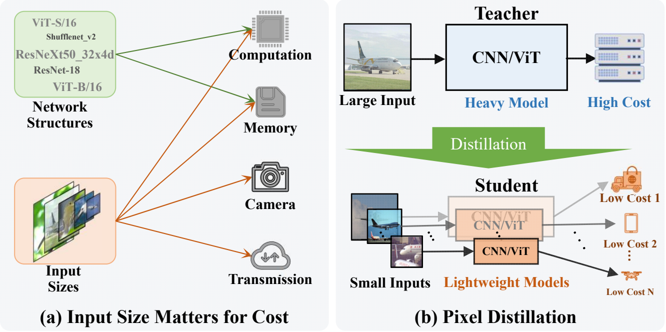
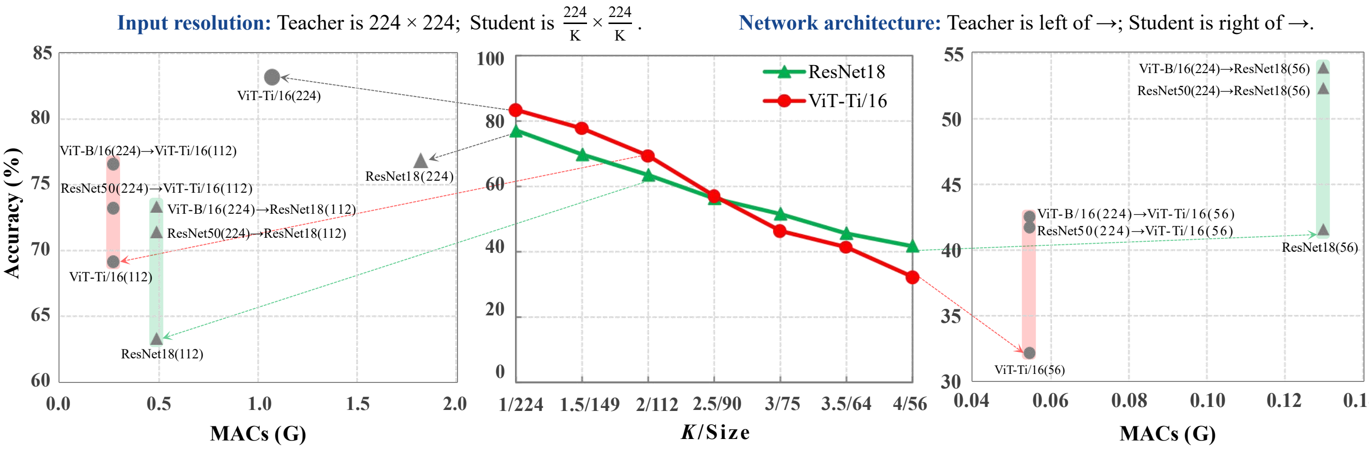


Fig. 1: (a) Compared to network architecture, input size has an impact on more kinds of costs, including requirements for cameras and transmission bandwidth. (b) Pixel distillation can provide more flexible cost control schemes for deployment by distilling knowledge across different input sizes and heterogeneous networks.

- *This work is partially supported by National Science and Technology Major Project (2022ZD0119004), National Natural Science Foundation of China (No.62202015, 62293543, 62322605, U21B2048), Anhui Provincial Key R&D Programmes (2023s07020001), the University Synergy Innovation Program of Anhui Province (GXXT-2022-052).*

- *Guangyu Guo is with Brain and Artificial Intelligence Laboratory, School of Automation, Northwestern Polytechnical University, Xi'an, China. Email: gyguo95@gmail.com*
- *Dingwen Zhang is with the Brain and Artificial Intelligence Laboratory, School of Automation, Northwestern Polytechnical University, Xi'an, China, and also with Xijing Hospital, The Fourth Military Medical University, Xi'an, China. Email: zhangdingwen2006yyy@gmail.com*
- *Longfei Han is with Beijing Technology And Business University, Beijing, China. Email: longfeihan@btbu.edu.cn*
- *Nian Liu is with the Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, United Arab. Emirates. E-mail: liunian228@gmail.com*
- *Ming-Ming Cheng is with TKLNDST, College of Computer Science, Nankai University, Tianjin, China. Email:cmm@nankai.edu.cn*
- *Junwei Han is with the Hefei Comprehensive National Science Center, Institute of Artificial Intelligence, Hefei, China. Email: junweihan2010@gmail.com*
- *Corresponding author: Dingwen Zhang, Junwei Han.*



(a) Pixel Distillation results when *K*=2 **(b) Student on different resolution (*K*=1:0.5:4)** **(c) Pixel Distillation results when *K*=4**

Fig. 2: Observations about pixel distillation. We report the accuracy (%) of baseline ResNet18 and ViT-Ti/16 under different input sizes in figure b, and the accuracy (%) *vs.* MACs (G) of our pixel distillation method when the input size is 112×112 (figure a) and 56×56 (figure c), respectively. **(b)** reports the performance of two student networks under seven input resolutions. **(a)** and **(c)** report the performance of our pixel distillation method under two input resolution settings, *i.e.*, $K = 2$ and $K = 4$. The arrows drawn from (b) represent the baseline performance without knowledge distillation.

greatly reduced by using a small input size. Meanwhile, in many real-world applications like some embedded systems, the devices might be only equipped with low-resolution cameras to keep a low cost of equipment. In conclusion, there is an urgent demand to enable well-trained deep models to fit on small images.

To solve these problems, we propose a new distillation framework, called pixel distillation, to achieve the best trade-off between the performance and the cost including the computational complexity, storage, and transmission. As shown in Fig. 1b, Pixel distillation generalizes the idea of knowledge distillation to the input level, where a large input is used by a heavy teacher model and a small size is used by a lightweight student model. A pixel distillation method should satisfy two criteria. The first is the **input adaptability criterion:** *the method can adapt to different small sizes for the input of students to guarantee flexibility of cost control schemes.* The second is the **architecture adaptability criterion:** *the method should be available to various architectures of the teacher and student to obtain better performance, including the case where the teacher and student belong to CNNs and ViTs, respectively.* The necessity of the architecture criterion comes from the phenomenon that different networks have varying adaptability to changes in the input size. As shown in Fig. 2b, ViT-Ti/16 [6] performs better when the input size is large, but its performance decreases faster than ResNet18 [2] when the input size becomes smaller. When the small size is set as 112×112, student ViT-Ti/16 can obtain better performance than ResNet18 with less computational complexity (Fig. 2a). However, when the input size is reduced to 56×56, student ResNet18 outperforms ViT-Ti/16 by 10% in terms of accuracy (Fig. 2c). Hence, a pixel distillation method should be adaptive to various network architectures to obtain students with better performance.

In this paper, we first propose a baseline for the proposed pixel distillation scheme for the image classification task, called vanilla PD, which satisfies the aforementioned two criteria (see Fig. 3). To be specific, on the assumption that the small input images in pixel distillation would lead to inadequate spatial information on the shallow features of the student, we propose an Input Spatial Representation Distillation (ISRD) mechanism to distill knowledge from the large input to help the input module of the student obtain richer representation. As illustrated in Fig. 3b, considering that our student network includes not only CNNs but also ViTs, we design a Generalized Spatial Feature Preprocess (GSFP) module as the encoder to transfer input spatial features from CNNs and ViTs into the same form. Moreover, as illustrated in Fig. 3c, the decoder of the ISRD can convert the encoded features of arbitrary volume into a pseudo large image, which makes the ISRD mechanism able to be used under any input size of the student. By combining ISRD with the previous prediction distillation methods, we can obtain a simple one-stage trained baseline method for pixel distillation, *i.e.*, the vanilla PD, which satisfies two necessary criteria.

Although the vanilla PD is able to achieve sufficiently good results, we hope that feature distillation can be used to further enhance the performance. Moreover, the gap between the teacher and student in pixel distillation not only comes from different model architectures but also varying input resolutions, which is larger than that in traditional knowledge distillation, distilling useful knowledge for the student will be more difficult [12], [13]. Therefore, we propose a two-stage distillation strategy where an assistant network is introduced into the classical Teacher-Student (TS) framework. As shown in Fig. 5, the proposed Teacher-Assistant-Student framework separates the pixel distillation process into the model compression stage and input compression stage. At each stage, it will be easier for the student to mimic the teacher than that in the one-stage distillation mechanism. Moreover, since the teacher and student have the same architecture in the input compression stage, TAS makes it easier to design a feature distillation mechanism to relieve the performance degradation caused by the small input. Finally, when we adapt the concept of pixel distillation to object detection, we observe that the variability in image resolution among object detectors correspondingly affects their output dimensions like the number of anchor boxes, complicating the preservation of knowledge from the teacher's detection head. To address this, we propose a strategy termed Aligned Feature for Preservation (AFP) for the assistant network. This strategy involves integrating an upsampling operation to match the feature dimension of the assistant network with those of the teacher during the first knowledge distillation stage. Subsequently, during knowledge transfer from assistant to student, we remove the upsampling step since both networks handle inputs of the same resolution. TAS can make the student network effectively leverage knowledge from both the heavy model and large input, and make it flexible to be utilized for more complex tasks.

One of the goals of our research is to evaluate the performance of our proposed method in realistic settings. Therefore, we choose three widely used datasets that reflect different challenges and characteristics of image classification tasks. The first two datasets are CUB-200-2011 [14] and FGCV-aircraft [15], which contain fine-grained categories of birds and aircraft respectively. The third dataset is ImageNet [16], which is a large-scale dataset with 1000 classes and millions of images. We conduct extensive experiments on these benchmarks to demonstrate the effectiveness and robustness of our method. Also, to evaluate the pixel distillation paradigm in a more complex task, *i.e.*, object detection, experiments are conducted on Pascal VOC [17] and MS-COCO 2017 dataset [18]. As shown in Fig. 2, on the CUB-200-2011 dataset, our proposed method can make ViT-Ti/16 with 112 × 112 input achieve comparable performance with ResNet18 that uses input images of size 224 × 224 (76.74% vs. 76.89%), while only 13% computational complexity is required (0.273G MACs vs. 1.82G MACs), and only 25% of the storage and transmission costs are needed.

To summarize, the contribution of this paper is fourfold:

- We present a new distillation scheme called pixel distillation, which provides an early attempt to establish a flexible KD scheme for edge devices with small input sizes.
- We present an input spatial representation distillation mechanism that adapts to input images of different small sizes and can be applied to common network architectures such as CNNs and ViTs.
- We propose a Teacher-Assistant-Student distillation framework that reduces the learning difficulty of the student in pixel distillation and enables feature distillation when the input size of the student is reduced.
- We adapt pixel distillation to object detection, crafting an aligned feature preservation strategy for the assistant network to tackle the challenge of inconsistent output dimensions due to varied image resolutions.
- Extensive experiments on image classification and object detection demonstrate the effectiveness and efficiency of the proposed distillation scheme.

2 Related Works and Relevance

In this section, we review several categories of the existing commonly used methods that can transfer knowledge from a strong source model to a weak target model. Besides, in Table 1 we provide the difference between our pixel distillation paradigm and with the previous method from aspects of testing efficiency, performance gain for the student, and adaptability to the input size and network architecture.

	FT	PKD	FKD	LR-KD	PD
Testing efficiency	fast	×slow	×slow	fast	√fast
Performance gain	×low	high	high	high	√high
Input adaptability	high	medium	×low	×low	√high
Architecture adaptability	×low	high	medium	×low	√high

TABLE 1: Illustration of the difference between the following tasks: fine-tuning (FT), knowledge distillation including prediction-based knowledge distillation (PKD) and feature-based knowledge distillation (FKD), low-resolution recognition with knowledge distillation (LR-KD), our proposed pixel distillation (PD).

2.1 Knowledge Distillation

Knowledge distillation in image classification: Hinton *et al.* first introduced the notion of knowledge distillation which aims to train a smaller model (*i.e.*, student) via learning from the cumbersome models (*i.e.*, teacher) [19]. The early knowledge distillation methods used the predicted score of the teacher model to guide the training of the student model. An essential way is to regard the predicted logits of the teacher model as the soft target of the student [19], [20], [21], [22]. Park *et al.* transfers mutual relations between different samples rather than the output of individual samples [21]. Yu *et al.* successfully applied knowledge distillation in metric learning [23]. Besides the predicted logits, many works have been proposed to guide the student by the intermediate representations of the teacher model [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37]. The fundamental problem for distilling intermediate representations is that the dimension of the feature map is different between the teacher and student models. Some previous works [27], [38], [39], [40] overcome these obstacles by building an adaptive module between hidden layers of teacher and student. Sergey *et al.* distills intermediate knowledge via matching the attention maps between the teacher and student [41]. Similarity Preserving (SP) [42] unified the dimension at the mini-batch level by matrix operation. Some existing methods re-designed the loss function, including activation transfer loss with boundaries formed by hidden neurons [43], Jacobian [44], instance graph [45]. Mirzadeh et al. [46] introduces a multi-step distillation approach by incorporating multiple assistant networks. However, in their method, the teacher, student, and all assistant networks are of homogeneous architectures, meaning they share a similar structural design or configuration.

Knowledge distillation in object detection: In recent days, knowledge distillation has been applied into more complex tasks, such as object detection [47], [48], [49], [50], [51], [52] and semantic segmentation [53], [54], [55], [56], [57], [58], *etc*. In this paper, we also evaluate our proposed pixel distillation method in object detection. Different from image classification, prediction of object detection contains more complex information. A common way is to distill knowledge from intermediate features of object detector [47], [48]. Knowledge from the Feature Pyramid Networks (FPN) [59] can also be used for distillation [60]. Recently, some works distill knowledge from the detection head of teacher object detector [51], [52], [61], [62], [63], which is more complex and efficient. ScaleKD [64] can distill knowledge between object detectors of different input resolutions, but the teacher and student are of the same architecture. UniKD [63] can transfer the knowledge in heterogeneous teacher-student pairs, but does not take input resolution into consideration.

As we illustrated in Table 1, classical knowledge distillation methods usually use the same input size for teacher and student, the efficiency improvement it can bring is limited to the reduction of model architecture. *However, the use of current distillation approaches is limited when the input size is different between the teacher model and student model*:

1) Prediction-based distillation methods have high input adaptability for the image classification task because the dimension of the output space is constant. However, using prediction-based distillation in object detection tasks is difficult because of changes in input resolution in inconsistent output dimensions, such as the number of proposals or anchors.
2) Although both the teacher and student belong to CNNs, some feature-distillation approaches cannot be directly used when the student uses smaller input [25], [41].
3) Difference in the input level leads to a larger gap between teacher and student, which results in performance degradation for some feature-based distillation methods [31], [42].
4) Due to the structural difference between CNNs and ViTs, most feature-based distillation methods can not be used where the teacher and student belong to CNNs and ViTs, respectively. To be specific, intermediate attention maps from ViT consist of a class token and several patch tokens, where the class token is used to learn class-specific information, and patch tokens are used to learn class-agnostic information. Meanwhile, an intermediate feature from CNN learns semantic-aware information. Therefore, the attention maps from ViT and the intermediate feature from CNN are not in the same form, which makes it difficult to use feature-based distillation methods.

Therefore, a novel distillation method is needed when the compression also occurs at the input level.

2.2 Fine-tuning and Low-resolution Recognition

Fine-tuning. One way to improve small image recognition is to use fine-tuning (FT) strategy, which adapts a model trained on large images to small images. However, fine-tuning has some limitations: it requires the same model architecture for both source and target models, and it usually only provides a small performance boost for the target model.

Low-resolution Recognition. Low-resolution (LR) recognition aims to achieve high performance with only LR input available in the inference process. In recent years, some works introduce knowledge distillation into the LR image recognition (*i.e.,* **LR-KD** in Table 2) [65], [66], [67], [68], [69], [70]. However, most methods do not focus on reducing the cost as their main purpose is to achieve better performance. For example, some of them up-sample the LR images to the same size as the HR images to use the existing distillation algorithm more conveniently [67], [71], while others use the same architecture as the teacher model [68], [69], [70], [72], [73], [74]. Furthermore, some works modify the network architecture by using fewer pooling layers or extra modules [65], [66], which makes it difficult to use large-scale pre-training models and reduces their generalization ability. As a result, these methods only perform well on small-scale databases or simple scenarios like face recognition. Taking the latest work in the field of LR face recognition—FMD [73]—as an example, FMD aims to enhance the performance of students with small input by using a teacher with large input, while maintaining the same architecture for the student. However, the FMD configuration restricts its use to situations where the input size of the teacher and the student are limited to 92×92 and 44×44, respectively. As a result, FMD fails to meet the criteria of either input adaptability or network adaptability.

Previous LR recognition methods, as discussed in [65], [66], [72], [73], are limited in their applicability to specific input size settings or network architectures due to their complex designs. In contrast, our pixel distillation paradigm offers a more flexible approach. By generalizing classical knowledge distillation in the input level, pixel distillation provides more options for deployment and can be adapted to various input sizes and network architectures for both the teacher and student models. This adaptability sets pixel distillation apart from previous LR recognition methods, which suffer from a lack of input and network adaptability.

3 Pixel Distillation

In this section, our goal is to address the pixel-level distillation problem within the image classification and object detection tasks. Initially, we will present a foundational overview of how knowledge distillation is applied to image classification. Then, we propose a straightforward pixel distillation baseline for image classification that we have developed using our novel input spatial representation distillation module. Subsequently, we explore integrating feature-level distillation with pixel distillation, in which we utilize the teacher-assistant-student framework. Finally, we adapt pixel distillation to object detection, and introduce an aligned feature preservation strategy for the assistant network, to tackle the challenge of inconsistent output dimensions caused by varied image resolutions.

3.1 Preliminary of KD in Image Classification

Traditional knowledge distillation approaches train a student model by learning the information from the teacher models. Based on the way to obtain the supervision information, we classify previous works in image classification into two categories: prediction-based methods and feature-based methods.

Prediction-based distillation methods train the student by using the class scores predicted by the teacher. One essential way is to regard the predicted logits of the teacher model as the soft target of the student [19], [20]. The loss is:

$$\mathcal{L}_{\mathrm{pkd}}(\mathbf{y},\mathbf{x}_{\mathrm{t}},\mathbf{x}_{\mathrm{s}}) = (1-\alpha)\mathcal{L}_{\mathrm{cls}}(\mathbf{y},\mathbf{x}_{\mathrm{s}}) + \alpha T^2\mathcal{L}_{\mathrm{kl}}(\mathbf{p}_{\mathrm{t}},\mathbf{p}_{\mathrm{s}}), \quad (1)$$

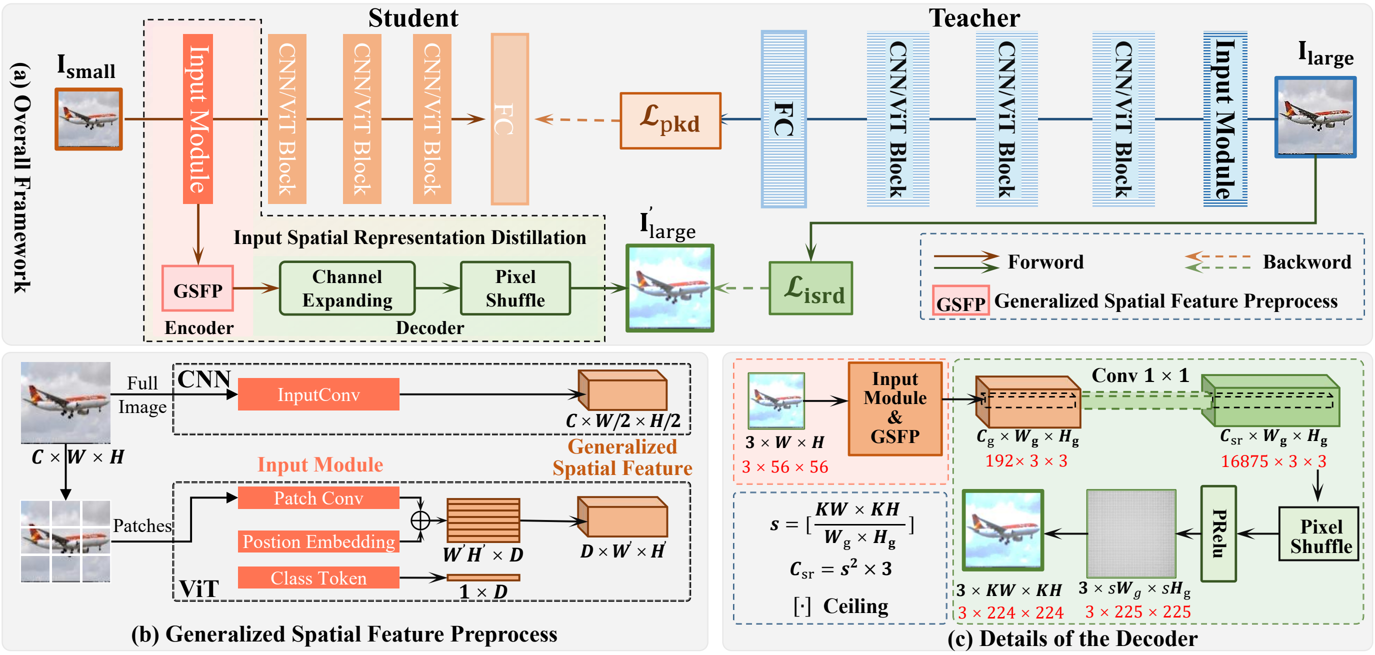


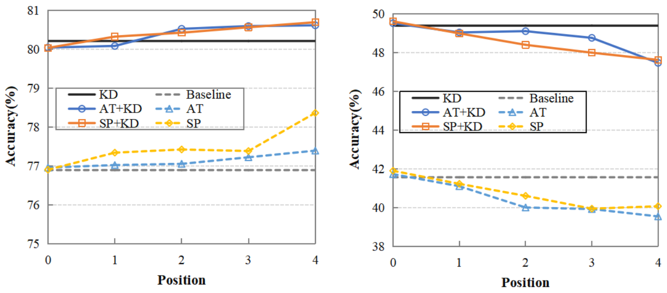
Fig. 3: Illustrations of the vanilla PD. **(a)** vanilla PD consists of a prediction distillation with an input spatial representation distillation (ISRD). ISRD aims to distill spatial information from the large images to train the input module of the student model. **(b)** The generalized spatial feature preprocess (GSFP) can transfer spatial features from CNNs and ViTs into the same form. **(c)** Details of the decoder of the ISRD, red text is an example when the backbone is ViT-Ti/16 and the input size of the student is 56 × 56.

where $\mathbf{p}_\mathrm{t} = \mathrm{softmax}(\frac{\mathbf{x}_\mathrm{t}}{T})$ is the class scores predicted by the teacher, $\mathbf{y}$ is the ground truth, $\mathbf{p}_\mathrm{s} = \log(\mathrm{softmax}(\frac{\mathbf{x}_\mathrm{s}}{T}))$, $\mathbf{x}_\mathrm{t}$ and $\mathbf{x}_\mathrm{s}$ are the predicted class scores of the teacher and student model, respectively, T is a temperature parameter, α is a hyperparameter to balance the classification loss L_cls and the Kullback–Leibler divergence loss L_kl.

Feature-based distillation methods guide the student model using intermediate representations from the teacher model. The learning process of these methods can be expressed as follows:

$$\mathcal{L}_\mathrm{fkd}(\mathcal{F}_\mathrm{t}, \mathcal{F}_\mathrm{s}, \mathbf{x}_\mathrm{s}) = \mathcal{L}_\mathrm{cls}(\mathbf{y}, \mathbf{x}_\mathrm{s}) + \beta \sum_{i \in \mathbf{B}} \delta(g_\mathrm{t}(\mathbf{F}_{\mathrm{t},i}), g_\mathrm{s}(\mathbf{F}_{\mathrm{s},i})), \tag{2}$$

where $\mathcal{F}_\mathrm{s} = \{\mathbf{F}_{\mathrm{s},1}, \mathbf{F}_{\mathrm{s},2}, ..., \mathbf{F}_{\mathrm{s},M}\}$ represents the intermediate features of student, while $\mathcal{F}_\mathrm{t} = \{\mathbf{F}_{\mathrm{t},1}, \mathbf{F}_{\mathrm{t},2}, ..., \mathbf{F}_{\mathrm{t},M}\}$ denotes the features of teacher. The variable M represents the number of blocks in the network. $\mathbf{B}$ is the set of selected features, which varies for different methods. $g_\mathrm{t}(\cdot)$ and $g_\mathrm{s}(\cdot)$ are functions to extract information from intermediate features. $\delta(\cdot)$ is the distance metric function. β is a hyperparameter to balance the classification loss and feature distillation loss.



(i) Knowledge distillation

(ii) Pixel distillation

Fig. 4: Study about the distillation position in traditional knowledge distillation (K=1) and our pixel distillation (K=4) in image classification. The teacher is ResNet50 with 224×224 input, and the student is ResNet18 with $\frac{224}{K} \times \frac{224}{K}$ input.

3.2 Building A Simple Baseline

In this paper, our objective is to train the student model with the help of the teacher model, where both the network architectures and the input size are different. The teacher model takes large images and utilizes heavy networks, while the student model takes small images as input and uses a lightweight network. To identify the best cost scheme, a pixel distillation method should adhere to two criteria: Firstly, it should be applicable to various architectures of both the teacher and student models, including different CNNs and ViTs. Secondly, the method should be adaptable to different small input sizes of the student. In this section, we introduce a simple one-stage trained baseline method vanilla PD that satisfies these two criteria.

3.2.1 *Framework of vanilla PD*

As shown in Fig. 3a, the proposed baseline consists of two distillation processes:

1) Following prediction-based distillation methods [19], [22], we use the logits of the teacher as a part of the supervision for the student. Prediction-based distillation methods are unaffected by network architecture and

Algorithm 1 Input Spatial Representation Distillation

```
Input: Input of the student model $\mathbf{I}_{\mathrm{lr}} \in \mathbb{R}^{3\times W\times H}$, Input of the teacher model $\mathbf{I}_{\mathrm{hr}} \in \mathbb{R}^{3\times KW\times KH}$
if stuent is CNN then
    Directly use the input feature of the student model: $\mathbf{F}_{\mathrm{g},0} = \mathbf{F}_{\mathrm{s},0}, \in \mathbb{R}^{C\times\frac{W}{2}\times\frac{H}{2}}$;
else if stuent is ViT then
    Extract patch embedding: $\mathbf{F}_{\mathrm{patch},0} \in \mathbb{R}^{W^{'}H^{'}\times D}$;
    Transform the form of patch embedding, obtain $\mathbf{F}_{\mathrm{g},0} \in \mathbb{R}^{D\times W^{'}\times H^{'}}$;
end if
Suppose the generalized input spatial feature as $\mathbf{F}_{\mathrm{g},0} \in \mathbb{R}^{C_{\mathrm{g}}\times W_{\mathrm{g}}\times H_{\mathrm{g}}}$;
Calculate scale factor: $s = [\frac{KW\times KH}{W_{\mathrm{g}}\times H_{\mathrm{g}}}]$;
Calculate expanded channel number: $C_{\mathrm{sr}} = 3s^2$;
Expand feature by a $1\times 1$ convolutional layer, obtain $\mathbf{F}_{\mathrm{sr},0} \in \mathbb{R}^{C_{\mathrm{sr}}\times W_{\mathrm{g}}\times H_{\mathrm{g}}}$;
Transform the expanded feature into 3 channels by the pixel shuffle operation, obtain $\mathbf{I}_{\mathrm{sr},0} \in \mathbb{R}^{3\times sW_{\mathrm{g}}\times sH_{\mathrm{g}}}$;
Obtain the pseudo large image $\mathbf{I}_{\mathrm{hr}}^{'} \in \mathbb{R}^{3\times KW\times KH}$ by a crop operation;
Output: $\mathbf{I}_{\mathrm{hr}}^{'}$
```

input, so they can naturally satisfy the abovementioned criteria.

2) An Input Spatial Representation Distillation (ISRD) mechanism is proposed to let the input module of the student learn valuable spatial knowledge from the large input of the teacher. ISRD needs to be designed carefully to satisfy the two criteria.

The loss of vanilla PD is defined as:

$$\mathcal{L}_{\text{vanilla}} = \mathcal{L}_{\text{pkd}} + \gamma\mathcal{L}_{\text{isrd}}, \tag{3}$$

where $\mathcal{L}_{\text{pkd}}$ is the loss of the prediction-based distillation, and $\mathcal{L}_{\text{isrd}}$ is the loss of the ISRD. γ is a hyperparameter to balance $\mathcal{L}_{\text{pkd}}$ and $\mathcal{L}_{\text{isrd}}$.

3.2.2 *Input Spatial Representation Distillation*

Based on Fig.4, it can be observed that the knowledge of the input convolution layer is more beneficial than that of the intermediate layers in pixel distillation. This is because the change of input image leads to larger gaps between intermediate features of the student and teacher, which will make it difficult for the student model to imitate the teacher model [12]. Considering the small input in pixel distillation would lead to inadequate information on the shallow features of the student, the proposed ISRD is used just after the input convolutions of CNN or ViT. As shown in Fig. 3a, the ISRD is an autoencoder that takes the input feature of the student as the input and outputs the large image. The encoder of ISRD transforms the spatial information of CNN and ViT into the same form, and the decoder of ISRD predicts the large image by the transformed feature. In this paper, we calculate the l_1 loss between the pseudo large image $\mathbf{I}_{\mathrm{hr}}^{'}$ and real large image $\mathbf{I}_{\mathrm{hr}}$ as the loss for ISRD module. Suppose the input of the student is an image with a width of W pixels and a height of H pixels, and the input of the teacher is an image with a width of KW pixels and a height of KH pixels, the loss of ISRD is defined as:

$$\mathcal{L}_{\text{isrd}} = \frac{1}{3\times KW\times KH}\|\,\mathbf{I}_{\mathrm{hr}}^{'} - \mathbf{I}_{\mathrm{hr}}\,\|_{1}^{1}, \tag{4}$$

we provide the detailed learning process of the ISRD in Algorithm 1.

Encoder of ISRD. The ISRD encoder is composed of a student input convolution layer and a GSFP operation. As we illustrated in Fig. 3b, the GSFP operation is a parameter-free operation, the student input convolution layer is the encoder's only learnable parameter. Both CNN and ViT use a convolution layer to map the input images into feature space, but the form of their input features are very different, so we need to transform the features of CNN and ViT into the same form to achieve a generalized distillation. As shown in Fig. 3b, given a small input $\mathbf{I}_{\mathrm{lr}} \in \mathbb{R}^{3\times W\times H}$, the feature map generated by the input module of CNN usually is $\mathbf{F}_{\mathrm{s},0}, \in \mathbb{R}^{C\times\frac{W}{2}\times\frac{H}{2}}$, where C is the channel number. Different from CNN, ViT splits the input image into patches and its input feature contains a series of patch tokens and one class token. The patch tokens $\mathbf{F}_{\mathrm{patch},0}, \in \mathbb{R}^{N\times D}$ are derived from the summation of patch features and position embeddings, which contain the spatial information of the input, where $N = W^{'}\times H^{'}$ is the number of patches and D is the hidden size of the tokens. In this paper we transform patch tokens into the size of $D\times W^{'}\times H^{'}$ to make it has the same form as the input feature of the CNN. For both CNN and ViT, We regard the generalized input spatial feature as $\mathbf{F}_{\mathrm{g},0} \in \mathbb{R}^{C_{\mathrm{g}}\times W_{\mathrm{g}}\times H_{\mathrm{g}}}$.

Decoder of ISRD. As shown in Fig. 3c, given the generalized input spatial feature $\mathbf{F}_{\mathrm{g}}$, we need to expand its volume the same as the large input $\mathbf{I}_{\mathrm{hr}} \in \mathbb{R}^{3\times KW\times KH}$. In this paper, we use a 1×1 convolutional layer to achieve channel expansion. To be specific, the scale factor s of the spatial size should be $\frac{KW\times KH}{W_{\mathrm{g}}\times H_{\mathrm{g}}}$, but in most cases, s is not an integer, so we use the round-up operation for s before obtaining the expanded channel number:

$$s = [\frac{KW\times KH}{W_{\mathrm{g}}\times H_{\mathrm{g}}}], \quad C_{\mathrm{sr}} = 3s^2, \tag{5}$$

where C_{sr} is the channel number of the extended feature, $[\cdot]$ is the round-up operation, the expanded input spatial feature is $\mathbf{F}_{\mathrm{sr},0} \in \mathbb{R}^{C_{\mathrm{sr}}\times W_{\mathrm{g}}\times H_{\mathrm{g}}}$. Then, we use a pixel shuffle operation [75] to transform the extended feature into a feature map of 3 channels and obtain $\mathbf{I}_{\mathrm{sr},0} \in \mathbb{R}^{3\times sW_{\mathrm{g}}\times sH_{\mathrm{g}}}$. Since we use the round-up operation when calculating the scale factor s, this map may be slightly larger than the large image, so we use a crop operation to obtain the final pseudo large image $\mathbf{I}_{\mathrm{hr}}^{'}$.

3.3 Teacher-Assistant-Student Framework.

In the classical teacher-student framework of knowledge distillation scheme, the teacher and student have the same input size and different network architecture. However, in the pixel distillation scheme, the teacher and student have different input sizes and network architecture, which makes it more difficult for the student to successfully mimic the teacher [12]. To reduce the learning difficulty for the student

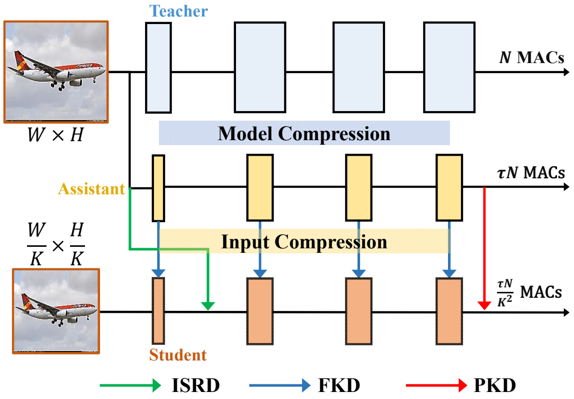


Fig. 5: Illustrations of the Teacher-Assistant-Student (TAS) framework for image classification task. The assistant model splits the pixel distillation into two stages: the model compression stage which reduces the computational cost by a factor of τ $(0 < \tau < 1)$ by using lightweight network architecture, and the input compression stage further reduces the input size and computational cost by a factor of $\frac{1}{K^2}$ by using small input.

in pixel distillation, this paper introduces an assistant network into the classical teacher-student framework to decouple the process of pixel distillation into the model compression stage and input compression stage. As shown in Fig. 5, the assistant network maintains the same large input size as the teacher model and has the same lightweight network architecture as the student. In the model compression stage, the assistant model is regarded as a student model to receive knowledge from the teacher model which has complex architecture. In the input compression stage, the assistant model is regarded as a teacher model to transfer spatial knowledge from the large input into the student model. The whole framework is called the Teacher-Assistant-Student (TAS) framework.

The proposed TAS framework will bring three-fold benefits for pixel distillation:

1) TAS is a two-stage learning framework and the learning difficulty of every single stage is smaller than the one-stage teacher-student framework, which will bring a higher performance gain for the student model.
2) Off-the-shelf knowledge distillation methods can be used in the model compression stage of the TAS framework.
3) In the input compression stage, since the assistant model and student model have the same network architecture, their features are of the same form and number, only the number of channels (CNN) or patches (ViT) is different, which makes it much easier to use feature distillation strategies.

In this paper, we design a simple feature distillation strategy that applies an upsampling operation to features from the assistant model. Suppose $\mathcal{F}_\text{s} = \{\mathbf{F}_{\text{s},1}, \mathbf{F}_{\text{s},2}, ..., \mathbf{F}_{\text{s},M}\}$ is the spatial features of a student model, $\mathcal{F}_\text{a} = \{\mathbf{F}_{\text{a},1}, \mathbf{F}_{\text{a},2}, ..., \mathbf{F}_{\text{a},M}\}$ is the spatial feature of the assistant model, M is the number of blocks in the network, the feature distillation loss of the student in the input compression stage is:

$$\mathcal{L}_{\text{icf}}(\mathcal{F}_\text{a}, \mathcal{F}_\text{s}) = \sum_{i \in \mathbf{B}} \delta(\text{UP}(\mathbf{F}_{\text{s},i}), \mathbf{F}_{\text{a},i}), \tag{6}$$

where $\mathbf{B}$ denotes the set of selected features. "UP" denotes the upsampling operation that is used to make the spatial size of the student feature the same as that of the assistant feature. To be specific, for feature maps from CNN, we directly upsample the feature map in the spatial dimension. For attention maps from ViT, we first expand the spatial dimension into two dimensions, and then apply the upsampling operation. The overall loss of the input compression in the TAS is:

$$\mathcal{L}_{\text{ic}} = \mathcal{L}_{\text{pkd}}(\mathbf{y}, \mathbf{x}_\text{a}, \mathbf{x}_\text{s}) + \gamma \mathcal{L}_{\text{isrd}} + \eta \mathcal{L}_{\text{icf}}(\mathcal{F}_\text{a}, \mathcal{F}_\text{s}), \tag{7}$$

where η is the loss weight for the feature distillation loss.

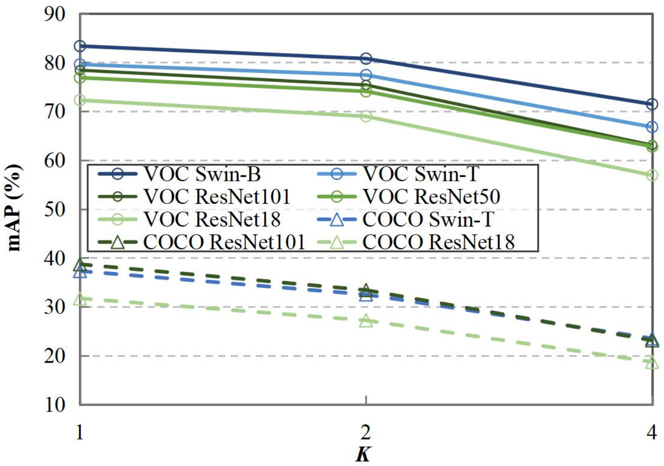


Fig. 6: Analysis of how input resolution affects the performance of object detection. We report the mAP (%) of RetinaNet [76] on the PASCAL VOC and COCO datasets.

3.4 Pixel Distillation in Object Detection

In this paper, we also extend our pixel distillation paradigm into the fields of object detection. We first analyze how the image resolution affects the performance of the object detectors and the distillation process. Then, we design an aligned feature for preservation (AFP) strategy to align the output dimensions of detectors at each stage.

Effect of input resolution: As shown in Fig. 6, we first analyze how the performance of the object detector is affected by the input resolution. On the PASCAL VOC and COCO datasets, We report the mAP (%) of RetinaNet [76] under several backbones. We observe that with a 4x reduction in resolution (K=4), the mAP for all examined models drops by at least 10%, demonstrating that input resolution significantly influences the effectiveness of object detectors. Furthermore, decreasing input resolution not only affects model performance but also alters output characteristics, such as the number of anchor boxes, thereby complicating the process of transferring knowledge from the teacher's detection head during distillation.

Aligned feature for preservation: As aforementioned before, decreasing input resolution will affect the output characteristics such as the number of anchor boxes, which makes

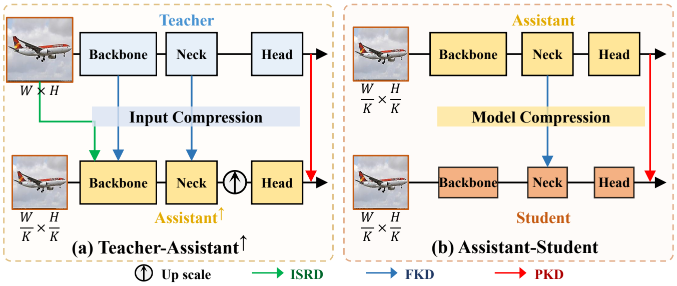


Fig. 7: The overall teacher-assistant-student framework for object detection involves employing the assistant network's features in distinct manners across two stages to align the output dimensions. 'Assistant$^{\uparrow}$' denotes that the features of the assistant are upscaled to match the dimensionality of the teacher's features.

it hard to perform pixel distillation in one stage if we want to preserve knowledge from the prediction head of the teacher. To address this issue, we have refined the two-stage teacher-assistant-student framework in this study. Fig. 7 illustrates that since the assistant network is not utilized during inference, its architecture can be adapted to counteract the effects of resolution reduction on the detector. Specifically, in the first input compression stage, both the teacher and assistant use the same backbone network. The assistant's features are upsampled to match the spatial resolution of the teacher's features, which we refer to as 'Assistant$^{\uparrow}$'. This approach allows the assistant detector to utilize the same anchoring schema as the teacher, facilitating the preservation of knowledge at the prediction level. The associated loss function for this object detection input compression stage is detailed as follows:

$$\mathcal{L}_{\text{od,ic}} = \mathcal{L}_{\text{pkd}} + \mathcal{L}_{\text{fkd}} + \gamma \mathcal{L}_{\text{isrd}} \tag{8}$$

where $\mathcal{L}_{\text{pkd}}$ and $\mathcal{L}_{\text{fkd}}$ denote the loss for the prediction distillation and feature distillation, respectively.

In the model compression phase, which is the second stage, both the teacher and student are provided with the same input, which allows for the removal of the feature upsampling operation within the assistant detector. Consequently, the assistant now adopts the same anchoring schema as the student. This alignment makes it possible to employ standard knowledge distillation techniques.

To sum up, by effectively manipulating the features and anchor configurations of the assistant network, our pixel distillation method is extended to the object detection task. Also, both prediction-based and feature-based distillation techniques can be used in each of the two stages. It is important to note that the input compression stage alters the structure of the assistant network and brings additional computational costs. Therefore, the sequence of the two stages in the Teacher-Assistant-Student (TAS) framework is not interchangeable for object detection tasks. The input compression stage must precede the model compression stage to ensure that the assistant's architecture is correctly configured for each stage.

4 Experiments

4.1 Settings

Datasets. To evaluate the performance of our proposed method in realistic settings, we choose three widely used datasets that reflect different challenges and characteristics of image classification tasks, *i.e.*, CUB (Caltech-UCSD Birds-200-2011) [14], Aircraft (FGCV-aircraft-2013b) [15], and ImageNet [16]. CUB is a fine-grained dataset that consists of 200 categories of birds, there are 5,994 training images and 5,794 testing images. Aircraft contains 100 categories of aircraft and each category has 100 images. The train, validation, and test set have 3,334, 3,333, and 3,333 images, respectively. ImageNet has 1,000 categories, each category contains approximately 1,300 training and 50 validation images per category. In total, it contains 129,395 and 5,000 images for train and validation, respectively. Besides, to evaluate our proposed method on the object detection task, we select two widely used benchmarks, *i.e.*, PASCAL VOC [17] and COCO 2017 dataset [18]. For PASCAL VOC, we use 5,000 trainval images in VOC2007 and 16,000 trainval images in VOC2012 for training, and 5,000 test images in VOC 2007 for evaluation. For COCO, we use 115,000 trainval135k images for training, and 5,000 minival set as validation.

Metrics. To evaluate the classification performance, we use Top-1 accuracy. For object detection, we report mean Average Precision (AP) as an evaluation metric. We also provide several metrics to evaluate how the input size and architecture complexity affect the cost of the student model. For input-related costs, we provide the storage and transmission (Stg/Trans.) for one image. For architecture-related costs, we provide the number of parameters (Param.). Moreover, we use the number of multiply–accumulate operations (MACs) to measure the computational complexity of the model, which is related to both the input size and architecture complexity.

We also propose a metric to evaluate the reduction of storage and transmission cost (Stg/Trans Red.), which is calculated as:

$$\text{Stg/Trans Red.} = 1 - \frac{\text{Stg/Trans. (Student)}}{\text{Stg/Trans. (Teacher)}}. \tag{9}$$

another metric is proposed to evaluate the reduction of computational complexity (Comput Red.), which is calculated as:

$$\text{Comput Red.} = 1 - \frac{\text{MACs (Student)}}{\text{MACs (Teacher)}}. \tag{10}$$

Implementation details. For the image classification task, we use mini-batch stochastic gradient descent (SGD) as the optimizer, and the momentum and the weight decay are set as 0.9 and 0.0005, respectively. On CUB and Aircraft, we set the learning rate as 0.01, 0.02, and 0.001 for ResNet, ShuffleNetV2, and ViT, respectively. We train the model by 120 epochs with batch size 64, the learning rate is reduced by a factor of 10 after every 30 epochs. On ImageNet, we set the learning rate as 0.001 and keep the remaining setting the same as CUB and Aircraft. The value of γ is determined by the architecture of the student network we analyze it in Section 4.4. The value of η is set as 10. For the object detection task, all models are implemented under the MMDetection [77] toolkit. Our code is implemented on the

TABLE 2: The configuration of teacher and student models for six experiment settings. We use different colors to present settings related **only to input** (Stg/Trans., Stg/Trans Red.), **only to architecture** (Param.), and **to both input and architecture** (MACs, Compute Red.).

					(a)	**(b)**	**(c)**	**(d)**	**(e)**	**(f)**
Teacher	Architecture→ Input↓			Network	ResNeXt50	ResNet50	ResNet18	ResNet50	ViT-B/16	ViT-B/16
				Param. (M)	25.03	25.56	11.69	25.56	86.57	86.57
	Size 224^2	Stg/Trans.(KB)	147	MACs (G)	4.260	4.110	1.820	4.110	16.850	16.850
Student	Architecture→ Input↓			Network	ResNet34	ResNet18	ShuffleNetV2 1.0	ViT-Ti/16	ResNet18	ViT-Ti/16
				Param. (M)	21.80	11.69	2.28	5.69	11.69	5.69
	K=2 Size 112^2	Stg/Trans.(KB)	36.75	MACs (G)	0.967	0.487	0.041	0.273	0.487	0.273
		Stg/Trans Red.	75.00%	Comput Red.	77.29%	88.16%	97.72%	93.36%	97.11%	98.38%
	K=4 Size 56^2	Stg/Trans.(KB)	9.1875	MACs (G)	0.268	0.130	0.012	0.055	0.130	0.055
		Stg/Trans Red.	93.75%	Comput Red.	93.71%	96.84%	99.33%	98.67%	99.23%	99.68%

TABLE 3: Results on the CUB dataset for setting (a) to (c). The best is shown in bold. Each experiment is repeated five times and we report the mean value.

	(a)		(b)		(c)	
	$K=2$	$K=4$	$K=2$	$K=4$	$K=2$	$K=4$
Teacher	81.84		80.46		76.89	
Baseline-FS	37.54	23.73	37.16	24.52	35.67	22.36
Baseline-FT	65.41	43.68	63.31	41.56	63.53	43.05
KD	70.40	49.32	69.26	49.38	65.69	44.09
AT	61.84	4.63	58.53	4.55	43.61	6.97
AT+KD	65.25	4.64	62.74	7.20	47.28	9.00
SP	68.35	42.31	65.96	40.07	64.35	40.45
SP+KD	70.41	48.01	69.39	47.61	65.69	42.54
IC	67.36	44.53	66.03	46.02	62.11	42.90
IC+KD	69.17	49.72	69.81	48.30	66.17	44.07
DKD	70.08	47.01	68.77	48.78	66.41	45.15
vanilla PD (One-Stage)	70.86	50.48	69.94	50.34	66.32	44.69
TAS (Two-Stage)	**72.59**	**51.52**	**71.65**	**52.44**	**67.67**	**45.93**

TABLE 4: Results on the CUB dataset for setting (d) to (e). The best is shown in bold. Each experiment is repeated five times and we report the mean value.

	(d)		(e)		(f)	
	$K=2$	$K=4$	$K=2$	$K=4$	$K=2$	$K=4$
Teacher	80.46		88.13		88.13	
Baseline-FS	9.49	6.55	37.16	24.52	9.49	6.55
Baseline-FT	69.14	32.14	63.31	41.56	69.14	32.14
KD	71.67	39.05	71.31	51.28	74.26	40.02
DKD	70.94	38.29	70.66	49.09	73.41	40.92
vanilla PD (One-Stage)	72.41	40.06	71.90	51.77	74.67	40.58
TAS (Two-Stage)	**73.46**	**41.93**	**73.38**	**54.17**	**76.74**	**42.46**

basis of PyTorch [78], and all experiments are carried out on an NVIDIA GeForce 3090 GPU.

4.2 Experimental Results of Image Classification

In this section, we compared our method with previous knowledge distillation methods including prediction distillation methods KD [19], DKD [22], feature distillation methods AT [41], SP [42], IC [31]. AT [41] is selected as the representative of methods requiring the same spatial size for features from teacher and student, we use average pooling to align feature maps following [79]. SP [42] is the representative of methods that do not require the same spatial size for teacher and student features, which can be directly used to resolve the pixel distillation problem. IC [31] is the representative of methods that contains an extra adaptive module to align features between teacher and student. We also provide the performance of the combination of each feature distillation method with the prediction distillation method [19].

Teacher-student pairs. In Table 2, we provide the detailed setting of our image classification task, which contains six teacher-student pairs with two down-sampling rate, including the model size (Params), computational complexity (MACs), and efficient ratio (Compute Red.). We use variants of ResNet [2], ShuffleNetV2 [80] and ViT [6] as the teacher and student models. In settings (a) to (c), both teacher and student belong to CNN. For settings (d) and (e), one of the teachers or students is CNN and another is ViT. For setting (f), both teacher and student models belong to ViT. With regard to the input, the spatial size of the large input is 224×224, while the spatial size of the small input is $\frac{224}{K} \times \frac{224}{K}$, we report the details of two down-sampling rates in Table 2, *i.e.*$K = 2$ and $K = 4$. Moreover, we report the performance of more down-sampling rates in Fig. 8.

Experiments on CUB and Aircraft. From Table 3 to 6, we report the performance for all settings on two fine-grained datasets. All the experiments are repeated five times and we report the mean value to avoid the influence of randomness on experimental results. We compared our methods with all aforementioned knowledge distillation methods for setting (a) to (c) where both teacher and student belong to CNN, and only prediction-based distillation methods are compared for setting (d) to (f) as ViT is involved. "Baseline-FS" denotes the student trained from scratch, and "Baseline-FT" denotes the student trained from pre-trained weights on ImageNet. Our proposed methods use KD as the prediction distillation method. From the results, we have the following observations:

1) Our one-stage trained baseline vanilla PD can stably provide performance gains over KD on all settings, whether the student model belongs to CNN or ViT. Moreover, in most settings, our one-stage trained vanilla PD outperforms the knowledge distillation methods, which demonstrates the effectiveness of our proposed vanilla PD.
2) The two-stage trained TAS framework enhances the performance of the baseline vanilla PD across all settings, providing additional performance gains for the student. This is because TAS reduces the learning difficulty of the student and introduces the feature distillation mechanism to relieve the performance degradation caused by the small input size.
3) Prediction-based distillation can provide stable performance gain for the student on all network architectures and input size settings.

TABLE 5: Results on the Aircraft dataset for setting (a) to (c). The best is shown in bold. Each experiment is repeated five times and we report the mean value.

	(a)		(b)		(c)	
	$K=2$	$K=4$	$K=2$	$K=4$	$K=2$	$K=4$
Teacher	86.26		85.30		79.96	
Baseline-FS	51.52	32.37	50.31	33.52	45.78	26.73
Baseline-FT	70.79	47.11	68.38	49.12	63.69	45.21
KD	73.65	53.60	73.70	55.14	66.30	46.92
AT	70.08	7.73	68.36	9.29	43.43	8.08
AT+KD	74.01	17.77	72.95	18.21	58.19	15.81
SP	71.69	41.40	70.15	39.68	61.19	10.76
SP+KD	74.43	52.05	73.49	50.20	66.40	37.79
IC	71.81	46.35	70.97	47.01	62.43	42.61
IC+KD	73.92	53.12	73.05	52.85	66.51	44.84
DKD	73.75	52.52	72.31	53.62	65.17	45.80
vanilla PD (One-Stage)	74.79	54.51	74.09	56.81	66.91	47.52
TAS (Two-Stage)	**76.51**	**56.17**	**75.72**	**58.31**	**67.84**	**48.67**

TABLE 6: Results on the Aircraft dataset for setting (d) to (e). The best is shown in bold. Each experiment is repeated five times and we report the mean value.

	(d)		(e)		(f)	
	$K=2$	$K=4$	$K=2$	$K=4$	$K=2$	$K=4$
Teacher	85.30		79.60		79.60	
Baseline-FS	9.38	5.71	50.31	33.52	9.38	5.71
Baseline-FT	57.35	31.89	68.38	49.12	57.35	31.89
KD	65.45	40.02	70.60	55.46	66.44	42.86
DKD	64.31	40.21	69.68	54.91	66.85	42.12
vanilla PD (One-Stage)	66.41	40.89	71.66	55.61	67.68	43.22
TAS (Two-Stage)	**67.78**	**41.88**	**72.74**	**57.54**	**68.85**	**44.65**

4) The performance of feature distillation is very unstable: SP+KD and IC+KD can only provide a light performance gain over KD on some settings when K=2, and all three feature distillation methods perform badly when the input size is too low (*i.e.*, K=4). This is because the teacher and student in pixel distillation have a larger gap than in knowledge distillation, which will make it difficult for the student to successfully mimic the teacher [12].
5) The performance of different models varies greatly with the change in input size and dataset. For instance, when K=4, student ViT-Ti/16 trained by ViT-B/16 (setting (f)) obtains the best performance on CUB (54.17%), but student ResNet18 trained by ResNet50 is the best on the Aircraft dataset.

Experiments on More Input Sizes. As shown in Fig. 8, we conduct experiments on more small input sizes to demonstrate the generalization ability of our methods on the input size. The teacher is ResNet50 with 224 × 224 input and the student is ResNet18. K is set from 1 to 4 with stride 0.5. We can observe that both the one-stage vanilla PD and two-stage TAS can obtain performance gains on all sizes.

Experiments on Imagenet. In Table 7 we report the performance on the ImageNet dataset. All models are trained from scratch. We compared our methods with previous knowledge distillation methods for setting (b), *i.e.*, the teacher is ResNet50 with input size 224 × 224 and the student is ResNet18 with input size 112 × 112 (K=2) and 56 × 56 (K=4). We report our performance based on two prediction distillation methods KD and DKD. For KD based method, the one-stage trained vanilla PD can bring 0.52% and 0.46% performance gains when K is 2 and 4, respectively. Also, when the prediction distillation method is DKD, vanilla PD can bring 0.34% and 0.56% performance gains when K is 2 and 4, respectively. Using two-stage trained TAS with a prediction distillation method DKD achieves the best performance, which outperforms the baseline student by 2.32% and 3.02%.

TABLE 7: Results on and ImageNet dataset for setting (b). Teacher is **ResNet50** and student is **ResNet18**.

Teacher	80.37	
	$K=2$	$K=4$
Student	62.04	50.81
KD	62.70	51.41
AT	57.77	30.34
AT+KD	58.18	35.34
SP	62.17	50.87
SP+KD	62.34	51.49
IC	62.28	51.16
IC+KD	62.59	51.70
DKD	63.56	52.28
vanilla PD (KD)	63.22	51.87
vanilla PD (DKD)	63.90	52.84
TAS (KD)	63.71	52.44
TAS (DKD)	**64.36**	**53.83**

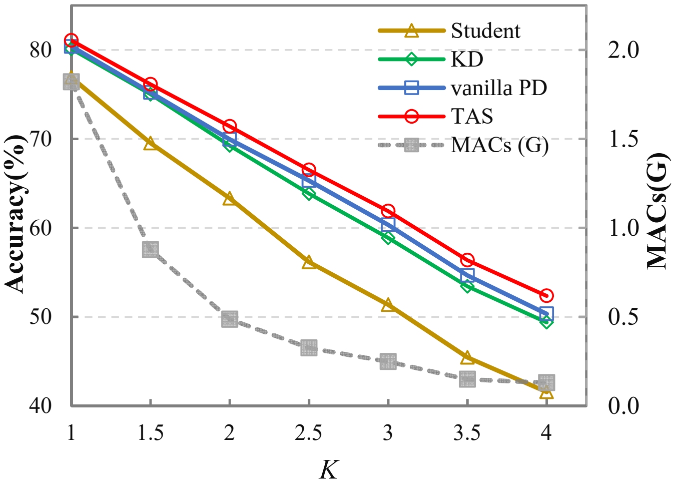


Fig. 8: Accuracy (%) and MACs(G) of more input sizes on the setting (b) of CUB datasets.

Comparision with super-resolution method. At first glance, an intuitive solution to the challenge of recognizing low-resolution images involves upscaling the input to a higher resolution, and subsequently processing these upscaled inputs with a model that has been trained on large inputs to make predictions. With this perspective, we introduce two baseline models in Table 8 denoted as experiment (ii). To be specific, in experiments **i** of Table 8, we provide the performance of the teacher network and the student network trained by HR images (Baseline-HR). Then, experiment (ii) represents the upscaling paradigms: pre-trained Baseline-HR models are used in inference, the LR inputs of students are upsampled into HR images via bilinear interpolation (Baseline-Bilinear) and the super-resolution model SwinIR-M [81], respectively. In experiments (iii), we provide the performance of students trained by LR images (Baseline-LR) and our methods. We can observe that the upscaling paradigm in experiments (ii) can outperform models with LR input in most cases, but it has two drawbacks that make it fundamentally different from pixel distillation: **1)** it needs more computational complexity that is caused by the HR

TABLE 8: Comparision with super-resolution method on CUB dataset with settings (b) and (f).

id	Method	(b) K=2	(b) K=4	(f) K=2	(f) K=4
i	Teacher Baseline-HR	80.46 76.89		88.13 83.19	
ii	Baseline-Bilinear Baseline-SwinIR	70.92 74.73	62.45 67.14	80.87 82.06	75.91 78.75
iii	Baseline-LR vanilla PD (One-Stage) TAS (Two-Stage)	63.31 69.94 71.65	41.56 50.34 52.44	69.14 74.67 76.74	32.14 40.58 42.46

TABLE 9: The configuration of teacher and student models for four experiment settings of object detection. We report model size (Params), computational complexity (FLOPs), and computational complexity reduction (Compute Red.)

		(g)	**(h)**	**(i)**	**(j)**
Teacher	Backbone Param. (M)	ResNet50 36.7	Swin-T 38.5	ResNet101 55.7	Swin-B 98.4
	FLOPs (G)	107	125	145	245
Student	Backbone Param. (M)	ResNet18 22.1	ResNet18 22.1	Swin-T 38.5	Swin-T 38.5
K=2	FLOPs (G) Compute Red.	20.2 81.2%	20.2 83.9%	34.2 76.4%	34.2 86.0%
K=4	FLOPs (G) Compute Red.	5.9 94.5%	5.9 95.3%	9.4 93.5%	9.4 96.2%

TABLE 10: Results (mAP) on the PASCAL VOC dataset from setting (g) to (j). The best is shown in bold.

CNN-based Student	**(g)**		**(h)**	
Teacher	76.9		79.6	
	K=2	K=4	K=2	K=4
Student	69.0	56.9	69.0	56.9
TAS-AFP (CrossKD)	72.3	60.3	72.6	61.3
TAS-AFP (CrossKD) + Cao *et al.*	72.5	60.3	73.4	61.1
TAS-AFP (CrossKD) + ISRD	**72.9**	**61.1**	**73.9**	**62.9**
Swin-based Student	**(i)**		**(j)**	
Teacher	78.4		83.4	
	K=2	K=4	K=2	K=4
Student	77.4	66.8	77.4	66.8
TAS-AFP (CrossKD)	77.2	67.3	78.3	69.4
TAS-AFP (CrossKD) + Cao *et al.*	76.7	67.6	78.6	68.9
TAS-AFP (CrossKD) + ISRD	**78.0**	**68.7**	**78.9**	**70.5**

TABLE 11: Results (mAP) on the COCO dataset for setting (h) and (i). The best is shown in bold.

	(h)		(i)	
Teacher	37.3		38.7	
	K=2	K=4	K=2	K=4
Student	27.2	18.7	32.5	23.5
TAS-AFP (CrossKD)	30.2	21.1	33.7	25.1
TAS-AFP (CrossKD) + Cao *et al.*	30.4	20.9	35.0	24.6
TAS-AFP (CrossKD) + ISRD	**30.6**	**22.1**	**35.6**	**25.9**

input of the student and the SR operation. Take setting (b) with $K = 2$ as an example, when the input resolution of ResNet-18 increases from 112×112 into 224×224, the MACs will increase from 0.487G into 1.820G, and the SR model SwinIR-M causes another 8.38G costs. **2)** The process of generation and prediction of SR images cannot be calculated in parallel, which means it will inevitably lead to additional calculation time.

4.3 Experimental Results of Object Detection

In this section, we compared our method with previous knowledge distillation methods in object detection, including prediction distillation method CrossKD [62], and the feature distillation methods proposed by Cao *et al.* [60] that distillation from the FPN.

Teacher-student pairs. In Table 9, we provide the detailed setting of the object detection task, which contains four teacher-student pairs with two down-sampling rates, including the model size (Params), computational complexity (FLOPs), and efficient ratio. Floating point operations (FLOPs) are used for computing computational complexity because we use the officially provided tools of MMDetection[1]. We use variants of ResNet [2], Swin transformer [7] as the teacher and student models. In settings (g), both teacher and student belong to ResNet. For settings (h) and (i), one of the teachers or students is ResNet and another is Swin transformer. For setting (j), both teacher and student models belong to Swin transformer. In Table 9, we report all details on the PASCAL VOC dataset, where the spatial size of the large input is 1000×600, and that of the small input is $\frac{1000}{K} \times \frac{600}{K}$, we report the details of two down-sampling rates, *i.e.*$K = 2$ and $K = 4$. For the COCO dataset, the spatial size of the large input is 1333×800. All the models utilize the RetinaNet [76] framework.

Experiments on PASCAL VOC and COCO. Table 10 and 11 report performance on the PASCAL VOC and COCO dataset, respectively. **TAS-AFP (CrossKD)** is a base method that is built by our TAS framework and AFP strategy, and the prediction-based distillation CrossKD [62] is utilized, we can observe that this base model can bring performance for all settings on both datasets. For example, on the setting (h) of PASCAL VOC, it can bring 4.4% mAP improvement when K=4 (56.9 vs. 61.3). Then, **TAS-AFP (CrossKD)+Cao *et al.*** introduce the FPN based feature distillation method [60], this method is not stable when the resolution is reduced. on the setting (i) of COCO with K=2, it can improve the mAP from 33.7% to 35.0%, but it brings performance degradation in most cases when K=4. Finally, **TAS-AFP (CrossKD)+ISRD** indicates we use our ISRD mechanism in the input compression stage, which can bring stable performance improvements under all settings.

4.4 Ablation Studies and Model Analysis

In this section, we provide ablation studies about the key components, the channel expanding layer of ISRD, the architecture of TAS, and hyperparameters.

Ablation study of the key learning components. As shown in Table 12, we conducted our ablation study on the CUB dataset. We select two settings, *i.e.*, setting (b) whose student is CNN, and setting (f) whose student is ViT. The experimental results show that there will be a large performance

[1] https://github.com/open-mmlab/mmdetection/blob/main/tools/analysis_tools/get_flops.py

TABLE 12: Ablation study of each component on setting (b) and (f) in the image classification task, we report performance on the CUB dataset.

One-stage			Two-Stage		(b)		(f)	
Baseline	KD	ISFR	TAS	ICF	$K=2$	$K=4$	$K=2$	$K=4$
√					63.31	41.56	69.14	32.14
√	√				69.26	49.38	74.26	40.02
√	√	√			69.94	50.34	74.67	40.58
√	√	√	√		70.31	51.40	75.87	41.64
√	√	√	√	√	71.65	52.44	76.79	42.46

gain via directly using traditional knowledge distillation (KD) [19] approach, which can improve the accuracy from 41.56% to 49.38% on setting (b) when K=4. Then, ISRD can provide 0.56% to 0.96% performance gain for all settings, by integrating KD and our proposed ISRD we could obtain a one-stage trained simple baseline that can provide stable performance improvement for students. Furthermore, introducing the assistant network to achieve a two-stage trained framework can bring stable performance gains as it can reduce the learning difficulty of the student. Finally, the simple feature distillation in the input compression stage (*i.e.*, ICF) can further reduce the performance degradation caused by the small input size. Such performance gain demonstrates that our proposed learning framework can work effectively with lightweight network architecture and small input sizes.

Ablation study about the kernel size of the channel expanding layer in ISRD. In the decoder of the proposed ISRD, a 1×1 convolution layer is used and expand the volume of the input feature, a natural question about this process is whether using a larger kernel size can help obtain higher performance. As shown in Fig. 9, on setting (b) of the CUB dataset, we conduct experiments about kernel size from 1×1 to 7×7. To avoid the influence of hyperparameter γ, experiments of each kernel size are conducted when γ increases from 10 to 100 with stride 10. We can observe that performance decreases when the kernel size increases from 1×1 to 7×7. The main reason is that the purpose of the ISRD is to transfer knowledge to the input module of the student, *i.e.*, the only one learnable parameter in the encoder of the ISRD, even though using stronger decoder can help the ISRD to obtain better pseudo large images, but the information received by encoders will become weaker and further lead to lower performance gains for the student.

Ablation studies about the architecture of the TAS. TAS separates the pixel distillation into model compression and input compression, these two compression process can exchange their order. In Table 13, we conduct experiments on settings (b) and (f) of CUB to explore the influence of this order. (1) is the traditional teacher-student framework, (2) and (3) are the proposed TAS framework. In experiment (2) we first perform the input compression process (vanilla PD) and then use the prediction distillation method KD to perform the model compression process. In experiment (3) we exchange the order of input compression and model compression. We can observe that the performance is better when we perform the model compression first, this is because input resolution has a greater impact on the performance. Hence, the performance of the assistant is too low if input compression is performed first.

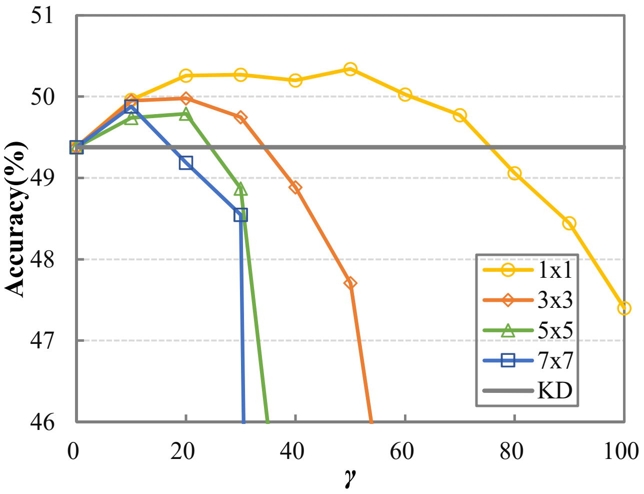


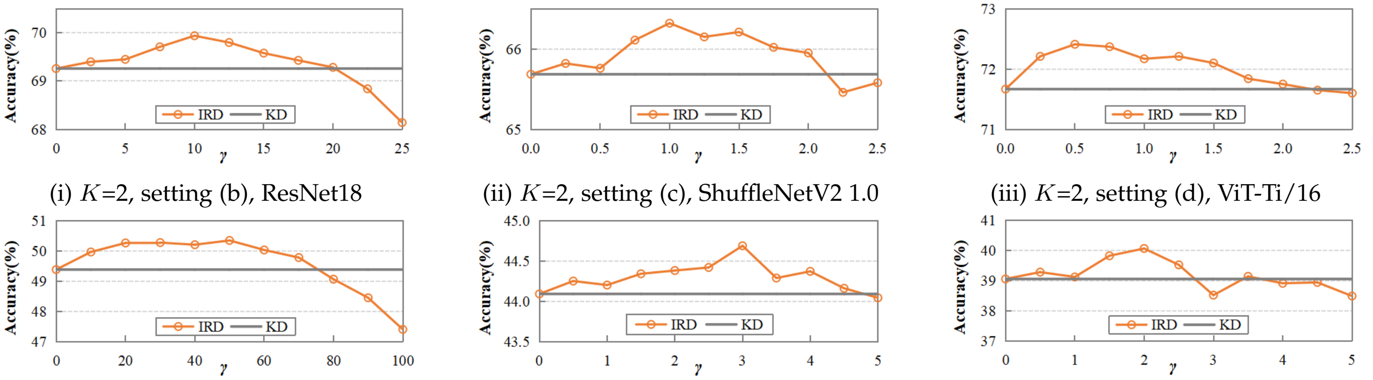
Fig. 9: Ablation study about the kernel size of the channel expanding layer in ISRD.

TABLE 13: Ablation study about the TAS framework in image classification. We provide results on the CUB dataset with settings (b) and (f).

	Method	(b)		(f)	
		$K=2$	$K=4$	$K=2$	$K=4$
(1)	vanilla PD (TS)	69.94	50.34	74.67	40.58
(2)	vanilla PD→KD (TAS)	69.74	49.74	75.21	40.861
(3)	KD→vanilla PD (TAS)	70.31	51.40	75.87	41.636

Ablation studies about hyperparameter γ. γ is the only hypermeter for the one-stage trained vanilla PD. As shown in Fig. 10, we conduct experiments about γ on setting (b) to (d) of the CUB dataset, their student is ResNet18, ShuffleNetV2 and ViT-Ti/16, respectively. We find the choice of γ is high relative to two factors, the kernel size of the input convolution layer and the volume of the input feature:

1) The kernel size of the input convolution layer determines how many parameters can be used to learn knowledge from the large images. When the kernel size of the input feature is large, we can set a large γ to obtain a better performance, and vice versa. For example, as illustrated in Fig. 10i and Fig. 10ii, the kernel size of the input convolution layer of the ResNet18 is $3 \times 7 \times 7 \times 64$, and the input convolution layer of the ShuffleNetV2 1.0 is $3 \times 3 \times 3 \times 24$. When the input size for them is same (112×112), the best γ for ResNet18 is 10 and the best γ for ShuffleNetV2 1.0 is 1.0.
2) The volume of the input feature determines how well the quality of the pseudo large images. When the kernel size and the volume of the input feature are large, we can use a large γ to obtain a better performance, and vice versa. One factor that affects the volume of the input feature is the network architecture: the input feature's volume of CNN is usually much larger than that of the ViT. For example, as shown in Fig. 10i and Fig. 10iii, the input feature's volume of the ResNet18 and ViT-Ti/16 is $28 \times 28 \times 64$ and 49×192, respectively. When both the input size (112×112) and teacher model (ResNet50) are the same for them, the best γ for ResNet18 is 10 and the best γ for ViT-Ti/16 is 0.5.



(i) K=2, setting (b), ResNet18

(ii) K=2, setting (c), ShuffleNetV2 1.0

(iii) K=2, setting (d), ViT-Ti/16

(iv) K=4, setting (b), student ResNet18

(v) K=4, setting (c), ShuffleNetV2 1.0

(vi) K=4, setting (d), ViT-Ti/16

Fig. 10: Ablation study about the hyperparameter γ on the CUB dataset under different teacher-student pairs and input sizes. For all experiments, the input size of the teacher and student is 224×224 and $\frac{224}{K} \times \frac{224}{K}$, respectively, In the caption of each figure, we provide **the value of K, the index of setting, and the architecture of the student** from left to right.

Another factor that affects the volume of the input feature is the input size, we can observe that the best γ in the first line of Fig. 10 is smaller than that in the second line, this is because the input size of the first line is larger ($K = 2$ *vs.* $K = 4$).

5 Conclusion

In this paper, we propose a novel pixel distillation that aims to distill knowledge from a teacher model with heavy architecture and large input size to student models that have a variety of lightweight network architectures and input of different small sizes, which can provide more flexible cost control schemes than traditional knowledge distillation scheme. We first provide a simple one-stag trained baseline for the classification task named vanilla PD, which can be adapted to sizes and different networks including CNN and ViT. Specifically, vanilla PD consists of a prediction-based distillation mechanism and a novel proposed input spatial representation distillation (ISRD) mechanism. ISRD can relieve the performance degradation due to the small input size by transferring information from the large inputs. Then we propose a teacher-assistant-student (TAS) framework to reduce the learning difficulty of students caused by the large gap between the teacher and student. TAS can also make it easier to relieve the performance degradation caused by small images by distilling knowledge in intermediate features. Experimental results demonstrate that the proposed method can improve the performance of models with various compact network architectures and small input sizes. Finally, we also apply the pixel distillation paradigm to a complex task, *i.e.*, object detection, to showcase its potential for application in more scenarios. In this phase, an Aligned Feature for Preservation (AFP) strategy is designed on the assistant network, which aligns the output dimensions of detectors at each stage by manipulating the scale of features before the detection head of the assistant network. In the future, we will apply the proposed distillation mechanism to other knowledge transfer tasks like [82], [83], [84]

References

[1] S. Karen and Z. Andrew, "Very deep convolutional networks for large-scale image recognition," in *Int. Conf. Learn. Represent.*, 2015.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 770–778.

[3] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 6848–6856.

[4] M.-H. Guo, C.-Z. Lu, Z.-N. Liu, M.-M. Cheng, and S.-M. Hu, "Visual attention network," *arXiv preprint arXiv:2202.09741*, 2022.

[5] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2022, pp. 11 976–11 986.

[6] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *Int. Conf. Learn. Represent.*, 2021.

[7] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Int. Conf. Comput. Vis.*, 2021, pp. 10 012–10 022.

[8] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Int. Conf. Mach. Learn.* PMLR, 2021, pp. 10 347–10 357.

[9] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[10] X. Wang, X. Zhang, Y. Cao, W. Wang, C. Shen, and T. Huang, "Seggpt: Segmenting everything in context," *arXiv preprint arXiv:2304.03284*, 2023.

[11] X. Zou, J. Yang, H. Zhang, F. Li, L. Li, J. Gao, and Y. J. Lee, "Segment everything everywhere all at once," *arXiv preprint arXiv:2304.06718*, 2023.

[12] J. H. Cho and B. Hariharan, "On the efficacy of knowledge distillation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 4794–4802.

[13] B. Qian, Y. Wang, H. Yin, R. Hong, and M. Wang, "Switchable online knowledge distillation," in *Eur. Conf. Comput. Vis.* Springer, 2022, pp. 449–466.

[14] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011.

[15] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi, "Fine-grained visual classification of aircraft," *arXiv preprint arXiv:1306.5151*, 2013.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 248–255.

[17] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *Int. J. Comput. Vis.*, vol. 88, pp. 303–338, 2010.

[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Eur. Conf. Comput. Vis.* Springer, 2014, pp. 740–755.

[19] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Adv. Neural Inform. Process. Syst. Worksh.*, 2015.

[20] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *Adv. Neural Inform. Process. Syst.*, 2014, pp. 2654–2662.

[21] W. Park, D. Kim, Y. Lu, and M. Cho, "Relational knowledge distillation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 3967–3976.

[22] B. Zhao, Q. Cui, R. Song, Y. Qiu, and J. Liang, "Decoupled knowledge distillation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 11 953–11 962.

[23] L. Yu, V. O. Yazici, X. Liu, J. v. d. Weijer, Y. Cheng, and A. Ramisa, "Learning metrics from teachers: Compact networks for image embedding," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 2907–2916.

[24] N. Passalis and A. Tefas, "Learning deep representations with probabilistic knowledge transfer," in *Eur. Conf. Comput. Vis.*, 2018, pp. 268–284.

[25] J. Kim, S. Park, and N. Kwak, "Paraphrasing complex network: Network compression via factor transfer," in *Adv. Neural Inform. Process. Syst.*, 2018, pp. 2760–2769.

[26] S. Ahn, S. X. Hu, A. Damianou, N. D. Lawrence, and Z. Dai, "Variational information distillation for knowledge transfer," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 9163–9171.

[27] A. Koratana, D. Kang, P. Bailis, and M. Zaharia, "Lit: Learned intermediate representation training for model compression," in *Int. Conf. Mach. Learn.*, 2019, pp. 3509–3518.

[28] Y. Guan, P. Zhao, B. Wang, Y. Zhang, C. Yao, K. Bian, and J. Tang, "Differentiable feature aggregation search for knowledge distillation," in *Eur. Conf. Comput. Vis.*, 2020, pp. 469–484.

[29] K. Yue, J. Deng, and F. Zhou, "Matching guided distillation," in *Eur. Conf. Comput. Vis.*, 2020, pp. 312–328.

[30] L. Zhang, Y. Shi, Z. Shi, K. Ma, and C. Bao, "Task-oriented feature distillation," *Adv. Neural Inform. Process. Syst.*, vol. 33, 2020.

[31] L. Liu, Q. Huang, S. Lin, H. Xie, B. Wang, X. Chang, and X. Liang, "Exploring inter-channel correlation for diversity-preserved knowledge distillation," in *Int. Conf. Comput. Vis.*, 2021, pp. 8271–8280.

[32] Y. Shang, B. Duan, Z. Zong, L. Nie, and Y. Yan, "Lipschitz continuity guided knowledge distillation," in *Int. Conf. Comput. Vis.*, 2021, pp. 10 675–10 684.

[33] L. Chen, D. Wang, Z. Gan, J. Liu, R. Henao, and L. Carin, "Wasserstein contrastive representation distillation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 16 296–16 305.

[34] P. Chen, S. Liu, H. Zhao, and J. Jia, "Distilling knowledge via knowledge review," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 5008–5017.

[35] J. Zhu, S. Tang, D. Chen, S. Yu, Y. Liu, M. Rong, A. Yang, and X. Wang, "Complementary relation contrastive distillation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 9260–9269.

[36] S. Lin, H. Xie, B. Wang, K. Yu, X. Chang, X. Liang, and G. Wang, "Knowledge distillation via the target-aware transformer," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 10 915–10 924.

[37] G. Guo, L. Han, L. Wang, D. Zhang, and J. Han, "Semantic-aware knowledge distillation with parameter-free feature uniformization," *Visual Intelligence*, vol. 1, no. 1, p. 6, 2023.

[38] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," in *Int. Conf. Learn. Represent.*, 2015.

[39] B. Heo, J. Kim, S. Yun, H. Park, N. Kwak, and J. Y. Choi, "A comprehensive overhaul of feature distillation," in *Int. Conf. Comput. Vis.*, 2019, pp. 1921–1930.

[40] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 4133–4141.

[41] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *Int. Conf. Learn. Represent.*, 2017.

[42] F. Tung and G. Mori, "Similarity-preserving knowledge distillation," in *Int. Conf. Comput. Vis.*, 2019, pp. 1365–1374.

[43] B. Heo, M. Lee, S. Yun, and J. Y. Choi, "Knowledge transfer via distillation of activation boundaries formed by hidden neurons," in *AAAI Conf. Art. Intell.*, vol. 33, 2019, pp. 3779–3787.

[44] S. Srinivas and F. Fleuret, "Knowledge transfer with jacobian matching," in *Int. Conf. Mach. Learn.*, 2018, pp. 4723–4731.

[45] Y. Liu, J. Cao, B. Li, C. Yuan, W. Hu, Y. Li, and Y. Duan, "Knowledge distillation via instance relationship graph," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 7096–7104.

[46] S. I. Mirzadeh, M. Farajtabar, A. Li, N. Levine, A. Matsukawa, and H. Ghasemzadeh, "Improved knowledge distillation via teacher assistant," in *AAAI Conf. Art. Intell.*, vol. 34, no. 04, 2020, pp. 5191–5198.

[47] Q. Li, S. Jin, and J. Yan, "Mimicking very efficient network for object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 6356–6364.

[48] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," in *Adv. Neural Inform. Process. Syst.*, 2017, pp. 742–751.

[49] Y. Wei, X. Pan, H. Qin, W. Ouyang, and J. Yan, "Quantization mimic: Towards very tiny cnn for object detection," in *Eur. Conf. Comput. Vis.*, 2018, pp. 267–283.

[50] T. Wang, L. Yuan, X. Zhang, and J. Feng, "Distilling object detectors with fine-grained feature imitation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 4933–4942.

[51] Z. Zheng, R. Ye, P. Wang, D. Ren, W. Zuo, Q. Hou, and M.-M. Cheng, "Localization distillation for dense object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

[52] Z. Zheng, R. Ye, Q. Hou, D. Ren, P. Wang, W. Zuo, and M.-M. Cheng, "Localization distillation for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–14, 2023.

[53] Y. Liu, K. Chen, C. Liu, Z. Qin, Z. Luo, and J. Wang, "Structured knowledge distillation for semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 2604–2613.

[54] N. Liu, N. Zhang, and J. Han, "Learning selective self-mutual attention for rgb-d saliency detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 13 756–13 765.

[55] C. Yang, H. Zhou, Z. An, X. Jiang, Y. Xu, and Q. Zhang, "Cross-image relational knowledge distillation for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 319–12 328.

[56] D. Ji, H. Wang, M. Tao, J. Huang, X.-S. Hua, and H. Lu, "Structural and statistical texture knowledge distillation for semantic segmentation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 16 876–16 885.

[57] C. Fang, L. Wang, D. Zhang, J. Xu, Y. Yuan, and J. Han, "Incremental cross-view mutual distillation for self-supervised medical ct synthesis," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 20 677–20 686.

[58] C. Fang, Q. Wang, L. Cheng, Z. Gao, C. Pan, Z. Cao, Z. Zheng, and D. Zhang, "Reliable mutual distillation for medical image segmentation under imperfect annotations," *IEEE Trans. Med. Imaging*, 2023.

[59] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 2117–2125.

[60] W. Cao, Y. Zhang, J. Gao, A. Cheng, K. Cheng, and J. Cheng, "Pkd: General distillation framework for object detectors via pearson correlation coefficient," *Adv. Neural Inform. Process. Syst.*, vol. 35, pp. 15 394–15 406, 2022.

[61] G. Li, X. Li, Y. Wang, S. Zhang, Y. Wu, and D. Liang, "Knowledge distillation for object detection via rank mimicking and prediction-guided feature imitation," in *AAAI Conf. Art. Intell.*, vol. 36, no. 2, 2022, pp. 1306–1313.

[62] J. Wang, Y. Chen, Z. Zheng, X. Li, M.-M. Cheng, and Q. Hou, "Crosskd: Cross-head knowledge distillation for dense object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024.

[63] S. Lao, G. Song, B. Liu, Y. Liu, and Y. Yang, "Unikd: Universal knowledge distillation for mimicking homogeneous or heterogeneous object detectors," in *Int. Conf. Comput. Vis.*, 2023, pp. 6362–6372.

[64] Y. Zhu, Q. Zhou, N. Liu, Z. Xu, Z. Ou, X. Mou, and J. Tang, "Scalekd: Distilling scale-aware knowledge in small object detector," in *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2023, pp. 19 723–19 733.

[65] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, "Perceptual generative adversarial networks for small object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 1222–1230.

[66] S. Ge, S. Zhao, C. Li, and J. Li, "Low-resolution face recognition in the wild via selective knowledge distillation," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 2051–2062, 2018.

[67] M. Wang, R. Liu, N. Hajime, A. Narishige, H. Uchida, and T. Matsunami, "Improved knowledge distillation for training fast low resolution face recognition model," in *Int. Conf. Comput. Vis. Worksh.*, 2019, pp. 0–0.

[68] L. Qi, J. Kuen, J. Gu, Z. Lin, Y. Wang, Y. Chen, Y. Li, and J. Jia, "Multi-scale aligned distillation for low-resolution detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 14 443–14 453.

[69] S. Shin, J. Lee, J. Lee, Y. Yu, and K. Lee, "Teaching where to look: Attention similarity knowledge distillation for low resolution face recognition," *arXiv preprint arXiv:2209.14498*, 2022.

[70] S. Ge, S. Zhao, C. Li, Y. Zhang, and J. Li, "Efficient low-resolution face recognition via bridge distillation," *IEEE Trans. Image Process.*, vol. 29, pp. 6898–6908, 2020.

[71] M. Zhu, K. Han, C. Zhang, J. Lin, and Y. Wang, "Low-resolution visual recognition via deep feature distillation," in *IEEE Int. Conf. Acoust. Speech SP*. IEEE, 2019, pp. 3762–3766.

[72] H. Chen, Y. Pei, H. Zhao, and Y. Huang, "Super-resolution guided knowledge distillation for low-resolution image classification," *Pattern Recog. Letters*, vol. 155, pp. 62–68, 2022.

[73] Z. Huang, S. Yang, M. Zhou, Z. Li, Z. Gong, and Y. Chen, "Feature map distillation of thin nets for low-resolution object recognition," *IEEE Trans. Image Process.*, vol. 31, pp. 1364–1379, 2022.

[74] B. Hu, S. Zhou, Z. Xiong, and F. Wu, "Cross-resolution distillation for efficient 3d medical image registration," *IEEE Trans. Circuit Syst. Video Technol.*, vol. 32, no. 10, pp. 7269–7283, 2022.

[75] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 1874–1883.

[76] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

[77] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu *et al.*, "Mmdetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.

[78] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *Adv. Neural Inform. Process. Syst. Worksh.*, 2017.

[79] Y. Tian, D. Krishnan, and P. Isola, "Contrastive representation distillation," in *Int. Conf. Learn. Represent.*, 2019.

[80] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Eur. Conf. Comput. Vis.*, 2018, pp. 116–131.

[81] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "Swinir: Image restoration using swin transformer," in *Int. Conf. Comput. Vis.*, 2021, pp. 1833–1844.

[82] D. Cheng, Y. Ji, D. Gong, Y. Li, N. Wang, J. Han, and D. Zhang, "Continual all-in-one adverse weather removal with knowledge replay on a unified network structure," *IEEE Trans. Multimedia*, 2024.

[83] D. Zhang, H. Li, W. Zeng, C. Fang, L. Cheng, M.-M. Cheng, and J. Han, "Weakly supervised semantic segmentation via alternate self-dual teaching," *IEEE Trans. Image Process.*, 2023.

[84] H. Li, D. Zhang, Y. Dai, N. Liu, L. Cheng, J. Li, J. Wang, and J. Han, "Gp-nerf: Generalized perception nerf for context-aware 3d scene understanding," *arXiv preprint arXiv:2311.11863*, 2023.