Recognition from Web Data: A Progressive Filtering Approach

Jufeng Yang, Xiaoxiao Sun, Yu-Kun Lai, Liang Zheng and Ming-Ming Cheng

***Abstract*—Leveraging the abundant number of web data is a promising strategy in addressing the problem of data lacking when training convolutional neural networks (CNNs). However, web images often contain incorrect tags, which may compromise the learned CNN model. To address this problem, this paper focuses on image classification and proposes to iterate between filtering out noisy web labels and fine-tuning the CNN model using the crawled web images. Overall, the proposed method benefits from the growing modeling capability of the learned model to correct labels for web images and learning from such new data to produce a more effective model. Our contribution is two-fold. First, we propose an iterative method that progressively improves the discriminative ability of CNNs and the accuracy of web image selection. This method is beneficial towards selecting high-quality web training images and expanding the training set as the model gets ameliorated. Second, since web images are usually complex and may not be accurately described by a single tag, we propose to assign a web image multiple labels to reduce the impact of hard label assignment. This labeling strategy mines more training samples to improve the CNN model. In the experiments, we crawl 0.5 million web images covering all categories of four public image classification datasets. Compared with the baseline which has no web images for training, we show that the proposed method brings notable improvement. We also report competitive recognition accuracy compared with the state of the art.**

***Index Terms*—Noisy web data, CNN, Progressive Filtering, Multiple Labels**

I. Introduction

THE success of convolutional neural networks (CNNs) is owing to sufficient labeled training data. However, labeling millions of images manually is very time-consuming and laborious, which can be practically impossible for many problems where a high level of expertise is needed. Lack of data is arguably the most significant obstacle to developing deep models for new tasks, so it is highly desirable and sometimes even necessary to train an effective CNN model with limited well-labeled data. To tackle this problem, some works [1], [2] consider the transferability of CNNs. They initialize parameters with a pre-trained model (*e.g.*, AlexNet [3], VGGNet [4] *etc.*) on the large-scale dataset ImageNet [5], and then fine-tune the model on the specific dataset. Meanwhile, as an alternative approach, some recent works [6], [7] have shown that fine-tuning a CNN model with extensive web data can be more effective than fine-tuning it merely on a small-scale clean dataset. Such works harvest images from the Internet and associate them with labels either from the keywords originally used for retrieval or tags extracted from their description texts.

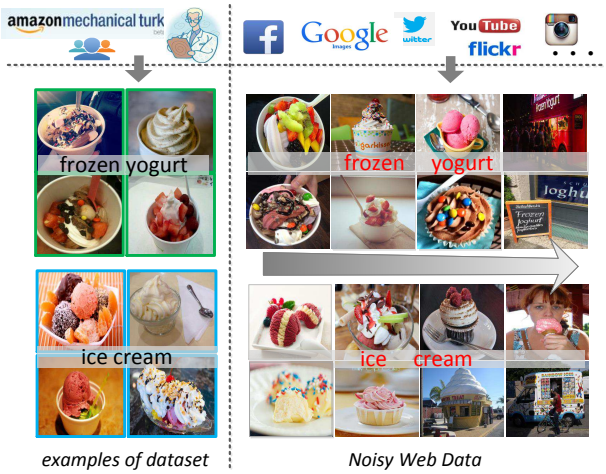


Fig. 1. Examples of the Food-101 dataset (left) and noisy web data (right) from the classes *ice cream* and *frozen yogurt*. On the left are images from the public dataset, and the right shows the images collected from the web by searching with the category names, listed in the decreasing order of reliability. We observe that images in the two middle columns of the web images are ambiguous *w.r.t.* their categories, and those in the rightmost column are outliers.

The major challenge of using web data is two-fold: given a corpus of web data, it is critical to retrieve images and their tags 1) accurately and 2) as many as possible. Without doing so, the selected web data may not accurately reflect the image content or have a limited number. The two challenges arise from the fact that web images have complex content and that the data distribution of web images can be distinct from the target dataset.

On the one hand, the content of web images is usually complex. Given its complex nature, it is likely that an image description does not accurately reflect the true content of the image [8]–[10]. For example, as demonstrated in Fig. 1, it poses difficulties to tell whether an image contains an ice cream or a frozen yogurt. In response to the inaccurate tags, Krause *et al.* [11] remove images that appear in search results for more than one category to reduce the effect of noise. Vo *et al.* [12] filter images by ranking the predicted results of classifiers trained on features exacted from a CNN. Nevertheless, such methods do not consider the complex

Manuscript received December 05, 2017; revised May 09, 2018; accepted June 18, 2018.

J. Yang, X. Sun and M.-M. Cheng are with College of Computer and Control Engineering, Nankai University, Tianjin, 300350, China. (Email: yangjufeng@nankai.edu.cn, sunxiaoxiaozrt@163.com, cmm@nankai.edu.cn)

Y.-K. Lai is with School of Computer Science and Informatics, Cardiff University, Wales, UK. (Email: Yukun.Lai@cs.cardiff.ac.uk)

L. Zheng is with Research School of Computer Science, Australian National University, Canberra, Australia. (Email: liangzheng06@gmail.com)

relationships between the tag and the content of web images. Specifically, real-world images contain richer content than images from public datasets. Therefore, a single tag may not be able to sufficiently describe the information of an image or the object related to the target task, especially when the image contains objects of multiple similar categories at the same time. However, since the label of real-world images is difficult to distinguish, forcing a single label loses potentially useful information, especially when the probabilities of an image belonging to multiple categories are similar.

On the other hand, the data distribution of web images can be so different from the target dataset that the basic model trained from the well-labeled target data makes incorrect predictions on the web data. When prediction error happens, the corresponding web data should be discarded to prevent error propagation. Meanwhile, without addressing the data distribution problem, it is the easier training samples that are more likely to be selected; the challenging training samples which are more useful for model robustness may well be missing, no matter how large the volume of the web data is. Therefore, the challenge of different data distributions may impair the full utilization of web data. In the literature, in an attempt to address this problem, a fine-tuning process can be conducted on the web data [6], [7]. This strategy however does not directly address the transfer problem [13] and thus does not make full use of the web data.

In light of the above discussions, this paper introduces a progressive learning approach to selecting possibly many web images with accurate labels. First, to address the data distribution gap, we propose to iterate between classifier fine-tuning and web tag estimation. Usually, model performance tends to improve with an increasing amount of training data. However, if we add a large volume of web data directly into well-labeled data to train a CNN model, the model will fit better with the large amount of web data rather than the well-labeled target dataset due to their different distributions [13]. In this paper, we first incorporate "high-quality" web data selected by the basic model in the training set to fine-tune the CNN model. Then, the fine-tuned model is in turn used to select web data. This process is applied iteratively. In the beginning, a few easy samples are selected from the web data. As the iterations progress, the CNN model grows stronger and more diverse and challenging web data can be selected, which further contributes to model refinement. In other words, compared with a static model, our model is refined iteratively with the filtered web data, and the model in turn provides a more reliable judgment for web data selection. Therefore, web data can be better utilized for CNN training.

Second, to address the problem of tag noise, we assign multiple labels to a web image to reduce the impact of hard label assignment. Specifically, given a web image, up to K class labels with the highest prediction scores are assigned to this image. If such labels have the similar confidence or if the label with the highest score is consistent with the original web tag of the image, we select this image as a new training sample. Experimental results indicate that, by using web data effectively, our method improves the recognition accuracy of CNN models effectively.

In summary, this paper claims three contributions.

- We propose an iterative filtering method that progressively learns from web data to improve the performance of CNN model.
- Instead of hard label assignment, images are assigned with multiple labels, which increases the recall of web training data selection.
- We have collected four web image datasets in correspondence to four public classification tasks. Extensive experiments demonstrate that our method yields competitive recognition accuracy against the state-of-the-art approaches.

II. Related Work

A. Deep Learning from Web Data

The training of deep models requires a large amount of well-labeled data, but they are generally expensive to obtain. To address this problem, recent works consider learning from web data, which is relatively convenient to obtain and contains a considerable level of visual information.

In this area, impressive improvement has been observed [6], [11], [12], [14]–[17]. For web data collections, Chen *et al.* [14] use a semi-supervised learning algorithm to find the relationships between common sense and labeled images of given categories. Schroff *et al.* [16] propose an automatic method for gathering hundreds of images for a given query class. These two works try to build visual datasets with minimum human effort, but the resulting datasets contain noisy labels. Meanwhile, the introduction of web data also improves the performance of deep models, which is verified by recent work [18]. To scale up to ImageNet-sized problems, Izadinia *et al.* [19] perform direct learning from image tags in the wild and achieve improved performance. Chen and Gupta [6] present a two-stage approach to training deep models by exploiting both noisy web data and the transferability of CNN. Alternatively, Xiao *et al.* [17] use a probabilistic framework to model the relationships among images, clean labels and noisy labels, and then train a model in an end-to-end structure. In addition, Krause *et al.* [11] download images from Google to form training sets for different tasks and filter such data with the human in the loop. They demonstrate the effectiveness of using noisy web data and the benefits of performing extra operations on noisy data, *e.g.*, filtering.

In this work, we also use web data for CNN training but we design a progressive learning framework. With reliable web data incrementally added, the classification capability of the learned model gradually improves. In the meantime, the improved model provides more robust and accurate predictions for web data. Further, in web data selection, we replace the previous one-to-one label assignment with a one-to-many strategy. This method aims to obtain more challenging and diverse training data from web images to train discriminative CNNs.

B. Progressive Learning

The strategy of progressive learning has been used in data mining [20], pattern recognition [21], computer vision [22]–[24], *etc*. When training data increases gradually, a progressive

learning method allows the data analysis systems to have the capability to learn progressively when new data is needed. Meanwhile, it is natural that knowledge is taught from easy to difficult in the form of a curriculum, leading to a learning paradigm called curriculum learning [25]. Later, self-paced learning [26] is proposed to embed easiness identification into the learning objective of curriculum learning.

This progressive paradigm has been used in many tasks. For example, Ma *et al.* [27] propose an alternative optimization process for an optimization model of self-paced curriculum learning, which effectively improves the standard co-training algorithm. Dong *et al.* [28] propose a method that iterates between model training and high-confidence sample selection, which obtains improvement in few-shot object detection. Fan *et al.* [23] improve the person re-identification [29] results by using a progressive method to train and rectify the model representation and clustering alternately. The iterative learning scheme has also been used in previous works for image clustering, in which the model is trained iteratively between clustering and representation learning [30]–[32]. For example, Yang *et al.* [30] learn representations and image clusters from an unlabeled image set, and optimize the two tasks iteratively. The two tasks interact with each other and are improved simultaneously, which results in a better unsupervised representation and well-separated clusters. Chang *et al*. [32] use Deep Adaptive Clustering to calculate the similarities of pairwise images based on the feature from a CNN, sample similar images, and then train the CNN by using the selected samples to improve the representation performance.

Inspired by the thought of iterative optimization and incremental improvement, the proposed progressive filtering approach exploits the relationship between the training of the model and distinguishing of noisy web data to gradually learn from the selected reliable web data. Different from these methods for image clustering, the proposed method not only needs to select reliable training data, but also needs to handle the content complexity of web images and the different data distributions during training. In general, each iteration of progressive learning can be treated as an optimization problem with the weights of the model updated as new data is fed in. In this work, we employ progressive learning to improve the performance of a CNN model when using web images with noisy tags. In each iteration, the recognition performance of the model is first improved by selected web data. Then, we utilize the recognition power of the progressively enhanced model to filter and correct web data, which contains new information and can be seen as "new data", although some of these images may have already been seen by the model. This method is more efficient than offline learning based methods where the entire set of web images is used to train the CNN model.

C. Web Datasets

Based on well-labeled visual datasets, several representative web image datasets have been established, which are summarized in Table I, in which the publicly available datasets are shown in **bold**. To meet the demand of different tasks, these datasets are collected with various scales from different web sources, *e.g*., Flickr, Instagram, *etc*. Krause *et al.* [11] collect datasets from Google Images and conduct two rounds of cleaning: active learning and human in the loop. Xiao *et al.* [17] establish a clothing dataset by searching images from shopping websites, which has both noisy labels and clean labels obtained by manual refinement. Xu *et al.* [37] use 200 bird species (CUB-200-2011) as keywords and download the top 100 images of search results. Web data collection is related to the technique of image retrieval [38]–[41], but the existing engines are mainly based on traditional keyword search techniques. Similar collection methods are also used in works [12], [19], [35], [36], [42] which utilize web data for specific tasks.

TABLE I
SOME EXISTING WEB DATASETS. THEY ARE COLLECTED FROM DIFFERENT SOURCES AND USED FOR CNN TRAINING. NOTE THAT THE YOUTUBE-8M DATASET IS COLLECTED FROM 8 MILLION URLS AND CONTAINS 1.9 BILLION VIDEO FRAMES.

Dataset	Year	# imgs	Source
Flickr-CIFAR [15]	2014	230173	Flickr
Flickr [6]	2015	1.2M	Flickr
Google [6]	2015	1.5M	Google
YFCC100M [19]	2015	99.3M	Flickr
Clothing [17]	2015	100000	Shopping sites
Sketch [33]	2016	191067	Google
Openimages [34]	2016	9.01M	Google
M-Flower-620 [35]	2016	20211	Instagram
YouTube-8M [36]	2016	1.9B	YouTube
Weakly (Bird) [37]	2016	200000	Flickr
Goldfince [11]	2016	9.8M	Google
Flickr-Bing 100 [12]	2017	416000	Flickr & Bing
Flickr-Bing 1K [12]	2017	3.12M	Flickr & Bing

In our work, to analyze the application of web data for classification tasks, we collect four new web datasets from a different perspective. We extend some manually labeled datasets with web data from Google Images, Flickr and Twitter, which cover classification tasks of a diverse range of targets, including objects (food and dog), scenes, and skin diseases. According to our observation, different resources (search engines, social network sites, specific web pages) have images with different characteristics. Therefore, web data from a specific web source may be more suitable for certain tasks than others. This observation will be evaluated in our experiment, which also demonstrates that different data distributions exist not only in target and web data, but also in different web resources. Furthermore, our method can suppress the impact of the different data distributions by progressive learning from web data.

III. METHOD

In this work, our goal is to effectively utilize easily obtained web images with noisy labels to train a CNN model for image classification.

We start our method with the traditional CNN fine-tuning process. Given a clean target dataset $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N}$, we

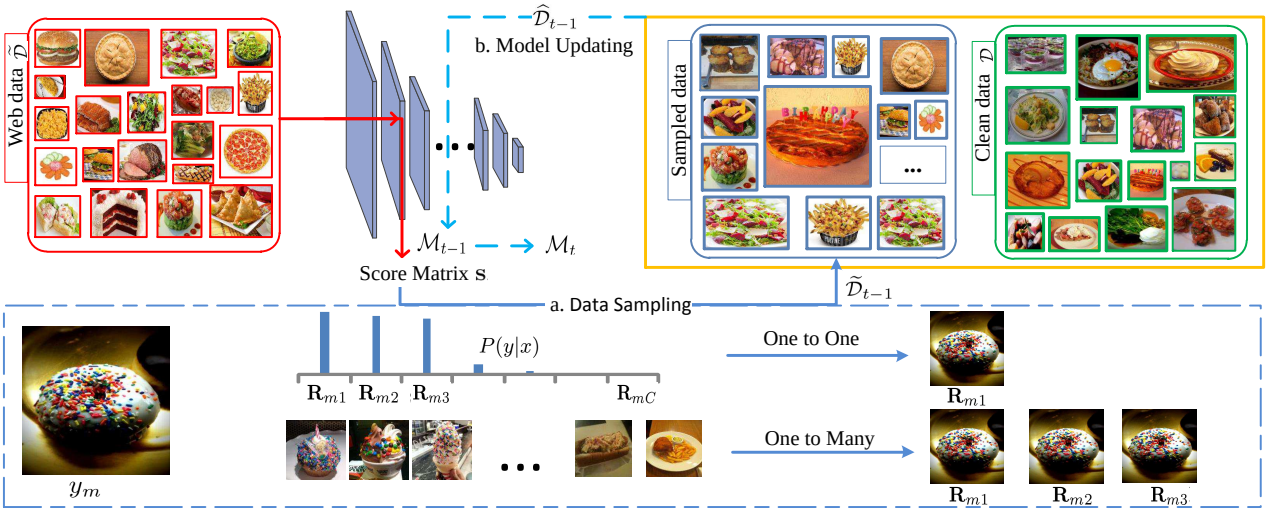


Fig. 2. Pipeline of the proposed progressive learning method, which includes two major steps, namely data sampling and model updating. For data sampling, we obtain $P(y|x)$, score matrix $\mathbf{S}$ and label matrix $\mathbf{R}$ of web data based on model $\mathcal{M}_{t-1}$, and use such information to obtain the dataset $\widetilde{\mathcal{D}}_{t-1}$ through a sampling scheme. Here, one to one and one to many are two types of label sampling strategies. For model updating, model $\mathcal{M}_t$ is initialized with the parameters of $\mathcal{M}_{t-1}$ and updated on the combined dataset $\widehat{\mathcal{D}}_{t-1}$ in the t-th iteration.

denote each data point as a D-dimensional vector $x_n \in \mathbb{R}^D$, and its label as $y_n \in \{1, 2, \cdots, C\}$. Here, N indicates the size of dataset $\mathcal{D}$, and C denotes the number of categories. We first train a CNN model $\mathcal{M}_0 : f(x; \boldsymbol{\theta}) \in \mathbb{R}^C$ using the clean dataset $\mathcal{D}$, where $\boldsymbol{\theta}$ represents the set of model parameters. Usually, $\boldsymbol{\theta}$ in $\mathcal{M}_0$ is initialized with a set of weights $\boldsymbol{\theta}_{pre}$ from a pre-trained model $\mathcal{M}_{pre}$, and then updated using stochastic gradient descent (SGD). As mentioned above, the data volume in $\mathcal{D}$ may be limited, so we employ a web dataset $\widetilde{\mathcal{D}}$ to improve the CNN model. The iteration of SGD updates current parameters $\boldsymbol{\theta}^*$ as:

$$\boldsymbol{\theta}^* = \boldsymbol{\theta} + \widehat{\gamma} \cdot \frac{1}{|\mathcal{D}^b|} \sum_{(x,y)\in\mathcal{D}^b} \nabla_{\boldsymbol{\theta}} \left[L(x, y) \right], \tag{1}$$

where $L(x, y)$ is a loss function, *e.g.*, softmax. $\nabla_{\boldsymbol{\theta}}$ is computed by gradient back-propagation. $\mathcal{D}^b$ is a mini-batch randomly drawn from the training dataset $\mathcal{D}$, and $\widehat{\gamma}$ is the learning rate.

A. Progressive Learning from Noisy Web Labels

The noisy dataset $\widetilde{\mathcal{D}} = \{(x_m, y_m)\}_{m=1}^M$ contains image/label pairs: the m-th image x_m and its corresponding web tag y_m. Here, M indicates the size of $\widetilde{\mathcal{D}}$. Fig. 2 shows the pipeline of our method, consisting of the following two major steps:

Web data selection for CNN training. In our method, we first select images with reliable tags to train our model, by utilizing the confidence of the tags. Specifically, given an instance x and the set of labels $\boldsymbol{l} = [1, 2, \cdots, C]$, the confidence for each label is $p(l_i|x), i = 1, ..., C$, where a higher probability corresponds to higher confidence. We use this information to select and correct labels for web images. Methods that can be used to estimate $p(l_i|x)$ include SVM, softmax, Bayes classifiers, *etc*. Considering the high discriminative ability of deep features, we use a softmax function to generate the probability for each web image x_m:

$$p(l_i|x_m) = \frac{e^{\boldsymbol{\theta}_i^\top x_m}}{\sum_j e^{\boldsymbol{\theta}_j^\top x_m}}. \tag{2}$$

Then, we obtain the confidence scores for the C categories $\boldsymbol{s}_m = \{p(l_i|x_m)\}_{i=1}^C$, and rank $\boldsymbol{s}_m$ in descending order. We also record the labels $\boldsymbol{r}_m$ corresponding to $\boldsymbol{s}_m$. For the web dataset $\widetilde{\mathcal{D}}$, we obtain its score matrix $\mathbf{S}=\{\boldsymbol{s}_m\}_{m=1}^M$, and the corresponding label matrix $\mathbf{R} = \{\boldsymbol{r}_m\}_{m=1}^M$.

For a web image x_m with tag y_m, if the predicted label is the same as the web tag, the web tag is preserved and x_m is selected for CNN training. Under the circumstance where the predicted label is different from the web tag, if the prediction confidence is higher than a threshold, we use the predicted label to replace the web tag, and x_m is selected for CNN training. Otherwise, x_m is rejected for training. As such, the corrected label $\widehat{y}_m$ is determined by

$$\widehat{y}_m = \begin{cases} y_m, & y_m = \mathbf{R}_{m1} \\ \mathbf{R}_{m1}, & y_m \neq \mathbf{R}_{m1}, \mathbf{S}_{m1} > \varepsilon \\ \emptyset, & \text{otherwise} \end{cases} \tag{3}$$

where ε is a threshold for label correction, and $\mathbf{R}_{m1}$ is the label with the largest confidence score $\mathbf{S}_{m1}$. In our experiments, ε is set to the classification accuracy on the validation dataset (see discussions in Section IV-D). $\widehat{y} = \emptyset$ means that the image will not be used for training, because it is considered as unreliable data. We employ Eq. 3 to select images for CNN fine-tuning.

Model updating. The progressive learning will update the model when new data is added. In the t-th iteration of our progressive learning, we initialize the weights of $\mathcal{M}_t$ on the model $\mathcal{M}_{t-1}$ and train $\mathcal{M}_t$ on the mixed dataset $\widehat{\mathcal{D}}_{t-1} = \mathcal{D} \cup \widetilde{\mathcal{D}}_{t-1}$, where $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^N$ is the clean

Algorithm 1 Progressive Learning from Web Data

Input:

Clean dataset: $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N}$;
The noisy web dataset: $\widetilde{\mathcal{D}} = \{(x_m, y_m)\}_{m=1}^{M}$;
The initialized network model $\mathcal{M}_{pre} : f(x; \boldsymbol{\theta}_{pre}) \in \mathbb{R}^C$.

Fine-tune a CNN model $\mathcal{M}_0$ based on $\mathcal{M}_{pre}$ using $\mathcal{D}$;
Calculate $\mathbf{S}$ and $\mathbf{R}$ for $\widetilde{\mathcal{D}}$ using model $\mathcal{M}_0$ with Eq. 2;
Update $\widetilde{\mathcal{D}}$ to obtain $\widetilde{\mathcal{D}}_0$ with Eq. 5;
$t \longleftarrow 0$;
repeat
  $t \longleftarrow t + 1$;
  repeat
    Update parameters $\boldsymbol{\theta}_t^*$ with Eq. 4 on each mini-batch $\widehat{\mathcal{D}}_{t-1}^b$, $\widehat{\mathcal{D}} = \mathcal{D} \cup \widetilde{\mathcal{D}}_{t-1}$;
  until loss function $L(x, y)$ has converged.
  Obtain model $\mathcal{M}_t$;
  Calculate $\mathbf{S}$ and $\mathbf{R}$ for $\widetilde{\mathcal{D}}$ using $\mathcal{M}_t$ based on Eq. 2;
  Update the dataset $\widetilde{\mathcal{D}}$ to obtain $\widetilde{\mathcal{D}}_t$ with Eq. 5;
until $\widetilde{\mathcal{D}}_t$ tends to be stable or the performance of $\mathcal{M}_t$ does not improve.

Output:

The trained model: $\mathcal{M}_t : f(x; \boldsymbol{\theta}_t) \in \mathbb{R}^C$.

dataset, $\widetilde{\mathcal{D}}_{t-1} = \{(x_m, \widehat{y}_m)\}_{m=1}^{\widetilde{M}_{t-1}}$ is the sampled web dataset of $(t-1)$-th iteration and $\widetilde{M}_{t-1}$ denotes the size of $\widetilde{\mathcal{D}}_{t-1}$. The parameters $\boldsymbol{\theta}_t^*$ of the model $\mathcal{M}_t$ is updated as follows,

$$\boldsymbol{\theta}_t^* = \boldsymbol{\theta}_t + \widehat{\gamma} \cdot \frac{1}{\left|\widehat{\mathcal{D}}_{t-1}^b\right|} \sum_{(x,y) \in \widehat{\mathcal{D}}_{t-1}^b} \nabla_{\boldsymbol{\theta}_t} \left[L(x, y)\right], \tag{4}$$

where $\boldsymbol{\theta}_t$ is initialized with the weights $\boldsymbol{\theta}_{t-1}$ of the model $\mathcal{M}_{t-1}$. The training of the model will be finished when the model is converged. Specifically, the iterative process of progressive learning repeats until either updated dataset $\widetilde{\mathcal{D}}_t$ is stable or the model $\mathcal{M}_t$ is not improved compared with $\mathcal{M}_{t-1}$.

B. One-to-Many Correction for Noisy Labels

As mentioned in Section I, the content of web images is usually complex. As illustrated at the bottom of Fig. 2, a web image may contain multiple objects and the categories of these objects are hard to disambiguate. Previous works usually employ a one-to-one label assignment for web images and select those images whose tag equals their assigned label. However, one-to-one label assignment and comparison only compare the image tag y_m with the top ranked label to obtain the final label, which may be inaccurate or may lose useful domain information for classification. To address this challenge, we replace the label assignment method of Eq. 3 with one-to-many label assignment,

$$\widehat{y}_m = \begin{cases} \mathbf{R}_{m1}, & \text{case 1} \\ \{\mathbf{R}_{mi}\}_{i=1}^{k}, & \text{case 2} \\ \emptyset, & \text{otherwise} \end{cases} \tag{5}$$

where k denotes the number of labels that are assigned to x_m. The situation belongs to case 1 or case 2 if and only if k satisfies the constraint $k \leq K$, where K is the maximum number of labels that are assigned to each web image. Otherwise, the image is excluded from the current training iteration, because the image is too ambiguous. Furthermore, case 1 requires $y_m = \mathbf{R}_{m1}$ or $\{y_m \neq \mathbf{R}_{m1}, \mathbf{S}_{m1} > \varepsilon\}$. That it, the top ranked label either matches the web tag or is sufficiently confident. Case 2 requires $\{y_m \neq \mathbf{R}_{m1}, \mathbf{S}_{m1} \leq \varepsilon\}$, and k is the maximum value that satisfies $\mathbf{S}_{m1} - \mathbf{S}_{mk} < \varepsilon/k$. That is the top k labels have sufficiently close confidence. Our method is summarized as Algorithm. 1.

TABLE II
STATISTICS OF THE TARGET AND WEB DATASETS. C IS THE NUMBER OF CLASSES. "TR/TE" DENOTES THE NUMBERS OF TRAINING AND TEST IMAGES IN EACH TARGET DATASET. WEB DATA REFERS TO THE IMAGES COLLECTED FROM THE INTERNET.

Dataset	Target Data			Web Data
	C	# imgs	tr/te	# imgs
SD-198 [43]	198	6,584	3,292 3,292	82,684
Stanford Dogs [44]	120	20,580	12,000 8,580	52,115
Food-101 [45]	101	101,000	75,750 25,250	240,096
MIT Indoor67 [46]	67	15,620	5,360 1,340	76,907

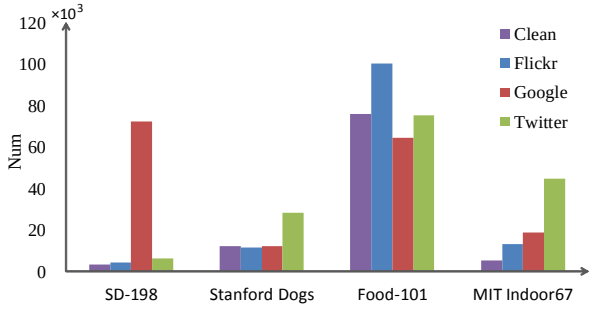


Fig. 3. Numbers of images collected from different sources, *i.e.*, Flickr, Google and Twitter. "Clean" is the data of the standard dataset.

IV. EXPERIMENTS

In this section, experiments are conducted on four classification tasks, including skin diseases, dogs, food and indoor scenes. The datasets and experimental setup are described in Section IV-A and Section IV-B. In addition, we analyze the quality of web data collected from different sources in Section IV-C. The discussion of important parameters is shown in Section IV-D, where we also evaluate the scale of clean training data. Finally, we evaluate the effectiveness of our method in Sections IV-E and IV-F.

A. Datasets

Following previous works on web data collection (Table I), we crawl images from Google Images, Flickr and Twitter. We show the total number of web images we collected for each

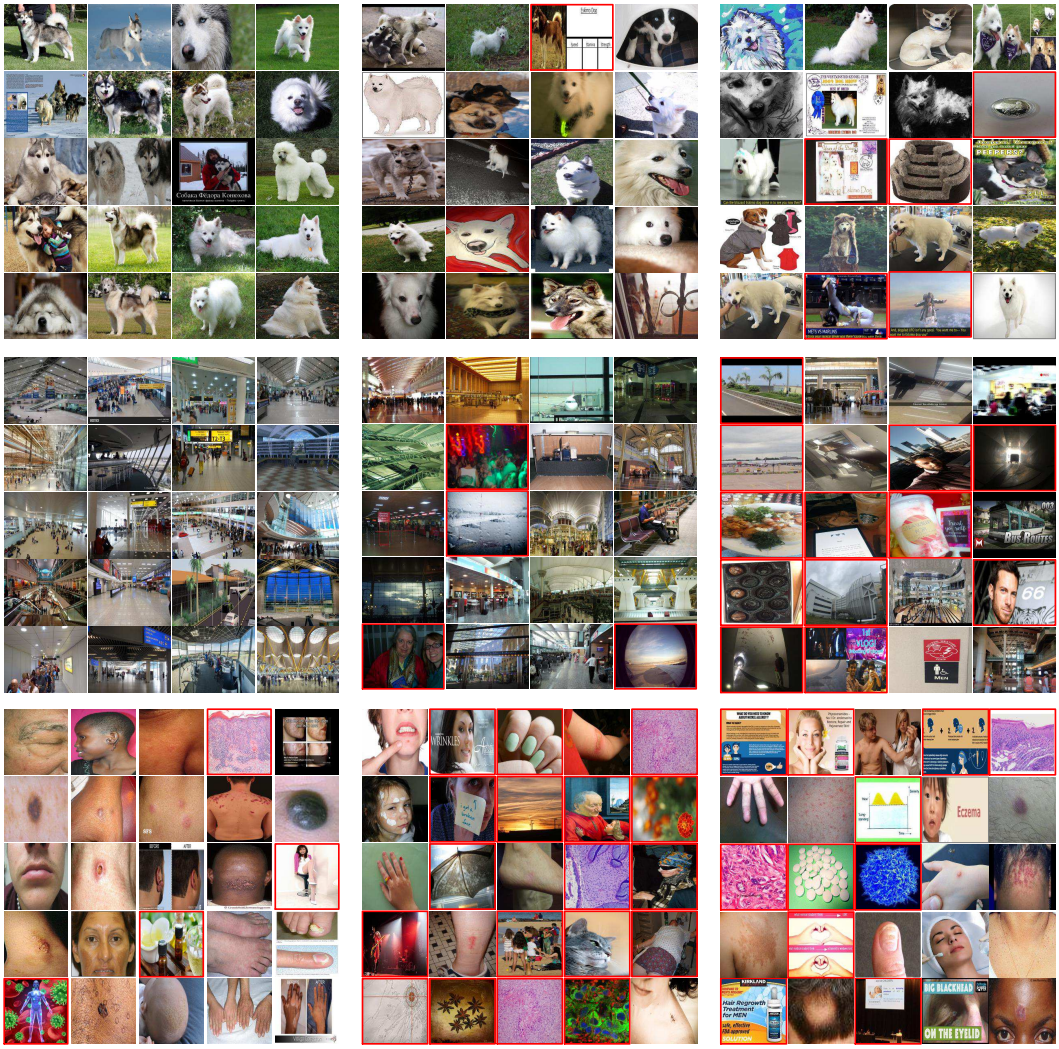


Fig. 4. Samples of collected web images of dogs (top), indoor scenes (middle) and skin diseases (bottom). All the images are retrieved by keyword search. Images with red boxes indicate they are outliers, which are almost irrelevant with classification tasks.

target dataset in Table II, and plot the numbers of images downloaded from different web sources in Fig. 3. In the downloading process, We first collect images by keyword search, where keywords correspond to the category labels in the public datasets. Then, we download images from the search results for the given class. Note that in our experiments the test datasets remain the same as the standard datasets: web images are only used for training. Meanwhile, to ensure fair comparisons, we remove the web images which are near duplicates of the images in the validation or the test sets. Moreover, we observe that familiar objects are easier to find and that these images contain less noise than uncommon things such as skin diseases (Fig. 4). However, the skin disease images may be more prone to tag errors, especially for data from social network sites. Therefore, the data sources will supply different quality of data for different tasks. In order to verify the analysis above, we evaluate different sources in Section IV-C.

B. Experiment Setup

Model. We mainly use four pre-trained models in the experiments, *i.e.*, AlexNet, CaffeNet, VGGNet and ResNet50, which exhibit good performance on many classification tasks [2], [47], fine-grained classification tasks [17], [48], [49], *etc*. These models are pre-trained on ImageNet [5]. The software package used in the experiments is Caffe [50]. Our models are trained using NVIDIA TITAN X GPUs. We set the mini-batch size to 64 for CaffeNet and AlexNet, 32 for VGGNet and 12 for ResNet50. We initialize the learning rate to 0.001 for food and dog classification, and 0.0001 for skin disease and indoor scene classification. The learning rate is reduced by a factor of 10 after 10 epochs. We keep training the model until convergence and set the max number of iterations to 40 epochs.

Label Ranking. We extract the features from fc8 layers of the model for web images, which is a C-dimensional feature vector representing the confidence of the predicted labels,

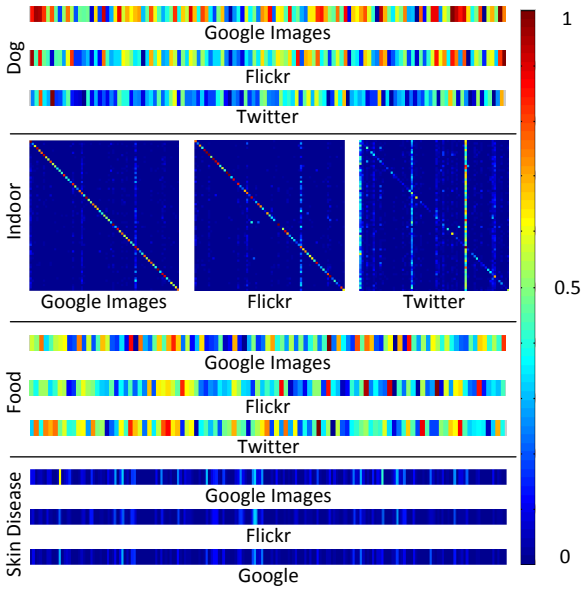


Fig. 5. Quality comparison of data from different sources. The quality of each category in the public datasets (Stanford Dogs, MIT Indoor67, Food-101 and SD-198) is shown using pseudo color (blue to red indicating worst to best). For indoor images, the confusion matrix of each source is visualized. These pictures are drawn based on the basic model fine-tuned from the VGGNet.

where C is the number of categories, *e.g.*, 120 for Stanford Dogs. In addition, we choose 10% of images as the validation data for each dataset, and set ε to the value of the classification accuracy on the validation data.

C. Comparison of Different Data Sources

In this section, we investigate the relationship between the target standard dataset and web sources. To evaluate the quality of web data, we predict the labels of web images by using the basic model $\mathcal{M}_0$ fine-tuned on the target data. The confusion matrix between the predicted labels and web tags provides an effective way to analyze the general quality of the collected datasets. We draw the confusion matrix of the web tag and predicted label on all the four tasks.

As shown in Fig. 5, We find that Google Images is generally of higher quality than the other two sources for *dog* images. This is probably because dog images from Flickr and Twitter are more likely to contain objects related to dogs but not dogs themselves, such as dog food and dog toys (see Fig. 4). The quality of *food* images from Twitter and Flickr is similar to that from Google Images due to the universality of food. According to the observation, incorrect search results of food include ingredients, menus, restaurants, *etc*. For the *indoor* dataset, we show the confusion matrix of each source in Fig. 5, in which Google Images is also of higher quality than Flickr and Twitter. Although scenes are very common in daily life, the indoor scene images from Flickr and Twitter are often selfies where the major part of the images is covered by people. The analysis is consistent with the typical examples presented in Fig. 4, where indoor scene images sampled from Google Images are better than the other two sources. Moreover, it is worth noticing that Google Images obtains less noisy retrieval results, especially in terms of outliers, which may benefit from the pre-processing performed by Google Image Search. Compared with the other two sources, the "good results" obtained by Google can also bring better performance for common classification tasks, *e.g.*, food recognition in Table III.

TABLE III
COMPARISON OF DIFFERENT WEB SOURCES ON INDOOR, FOOD, AND SKIN DISEASE DATASETS. WE DOWNLOAD IMAGES USING CLASS TAGS FROM TARGET DATASETS AND FINE-TUNE THE MODEL FROM VGGNET PRE-TRAINED ON IMAGENET. "ALL" MEANS IMAGES FROM ALL THE THREE SOURCES ARE USED.

Tasks		Google	Flickr	Twitter	All
Indoor	# imgs	19,176	13,360	44,892	77,428
	Acc (%)	**72.09**	71.19	70.90	72.01
Food	# imgs	64,444	100,314	75,311	240.069
	Acc (%)	75.57	76.83	76.13	**76.98**
Skin Disease	# imgs	74,774	4,273	6,171	85,218
	Acc (%)	51.26	47.68	46.82	**51.58**

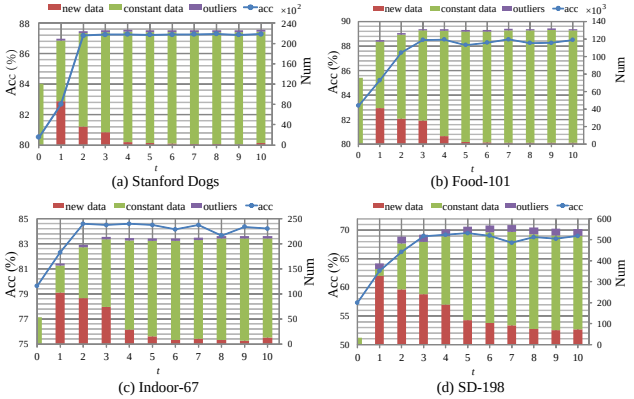


Fig. 6. Classification accuracies (Acc) of different iterations. Web images collected from Google are used to help with training of Resnet 50. "Num" denotes the number of training images, "new data" indicates the web images updated by the data sampling algorithm and one-to-many correction. "constant data" indicates the images used in both the current and previous iterations. The "outliers" are selected from the images which are included in the last training iteration but excluded in the current iteration due to their relatively low confidence scores.

However, the quality of the skin disease data from all the three sources is not good, because skin disease recognition needs more expert knowledge. Search engines may not process the data of this kind of tasks well, since except for the noisy labels, there are many outliers for skin diseases, which can be seen in Fig. 4, *e.g.*, drug advertisements, pathologic portrait and posters of prevention campaigns, *etc*.

We also compare the web sources by fine-tuning VGGNet using data of *food*, *indoor* and *skin disease* from the three sources. Table III shows the results. When utilizing web images for indoor scene classification, the accuracy of using Google Images alone is higher than the results of using web images from all the three sources, which indicates the negative

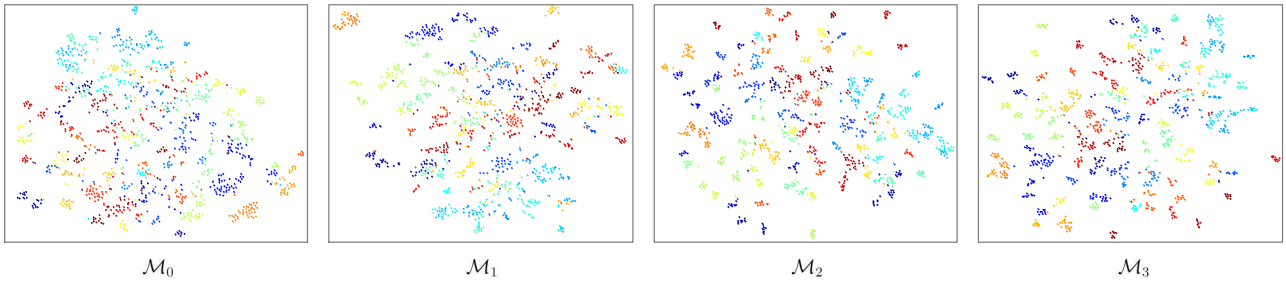


Fig. 7. Feature embedding visualizations of Stanford Dogs using t-SNE [51]. The features are exacted from the validation set based on $\mathcal{M}_0$, $\mathcal{M}_1$, $\mathcal{M}_2$ and $\mathcal{M}_3$, respectively.

TABLE IV
RECOGNITION ACCURACIES ON STANFORD DOGS + GOOGLE IMAGES WITH DIFFERENT K USING VGGNET. K IS THE MAXIMUM NUMBER OF ASSIGNED LABELS FOR EACH WEB IMAGE IN ONE-TO-MANY CORRECTION AND $K = 0$ CORRESPONDS TO CASE 1 IN ONE-TO-ONE CORRECTION. "NUM" MEANS THE NUMBER OF IMAGES USED FOR TRAINING. "PERCENTAGE" REPRESENTS THE PERCENTAGE OF USED IMAGES *w.r.t.* ALL THE WEB IMAGES. "NUM+" MEANS THE NUMBER OF IMAGES WITH MORE THAN ONE LABELS PRESERVED.

K	Num	Percentage	Num+	Acc (%)
0	6881	56.14	0	79.70
1	7020	57.28	0	80.23
2	8674	70.78	1654	**82.47**
3	9491	77.45	3288	82.39
4	9930	81.03	4650	82.43
5	10238	83.54	5837	81.98

effect of the other two sources of web data. The observations above can provide valuable insights for research that intends to employ web data for specific tasks. On the other hand, in order to demonstrate the robustness of the proposed method on processing and utilizing noisy web data, we employ all the web images from different sources in the following experiments in this paper.

D. Important Parameters and Evaluations

1) Parameter t of Iterations in Incremental Learning: In this section, we discuss the impact of iterations t in the proposed method. We use the basic model $\mathcal{M}_0$ to extract the fc8 features for all web data as the initial confidence scores of labels. To prevent over-fitting, we add some outliers in each round, which have high confidence in the previous iteration, but relatively low confidence in this round. The number of iterations t in our experiments is determined by the amount of updated data and the feedback (*i.e.*, loss, accuracy) of the training process.

In Fig. 6, we show the numbers of updated images as well as the recognition accuracies for 10 iterations on the Stanford Dogs, Food-101, Indoor-67 and SD-198 datasets. Take dog recognition as an example, the performance is significantly improved when a relatively large number of new images (red) are selected and used for training. In later rounds, when fewer new images are selected, the classification accuracy tends to be stable. These results show that the proposed method is capable of training a more accurate model from noisy data in 3–4 iterations. For the training process, except for $\mathcal{M}_1$ which is trained in about 12h (hours), $\mathcal{M}_2$ and $\mathcal{M}_3$ only need about 6h$-$9h on ResNet50. In our work, we set t to 3 which provides a good balance between accuracy and training efficiency.

Meanwhile, Fig. 7 visualizes the feature embedding of the features from the validation set of Stanford Dogs based on the model of different iterations. As can be seen, the discriminative capability of the features from $\mathcal{M}_0$, $\mathcal{M}_1$, $\mathcal{M}_2$ and $\mathcal{M}_3$ becomes better gradually along with the iteration. It is worth noting that the improvements are obvious for early iterations, *e.g.*, 1st and 2nd iterations. For later iterations, *e.g.*, the 3rd iteration, the change is limited due to fewer selected new imaged. Overall, compared with the model $\mathcal{M}_0$, the model after iterations has a more discriminative feature space for classification.

2) Parameter K: In this section, we discuss the effect of one-to-many correction for noisy web images. As previously mentioned, there exist tag ambiguities of similar categories for web images, so filtering and correcting web images by comparing web tags with predicted labels will result in severe information loss.

As shown in Fig. 6, in the iterative learning process, the improvement of classification accuracy is proportional to the change of training data. Using one-to-many correction instead of one-to-one intensifies the change, which is effective for model training. Table IV reports the effect of K on the Stanford Dogs dataset. $K = 0$ corresponds to case 1 in one-to-one correction, which is a strict constraint for selecting web data and gets a worse result than relaxing the constraint by adding case 2 in Eq. 5. As K increases, more training pairs (images and their corresponding labels) are included, which indicates that some web images attached with ambiguous labels are identified which may carry useful information. However, bigger K also means that more noise will be introduced. We set $K = 2$ in our experiments based on the results in Table IV for a good balance, which clearly outperforms one-to-one correction ($K = 1$) and can reduce the influence of hard label assignment.

3) Threshold ε in One-to-Many Correction: Threshold ε is used to control the selection of "high-quality" web data by comparing with the prediction results derived from the basic model $\mathcal{M}_0$. There are several elements which need to be considered when deciding ε: 1) if ε is set to a small value,

TABLE V
PERFORMANCE ON DIFFERENT SCALES OF CLEAN TRAINING SET ON STANFORD DOGS. THE NUMBER OF TRAINING IMAGES IN EACH SCALE IS $120 \times i$, WHERE $i = 1, 2 \cdots, 10$ (VERY SMALL SCALE); $20, 30, \cdots, 90$ (LARGER SCALE), IN WHICH i IS THE NUMBER OF IMAGES USED OF EACH CATEGORY. THE EXPERIMENTS ARE CONDUCTED ON RESNET50.

i	1	2	3	4	5	6	7	8	9	10	20	30	40	50	60	70	80	90
$\mathcal{D}$+ft	16.35	30.76	39.78	50.49	55.72	59.91	62.24	63.52	66.30	66.85	73.02	74.75	77.15	77.98	78.21	78.79	78.87	79.47
$\widehat{\mathcal{D}}$+ft	**74.58**	75.07	75.59	76.46	76.72	77.37	77.62	76.67	77.43	75.35	78.11	78.89	78.42	78.94	80.10	79.75	80.01	80.71
Ours	73.17	**76.64**	**77.21**	**79.99**	**80.44**	**80.78**	**80.38**	**80.81**	**80.74**	**81.26**	**82.38**	**82.53**	**83.01**	**83.56**	**84.12**	**85.33**	**85.79**	**86.34**

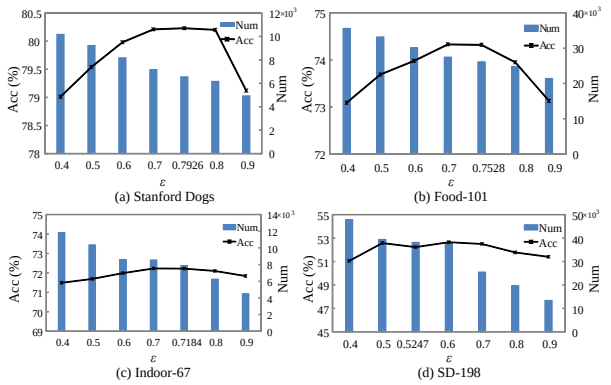


Fig. 8. Effectiveness of the parameter ε on Google data based on VGGNet in 1-st iterations. "Num" is the number of selected images by the progressive filtering method when ε is set to different values. "Acc" is classification accuracy on the validation set of the Stanford Dogs, Food-101, Indoor-67 and SD-198 dataset.

more training data will be used but the data is more noisy; 2) if ε is set to a large value, many images will be deleted, but the training data contains less noise; 3) the reliability of web label is related with the difficulty level of task, which has been discussed in Section IV-C. Therefore, ε should be a trade-off between the quantity and quality of the selected web data. We evaluate the performance of our method with changing ε in Fig. 8. As can be seen, the proposed method is not sensitive to ε when the values of ε belongs to an interval $[\varepsilon^* - \delta, \varepsilon^* + \delta], 0 \leq \delta < 0.1$ (ε^* represents the optimal solution). Meanwhile, we find from Fig. 8 that the classification accuracy of the basic model on the validation set is contained in the confidence interval (*i.e.*, 0.7926, 0.7528, 0.7184 and 0.5247 for the four tasks, respectively). Therefore, we set ε as the recognition performance on the validation data to adapt to each specific task. This explains the setting of parameter ε in Section III-A.

4) Evaluation of the scale of clean training data: Table V shows the results that the initial model $\mathcal{M}_0$ is trained on different amounts of training data from Stanford Dogs. As can be seen, the proposed method will begin to bootstrap the training when the clean training data increases slightly (compared with one image per category). For the smallest scale (120 images for dog recognition), it trains a basic model with an accuracy of only 16.34%, *i.e.*, the selection capability of this model is very limited. Therefore, the performance of the final model is largely decided by the distribution of web data rather than the clean target data. Meanwhile, different tasks have disparate properties and difficulties, so the initial amount of training data can also be different. Based on these results, we believe the proposed method can work effectively by using limited images, *e.g.*, fewer than 10 images per category, to train M_0, and this number of data is easy to collect in practice.

E. Analysis of Results for Different Tasks

In this section, we discuss the classification results on different tasks. With the setup and the parameters illustrated above, we conduct experiments on four datasets, and the results are shown in Table VI. The results of basic models are shown in Baseline, including models fine-tuned with clean data and noisy web data, where the performance of the model with web data is better than only using clean data except for skin disease classification (#1 and #2). In particular, for skin disease classification, the standard dataset is comparatively small and the web skin disease images have more noisy labels than other tasks. The accuracies of $D_{mix} + ft$ are lower than using clean data only on skin disease classification, which is consistent with the analysis in Section IV-C and the information shown in Fig. 3 and Fig. 4. For example, Fig. 4 shows the web skin disease images from Flickr and Twitter. They have a higher degree of label noise than other tasks, which results in lower accuracy on $D_{mix} + ft$. Compared with skin diseases, the quality of images from search engines and social network sites for food and dogs is acceptable because they are more common in daily life and people are also more willing to share this kind of images on the Internet.

Furthermore, the improvement of accuracy against the baseline for different tasks varies, *e.g.*, the improvement on Alexnet of skin diseases is around 4%, while for food classification, it is near 10%. Meanwhile, the experimental results on food and dogs conform to the expectation. For different models, adding web images can improve the performance of the model (#2, #13, #23 and #31 of food and dog). However, after simple filtering [53], [54], the accuracy may be reduced (*e.g.*, #14 and #32 of food, #24 of dog) because some useful images are wrongly removed. For the dog dataset, the first round filtering removes almost 40% web images, which contain many useful images with abundant domain information. This is also evidenced by results in the first two lines of Table IV, so the classification accuracy becomes worse after filtering. The proposed method can prevent the above case from happening and the results (Baseline+) are better than $\mathcal{D}_{filter}$+ft. Since, we employ a progressive filtering method to process web data

TABLE VI
ACCURACIES (%) ON FOUR DATASETS WITH DIFFERENT METHODS. $\mathcal{D}_{clean}$ AND $\mathcal{D}_{mix}$ REPRESENT THE TARGET DATASET AND THE MIXED DATASET, RESPECTIVELY. $\mathcal{D}_{filter}$ DENOTES THAT THE WEB DATA IS FILTERED FROM $\mathcal{D}_{mix}$. "FT" MEANS FINE-TUNING. IN THE LAST COLUMN, (+L-DOG) MEANS THE USED WEB DATA IS L-DOG [11]. BASELINE+A INDICATES THAT PROGRESSIVE FILTERING IS EMPLOYED, AND +B SHOWS ONE-TO-MANY CORRECTION IS USED (INSTEAD OF ONE-TO-ONE CORRECTION). "$*$" IS OBTAINED BY EMPLOYING THE MODIFICATION OF DOMAIN ADAPTATION MODEL [52] TO FURTHER REDUCE THE EFFECT OF DIFFERENT DATA DISTRIBUTION. HERE, WE REPLACE THE LOGISTIC REGRESSION LOSS WITH SOFTMAX LOSS (THE LOG-LIKELIHOOD LOSS WITH BINOMIAL CROSS-ENTROPY RESPECTIVELY).

#	Type	Method	Model	Test Accuracy				
				SD-198	Food-101	Dogs-120	Indoor-67	Dogs-120 (+L-Dog)
1	Baseline	$\mathcal{D}_{clean}$+ft	Alexnet	50.85	65.93	63.57	65.52	63.57
2		$\mathcal{D}_{mix}$ +ft		42.71	69.71	65.63	69.63	64.84
3		$\mathcal{D}_{filter}$+ft		50.92	69.89	67.95	63.58	66.16
4	Previous work	Bottom-up [53]		44.26	70.29	72.17	64.18	71.59
6		Pseudo-label [54]		44.02	69.36	70.32	63.88	71.01
7		Weakly [55]		45.44	71.10	73.00	64.85	73.64
8	Ours	Baseline+A		52.76	73.81	72.30	69.63	71.42
9		Baseline+B		51.06	70.64	69.43	65.47	69.35
10		Baseline+A+B		53.46	73.78	73.99	70.30	73.20
11		Baseline+A+B+$*$		**54.37**	**75.65**	**75.52**	**71.12**	**74.93**
12	Baseline	$\mathcal{D}_{clean}$+ft	Caffenet	51.34	66.61	63.19	65.22	63.19
13		$\mathcal{D}_{mix}$ +ft		44.74	69.25	66.08	67.99	65.34
14		$\mathcal{D}_{filter}$+ft		51.43	68.48	69.56	63.51	65.90
15	Previous work	Boosting [56]		45.99	72.53	73.49	65.60	73.28
16		PGM [17]		46.38	73.14	72.63	65.30	71.83
17		WSL [6]		46.99	73.21	73.52	65.60	73.79
18	Ours	Baseline+A		51.97	73.57	73.05	68.21	72.64
19		Baseline+B		51.55	71.13	72.34	65.45	71.22
20		Baseline+A+B		53.01	75.24	73.68	69.03	73.80
21		Baseline+A+B$*$		**53.95**	**76.92**	**75.74**	**71.19**	**75.63**
22	Baseline	$\mathcal{D}_{clean}$+ft	VGGNet	55.19	74.32	78.29	71.79	78.68
23		$\mathcal{D}_{mix}$ +ft		51.58	76.98	81.03	72.01	77.54
24		$\mathcal{D}_{filter}$+ft		53.31	78.24	79.70	72.16	79.57
25	Previous work	Harnessing [12]		54.50	79.02	78.45	70.00	78.31
26	Ours	Baseline+A		56.23	79.59	83.12	72.46	81.95
27		Baseline+B		55.47	78.71	82.47	72.24	80.66
28		Baseline+A+B		57.44	79.93	83.72	73.51	82.51
29		Baseline+A+B+$*$		**59.66**	**81.32**	**84.36**	**75.97**	**83.75**
30	Baseline	$\mathcal{D}_{clean}$+ft	Resnet50	57.35	83.14	80.51	79.63	80.51
31		$\mathcal{D}_{mix}$ +ft		54.22	85.21	81.43	82.31	82.07
32		$\mathcal{D}_{filter}$+ft		63.49	86.10	82.62	81.34	83.61
33	Previous work	Goldfince [11]		65.74	86.75	85.90	83.43	85.48
34	Ours	Baseline+A		65.67	88.58	84.57	82.54	84.57
35		Baseline+B		64.19	86.47	83.26	82.24	83.93
36		Baseline+A+B		67.25	88.96	85.93	83.58	85.69
37		Baseline+A+B+$*$		**70.56**	**89.77**	**87.36**	**84.78**	**86.94**

iteratively, the useful images have more chances to be chosen, and the model will be more robust to noise.

In contrast to dog images which have specific objects, indoor scene images have a wide variety of content which often contain salient people and other obstructions in the center of the images, so it is difficult to improve the performance of recognition with typical filtering strategies [53], [54]. However, the proposed algorithm can boost the classification accuracy on the indoor scenes dataset. As shown in Table VI, our method achieves an accuracy of 84.78%, outperforming other methods. Since the challenging data will be added gradually, the learned model can recognize complex scenes increasingly.

Finally, in order to verify the robustness of our method, we also conduct experiments on the L-Dog dataset [11], which is a publicly available noisy dataset for dog recognition. Note that, we only use a subset of L-Dog dataset, in which the categories are the same with the Stanford Dogs dataset. The results are consistent with those of our collected web data. Moreover, no matter which deep model (AlexNet, CaffeNet, VggNet, ResNet) is employed, the proposed algorithm shows its superiority consistently.

F. Comparison with the State-of-the-Art

In Table VII, we compare the proposed method with other state-of-the-art approaches. The proposed method performs favorably against other methods on different tasks. Specifically, for dog recognition, [11] employs multiple crops and

TABLE VII
COMPARISON WITH STATE-OF-THE-ART METHODS (INCLUDING THOSE USING WEB DATA, *e.g.*, [11]) ON FOUR PUBLIC DATASETS. BOLD VALUES CORRESPOND TO THE BEST ACCURACY (%) FOR EACH DATASET.

SD-198		Food-101		Stanford Dogs		MIT Indoor	
Method	Acc (%)	Method	Acc (%)	Method	Acc (%)	Method	Acc (%)
Caffe [43]	42.31	Random Forest [45]	50.76	NAC [57]	68.61	IFV+DMS [58]	66.87
VGG [43]	37.91	SNN [59]	69.90	FoF-Weakly [60]	71.40	FB/REF [12]	61.60
Caffe+ft [43]	46.69	DCNN [45]	56.40	PDFS [61]	71.96	CL-45C [62]	68.80
VGG+ft [43]	50.27	CNNFM [63]	58.49	FB/REF [12]	73.10	MLVED [64]	69.69
NPT [65]	52.19	DCNN+ft [45]	68.44	FOAF+ft [66]	74.49	Hybrid-CNN [67]	70.80
CSDR [68]	56.47	PTFT [69]	70.41	MagNet [70]	75.10	CNN+G [64]	70.46
Ours (Resnet50)	**70.56**	Im2Calories [71]	79.00	RED-OSSVR(vs) [72]	79.50	S-NN [59]	72.20
		ResNet50+ft	84.31	Weakly-S [73]	80.43	SFV [74]	72.86
		ResNet110+ft	84.88	Inception-v3 [11]	80.60	MPP+DSFL [75]	80.78
		Inception-v3 [76]	88.28	Goldfince [11]	85.90	Double fully hybrid [77]	80.97
		Ours (Resnet50)	**89.77**	Ours (Resnet50)	**87.36**	Ours (Resnet50)	**84.78**

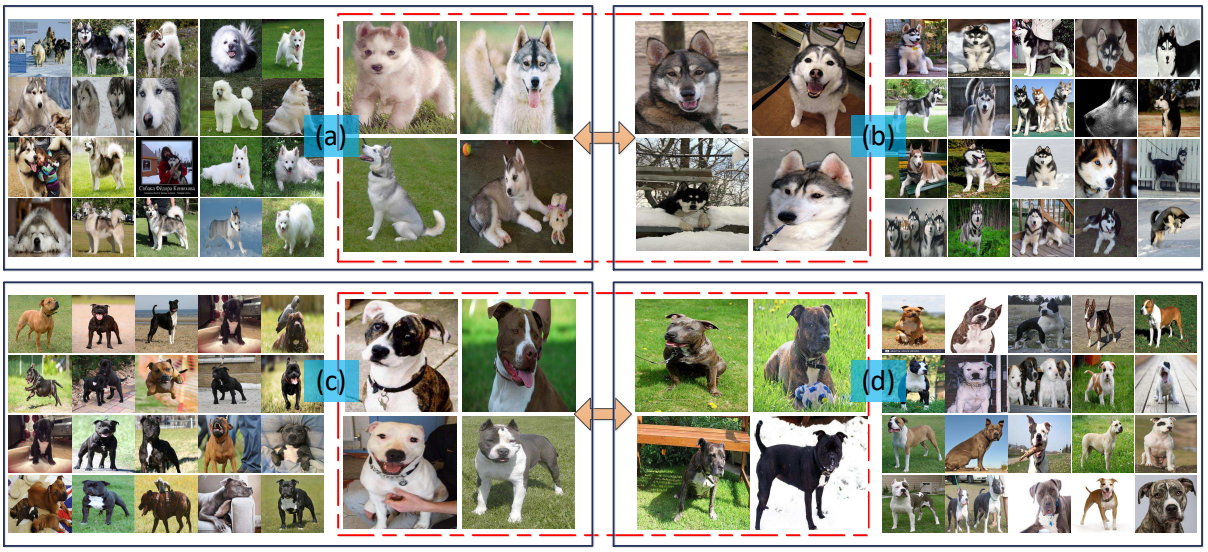


Fig. 9. Examples from four dog categories. The images in the same red rectangle are samples misclassified by prior work [11]. With the help of noisy web data, our proposed method can distinguish images from classes (a) and (b). However, we fail to recognize the dogs from classes (c) and (d) because none of the collected noisy data looks like the test images.

a much larger web dataset (both in terms of category and image numbers). Our method does not require additional categories while improves the accuracy by about 1.5% compared with [11] (from 85.90% to 87.36%). Fig. 9 shows the examples misclassified by [11] and correctly recognized by our method. Since the web images sampled by the proposed method can cover the characteristics of both categories, the trained model can recognize the images with the similar appearance by exploiting web data. Overall, these results indicate that progressive filtering and one-to-many correction are effective in extracting meaningful information from web data to improve the performance of CNN models.

V. CONCLUSIONS

In this paper, we present a novel progressive filtering method that effectively exploits web images for various image classification tasks. Moreover, a one-to-many label assignment strategy is employed for data correction based on the confidence values of labels and the tags of images. The method performs well in a variety of image classification tasks and the results are competitive to the state of the art.

ACKNOWLEDGMENT

This work was supported by the Natural Science Foundation of Tianjin, China (No.18JCYBJC15400), the Open Project Program of the National Laboratory of Pattern Recognition (NLPR), and the Fundamental Research Funds for the Central Universities.

REFERENCES

[1] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 1

[2] L. Yang, P. Luo, C. Change Loy, and X. Tang, "A large-scale car dataset for fine-grained categorization and verification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 1, 6

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012. 1

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015. 1

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 1, 6

[6] X. Chen and A. Gupta, "Webly supervised learning of convolutional networks," in *IEEE International Conference on Computer Vision*, 2015. 1, 2, 3, 10

[7] Z. Xu, S. Huang, Y. Zhang, and D. Tao, "Augmenting strong supervision using web data for fine-grained categorization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 1, 2

[8] L. Castrejon, Y. Aytar, C. Vondrick, H. Pirsiavash, and A. Torralba, "Learning aligned cross-modal representations from weakly aligned data," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1

[9] Y. Hu, L. Zheng, Y. Yang, and Y. Huang, "Twitter100k: A real-world dataset for weakly supervised cross-media retrieval," *IEEE Transactions on Multimedia*, vol. 20, no. 4, pp. 927–938, 2018. 1

[10] L. Zheng, S. Wang, and Q. Tian, "Coupled binary embedding for large-scale image retrieval," *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3368–3380, 2014. 1

[11] J. Krause, B. Sapp, A. Howard, H. Zhou, A. Toshev, T. Duerig, J. Philbin, and L. Fei-Fei, "The unreasonable effectiveness of noisy data for fine-grained recognition," in *European Conference on Computer Vision*, 2016. 1, 2, 3, 10, 11

[12] P. D. Vo, A. Ginsca, H. Le Borgne, and A. Popescu, "Harnessing noisy web images for deep representation," *Computer Vision and Image Understanding*, 2017. 1, 2, 3, 10, 11

[13] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010. 2

[14] X. Chen, A. Shrivastava, and A. Gupta, "Neil: Extracting visual knowledge from web data," in *IEEE International Conference on Computer Vision*, 2013. 2

[15] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik, "A multi-view embedding space for modeling internet images, tags, and their semantics," *International Journal of Computer Vision*, vol. 106, no. 2, pp. 210–233, 2014. 2, 3

[16] F. Schroff, A. Criminisi, and A. Zisserman, "Harvesting image databases from the web," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 754–766, 2011. 2

[17] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, "Learning from massive noisy labeled data for image classification," in *IEEE International Conference on Computer Vision*, 2015. 2, 3, 6, 10

[18] G. Van Horn, S. Branson, R. Farrell, S. Haber, J. Barry, P. Ipeirotis, P. Perona, and S. Belongie, "Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 2

[19] H. Izadinia, B. C. Russell, A. Farhadi, M. D. Hoffman, and A. Hertzmann, "Deep classifiers from image tags in the wild," in *Workshop on MMCommons*, 2015. 2, 3

[20] J. C. Schlimmer and R. H. Granger, "Incremental learning from noisy data." *Machine Learning*, vol. 1, no. 3, pp. 317–354, 1986. 2

[21] T. Poggio and G. Cauwenberghs, "Incremental and decremental support vector machine learning," in *Advances in Neural Information Processing Systems*, 2001. 2

[22] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, 2008. 2

[23] H. Fan, L. Zheng, and Y. Yang, "Unsupervised person re-identification: Clustering and fine-tuning," *arXiv preprint arXiv:1705.10444*, 2017. 2, 3

[24] M. Ye, A. J. Ma, L. Zheng, J. Li, and P. C. Yuen, "Dynamic label graph matching for unsupervised video re-identification," in *IEEE International Conference on Computer Vision*, 2017. 2

[25] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *International Conference on Machine Learning*, 2009. 3

[26] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Advances in Neural Information Processing Systems*, 2010. 3

[27] F. Ma, D. Meng, Q. Xie, Z. Li, and X. Dong, "Self-paced co-training," in *International Conference on Machine Learning*, 2017. 3

[28] X. Dong, L. Zheng, F. Ma, Y. Yang, and D. Meng, "Few-shot object detection," *arXiv preprint arXiv:1706.08249*, 2017. 3

[29] Z. Zheng, L. Zheng, and Y. Yang, "A discriminatively learned cnn embedding for person reidentification," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 14, no. 1, p. 13, 2017. 3

[30] J. Yang, D. Parikh, and D. Batra, "Joint unsupervised learning of deep representations and image clusters," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3

[31] K. G. Dizaji, A. Herandi, C. Deng, W. Cai, and H. Huang, "Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization," in *IEEE International Conference on Computer Vision*, 2017. 3

[32] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, "Deep adaptive image clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3

[33] H. Zhang, S. Liu, C. Zhang, W. Ren, R. Wang, and X. Cao, "Sketchnet: Sketch classification with web images," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3

[34] I. Krasin, T. Duerig, N. Alldrin, A. Veit, S. Abu-El-Haija, Sami andBelongie, D. Cai, Z. Feng, V. Ferrari, V. Gomes, A. Gupta, D. Narayanan, C. Sun, G. Chechik, and K. Murphy, "Openimages: A public dataset for large-scale multi-label and multi-class image classification." *Dataset available from https://github.com/openimages*, 2016. 3

[35] Y. Cui, F. Zhou, Y. Lin, and S. J. Belongie, "Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 3

[36] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "Youtube-8m: A large-scale video classification benchmark," *arXiv:1609.08675*, 2016. 3

[37] Z. Xu, S. Huang, Y. Zhang, and D. Tao, "Webly-supervised fine-grained visual categorization via deep domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 2028–2041, 2016. 3

[38] L. Zheng, S. Wang, Z. Liu, and Q. Tian, "Fast image retrieval: Query pruning and early termination," *IEEE Transactions on Multimedia*, vol. 17, no. 5, pp. 648–659, 2015. 3

[39] L. Zheng, S. Wang, J. Wang, and Q. Tian, "Accurate image search with multi-scale contextual evidences," *International Journal of Computer Vision*, vol. 120, no. 1, pp. 1–13, 2016. 3

[40] L. Zheng, Y. Yang, and Q. Tian, "SIFT meets CNN: A decade survey of instance retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 3

[41] L. Zheng, S. Wang, and Q. Tian, "L_p-norm IDF for scalable image retrieval," *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3604–3617, 2014. 3

[42] Q. You, J. Luo, H. Jin, and J. Yang, "Building a large scale dataset for image emotion recognition: The fine print and the benchmark," in *AAAI Conference on Artificial Intelligence*, 2016. 3

[43] X. Sun, J. Yang, M. Sun, and K. Wang, "A benchmark for automatic visual classification of clinical skin disease images," in *European Conference on Computer Vision*, 2016. 5, 11

[44] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei, "Novel dataset for fine-grained image categorization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 5

[45] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101–mining discriminative components with random forests," in *European Conference on Computer Vision*, 2014. 5, 11

[46] A. T. Ariadna Quattoni, "Recognizing indoor scenes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 5

[47] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 6

[48] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang, "The application of two-level attention models in deep convolutional neural network for fine-grained image classification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 6

[49] M. Simon and E. Rodner, "Neural activation constellations: Unsupervised part model discovery with convolutional networks," in *IEEE International Conference on Computer Vision*, 2015. 6

[50] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *ACM International Conference on Multimedia*, 2014. 6

[51] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008. 8

[52] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International Conference on Machine Learning*, 2015. 10

[53] S. Sukhbaatar and R. Fergus, "Learning from noisy labels with deep neural networks," *arXiv:1406.2080*, 2014. 9, 10

[54] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *International Conference on Machine Learning Workshop*, 2013. 9, 10

[55] A. Joulin, L. van der Maaten, A. Jabri, and N. Vasilache, "Learning visual features from large weakly supervised data," in *European Conference on Computer Vision*, 2016. 10

[56] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus, "Training convolutional networks with noisy labels," *arXiv:1406.2080*, 2014. 10

[57] M. Simon and E. Rodner, "Neural activation constellations: Unsupervised part model discovery with convolutional networks," in *IEEE International Conference on Computer Vision*, 2015. 11

[58] C. Doersch, A. Gupta, and A. A. Efros, "Mid-level visual element discovery as discriminative mode seeking," in *Advances in Neural Information Processing Systems*, 2013. 11

[59] M. Mohammadi and S. Das, "SNN: stacked neural networks," *CoRR*, vol. abs/1605.08512, 2016. 11

[60] Z. Xu, D. Tao, S. Huang, and Y. Zhang, "Friend or foe: Fine-grained categorization with weak supervision," *IEEE Transactions on Image Processing*, vol. 26, no. 1, pp. 135–146, 2017. 11

[61] X. Zhang, H. Xiong, W. Zhou, W. Lin, and Q. Tian, "Picking deep filter responses for fine-grained image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 11

[62] L. Liu, C. Shen, and A. V. D. Hengel, "The treasure beneath convolutional layers: Cross-convolutional-layer pooling for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 11

[63] A. Tatsuma and M. Aono, "Food image recognition using covariance of convolutional layer feature maps," *IEICE Transactions on Information and Systems*, no. 6, pp. 1711–1715, 2016. 11

[64] Y. Li, L. Liu, C. Shen, and A. van den Hengel, "Mid-level deep pattern mining," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 11

[65] C. Goring, E. Rodner, A. Freytag, and J. Denzler, "Nonparametric part transfer for fine-grained recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 11

[66] X. Zhang, H. Xiong, W. Zhou, and Q. Tian, "Fused one-vs-all features with semantic alignments for fine-grained visual categorization," *IEEE Transactions on Image Processing*, vol. 25, no. 2, pp. 878–892, 2016. 11

[67] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in Neural Information Processing Systems*, 2014. 11

[68] J. Yang, X. Sun, L. Jie, and R. Paul, "Clinical skin lesion diagnosis using representations inspired by dermatologist criteria," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 11

[69] K. Yanai and Y. Kawano, "Food image recognition using deep convolutional network with pre-training and fine-tuning," in *IEEE International Conference on Multimedia and Expo Workshop*, 2015. 11

[70] O. Rippel, M. Paluri, P. Dollar, and L. Bourdev, "Metric learning with adaptive density discrimination," *arXiv:1511.05939*, 2015. 11

[71] A. Meyers, N. Johnston, V. Rathod, A. Korattikara, A. Gorban, N. Silberman, S. Guadarrama, G. Papandreou, J. Huang, and K. P. Murphy, "Im2calories: towards an automated mobile vision food diary," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 11

[72] K. Chen and Z. Zhang, "Learning to classify fine-grained categories with privileged visual-semantic misalignment," *IEEE Transactions on Big Data*, vol. 3, pp. 37–43, 2016. 11

[73] Y. Zhang, X. S. Wei, J. Wu, and J. Cai, "Weakly supervised fine-grained categorization with part-based image representation," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1713–1725, 2016. 11

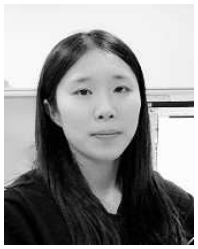
[74] M. Dixit, S. Chen, D. Gao, N. Rasiwasia, and N. Vasconcelos, "Scene classification with semantic fisher vectors," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 11

[75] D. Yoo, S. Park, J.-Y. Lee, and I. So Kweon, "Multi-scale pyramid pooling for deep convolutional representation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 11

[76] H. Hassannejad, G. Matrella, P. Ciampolini, I. De Munari, M. Mordonini, and S. Cagnoni, "Food image recognition using very deep convolutional networks," in *International Workshop on Multimedia Assisted Dietary Management*. ACM, 2016, pp. 41–49. 11

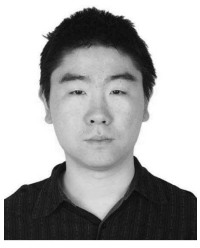
[77] L. Herranz, S. Jiang, and X. Li, "Scene recognition with cnns: objects, scales and dataset bias," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 11

Jufeng Yang is an associate professor in the College of Computer and Control Engineering, Nankai University. He received the PhD degree from Nankai University in 2009. From 2015 to 2016, he was working at the Vision and Learning Lab, University of California, Merced. His research falls in the field of computer vision, machine learning and multimedia.



Xiaoxiao Sun is currently a Master student with the College of Computer and Control Engineering, Nankai University. Her current research interests include computer vision, machine learning, pattern recognition and deep learning.

Yu-Kun Lai received his bachelors and Ph.D. degrees in computer science from Tsinghua University, China, in 2003 and 2008, respectively. He is currently a senior lecturer at the School of Computer Science & Informatics, Cardiff University. His research interests include Computer Graphics, Computer Vision, Geometry Processing and Image Processing. He is on the editorial board of *The Visual Computer*.



Liang Zheng received the Ph.D degree in Electronic Engineering from Tsinghua University, China, in 2015, and the B.E. degree in Life Science from Tsinghua University, China, in 2010. He is currently a Computer Science Futures Fellow and Lecturer in the Research School of Computer Science in the Australian National University. He was a postdoc researcher in University of Technology Sydney, Australia. His research interests are image retrieval, person re-identification and deep learning.

Ming-Ming Cheng received his PhD degree from Tsinghua University in 2012. Then he did 2 years research fellow, with Prof. Philip Torr in Oxford. He is now a professor at Nankai University, leading the Media Computing Lab. His research interests include computer graphics, computer vision, and image processing. He received research awards including ACM China Rising Star Award, IBM Global SUR Award, CCF-Intel Young Faculty Researcher Program, *etc*.