Learning from Web Data using Adversarial Discriminative Neural Networks for Fine-Grained Classification

Xiaoxiao Sun, Liyi Chen, Jufeng Yang*
College of Computer Science, Nankai University, Tianjin 300350, China
*corresponding author: yangjufeng@nankai.edu.cn

Abstract

Fine-grained classification is absorbed in recognizing the subordinate categories of one field, which need a large number of labeled images, while it is expensive to label these images. Utilizing web data has been an attractive option to meet the demands of training data for convolutional neural networks (CNNs), especially when the well-labeled data is not enough. However, directly training on such easily obtained images often leads to unsatisfactory performance due to factors such as noisy labels. This has been conventionally addressed by reducing the noise level of web data. In this paper, we take a fundamentally different view and propose an adversarial discriminative loss to advocate representation coherence between standard and web data. This is further encapsulated in a simple, scalable and end-to-end trainable multi-task learning framework. We experiment on three public datasets using large-scale web data to evaluate the effectiveness and generalizability of the proposed approach. Extensive experiments demonstrate that our approach performs favorably against the state-of-the-art methods.

Introduction

Deep learning has shown impressive improvement on many computer vision tasks, *e.g.*, general image classification, object detection and scene recognition *etc*. Many fine-grained classification works using convolutional neural networks (CNNs) also achieved surprising results (Xiao et al. 2015). The success of CNNs is inseparable from large-scale well-annotated image datasets. Nevertheless, CNNs are data-hungry. (Oquab et al. 2014; Yang et al. 2015) consider the transferability of CNNs by firstly initializing parameters with a pre-trained model (*e.g.*, AlexNet, VggNet) generated on a large-scale dataset and then fine-tuning it on a target well-labeled dataset. This is partially mitigated by employing pre-trained models, but a large-scale dataset is still necessary for better domain adaptation. Hence, they are becoming the de facto standard for tasks where it is possible to collect large well-annotated training sets, often by crowd-sourcing manual annotations. However, manual labeling is costly, time-consuming and error-prone, raises privacy concerns, and requires massive human intervention for every new task.

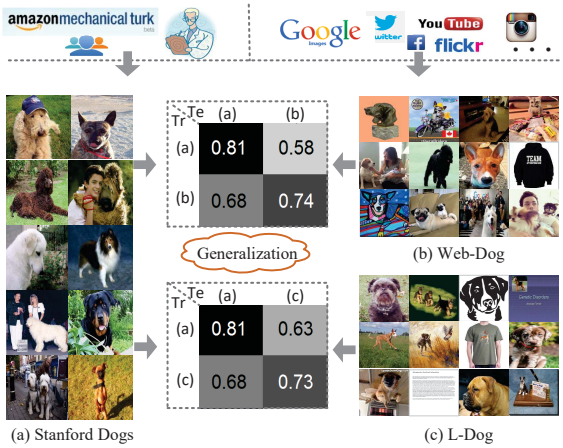

Figure 1: Examples from (a) the Stanford Dogs dataset, (b) the web images (Web-Dog) and (c) the L-Dog dataset. Note the Stanford Dogs dataset is well-labeled by users, while the other two datasets are labeled with keywords on the web. The cross dataset training-testing accuracies are shown in the center, where "Tr" and "Te" indicate training set and test set respectively. The gap between the results of Tr and Te on the **same** dataset and those of Tr and Te on **different** datasets shows that these datasets are not generalized well to each other.

As an alternative approach, using network can collect a large number of images more quickly and easily. Although it is inevitable that the web data has some noise, the large number of network data can make up for this deficiency. Some recent works (Chen and Gupta 2015; Xu et al. 2015; Gong and Wang 2017) have shown that fine-tuning CNNs with extensive web data can be more effective than only with a small-scale clean dataset. Different from work focusing on reducing the noisy level of web data, we consider that the content of the web image is usually complicated comparing with the clean dataset. For example, a web image contains several objects including a target object, where the target object is located at the edge of the image or has a small size, which makes the target object difficult to be distinguished. Therefore, there is a gap between web images and well-labeled images, which may be produced by

different levels of information complexity (web images often contain richer content than clean training images), various localizations and scales of objects, and the image label differences (noisy labels), *etc*. To effectively utilize web data, previous work focuses on filtering web data to remove noisy images with incorrect tags as much as possible. Interestingly, (Joulin et al. 2016) suggests that if the amount of web data is large, "label cleansing" may not be necessary even if the network data is noisy.

In this paper, we address this problem by exploring the gap between web and well-labeled data which is attributed to many factors including noisy labels, and differences in object locations, object sizes, viewing angles, scenes, background clutter, *etc*. For a quick preview, the cross dataset training-testing results in Figure 1 demonstrate the influence of the gap between the web and standard datasets. The model fine-tuned from Resnet50 on Stanford Dogs dataset (Aditya et al. 2011) has the testing accuracy of 0.81 on itself, but only gets 0.58 on the web data (Yang et al. 2018) and 0.63 on the L-Dog dataset. Moreover, the model trained directly on web data and tested on Stanford Dogs dataset gets an accuracy of 0.68, which is greatly lower than 0.81 (both training and testing are on the standard dataset). Here, L-Dogd[1] is collected for dog classification in Goldfince dataset (Krause et al. 2016). In addition, we also conduct the same experiment on Food-101 (Bossard, Guillaumin, and Van Gool 2014) and MIT Indoor 67 (Ariadna and Antonio 2009) dataset, training on web data and testing on standard data get an accuracy of 0.74 (the result of train and test on standard data is 0.84) and 0.66 (the result of train and test on standard data is 0.80) respectively. To reduce the influence of the gap between different datasets, we propose a novel CNN architecture that jointly optimizes the two classifiers operating on the representation: (i) the label predictor, which is used during both training and testing stages for object classification tasks, and (ii) the source classifier that distinguishes the data from the standard and web datasets during the training stage. The source classifier predicts the binary datasets label, and the corresponding loss–adversarial loss encourages the learned representations to be coherent between different datasets. The two classifiers mentioned above are well encapsulated in a multi-task learning framework where the second task helps to regularize the learning of the first one towards more generalizable performance.

We make the following contributions:

(1) We propose a jointly optimized deep architecture towards overcoming the dataset gap between easily acquired web images and the well-labeled data from standard datasets.

(2) Extensive experiments show that the proposed method is simple yet powerful and achieves state-of-the-art classification results on the Food-101 (Bossard, Guillaumin, and Van Gool 2014), Stanford Dogs (Aditya et al. 2011) and MIT Indoor 67 (Ariadna and Antonio 2009) datasets.

[1]It consists of 515 dog categories, and for experiments in our work, we use its 43,342 images containing the same 120 classes as the Stanford Dogs dataset.

Related Work

Deep Learning from Noisy Web Data

In the field of fine-grained classification, the training of deep models requires a large amount of well-labeled data, so the researchers pay more attention to collect a large scale dataset. However, in the majority of the fine-grained tasks, labeling datasets relies on expert knowledge, which is generally difficult and expensive to obtain.

To address this problem, recent works consider learning from web data, which is much cheaper to be obtained in general. However, directly training on such automatically harvested web data is usually difficult to get satisfactory performance. To address this drawback, learning with noisy labels has been extensively studied in the AI related tasks. Recent works (Vo et al. 2017; Chen and Gupta 2015) have achieved superior results in their tasks by using web data. Chen *et al.* (Chen, Shrivastava, and Gupta 2013) use a semi-supervised learning algorithm to find the relationships between common sense and labeled images of given categories. Schroff *et al.* (Schroff, Criminisi, and Zisserman 2011) 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. However, such data contains many noisy labels. Nevertheless, the introduction of web data improves the performance of deep models, which is verified by recent work. Chen and Gupta (Chen and Gupta 2015) present a two-stage approach to train deep models by exploiting both noisy web data and the transferability of CNNs. Xu *et al*. (Xu et al. 2015) propose a method that collects a great number of part patches from inexhaustible and weakly supervised web images to augment the training set, which generates more discriminative CNNs feature representations and improves classification accuracy. In order to transfer more knowledge from existing datasets to weakly supervised web images, Xu *et al*. (Xu et al. 2018) propose a semi-supervised method that utilizes both standard image-level labels and detailed annotations (*i.e*., object bounding boxes and part landmarks). Niu *et al*. (Niu, Veeraraghavan, and Sabharwal 2018) design a framework using both auxiliary labeled categories and web images to predict the categories of test images which have no connection with well-labeled training images. Alternatively, Xiao *et al.* (Xiao et al. 2015) use a probabilistic framework to model the relationships among images, clean labels and noisy labels in an end-to-end structure. They demonstrate the effectiveness of using noisy web data and the benefits of performing extra operations on noisy data, *e.g*., filtering. However, the results in (Joulin et al. 2016; Krause et al. 2016) suggest that data cleaning cannot bring noticeable improvements.

Although we take inspirations from the methods mentioned above, we take a fundamentally different view and propose a multi-task strategy that learns both generalizable and coherent feature representations between standard and web data. The proposed method is simple, scalable and end-to-end trainable.

Adversarial Learning Methods

In recent years, some works (Gong, Grauman, and Sha 2013; Ganin et al. 2015) employ joint learning to mitigate the negative influence of the gap when building the mapping between the source and target domains. In existing joint learning works, some approaches perform this by re-weighting or selecting samples from the source domain (Gong, Grauman, and Sha 2013), and others match the feature representations of the source and target tasks using adversarial learning (Ganin et al. 2015; Tzeng et al. 2015; 2017). While the loss formulation for adversarial learning stays consistent between most approaches, various objectives have been applied for the encoder, *e.g.*, confusion loss (Tzeng et al. 2015) and minimax formulation (Ganin et al. 2015), *etc*.

To learn from web data, we also demand to reduce the influence of the difference between web data and target data. However, to the best of our knowledge, no previous work considers learning with web data from the point of bridging the gap between web and standard data. In existing joint learning works, some approaches perform this by re-weighting or selecting samples from the source domain (Gong, Grauman, and Sha 2013), and others match the feature representation of the source and target tasks. Other methods (Tzeng et al. 2017) have chosen an adversarial loss to minimize domain shift, learning a representation that is simultaneously discriminative of source labels while not being able to distinguish between domains. In this paper, these works motivate us to explore the feasibility of employing adversarial learning to advocate feature coherence between large-scale easily-obtained noisy web data and limited well-annotated standard data for more generalizable performance.

Method

Figure 2 shows the pipeline of our proposed architecture. Given a target learning task with standard dataset $\mathcal{D}_s = \{(x^s_{n_s}, y^s_{n_s})\}_{n_s=1}^{N_s}$, the representation of n_s-th image $x^s_{n_s}$ has the tag $y^s_{n_s} \in \{1, ..., C\}$, where C is the number of classes and N_s is the number of images, which has limited training data. The web dataset $\mathcal{D}_w = \{(x^w_{n_w}, y^w_{n_w})\}_{n_w=1}^{N_w}$ is collected from web and used to help train the CNN model, where N_w is the number of images, the representation $x^w_{n_w}$ of n_w-th image has the tag $y^w_{n_w}$. Our method takes a standard dataset $\mathcal{D}_s$ (green box in Figure 2) and web data $\mathcal{D}_w$ (red box in Figure 2) as input. We also denote the images in the combined dataset $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N}$ with a label d_n, $n = 1, ..., N$, $N = N_s + N_w$. d_n is a binary variable indicating where x_n comes from $d_n = 1$ if the image is from the standard dataset and $d_n = 0$ for web images. Our goal is to transfer rich information from abounding and easily obtained $\mathcal{D}_w \in \mathcal{D}$ for better performance on $\mathcal{D}_s$.

Standard Classification The proposed method is generic and independent of the convolutional network backbone structure. More specifically, consider a CNN with parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta_f}, \boldsymbol{\theta_c}\}$, where $\boldsymbol{\theta_f}$ stands for the parameters for feature learning (*e.g.*, convolutional layers), and $\boldsymbol{\theta_c}$ represents the parameters of a C-way classifier, where C is the number of categories. For the classification part, we use the standard *softmax loss* as follows:

$$L_{\mathbf{c}}(x, y; \boldsymbol{\theta_f}, \boldsymbol{\theta_c}) = -\left[\sum_{j=1}^{C} \mathbf{1}(y = j) \log \frac{e^{\{\boldsymbol{\theta_f}, \boldsymbol{\theta_c}\}_j^\top x}}{\sum_{k=1}^{C} e^{\{\boldsymbol{\theta_f}, \boldsymbol{\theta_c}\}_k^\top x}}\right], \tag{1}$$

where x is the representation of an image with the label y. The indicator function $\mathbf{1}(a) = 1$ if a is true, and $\mathbf{1}(a) = 0$ otherwise.

We use both the standard data and web data to train $\boldsymbol{\theta_f}$ and $\boldsymbol{\theta_c}$. All the parameters can be well-optimized with both the standard dataset and the additional rich web data. Using a large amount of web data for training can improve the classification performance of the standard dataset. However, they may also fit better with a large amount of web data than standard data due to the gap between the two kinds of data. The inevitable noise data in the web dataset will also affect the model during the training process, and interfere with the classification effect of the standard dataset. We will jointly optimize the source classification task in the later section to minimize the negative influence mentioned above.

Source Classification To bridge the gap between web and standard data, we add a domain classifier to optimize the representation $\boldsymbol{\theta_f}$ by maximizing the loss $L_{\mathbf{d}}$ of source classification. We first define a logistic function for the domain classifier as follows:

$$g(x, \boldsymbol{\theta_d}, \boldsymbol{\theta_f}) = \frac{1}{1 + e^{-\{\boldsymbol{\theta_d}, \boldsymbol{\theta_f}\}^\top x}}, \tag{2}$$

where $\boldsymbol{\theta_d}$ represents the parameters of the source recognition. Then, the corresponding *log-likelihood loss* function for domain classification is defined as:

$$\begin{aligned} L_{\mathbf{d}}(d; g(x, \boldsymbol{\theta_d}, \boldsymbol{\theta_f})) = \sum_{i=1}^{M^b} & -d_i \log(g(x, \boldsymbol{\theta_d}, \boldsymbol{\theta_f})) \\ & - (1 - d_i) \log(1 - g(x, \boldsymbol{\theta_d}, \boldsymbol{\theta_f})), \end{aligned} \tag{3}$$

where $d_i = 1$ when the input image comes from the target dataset, and $d_i = 0$ when it is from the web dataset. M^b is the mini-batch size. When the model accurately identifies the source of an image, the value of the source classification loss $L_{\mathbf{d}}$ will decline. Otherwise, when the model is difficult to distinguish the source of the image, *i.e.*, the gap between the web dataset and the standard dataset is narrowed, the loss will increase.

Multi-task Learning In a nutshell, the goal of our method then becomes seeking the parameters $\boldsymbol{\theta_f}, \boldsymbol{\theta_c}, \boldsymbol{\theta_d}$ that minimize the joint loss function as shown in Figure 2:

$$L(\boldsymbol{\theta_f}, \boldsymbol{\theta_c}, \boldsymbol{\theta_d}) = L_{\mathbf{c}} - \lambda L_{\mathbf{d}}, \tag{4}$$

The parameter λ controls the trade-off between the two losses which will be discussed in the experiment section. On one hand, both parameters $\boldsymbol{\theta_f}$ and $\boldsymbol{\theta_c}$ can be well optimized with both the standard dataset and the rich web data. On the other hand, the source classification task which corresponds to the second part in Eq. 4, seeks for representation coherent (*i.e.*, the feature representations are not distinguished between standard and web datasets) and essentially regularize the learning of $\boldsymbol{\theta_f}$ towards more generalizable performance.

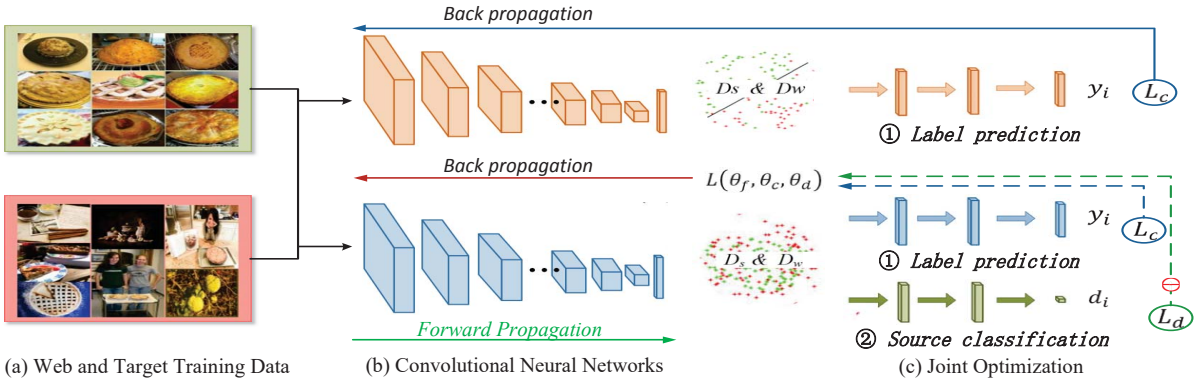

Figure 2: Overview of the proposed method. (a) web and target training data is the input to (b) convolutional neural networks. (c) joint optimization for label prediction and source classification. $L_{\mathbf{c}}$ is the loss for label prediction and $L_{\mathbf{d}}$ is the loss for source classification. $L(\boldsymbol{\theta}_{\mathbf{f}}, \boldsymbol{\theta}_{\mathbf{c}}, \boldsymbol{\theta}_{\mathbf{d}})$ is the joint loss, which influences the parameters by back propagation. $L_{\mathbf{d}}$ is preceded by a minus sign, so $\boldsymbol{\theta}_{\mathbf{f}}$ aims to maximize $L_{\mathbf{d}}$, which means that the feature from the shared convolutional layers becomes more and more consistent for web and target data.

For the objective function, we give the clarification as follows: Our goal is to optimize the parameters $(\boldsymbol{\theta}'_{\mathbf{f}}, \boldsymbol{\theta}'_{\mathbf{c}}, \boldsymbol{\theta}'_{\mathbf{d}})$ that deliver a saddle point of the objective function in Eq. 4:

$$(\boldsymbol{\theta}'_{\mathbf{f}}, \boldsymbol{\theta}'_{\mathbf{c}}) = \arg\min L_{\mathbf{fcd}}(\boldsymbol{\theta}_{\mathbf{f}}, \boldsymbol{\theta}_{\mathbf{c}}, \boldsymbol{\theta}'_{\mathbf{d}}), \tag{5}$$

and

$$\boldsymbol{\theta}'_{\mathbf{d}} = \arg\max L_{\mathbf{fcd}}(\boldsymbol{\theta}'_{\mathbf{f}}, \boldsymbol{\theta}'_{\mathbf{c}}, \boldsymbol{\theta}_{\mathbf{d}}). \tag{6}$$

The saddle point (Eqs. 5 and 6) will be found as a stationary point of the following stochastic updates:

$$\boldsymbol{\theta}_{\mathbf{f}} \leftarrow \boldsymbol{\theta}_{\mathbf{f}} - lr(\frac{\partial L_{\mathbf{c}}}{\partial \boldsymbol{\theta}_{\mathbf{c}}} - \lambda \frac{\partial L_{\mathbf{d}}}{\partial \boldsymbol{\theta}_{\mathbf{f}}}), \tag{7}$$

where lr is the learning rate.

$$\boldsymbol{\theta}_{\mathbf{c}} \leftarrow \boldsymbol{\theta}_{\mathbf{c}} - lr \frac{\partial L_{\mathbf{c}}}{\partial \boldsymbol{\theta}_{\mathbf{c}}}, \tag{8}$$

$$\boldsymbol{\theta}_{\mathbf{d}} \leftarrow \boldsymbol{\theta}_{\mathbf{d}} - lr \frac{\partial L_{\mathbf{d}}}{\partial \boldsymbol{\theta}_{\mathbf{d}}}. \tag{9}$$

During the training, for $\boldsymbol{\theta}_{\mathbf{f}}$, the partial derivatives are downstream $L_{\mathbf{d}}$ in Eq. 7, and the layer parameters that are upstream $\boldsymbol{\theta}_{\mathbf{f}}$ get multiplied by $-\lambda$, *i.e.*, $\frac{\partial L_{\mathbf{d}}}{\partial \boldsymbol{\theta}_{\mathbf{f}}}$ is effectively replaced by $-\lambda \frac{\partial L_{\mathbf{d}}}{\partial \boldsymbol{\theta}_{\mathbf{f}}}$. So, running stochastic gradient descent (SGD) on the model will implement the updates (Eq. 7 and Eq. 9) and can converge to a saddle point of the objective function. Note that, mathematically, we can formally process $-\lambda \frac{\partial L_{\mathbf{d}}}{\partial \boldsymbol{\theta}_{\mathbf{f}}}$ by a "pseudo function" $F(x)$ defined by two functions to realize the forward-backpropagation:

$$F(x) = -\lambda x \quad \text{and} \quad \frac{\mathrm{d}F}{\mathrm{d}x} = -\lambda. \tag{10}$$

We can take the item $-\lambda L_{\mathbf{d}}(d, g(x, \boldsymbol{\theta}_{\mathbf{d}}, \boldsymbol{\theta}_{\mathbf{f}}))$ in Eq. 4 as $L_{\mathbf{d}}(d, g(F(x, \boldsymbol{\theta}_{\mathbf{f}}), \boldsymbol{\theta}_{\mathbf{d}}))$, so Eqs. 7 and 9 can be updated by SGD. At the saddle point (Eqs. 5 and 6), the parameters $\boldsymbol{\theta}_{\mathbf{d}}$ of $L_{\mathbf{d}}$ minimize the domain classification loss (with the minus sign) to maximize the $L_{\mathbf{fcd}}$. The feature mapping parameters $\boldsymbol{\theta}_{\mathbf{f}}$ minimize the label prediction loss (the features are discriminative), while maximizing the source classification loss (the features are domain-invariant).

Experiments

We first investigate the feasibility of the proposed method across various classification tasks (including dogs, food and indoor scenes) and different network structures (such as CaffeNet, AlexNet, VggNet, ResNet). Then, we compare our approach with other state-of-the-art methods.

Experiment Setup

Datasets We experiment on three well-labeled datasets to evaluate the performance of our method, involving various representative fine-grained classification tasks, *i.e.*, dog, food, and indoor scene. Stanford Dogs (Aditya et al. 2011) consists of 20,580 images of 120 breeds of dogs, in which the training set has 12,000 images and the test set has 8,580 images. The Food-101 (Bossard, Guillaumin, and Van Gool 2014) dataset collects 101,000 food images with 101 categories, one quarter of which is the test set, and the rest is the training set. MIT Indoor 67 (Ariadna and Antonio 2009) contains 67 indoor scenes with a total of 15,620 images. Following existing evaluating protocol (Ariadna and Antonio 2009), we use a subset of the dataset with 5,360 training images and 1,340 test images.

Furthermore, we use three large-scale web datasets in training that proposed by Yang *et al.* (Yang et al. 2018). They download images from Google, Flickr and Twitter, by conducting a keyword search, where keywords correspond to the category labels in the public datasets. Then, they select images from search results as web data for the given class. According to three original standard fine-grained classification datasets, they collect 52,115 dog images, 240,096 food images, and 76,907 indoor scene images separately. Note that in our experiments the test datasets are the same from the original standard datasets.

Models The method we proposed in the paper has good portability, and it can be easily applied to different CNN models. The major pre-trained models used in our experiments are AlexNet, CaffeNet, VggNet, and ResNet50,

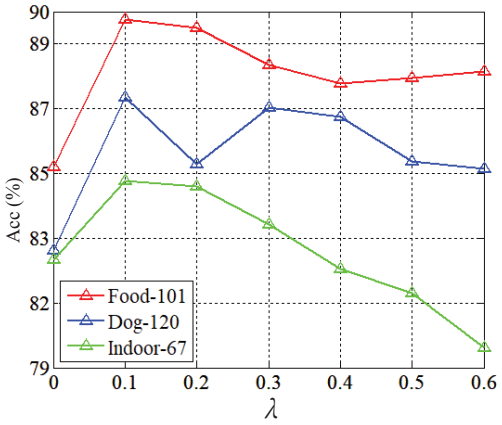

Figure 3: Trade-off parameter λ. We set λ to different values and conduct experiments on Food-101, Dog-120 (Stanford Dogs) and Indoor-67 datasets. As shown, big λ is likely to lead to low object classification accuracies, because the model tends to fit the source classification task.

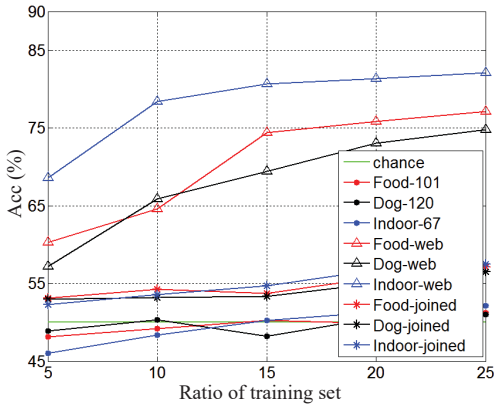

Figure 4: CNN plays "Name That Dataset", performing a 2-way classification to recognize image source. The ratio of training set indicates the proportion of images in each dataset used for training. Food-101, Dog-120, and Indoor-67 denote that only the images in the standard datasets are used. "-web" means both standard and web data are employed. "-jointed" uses the proposed jointly optimization strategy to learn consistent feature on both kinds of datasets.

which have good performance on many classification tasks. Since the pre-trained CNN model on ImageNet has shown state-of-the-art performance in many works, and fine-tuning a CNN model on a new dataset can often obtain better results, we fine-tune the initial models on both standard and the web datasets in experiments. As shown in Figure 2, the extra layer is similar to the basic architecture and consists of three additional fully connected (fc) layers using ReLU: 4096-4096-2 hidden units for Alexnet, Caffenet, VggNet-16 (one additional fc layer with 2 hidden units for Resnet50). The framework in our experiments is Caffe, and our models are trained on NVIDIA TITAN X GPUs. We set the mini-batch size to 64 for CaffeNet and AlexNet, 32 for VggNet and 12 for ResNet50, and initialize the learning rate to 0.001 for food and dog classification, and 0.0001 for indoor scene classification task. The learning rate is reduced after 20K iterations. The parameter λ will be discussed in the later section. We keep training the model until convergence and set the max iteration number to 200K.

The Parameter λ

In this section, we discuss the value of the parameter λ that controls the trade-off between the two tasks. The value influences the parameters of convolutional layers and further shapes the features from these layers. By setting λ to different values, we obtain the experimental results in three classification tasks. The classification accuracies on validation datasets (10% of training datasets) for different tasks are shown in Figure 3.

We can easily observe an opposite "U" trend in the results. On one hand, setting $\lambda = 0$ ignores the dataset discrepancy and suffers from large intra-class variations. It is equivalent to the general classification model that discards the part of the source classification. The gap between standard and web datasets limits the effectiveness of using large-scale network data, and may even be worse than the performance that only using standard datasets. On the other hand, employing a relatively larger value for $\lambda = 0$ could bias the network towards a trivial solution where nearly the same representation is achieved for both datasets. Since the parameter λ is used to weigh the two classification tasks, a larger value causes the task of this paper to be more inclined to focus on the source of the image. Therefore, it can be expected that experiments using a larger λ will affect the effect of fine-grained classification. As shown in the Figure 3, when $\lambda = 0$, the experimental result is lower than the classification accuracy using other values in the Food-101 dataset and the Dog-120 dataset. Similar to the Indoor-67 dataset, when the value of λ is between 0.1 and 0.3, the experimental result is obviously better than the result of $\lambda = 0$. In addition, as the value increases, when the λ exceeds 0.3, the performance of the model generally decreases. In particular, there is a significant downward trend in the Indoor-67 dataset. Indeed, we empirically find that setting $\lambda = 0.1$ leads to satisfactory results in all the three classification tasks. According to the results, we set $\lambda = 0.1$ in our experiments.

Representation Coherence

To further prove the existence of the gap between web and standard datasets, we conduct experiments on all three tasks. We choose the same amount of data from each class to avoid imbalanced data distribution. For example, images from the Food-101 training set are labeled as 0 and web data are labeled as 1 to train a 2-way model and the ratio between training and test sets is 3:1. That is called the "Name That Dataset" experiment. As shown in Figure 4, with increasing training data, the web and standard data are easier to be separated and there is no evidence of saturation. For comparison, we also conduct the experiment on Food-101 by labeling half of the sampled images as 0 and the rest as 1, and

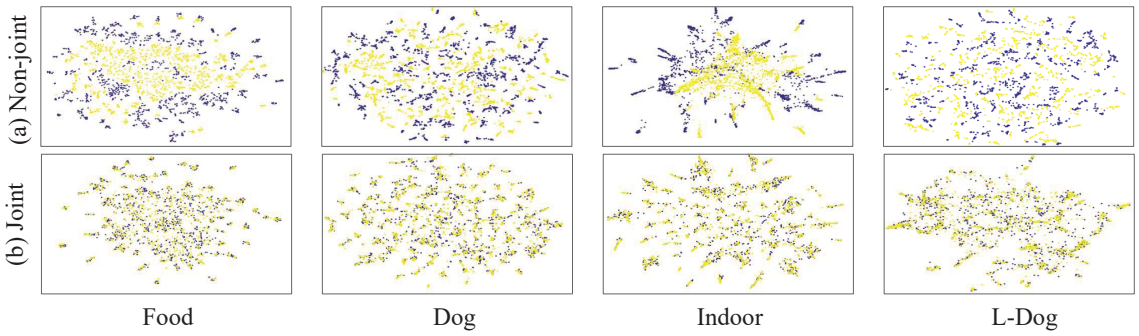

Figure 5: Effect of adaptation on the distribution of the extracted features. The figure shows t-SNE visualizations of the CNN's activations (a) in the case when no joint optimization is performed and (b) in the case when our joint loss is incorporated into training. Blue points correspond to the examples from web dataset, while yellow ones correspond to the standard dataset. For all tasks, the two distributions of features derived by our method are closer than others.

Table 1: Accuracies (%) on four datasets with different methods. $\mathcal{D}_{clean}$ and $\mathcal{D}_{mix}$ represent clean data and mixed data (including clean and web images). $\mathcal{D}_{filter}$ denotes that the web data is filtered from $\mathcal{D}_{mix}$. The "ft" means the fine-tuning process. L-Dogs refers to the web dataset from Goldfince is used for boosting the performance of Stanford Dogs.

#	Method	Model	Test Accuracy (%)			
			Food	Indoor	Dogs	L-Dogs
1	$\mathcal{D}_{clean}$+ft	AlexNet	65.93	65.53	63.57	63.57
2	$\mathcal{D}_{mix}$ +ft		69.71	69.60	65.63	64.84
3	$\mathcal{D}_{filter}$+ft		69.89	66.25	67.95	66.16
4	Bottom-up		70.29	66.79	72.17	71.59
5	Pseudo-label		69.36	67.35	70.32	71.01
6	Weakly		71.10	67.82	73.88	73.64
7	Ours		**73.78**	**71.21**	**75.26**	**74.58**
8	$\mathcal{D}_{clean}$+ft	CaffeNet	66.61	65.24	63.19	63.19
9	$\mathcal{D}_{mix}$ +ft		69.25	68.00	66.08	65.34
10	$\mathcal{D}_{filter}$+ft		68.48	63.53	69.56	65.90
11	Boosting		72.53	65.56	73.49	73.28
12	PGM		73.14	65.29	72.63	71.83
13	WSL		73.21	65.58	73.52	73.79
14	Ours		**74.78**	**68.40**	**74.93**	**75.25**
15	$\mathcal{D}_{clean}$+ft	VggNet	74.32	71.81	78.29	78.68
16	$\mathcal{D}_{mix}$ +ft		76.98	72.00	81.03	77.54
17	$\mathcal{D}_{filter}$+ft		78.24	72.04	79.70	79.57
18	Harnessing		79.02	72.48	78.45	79.92
19	Ours		**82.94**	**76.12**	**84.92**	**82.55**
20	$\mathcal{D}_{clean}$+ft	ResNet50	84.31	79.63	80.51	80.51
21	$\mathcal{D}_{mix}$ +ft		85.21	82.35	81.43	82.07
22	$\mathcal{D}_{filter}$+ft		86.10	81.32	82.62	83.61
23	Goldfince		86.75	83.47	85.90	85.48
24	Ours		**89.35**	**84.59**	**87.38**	**86.64**

the classification accuracy is reasonably stable. The results illustrate that the trained model can recognize the datasets from different sources, even though their nature is similar for food classification, which is consistent with the results in Figure 1 (the poor cross dataset generalization). Inspired by these results, we set our goal to mitigate the gap between web and target datasets. The proposed approach bridges the gap between the web and the target datasets by advocating representation coherence between both standard and web datasets. In this way, the parameter learning of the convolutional layers can also be enhanced. For a human, it is hard to distinguish the two sources, but for the CNN model trained on the web and target datasets, these domains could be distinct, which has been illustrated in Figure 4. Employing joint learning approaches can learn better parameters of convolutional layers, *i.e.*, a more robust representation of data.

We extract features from the fc7 layer using a non-joint model and visualize the feature embedding, where the blue points in Figure 5 correspond to the web examples, while yellow ones correspond to the target data. The proposed approach with joint learning effectively leads to more coherent representation, as we expected. As shown in Figure 5 (a), through a non-joint model (*i.e.*, basic model), images from different domains are separated. As can be seen in Figure 5 (b), our method leads to similar distributions for both dataset, indicating that the top layers of the CNN model are trained with source invariance. Our representation coherence is accomplished through standard back propagation training for web data and target data, which is also scalable and can be incorporated into other deep learning models. Figure 4 shows that the classification accuracies of "data-web" and standard data tend to rise. On the contrary, after joint learning, the relative dataset gap becomes lower for "data-joint" and standard data. With added training data, the classification accuracies keep leveling around the chance 50% for "data-joint", which are similar with the accuracy on the original standard datasets, *i.e.*, Food-101, Dog-120, and Indoor-67. The results illustrate that, after joint learning, the representations for web and standard data become similar.

Analysis of Results for Different Tasks

In this section, we discuss the classification results on different tasks. With the setup and the parameters discussed above, we conduct experiments on three datasets, and the results are shown in Table 1. For the basic models, including models fine-tuned with clean data (standard training set) and web data, the results using mixed data outperform using clean data alone, *e.g.*, #1 and #2. We also show the results after filtering (by removing web data, which has a tag different from the predicted label by the basic model, referred to

Table 2: Classification performance comparison with state-of-the-art methods on three public datasets. Bold values correspond to the best accuracy (%) per dataset. As shown, the proposed method outperforms the state-of-the-art approaches on all the tasks.

Food-101		Stanford Dogs		MIT Indoor 67	
Method	Acc (%)	Method	Acc (%)	Method	Acc (%)
(Bossard, Guillaumin, and Van Gool 2014)	50.76	(Huang et al. 2017)	78.30	(Milad and Subhasis 2016)	72.20
(Bossard, Guillaumin, and Van Gool 2014)	56.40	(Wei et al. 2017)	78.86	(Dixit et al. 2015)	72.86
(Meyers et al. 2015)	79.00	(Chen and Zhang 2016)	79.50	(Lin, RoyChowdhury, and Maji 2018)	79.00
(Li et al. 2018)	82.60	(Zhang et al. 2016)	80.43	(Zhou et al. 2018)	79.76
(Wei et al. 2018)	85.70	(Dubey et al. 2018)	83.75	(Yoo et al. 2015)	80.78
(Guo et al. 2018)	87.30	(Niu, Veeraraghavan, and Sabharwal 2018)	85.16	(Herranz, Jiang, and Li 2016)	80.97
(Hassannejad et al. 2016)	88.28	(Krause et al. 2016)	85.90	(Guo et al. 2017)	83.75
Ours (Resnet50)	**89.35**	Ours (Resnet50)	**87.07**	Ours (Resnet50)	**84.59**

as $D_{clean}+ft$), and previous works: Bottom-up (Sukhbaatar and Fergus 2014), Pseudo-label (Lee 2013), Weakly (Joulin et al. 2016), Boosting (Sukhbaatar et al. 2014), PGM (Xiao et al. 2015), WSL (Chen and Gupta 2015), Harnessing (Vo et al. 2017), Goldfince (Krause et al. 2016) that also employ and process web data for training CNN models.

Different from the above methods which focus on data pre-processing, we optimize the model to learn from the web and standard data by reducing the influence of dataset gap. For our method, the improvement of accuracy against the baseline ($D_{clean} + ft$) for different tasks varies, *e.g.*, the improvement on CaffeNet of the indoor scene is around 3%, and for food classification, it is about 8%. Meanwhile, the experimental results on food and dog conform to our expectation. For different models, just adding web images can improve the performance of the model (#2, #9, #16, and #21 of food and dog). However, after simple filtering (Sukhbaatar and Fergus 2014; Lee 2013), the accuracy may drop (*e.g.*, #3 and #10 of food and indoor, #17 of dog) because some useful images are wrongly removed. For the dog dataset, the filtering removes almost 40% web images, while the size is still larger than the original Stanford Dogs dataset.

Furthermore, 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 (Sukhbaatar and Fergus 2014; Lee 2013). However, our proposed algorithm can boost the classification accuracy on the indoor scene dataset. As shown in Table 1, our method achieves 84.59% accuracy, outperforming other methods. Finally, to verify the robustness of our method, we also conduct experiments on the L-Dog dataset, which is a publicly available noisy dataset for dog recognition. The results are consistent with those of web data. Moreover, we can find that the proposed method is generic and independent of the network structure backbone.

Comparison with State of the Art

In Table 2, we compare our method with other state-of-the-art approaches on different datasets. As can be seen, our method performs favorably against other methods for different tasks. For example, in dog classification task, (Krause et al. 2016) employs multiple crops and a larger web dataset with additional categories. Our method does not require additional categories while improving accuracy by 1.17% comparing with the method proposed by Krause *et al*. (Krause et al. 2016) (from 85.90% to 87.07%). Therefore, these results indicate that the proposed joint learning method is effective at bridging the gap between web and standard datasets to improve the performance of CNN models. Meanwhile, simplicity is central to our design and the strategies adopted in the proposed method are almost complementary to many other advanced approaches, such as employing multiple crops as done in (Krause et al. 2016), using additional categories for extras regularization adopted in (Krause et al. 2016) and leveraging additional annotations such as bounding boxes used in (Zhang et al. 2016).

Conclusion

In this paper, we firstly show that there exists a gap between the web and the standard datasets, which will inhibit the training of parameters in convolutional layers when both of them are utilized. To address this problem, we present a novel multi-task learning framework that effectively exploits web images for various fine-grained classification tasks. An adversarial discriminative loss is proposed to advocate representation coherence between standard and web data. To evaluate the effectiveness and generalization capability of our approach, we experiment on three public datasets, involving food, dog, and indoor scene classification tasks. In the experiment, we use large-scale web images and standard datasets to conduct experiments on different CNN models. Extensive experiments demonstrate that our approach performs favorably against the state-of-the-art methods.

Acknowledgments

This work was supported by the NSFC (NO. 61876094), 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

Aditya, K.; Nityananda, J.; Bangpeng, Y.; and Li, F.-F. 2011. Novel dataset for fine-grained image categorization. In *CVPR*.

Ariadna, Q., and Antonio, T. 2009. Recognizing indoor scenes. In *CVPR*.

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

Chen, X., and Gupta, A. 2015. Webly supervised learning of convolutional networks. In *ICCV*.

Chen, K., and Zhang, Z. 2016. Learning to classify fine-grained categories with privileged visual-semantic misalignment. *IEEE Trans. Big Data*.

Chen, X.; Shrivastava, A.; and Gupta, A. 2013. Neil: Extracting visual knowledge from web data. In *ICCV*.

Dixit, M.; Chen, S.; Gao, D.; Rasiwasia, N.; and Vasconcelos, N. 2015. Scene classification with semantic fisher vectors. In *CVPR*.

Dubey, A.; Gupta, O.; Guo, P.; Raskar, R.; Farrell, R.; and Naik, N. 2018. Pairwise confusion for fine-grained visual classification. In *ECCV*.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. 2015. Domain-adversarial training of neural networks. *J. Machine Learning Research* 17(1):2096–2030.

Gong, D., and Wang, D. Z. 2017. Extracting visual knowledge from the web with multimodal learning. In *IJCAI*.

Gong, B.; Grauman, K.; and Sha, F. 2013. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *ICML*.

Guo, S.; Huang, W.; Wang, L.; and Qiao, Y. 2017. Locally supervised deep hybrid model for scene recognition. *IEEE Trans. Image Processing* 26(2):808–820.

Guo, S.; Huang, W.; Zhang, H.; Zhuang, C.; Dong, D.; Scott, M. R.; and Huang, D. 2018. Curriculumnet: Weakly supervised learning from large-scale web images. In *ECCV*.

Hassannejad, H.; Matrella, G.; Ciampolini, P.; De Munari, I.; Mordonini, M.; and Cagnoni, S. 2016. Food image recognition using very deep convolutional networks. In *MADiMa*.

Herranz, L.; Jiang, S.; and Li, X. 2016. Scene recognition with cnns: Objects, scales and dataset bias. In *CVPR*.

Huang, C.; Li, H.; Xie, Y.; Wu, Q.; and Luo, B. 2017. Pbc: Polygon-based classifier for fine-grained categorization. *IEEE Transactions on Multimedia* 19(4):673–684.

Joulin, A.; van der Maaten, L.; Jabri, A.; and Vasilache, N. 2016. Learning visual features from large weakly supervised data. In *ECCV*.

Krause, J.; Sapp, B.; Howard, A.; Zhou, H.; Toshev, A.; Duerig, T.; Philbin, J.; and Fei-Fei, L. 2016. The unreasonable effectiveness of noisy data for fine-grained recognition. In *ECCV*.

Lee, D.-H. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop*.

Li, Z.; Zhu, X.; Wang, L.; and Guo, P. 2018. Image classification using convolutional neural networks and kernel extreme learning machines. In *ICIP*.

Lin, T.-Y.; RoyChowdhury, A.; and Maji, S. 2018. Bilinear convolutional neural networks for fine-grained visual recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence* 40(6):1309–1322.

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 *CVPR*.

Milad, M., and Subhasis, D. 2016. SNN: stacked neural networks. *arXiv:1605.08512*.

Niu, L.; Veeraraghavan, A.; and Sabharwal, A. 2018. Webly supervised learning meets zero-shot learning: A hybrid approach for fine-grained classification. In *CVPR*.

Oquab, M.; Bottou, L.; Laptev, I.; and Sivic, J. 2014. Learning and transferring mid-level image pepresentations using convolutional neural networks. In *CVPR*.

Schroff, F.; Criminisi, A.; and Zisserman, A. 2011. Harvesting image databases from the web. *IEEE Trans. Pattern Analysis and Machine Intelligence* 33(4):754–766.

Sukhbaatar, S., and Fergus, R. 2014. Learning from noisy labels with deep neural networks. *arXiv:1406.2080*.

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

Tzeng, E.; Hoffman, J.; Darrell, T.; and Saenko, K. 2015. Simultaneous deep transfer across domains and tasks. In *ICCV*.

Tzeng, E.; Hoffman, J.; Saenko, K.; and Darrell, T. 2017. Adversarial discriminative domain adaptation. In *CVPR*.

Vo, P. D.; Ginsca, A.; Le Borgne, H.; and Popescu, A. 2017. Harnessing noisy web images for deep representation. *Comp. Vis. Image Unders.*

Wei, X.-S.; Luo, J.-H.; Wu, J.; and Zhou, Z.-H. 2017. Selective convolutional descriptor aggregation for fine-grained image retrieval. *IEEE Trans. Image Processing*.

Wei, X.; Zhang, Y.; Gong, Y.; Zhang, J.; and Zheng, N. 2018. Grassmann pooling as compact homogeneous bilinear pooling for fine-grained visual classification. In *ECCV*.

Xiao, T.; Xia, T.; Yang, Y.; Huang, C.; and Wang, X. 2015. Learning from massive noisy labeled data for image classification. In *ICCV*.

Xu, Z.; Huang, S.; Zhang, Y.; and Tao, D. 2015. Augmenting strong supervision using web data for fine-grained categorization. In *CVPR*.

Xu, Z.; Huang, S.; Zhang, Y.; and Tao, D. 2018. Webly-supervised fine-grained visual categorization via deep domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40(5):1100–1113.

Yang, L.; Luo, P.; Change Loy, C.; and Tang, X. 2015. A large-scale car dataset for fine-grained categorization and verification. In *CVPR*.

Yang, J.; Sun, X.; Lai, Y.-K.; Zheng, L.; and Cheng, M.-M. 2018. Recognition from web data: A progressive filtering approach. *IEEE Transactions on Image Processing* 27(11):5303–5315.

Yoo, D.; Park, S.; Lee, J.-Y.; and So Kweon, I. 2015. Multi-scale pyramid pooling for deep convolutional representation. In *CVPR*.

Zhang, Y.; Wei, X. S.; Wu, J.; and Cai, J. 2016. Weakly supervised fine-grained categorization with part-based image representation. *IEEE Trans. Image Processing* 25(4):1713–1725.

Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; and Torralba, A. 2018. Places: A 10 million image database for scene recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence* 40(6):1452–1464.