Learn More for Food Recognition via Progressive Self-Distillation

Yaohui Zhu[1*], **Linhu Liu**[2*], **Jiang Tian**[2]

[1] School of Artifical Intelligence, Beijing Normal University, Beijing 10875, China
[2] AI Lab, Lenovo Research, Beijing, China
yaohui.zhu@bnu.edu.cn, {liulh7, tianjiang1}@lenovo.com

Abstract

Food recognition has a wide range of applications, such as health-aware recommendation and self-service restaurants. Most previous methods of food recognition firstly locate informative regions in some weakly-supervised manners and then aggregate their features. However, location errors of informative regions limit the effectiveness of these methods to some extent. Instead of locating multiple regions, we propose a Progressive Self-Distillation (PSD) method, which progressively enhances the ability of network to mine more details for food recognition. The training of PSD simultaneously contains multiple self-distillations, in which a teacher network and a student network share the same embedding network. Since the student network receives a modified image from its teacher network by masking some informative regions, the teacher network outputs stronger semantic representations than the student network. Guided by such teacher network with stronger semantics, the student network is encouraged to mine more useful regions from the modified image by enhancing its own ability. The ability of the teacher network is also enhanced with the shared embedding network. By using progressive training, the teacher network incrementally improves its ability to mine more discriminative regions. In inference phase, only the teacher network is used without the help of the student network. Extensive experiments on three datasets demonstrate the effectiveness of our proposed method and state-of-the-art performance.

Introduction

Food is a necessity of human life and the foundation of human experience. As a basic research in food field, food recognition has a wide range of applications such as visual food choice (Chen et al. 2020a), health-aware recommendation (Nag, Pandey, and Jain 2017) and self-service restaurants (Aguilar et al. 2018).

Food recognition belongs to fine-grained recognition, which refers to the task of distinguishing subordinate categories, such as birds, dogs and cars. Most previous methods of food recognition (Martinel and Foresti 2018; Jianing et al. 2019; Min et al. 2020, 2019; Wang et al. 2022) mainly follow the key solution of fine-grained recognition,
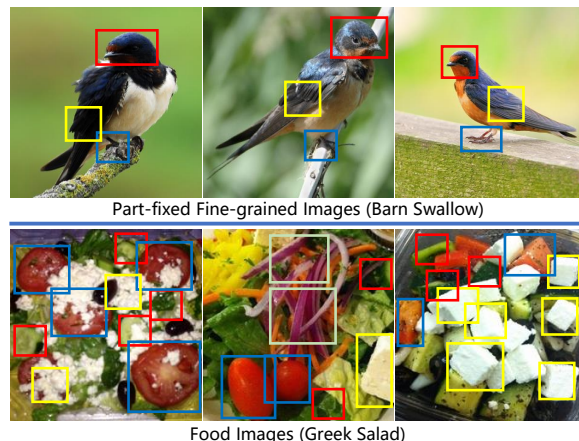
---

*These authors contributed equally.

Figure 1: Some fine-grained bird images and food images. There may be three discriminative regions in 'Barn Swallow' (see bounding boxes), while 'Greek Salad' contains lots of similar regions, each of which is possibly discriminative. Therefore, we think that food images have more informative regions than part-fixed fine-grained images.

which firstly locates informative regions in some weakly-supervised manners, and then aggregates features of these regions. Although these methods achieve promising performance, they possibly suffer from location errors of those informative regions, leading to the limited effectiveness to some extent.

In comparison with part-fixed fine-grained images such as birds, food images include a variety of similar ingredients stacked together. As shown in Figure 1, there are some semantic parts (e.g., head and claw) in 'Barn Swallow', while lots of sliced material (e.g., cucumbers and cheese), fruit and vegetable are stacked together in 'Greek Salad'. An intuitive hypothesis is that food images have more informative regions than part-fixed fine-grained images. However, it is challenging to capture so many informative regions under the existing training mechanism of vanilla networks, since training these regions together can not effectively learn each of them.

In this paper, we propose a self-boosting training mechanism, called Progressive Self-Distillation (PSD), to learn

more details for food recognition. Instead of aggregating multiple detected local features, the proposed PSD progressively enhances the ability of network to mine other informative regions from an image with some informative regions masked. To be more specific, the training of PSD simultaneously contains multiple self-distillations. In each self-distillation, a teacher network and a student network share an embedding network. The student network receives a modified image from its teacher network by removing some informative regions. Thus, the teacher network outputs stronger semantic representations than the student network. Guided by such teacher network with stronger semantics, the student network is encouraged to mine other discriminative regions from modified images by improving its own ability. Correspondingly, the ability of the teacher network is also enhanced with the shared embedding network. In the next self-distillation, more informative regions are masked in the input image of the student network based on the previous self-distillation. The multiple self-distillations are learned together by adjusting their weights with a ramp-up function to achieve progressive training. In this way, the teacher network incrementally improves its ability to mine more discriminative regions. Only the teacher network is used for inference without the help of the student network.

The proposed method can be flexibly implemented with different architectures of the embedding network. Two representative architectures are employed, including convolution neural networks (CNNs) and vision Transformers (e.g., Swin Transformer (Liu et al. 2021)). Comprehensive experiments on three benchmark datasets demonstrate the effectiveness and superiority of our method.

The contributions of our paper are summarized as follows:

- We propose a progressive self-distillation method, which progressively mines more informative regions in a self-boosting manner for food recognition.

- The proposed method is flexible to architectures of the embedding network including convolution neural networks and vision Transformers.

- We conduct extensive evaluation on three popular food benchmark datasets to verify the effectiveness of the proposed method.

Related Work

Food Recognition

In the earlier years, some works use hand-crafted features (Lowe 2004) or combination of hand-crafted features (Martinel et al. 2015) for food recognition. With the development of CNNs (He et al. 2016), some works (Kagaya, Aizawa, and Ogawa 2014) directly employ various types of CNNs for food recognition.

Recently, some works (Martinel and Foresti 2018; Jianing et al. 2019; Min et al. 2020, 2019) follow the key solution of fine-grained recognition, which firstly locates informative regions in some weakly-supervised manners, and then combines features of these regions with a global feature for food recognition. For examples, the information of ingredient (Min et al. 2019) is leveraged to extract local features,

and these local features and a global feature are combined to recognize food images. A slice network (Martinel and Foresti 2018) is proposed to capture specific vertical food layers, and then combine the features of slice network with ones from backbone network for food recognition. Although obtaining the impressive performance, these methods possibly locate some incorrect regions, which limits their effectiveness. Moreover, these methods require to locate multiple regions and aggregate features of these regions, resulting in high computational overhead in both training phase and inference phase. Different from the above existing works of aggregating multiple regions, the proposed PSD learns more informative regions in a self-boosting manner. Compared with a standard classification network, the PSD does not bring extra computational overhead in the inference phase.

Self-supervised Learning

The approaches of self-supervised learning focus on designing auxiliary objectives to learn useful feature representations by using the structure of the data itself. The auxiliary objectives can be handcrafted pretext tasks, such as relative patch prediction (Doersch, Gupta, and Efros 2015), solving jigsaw puzzles (Noroozi and Favaro 2016) and rotation prediction (Komodakis and Gidaris 2018). Although these methods learn useful feature representations with big networks and long training (Kolesnikov, Zhai, and Beyer 2019), they heavily rely on somewhat adhoc pretext tasks, which limits the generalization ability of learned representations to some extent.

Recently, contrastive learning (Hadsell, Chopra, and Le-Cun 2006) is increasingly successful in self-supervised learning. The core idea of contrastive learning is to attract the positive sample pairs and repulse the negative sample pairs. In practice, contrastive learning methods benefit from a large number of negative samples, which can be maintained in a memory bank (Wu et al. 2018). Without requiring specialized architectures or a memory bank, a simple contrastive self-supervised learning (Chen et al. 2020b) is proposed while this work requires a large batch size to work well. And a dynamic dictionary (He et al. 2020) is used with a queue and a moving-averaged encoder, building a large and consistent dictionary on-the-fly. To eliminate the requirement of negative samples for reducing memory consumption, a series of works (Chen and He 2021; Grill et al. 2020) retain siamese architectures to learn invariant features by matching positive samples, and employ a stop-gradient operation to prevent model from collapsing. Interesting, the representations from contrastive self-supervised pre-training can outperform their supervised counterparts in certain tasks. In this paper, we borrow contrastive learning for food recognition at semantic levels in a self-boosting manner to some extent.

Knowledge Distillation

The concept of knowledge distillation was firstly proposed in (Hinton et al. 2015). In a learning paradigm of knowledge distillation, a bigger teacher network guides the training of a smaller student network to transfer its knowledge to
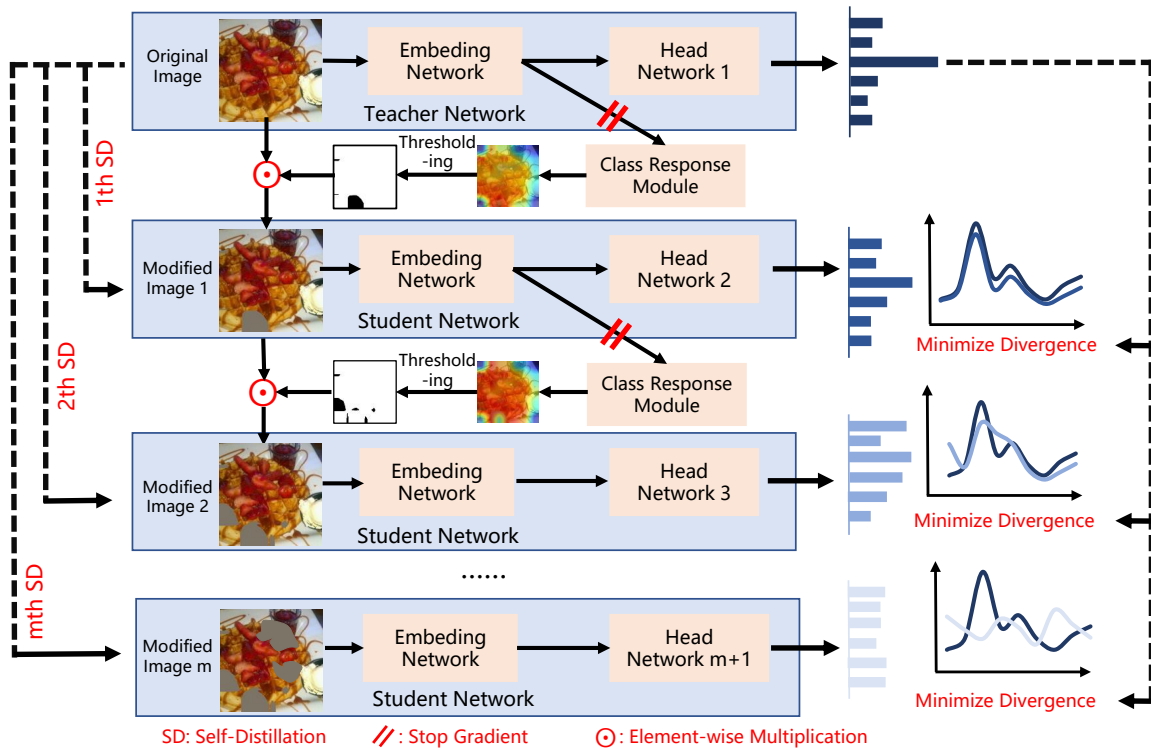
Figure 2: The training pipeline of the proposed method PSD. The PSD contains multiple self-distillations, and in each self-distillation, a teacher network and a student network share the same embedding network but with different head networks.

the student network. The knowledge can be distilled via semantic distributions (Tung and Mori 2019; Zhao et al. 2022; Chen et al. 2022) or intermediate features (Heo et al. 2019; Tian, Krishnan, and Isola 2020). To enhance efficiency and effectiveness in knowledge transferring, self knowledge distillation is proposed to utilize knowledge from itself, without the involvement of extra networks. For example, aggregation of various distortion data is used to achieve self-distillation (Lee, Hwang, and Shin 2020). Motivated by this, we propose a progressive self-distillation method to mine more details for food recognition.

The Proposed Method

The proposed PSD progressively enhances the ability of network to mine more discriminative regions for food recognition. As shown in Figure 2, the training pipeline of PSD includes multiple self-distillations. In this section, we first present the framework of self-distillation, and then introduce how to organize multiple self-distillations for progressive training. Finally, the method implementation is provided.

Self-Distillation

In self-distillation, there are a teacher network and a student network, as shown in Figure 2. The teacher network and the student network share the same embedding network but with corresponding head networks. In the teacher network, an original image is inputted. An input image of a student network is modified from its teacher network by re-

moving some informative regions. Next, we will introduce how to obtain an input image of a student network, and how to distill.

Locating Discriminative Region. The discriminative regions are located with high class responses of feature map. Given an input image x, an embedding network is used to extract a feature map $f(x; \theta) = S \in \mathbf{R}^{H \times W \times D}$, where $f(; \theta)$ represents an embedding network with parameter θ, D is the number of feature channels, and $H \times W$ is the spatial size of the feature map. When using vision Transformer as an embedding network, the tokens $T \times D$ are reshaped to a $H \times W \times D$ feature map. The feature map S is then fed to a class response module, which contains a global average pooling layer followed by a fully connected layer. The weight matrix of the fully connected layer is $\Theta \in \mathbf{R}^{D \times C}$, where C is the number of food categories. Then a Class Response Map (CRM) of the c-th category M_c is computed as:

$$M_c = \sum_{d=1}^{D} \Theta_{d,c} \times S_d \qquad (1)$$

where $\Theta_{d,c}$ represents the d-th weight for the c-th category in Θ, and S_d is a d-th feature map in S. The weight Θ needs to be learned, and the optimization loss is:

$$L_l(x, y) = L_{ce}(\text{GAP}(f(x; \theta)) * \Theta, y) \qquad (2)$$

where $\text{GAP}()$ is a global average pooling operation, $*$ is a matrix multiplication operation, $L_{ce}(,)$ is a cross-entropy loss, and y is the ground truth label.
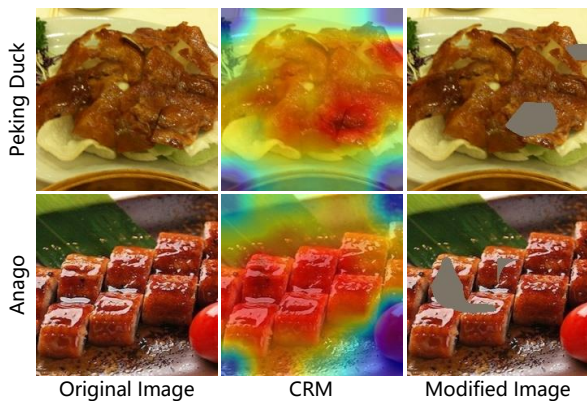
Figure 3: Some examples of original images, CRM and modified images.

The location of high value in the class response map M_c can represent discriminative area. Therefore, these high values in M_c is used to locate discriminative regions. A certain percentile η in M_c is used as a threshold. Supposing the value of the percentile is $\hat{w}_m$, the location of discriminative regions for the c-th category is calculated as:

$$Loc_c = \begin{cases} 1, & w_m \geq \hat{w}_m \\ 0, & w_m < \hat{w}_m \end{cases} \quad (3)$$

where $w_m \in M_c$.

Masking Discriminative Region. The ground truth label is used to obtain the location of discriminative regions Loc_c, and the discriminative regions are removed in image level. The formulation of this process is defined as follows:

$$\bar{x} = (1 - Loc_c) \odot x, \quad (4)$$

where $\odot$ means an element-wise multiplication, and $\bar{x}$ is a modified image. As shown in Figure 3, the discriminative regions are masked to form modified images.

Distillation Objective. The image x is fed into a teacher network, and the modified image $\bar{x}$ is fed into a student network. Therefore, the teacher network outputs stronger semantic information than the student network. The optimization goal is to minimize difference in output distribution between the teacher network and the student network, and the loss is as follows:

$$L_d(x, \bar{x}) = D_{KL}(h(f(x; \theta); \phi_t), h(f(\bar{x}; \theta); \phi_s))) \quad (5)$$

where $D_{KL}(,)$ is Kullback-Leibler divergence, $h(;)$ represents a head network, and ϕ_t and ϕ_s are parameters of $h(;)$ in the teacher network and the student network, respectively.

Guided by stronger semantic information, the student network is encouraged to mine other discriminative regions from the modified image $\bar{x}$, improving the ability of the student network. The ability of the teacher network is also enhanced since the teacher network and the student network share the same embedding network.

Progressive Training

Progressive training methodology has been widely utilized for image generation tasks (Ahn, Kang, and Sohn 2018; Karras, Laine, and Aila 2019). This methodology starts with

generating low-resolution images, and then gradually increases the generated resolution. In this methodology, an easy task is completed first, and then the difficulty of the task is gradually increased. Our work uses a progressive training methodology to organize multiple self-distillations. In these self-distillations, an input image of the student network is a modified image of student network in the previous self-distillation, except for the first time. In the first self-distillation, the input image of student network is a modified image of the teacher network.

The classification loss is:

$$L_g(x, y) = L_{ce}(h(f(x; \theta); \phi_t), y) \quad (6)$$

Combining with the classification loss, the locating loss L_l in Eq. 2 and the distillation loss L_d in Eq. 5, the final optimization loss is:

$$L = L_g(x, y) + \sum_{i=1}^{m}(\omega_l \, L_l(\bar{x}_{i-1}, y) + \omega_d L_d(x, \bar{x}_i)) \quad (7)$$

where $\bar{x}_i$ is an input image of the student network in i-th self-distillation, $\bar{x}_0$ is x, m is the number of self-distillation, and ω_l and ω_d are balance parameters. Instead of using a step-by-step progressive training like (Ahn, Kang, and Sohn 2018; Karras, Laine, and Aila 2019), multiple self-distillations are learned together by adjusting the value of ω_d during training phase. The ω_d starts from a small value to a fixed value α with a ramp-up function (Laine and Aila 2016), and it can be formalized as follows:

$$\omega_d = \begin{cases} \alpha * exp(-5(1 - \frac{e}{\beta})^2), & e < \beta \\ \alpha, & e \geq \beta \end{cases} \quad (8)$$

where e denotes the current epoch during training phase, α is a scalar, and β is an integer.

At the begin of training, the predictions of the teacher network may be incorrect. A small value of ω_d can prevent the student network learning incorrect knowledge from the teacher network. After the predictions of the teacher network become confident, a big value of ω_d can encourage the student network to learn correct knowledge from the teacher network. Using a ramp-up ω_d makes multiple self-distillations of different difficulties learned together. By using this progressive training, the teacher network incrementally improves the ability to mine more discriminative regions for food recognition.

Implementation

The embedding network can be implemented with the vast majority of networks. In this work, we use two types of network architectures including vision transformers and CNN-based networks. The head network contains a global average pooling layer and a fully connected layer. The class response module acts on the feature map of the third hidden layer in the embedding network. A stop-gradient operation is performed on the feature map before inputting the class response module. Therefore, the loss in class response module (i.e., Eq. 2) does not affect the training of the embedding network. Although our method PSD requires multi-stage network calculation in training phase, only one stage network

calculation (i.e., the teacher network) is used for classification in inference phase. Compared with a standard classification network, the PSD does not bring additional computational overhead in the inference phase.

Experiments

We will validate the effectiveness of the proposed method by answering the following two questions: Q1. Is the proposed method PSD effective for food recognition? Q2. Does the proposed method PSD learn more discriminative regions? In this section, the experimental setup is firstly introduced. Next, the two questions are answered in experimental results and analysis. Finally, the further analysis is provided.

Experimental Setup

Datasets. We validate our method on three commonly used food datasets.

- **ETHZ Food-101** (Bossard, Guillaumin, and Van Gool 2014) contains 101,000 images with 101 food categories. Each category has 1,000 images including 750 training images and 250 test images.

- **Vireo Food-172** (Chen and Ngo 2016) contains 110,241 food images from 172 categories. Following commonly used splits, 60%, 10%, 30% images of each food category are randomly selected for training, validation and testing, respectively.

- **ISIA Food-500** (Min et al. 2020) consists of 399,726 images with 500 categories. The average number of images per category is about 800. This dataset is divided into 60%, 10% and 30% images for training, validation and testing, respectively.

Implementation Details. All experiments are implemented on the Pytorch platform with one Nvidia A100 GPU. The input image size is set to 224×224 in all experiments. For fair comparisons and re-implementations, the same random seed is used to eliminate the training bias in all experiments. Top1 accuracy and Top5 accuracy are used as evaluation metrics. On Vireo Food-172 and ISIA Food-500, the model of the highest performance on validation set is used for test. On ETHZ Food-101, since there is no validation set, the last model is used for test. We set a percentile $\eta = 5\%$ in M_c as a threshold, $\omega_l = 1$, the ramp-up epochs $\beta = 5$ in Eq. 7, and the number of self-distillation $m = 2$ in all experiments.

When employing Swin-B (Liu et al. 2021) as an embedding network, the model is optimized by adamw (Kingma and Ba 2014) algorithm with an initial learning rate of 5×10^{-5} and a weight decay of 10^{-8}. The model is initialized with ImageNet-22K pre-trained model. A cosine decay learning rate scheduler with 5 epochs of warm-up is used. The total number of training epochs is 50, and a batch size of 42 and gradient clipping with a max norm of 5 are used. Similar to the work (Liu et al. 2021), gradient accumulation step of 2 is used to reduce GPU consumption and stochastic depth ratio of 0.2 is adopted. In Eq. 8, $\alpha = 2.0$.

When employing DenseNet161 (Huang et al. 2017) as an embedding network, the model is optimized using stochastic gradient descent with a momentum of 0.9 and a weight

Method		Top1	Top5
MF	WISeR (Martinel and Foresti 2018)[†]	90.27	98.71
	IG-CMAN (Min et al. 2019)	90.37	98.42
	PAR-Net (Jianing et al. 2019)[†]	90.40	-
	MSMVFA (Jiang et al. 2019)	90.59	98.25
	SGLANet (Min et al. 2020)[†]	90.92	98.24
	IGRL (Wang et al. 2022)	92.36	98.68
SF	SFLR (Bolanos and Radeva 2017)	79.20	94.11
	SENet154 (Hu and Shen 2018)	88.62	97.57
	DLA (Yu et al. 2018)[*]	90.00	-
	Incep-Res-v2 (Yin et al. 2018)	90.40	-
	Incep-v4 (Kornblith et al. 2019)[*]	90.80	-
	GPipe (Huang et al. 2019)	93.00	-
	EfficientNet (Tan and Le 2019)	93.00	-
	Grafit (Touvron et al. 2021)	93.70	-
	DenseNet161 (Huang et al. 2017)	86.93	97.10
	DenseNet161+PSD (our)	87.40	97.20
	Swin-B (Liu et al. 2021)	93.91	99.03
	Swin-B+PSD (our)	**94.56**	**99.34**

Table 1: Accuracy (%) comparison on ETHZ Food-101. [†]: 10 crop images for test. [*]: 448×448 resolution images. MF: multi-feature aggregation. SF: single feature for test.

decay of 10^{-4}. The model is initialized with ImageNet-1K pre-trained model. The learning rate is initially set to 10^{-3} and divided by 10 after 10 epochs. The total number of training epochs is 30, and the batch size is 42. In Eq. 8, $\alpha = 1.0$.

Experimental Results and Aanlysis

Evaluation on Three Benchmark Datasets. The experimental results on ETHZ Food-101 are illustrated on Table 1. We build a strong baseline with Swin Transformer (Liu et al. 2021). The baseline method Swin-B has already outperformed some previous best methods, including some methods of multi-feature aggregation (e.g., SGLANet (Min et al. 2020) and PAR-Net (Jianing et al. 2019)) and some methods of using advanced architectures (e.g., Grafit (Touvron et al. 2021) and GPipe (Huang et al. 2019)). Although a strong baseline is achieved with the Swin-B, the proposed PSD still obtains 0.65% improvements in Top1 accuracy and 0.31% improvements in Top5 accuracy, exhibiting its superiority. When using DenseNet161, the proposed PSD obtains about 0.5% gains in Top1 accuracy.

The experimental results on Vireo Food-172 are illustrated on Table 2. A strong baseline is built with Swin-B, which outperforms the best method MVANET (Liang et al. 2021) by 1.67% in Top1 accuracy, and is comparable with MVANET in Top5 accuracy. Based on this baseline, the proposed Swin-B+PSD still obtains performance improvements. When using DenseNet161, the proposed PSD achieves 0.75% gains in Top1 accuracy.

The experimental results on ISIA Food-500 are illustrated on Table 3. A strong baseline is also built with Swin-B, which outperforms the best method TPSKG (Liu and Wang 2022) by about 1.9% in Top1 accuracy and SGLANet (Min et al. 2020) by over 1.0% in Top5 accuracy. Note that TPSKG also employs vision Transformer as backbone and use

Method		Top1	Top5
MF	IG-CMAN (Min et al. 2019)	90.63	98.40
	PAR-Net (Jianing et al. 2019)[†]	90.20	-
	MSMVFA (Jiang et al. 2019)	90.61	98.31
	AFN (Liu et al. 2020)	89.54	98.05
	SGLANet (Min et al. 2020)[†]	90.98	98.35
	MVANET (Liang et al. 2021)	91.08	98.86
SF	VGG16 (Szegedy et al. 2015)	80.41	95.59
	MTDCNN (Chen and Ngo 2016)	82.06	95.88
	SENet154 (Hu and Shen 2018)	88.71	97.74
	DenseNet161 (Huang et al. 2017)	88.25	97.53
	DenseNet161+PSD (our)	89.00	97.70
	Swin-B (Liu et al. 2021)	92.75	98.71
	Swin-B+PSD (our)	**92.91**	**99.08**

Table 2: Accuracy (%) comparison on Vireo Food-172. [†]: 10 crop images for test.

Method		Top1	Top5
MF	NTS-NET (Yang et al. 2018)	63.66	88.48
	WS-DAN (Hu and Qi 2019)	60.67	86.48
	SGLANet (Min et al. 2020)	64.74	89.12
SF	VGG16 (Szegedy et al. 2015)	55.22	82.77
	GoogLeNet (Meyers et al. 2015)	56.03	83.42
	ResNet152 (He et al. 2016)	57.03	83.80
	WRN50 (Sergey and Nikos 2016)	60.08	85.98
	SENet154 (Hu and Shen 2018)	63.83	88.61
	DCL (Chen et al. 2019)	64.10	88.77
	TPSKG (Liu and Wang 2022)*	65.40	-
	DenseNet161 (Huang et al. 2017)	60.05	86.09
	DenseNet161+PSD (our)	60.94	87.33
	Swin-B (Liu et al. 2021)	67.32	90.18
	Swin-B+PSD (our)	**70.10**	**92.75**

Table 3: Accuracy (%) comparison on ISIA Food-500. *: 448×448 resolution images.

Figure 4: Some visualizations of Class Response Map.
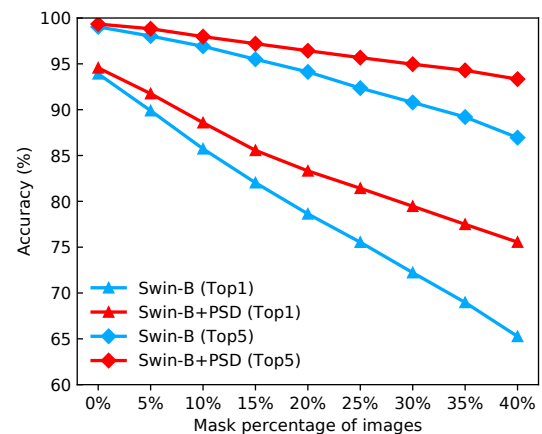
Figure 5: Accuracy of masked images on ETHZ Food-101.

higher resolution images. Despite achieving a strong baseline, the proposed method still obtains relatively large performance gains, i.e., 2.78% improvements in Top1 accuracy and 2.57% improvements in Top5 accuracy, validating the advantage of the proposed method. When using DenseNet161, the proposed PSD achieves about 0.9% gains in Top1 accuracy and 1.24% gains in Top5 accuracy.

To summarize, the proposed method PSD obtains performance gains on the above three datasets (e.g., ETHZ Food-101, Vireo Food-172 and ISIA Food-500) in two architectures (e.g., Swin-B and DenseNet161). This can validate that the proposed method is effective for food recognition (Q1).

Visualization of Response Map. To demonstrate more informative regions learned by the proposed PSD, we visualize the class response map of both Swin-B and Swin-B+PSD, as shown in Figure 4. We can obviously observe that the proposed Swin-B+PSD obtains more high-response regions than Swin-B. These high-response regions are also discriminative, which can explain that the proposed method learns more discriminative regions (Q2).
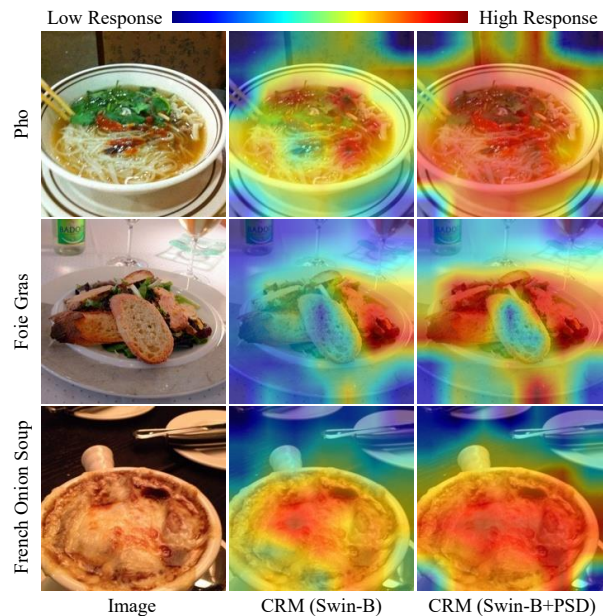
Test on Masked Images. Furthermore, we test the trained model with modified images where some regions are randomly masked. The mask percentage for each test image ranges from 0% to 40%. The experimental results on ETHZ Food-101 are exhibited in Figure 5. Both the baseline method Swin-B and the proposed Swin-B+PSD degrade performance with increasing mask percentage of each test image. Top1 and Top5 accuracy of Swin-B drops over 25% and 10%, respectively. In contrast, Top1 and Top5 accuracy of Swin-B+PSD drops less than 20%, and 7%, respectively. The performance drop of the proposed PSD is lower than the baseline method in both Top1 accuracy and Top5 accuracy. This can explain that the proposed PSD learns more effective features than the baseline method from masked images. To some extent, this also can explain that the proposed PSD learns more discriminative regions (Q2).

Method	Food-101 (%)		Food-500 (%)	
	Top1	Top5	Top1	Top5
Swin-B	93.91	99.03	67.32	90.18
Swin-B+PSD ($m=1$)	94.46	99.25	69.47	92.24
Swin-B+PSD ($m=2$)	94.56	99.34	70.10	92.75
Swin-B+PSD ($m=3$)	94.44	99.30	69.94	92.63

Table 4: Comparisons with numbers of self-distillations.

Method	Food-101 (%)		Food-500 (%)	
	Top1	Top5	Top1	Top5
Swin-B+PSD (SbS)	94.40	99.30	69.30	92.01
Swin-B+PSD (HS)	94.39	99.28	69.79	92.47
Swin-B+PSD (our)	94.56	99.34	70.10	92.75
Swin-B	93.91	99.03	67.32	90.18
Swin-B (DA Mask5%)	93.87	98.90	67.41	90.13
Swin-B (DA Mask10%)	93.88	98.94	67.43	90.12

Table 5: Comparisons with progressive training and Data Augmentation (DA). SbS: step-by-step. HS:head shared.

Further Analysis

Comparisons with Numbers of Self-distillations. The experimental results of different numbers of self-distillations on ETHZ Food-101 and ISIA Food-500 are illustrated in Table 4. Compared with the baseline method Swin-B, the proposed method equipped with 1, 2, or 3 self-distillations obtains improvements, demonstrating the effectiveness of the proposed method. Among them, the proposed method equipped with 2 self-distillations achieves the best performance. Based on two self-distillations, adding a third self-distillation results in performance drops. The third self-distillation is a tougher task than the first two, since it removes more informative regions. This tough task possibly increases the difficulty of training.

Comparisons with Settings of Progressive Training. We consider the following two variants of progressive training, including i) step-by-step progressive training: multiple self-distillations are trained from first to last, ii) head shared progressive training: the teacher network and the student network share the head network. The experimental results of two variants are illustrated at the top of Table 5. Our progressive training obtains better performance than the two variants, demonstrating its advantages.

Comparisons with Data Augmentation. Masking discriminative regions from images with the class response module is considered as a manner of data augmentation. The experimental results of this data augmentation are illustrated at the bottom of Table 5. This data augmentation does not bring performance improvements based on Swin-B. This can illustrate that the performance improvement obtained by our method PSD is not due to the use of modified images.

Comparisons with Other Learning Methods. We compare the proposed method with other learning methods BAN (Furlanello et al. 2018) and MutL (Zhang et al. 2018). The two works use the same backbone (i.e., Swin-B), batch size and the settings of optimizer as ours. Experimental results are shown in Table 6. The proposed method obtains higher

Method	Food-101 (%)		Food-500 (%)	
	Top1	Top5	Top1	Top5
BAN	94.10	99.02	68.01	91.15
MutL	94.29	99.23	69.23	92.22
Our	94.56	99.34	70.10	92.75

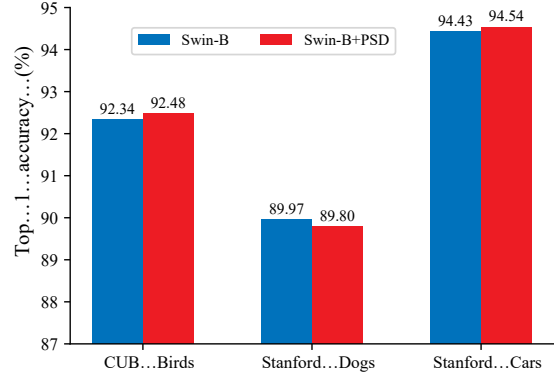Table 6: Comparsions with other learning methods.

Figure 6: Top1 accuracy comparison on CUB Birds, Stanford Dogs, and Stanford Cars.

accuracy than the two works on ETHZ Food-101 and ISIA Food-500, illustrating its superiority.

Experiments on Part-fixed Fine-grained Recognition. We employ the proposed method on three commonly used fine-grained datasets (i.e., CUB Birds (Wah et al. 2011), Stanford Dogs (Khosla et al. 2011) and Stanford Cars (Krause et al. 2013)). The experimental results are shown in Figure 6. Based on Swin-B, our method hardly obtains performance gains on three part-fixed fine-grained datasets. Compared with food images, the number of discriminative regions in part-fixed fine-grained images is limited, making it difficult to mine more discriminative details.

Conclusions

In this paper, we propose a Progressive Self-Distillation method (PSD) to learn more details for food recognition. Instead of locating multiple regions, the proposed PSD progressively enhances the ability of network to mine more informative regions. The training of PSD simultaneously contains multiple self-distillations. In each self-distillation, the student network is guided with stronger semantic representations from the teacher network to improve its ability. The ability of the teacher network also be improved with a shared embedding network in the student network. By using progressive training to organize multiple self-distillations, the teacher network incrementally improves the ability to mine more discriminative regions. Comprehensive experiments on three benchmark datasets demonstrate the effectiveness of our proposed method.

Acknowledgments

References

Aguilar, E.; Remeseiro, B.; Bolaños, M.; and Radeva, P. 2018. Grab, Pay and Eat:Semantic Food Detection for Smart Restaurants. *IEEE TMM*, 20(12): 3266–3275.

Ahn, N.; Kang, B.; and Sohn, K.-A. 2018. Image super-resolution via progressive cascading residual network. In *CVPR Workshops*, 791–799.

Bolanos, M.; and Radeva, P. 2017. Simultaneous food localization and recognition. In *ICPR*, 3140–3145.

Bossard, L.; Guillaumin, M.; and Van Gool, L. 2014. Food-101–mining discriminative components with random forests. In *ECCV*, 446–461.

Chen, D.; Mei, J.-P.; Zhang, H.; Wang, C.; Feng, Y.; and Chen, C. 2022. Knowledge Distillation with the Reused Teacher Classifier. In *CVPR*, 11933–11942.

Chen, J.; and Ngo, C.-W. 2016. Deep-based ingredient recognition for cooking recipe retrieval. In *ACM MM*, 32–41.

Chen, J.; Pan, L.; Wei, Z.; Wang, X.; Ngo, C.-W.; and Chua, T.-S. 2020a. Zero-shot ingredient recognition by multi-relational graph convolutional network. In *AAAI*, volume 34, 10542–10550.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020b. A simple framework for contrastive learning of visual representations. In *ICML*, 1597–1607. PMLR.

Chen, X.; and He, K. 2021. Exploring simple siamese representation learning. In *CVPR*, 15750–15758.

Chen, Y.; Bai, Y.; Zhang, W.; and Mei, T. 2019. Destruction and Construction Learning for Fine-Grained Image Recognition. In *CVPR*, 5157–5166.

Doersch, C.; Gupta, A.; and Efros, A. A. 2015. Unsupervised visual representation learning by context prediction. In *ICCV*, 1422–1430.

Furlanello, T.; Lipton, Z.; Tschannen, M.; Itti, L.; and Anandkumar, A. 2018. Born again neural networks. In *ICML*, 1607–1616.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. In *NeurIPS*, 21271–21284.

Hadsell, R.; Chopra, S.; and LeCun, Y. 2006. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 1735–1742.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 9729–9738.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *CVPR*, 770–778.

Heo, B.; Lee, M.; Yun, S.; and Choi, J. Y. 2019. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *AAAI*, volume 33, 3779–3787.

Hinton, G.; Vinyals, O.; Dean, J.; et al. 2015. Distilling the knowledge in a neural network. In *NeurIPS workshop*.

Hu, J.; and Shen, L. 2018. Squeeze-and-Excitation Networks. In *CVPR*, 7132–7141.

Hu, T.; and Qi, H. 2019. See Better Before Looking Closer: Weakly Supervised Data Augmentation Network for Fine-Grained Visual Classification. volume abs/1901.09891.

Huang, G.; Liu, Z.; Weinberger, K. Q.; and van der Maaten, L. 2017. Densely connected convolutional networks. In *CVPR*, 2261–2269.

Huang, Y.; Cheng, Y.; Bapna, A.; Firat, O.; Chen, D.; Chen, M.; Lee, H.; Ngiam, J.; Le, Q. V.; Wu, Y.; et al. 2019. Gpipe: Efficient training of giant neural networks using pipeline parallelism. In *NeurIPS*, 21271–21284.

Jiang, S.; Min, W.; Liu, L.; and Luo, Z. 2019. Multi-Scale Multi-View Deep Feature Aggregation for Food Recognition. *IEEE TIP*, 29(1): 265–276.

Jianing, Q.; Frank, P.-W. L.; Yingnan, S.; Siyao, W.; and Benny, L. 2019. Mining Discriminative Food Regions for Accurate Food Recognition. In *BMVC*.

Kagaya, H.; Aizawa, K.; and Ogawa, M. 2014. Food detection and recognition using convolutional neural network. In *ACM MM*, 1085–1088.

Karras, T.; Laine, S.; and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. In *CVPR*, 4401–4410.

Khosla, A.; Jayadevaprakash, N.; Yao, B.; and Fei-Fei, L. 2011. Novel Dataset for Fine-Grained Image Categorization. In *CVPR Workshops*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kolesnikov, A.; Zhai, X.; and Beyer, L. 2019. Revisiting self-supervised visual representation learning. In *CVPR*, 1920–1929.

Komodakis, N.; and Gidaris, S. 2018. Unsupervised representation learning by predicting image rotations. In *ICLR*.

Kornblith, S.; Shlens, J.; and Le, Q. 2019. Do Better ImageNet Models Transfer Better? In *CVPR*, 2661–2671.

Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3d object representations for fine-grained categorization. In *ICCV Workshops*, 554–561.

Laine, S.; and Aila, T. 2016. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*.

Lee, H.; Hwang, S. J.; and Shin, J. 2020. Self-supervised label augmentation via input transformations. In *ICML*, 5714–5724.

Liang, H.; Wen, G.; Hu, Y.; Luo, M.; Yang, P.; and Xu, Y. 2021. MVANet: Multi-Task Guided Multi-View Attention Network for Chinese Food Recognition. *IEEE TMM*, 23: 3551–3561.

Liu, C.; Liang, Y.; Xue, Y.; Qian, X.; and Fu, J. 2020. Food and ingredient joint learning for fine-grained recognition. *IEEE TCSVT*, 31(6): 2480–2493.

Liu, X.; and Wang, L. 2022. Transformer with peak suppression and knowledge guidance for fine-grained image recognition. *Neurocomputing*, 492: 137–149.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 10012–10022.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2): 91–110.

Martinel, N.; and Foresti, G. L. 2018. Wide-slice residual networks for food recognition. In *WACV*, 567–576.

Martinel, N.; Piciarelli, C.; Micheloni, C.; and Luca Foresti, G. 2015. A structured committee for food recognition. In *ICCV Workshops*, 92–100.

Meyers, A.; Johnston, N.; Rathod, V.; Korattikara, A.; Gorban, A.; Silberman, N.; Guadarrama, S.; Papandreou, G.; Huang, J.; and Murphy, K. P. 2015. Im2Calories: towards an automated mobile vision food diary. In *ICCV*, 1233–1241.

Min, W.; Liu, L.; Luo, Z.; and Jiang, S. 2019. Ingredient-Guided Cascaded Multi-Attention Network for Food Recognition. In *ACM MM*, 1331–1339.

Min, W.; Liu, L.; Wang, Z.; Luo, Z.; Wei, X.; Wei, X.; and Jiang, S. 2020. Isia food-500: A dataset for large-scale food recognition via stacked global-local attention network. In *ACM MM*, 393–401.

Nag, N.; Pandey, V.; and Jain, R. 2017. Health multimedia: Lifestyle recommendations based on diverse observations. In *ICMR*, 99–106.

Noroozi, M.; and Favaro, P. 2016. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 69–84.

Sergey, Z.; and Nikos, K. 2016. Wide Residual Networks. In *BMVC*, 87.1–87.12.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *CVPR*, 1–9.

Tan, M.; and Le, Q. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 6105–6114.

Tian, Y.; Krishnan, D.; and Isola, P. 2020. Contrastive Representation Distillation. In *ICRL*.

Touvron, H.; Sablayrolles, A.; Douze, M.; Cord, M.; and Jégou, H. 2021. Grafit: Learning fine-grained image representations with coarse labels. In *ICCV*, 874–884.

Tung, F.; and Mori, G. 2019. Similarity-preserving knowledge distillation. In *ICCV*, 1365–1374.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The caltech-ucsd birds-200-2011 dataset. *California Institute of Technology*.

Wang, Z.; Min, W.; Li, Z.; Kang, L.; Wei, X.; Wei, X.; and Jiang, S. 2022. Ingredient-Guided Region Discovery and Relationship Modeling for Food Category-Ingredient Prediction. *IEEE TIP*, 31: 5214–5226.

Wu, Z.; Xiong, Y.; Yu, S. X.; and Lin, D. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 3733–3742.

Yang, Z.; Luo, T.; Wang, D.; Hu, Z.; Gao, J.; and Wang, L. 2018. Learning to Navigate for Fine-Grained Classification. In *ECCV*, 438–454.

Yin, C.; Yang, S.; Chen, S.; Andrew, H.; and Serge, B. 2018. Large Scale Fine-Grained Categorization and Domain-Specific Transfer Learning. In *CVPR*, 4109–4118.

Yu, F.; Wang, D.; Shelhamer, E.; and Darrell, T. 2018. Deep Layer Aggregation. In *CVPR*, 2403–2412.

Zhang, Y.; Xiang, T.; Hospedales, T. M.; and Lu, H. 2018. Deep mutual learning. In *CVPR*, 4320–4328.

Zhao, B.; Cui, Q.; Song, R.; Qiu, Y.; and Liang, J. 2022. Decoupled Knowledge Distillation. In *CVPR*, 11953–11962.