Relaxing from Vocabulary: Robust Weakly-Supervised Deep Learning for Vocabulary-Free Image Tagging*

Jianlong Fu[1,2], Yue Wu[3], Tao Mei[2], Jinqiao Wang[1], Hanqing Lu[1] and Yong Rui[2]

[1]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China

[2]Microsoft Research, Beijing, China

[3]University of Science and Technology of China, Hefei, China

[2]{jianf, tmei, yongrui}@microsoft.com, [1]{jqwang, luhq}@nlpr.ia.ac.cn, [3]wye@mail.ustc.edu.cn

Abstract

The development of deep learning has empowered machines with comparable capability of recognizing limited image categories to human beings. However, most existing approaches heavily rely on human-curated training data, which hinders the scalability to large and unlabeled vocabularies in image tagging. In this paper, we propose a weakly-supervised deep learning model which can be trained from the readily available Web images to relax the dependence on human labors and scale up to arbitrary tags (categories). Specifically, based on the assumption that features of true samples in a category tend to be similar and noises tend to be variant, we embed the feature map of the last deep layer into a new affinity representation, and further minimize the discrepancy between the affinity representation and its low-rank approximation. The discrepancy is finally transformed into the objective function to give relevance feedback to back propagation. Experiments show that we can achieve a performance gain of 14.0% *in terms of a semantic-based relevance metric in image tagging with 63,043 tags from the WordNet, against the typical deep model trained on the ImageNet 1,000 vocabulary set.*

1. Introduction

More recently, deep learning has achieved comparable accuracy to human beings in image categorization tasks on the limited vocabulary [10]. However, this result is far from many real-world applications, such as image tagging, where we often need tens of thousands of tags to describe the various image content [5, 8]. One of the major challenges is to acquire sufficient and high-quality training data for a large vocabulary, which is often too expensive to obtain. For example, it took more than 25,000 AMT[1] workers about one year to construct the entire ImageNet dataset [6] (about 22,000 categories and 14.2 million images). Despite of its wide adoption in research communities, ImageNet is still a small subset of the nouns in WordNet[2]. There are huge numbers of categories left unlabeled, making the existing deep learning models hard to scale up. Therefore, how to scale deep learning approaches to large and arbitrary categories without enormous human-cost appears to be a challenging yet urgent problem.

With the success of commercial image search engines, learning from the Web has demonstrated one of the most effective solutions to collect massive training data [4, 9, 22].

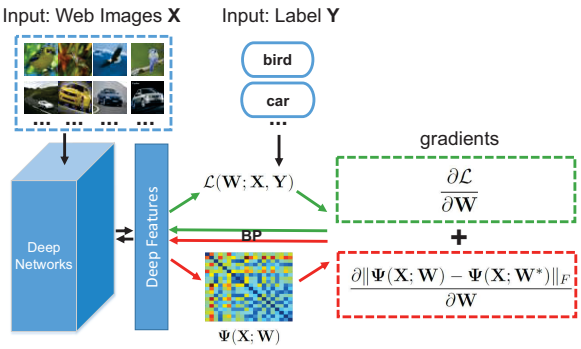

Figure 1: The illustration of the proposed model. The deep network is trained not only by the label supervision with loss $\mathcal{L}$, but also the minimization of the discrepancy between the affinity representation $\mathbf{\Psi}(\mathbf{X};\mathbf{W})$ and its low-rank approximation $\mathbf{\Psi}(\mathbf{X};\mathbf{W}^*)$. Note that a traditional CNN model only follows the flowchart of the top green part, without the feature relevance feedback indicated in the bottom red part. Details are in Sec. 3. [Best viewed in color]

* This work was performed when Jianlong Fu and Yue Wu were visiting Microsoft Research as research interns. The first two authors contributed equally to this work.

[1]https://www.mturk.com/mturk/welcome

[2]http://wordnet.princeton.edu/

Despite of the convenience using Web images to train models, the performance degradation is inevitable due to the noises in Web image search results. A conventional deep learning network is sensitive to noisy training images, as it tries to fit all the training data without distinguishing the authenticity of their labels. According to our experiments, when 30% of the training images are mislabeled, the accuracy of a conventional deep network drops at least 20% in CIFAR-10 dataset. Therefore, designing a noise-robust deep network is imperative to attenuate the influence of the noises in Web images.

Although previous works have studied how to perform the weakly-supervised object recognition or localization if the accurate image-level labels can be provided [19, 24], how to suppress the image-level noise effect has not been fully explored yet. In this paper, we propose a robust weakly-supervised deep learning network with the noisy Web training data for image tagging. As the Web data is readily available, the proposed approach can scale to arbitrary and unlabeled categories without heavy human effort. To achieve this goal, we first start from embedding the feature map of the last deep layer into a new affinity representation that essentially explores the similarities among the deep features of training samples. Second, by adopting the "few and different" assumption about the noises, we minimize the discrepancy between the affinity representation and its low-rank approximation. Third, this discrepancy is further transformed into the objective function to give those "few and different" noisy samples low-level authorities in training.

The advantages of the proposed method are three folds. First, except for the label supervision, we utilize the mutual relationship of features as feedback in our formulation. In this way, the learning process is mainly driven by the dominant correct samples. To the best of our knowledge, this idea has not been exploited by previous deep learning works. Second, we conduct image tagging with the largest vocabulary set of about 63,000 tags from the WordNet, and achieve a significant improvement against the typical deep learning model trained on the ImageNet 1,000 vocabulary set. Third, our improvement is network-independent, so that with the help of our model, any existing deep learning network can be readily extended to unlabeled categories. An illustration of the proposed model is shown in Fig. 1.

The rest of the paper is organized as follows. Sec.2 reviews related works. In Sec.3, we introduce the proposed approach and implementation details. The performance is evaluated in Sec.4. Sec.5 concludes this paper.

2. Related Work

There are two schemes to handle the data noises in deep learning. One aims to remove the noisy data before training by preprocessing. The other is designed to make the deep network itself robust to noises.

The preprocessing methods can be implemented either by the conventional outlier detection, or by the pre-training strategy in deep learning. First, the specific methods in outlier detection include PCA, Robust PCA [3], Robust Kernel PCA [25], probabilistic modeling [11] and one-class SVM [14]. These methods regard the outliers as those "few and different" samples. However, the challenge of these methods is to distinguish "hard samples" from the truly noisy samples. Second, recovering the clean training samples by a layer-wise autoencoder or denoising autoencoder [23] in the pre-training and then initializing a deep network by the pre-trained model parameters is an effective method to remove global noises, which has been used in face parsing [15]. However, these methods are mainly designed for cases where noises are contained in correct images (e.g., background noises), while noises in web images are often those mislabeled.

To train a robust deep learning model on noisy training data, J.Larsen *et al.* proposed one of the pioneer works which added noise modeling into the neural networks [13]. However, they make a symmetric label noise assumption, which is often not true in real applications. V.Mnih *et al.* proposed to label aerial images from noisy data where only a binary classification was considered [17]. The most related work to ours was proposed by S.Sukhbaatar *et al.* who introduced an extra noise layer as a part of training process in multi-class image classification [21]. They first trained a base model on noisy training data with several iterations, then activated the extra noise layer to absorb the noise from the learned base model.

Compared with previous works, we propose a holistic noise-robust model that handles noisy samples softly by limiting their contributions in the learning process according to their affinity to other samples. Besides, the algorithm can affect the whole back propagation, rather than simply relying on a certain layer.

3. Weakly-Supervised Deep Learning Model

Our goal is to design a noise-robust deep learning algorithm. We use the convolutional deep neural network (CNN) [12] for its state-of-the-art performance in image categorization. We will first analyze its limitation on noisy training samples and then propose the weakly-supervised deep learning model.

3.1. Traditional CNN Model

The first several layers of the traditional CNN model are convolutional and the remaining layers are fully-connected. The exact number of layers generally depends on specific tasks. The output of the last fully-connected layer is considered as an input to a softmax classifier which can generate a distribution over the final category labels. Let

$\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N]$ be the matrix of training data, where $\mathbf{x}_i$ is the feature vector of the i^{th} image. N is the number of images. Denote $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_N]^T \in \{0,1\}^{N\times K}$, where $\mathbf{y}_i \in \{0,1\}^{K\times 1}$ is the cluster indicator vector for $\mathbf{x}_i$. K is the number of categories. There are M layers in total and $\mathbf{W} = \{\mathbf{W}^{(1)}, ..., \mathbf{W}^{(M)}\}$ are the model parameters. In each layer, we absorb the bias term into the weights and denote them as a whole. $\mathbf{W}^{(m)} = [\mathbf{w}_1^{(m)}, ..., \mathbf{w}_{d_m}^{(m)}]^T \in R^{d_m \times d_{m-1}}$, where $\mathbf{w}_i^{(m)} \in R^{d_{m-1}}$, d_{m-1} is the dimension of the $(m-1)^{th}$ feature map. $\mathbf{Z}^{(m)}(\mathbf{X}) = [\mathbf{z}^{(m)}(\mathbf{x}_i), ..., \mathbf{z}^{(m)}(\mathbf{x}_N)]^T \in R^{N\times d_m}$ is the feature map produced by the m^{th} layer.

The goal is to minimize the following objective function in the form of a softmax regression with weight decay:

$$\mathcal{L}(\mathbf{W}; \mathbf{X}, \mathbf{Y}) = -\frac{1}{N}[\sum_{i=1}^{N}\sum_{j=1}^{K} \mathbf{1}_{Y_{ij}}(j) \log p(Y_{ij} = 1|\mathbf{x}_i; \mathbf{W})] + \frac{\beta}{2}||\mathbf{W}||_F, \quad (1)$$

where Y_{ij} is the $(i,j)^{th}$ entry of $\mathbf{Y}$. $\mathbf{1}_{Y_{ij}}(j)$ is the indicator function such that $\mathbf{1}_{Y_{ij}}(j) = 1$ if $Y_{ij} = 1$, otherwise zero. β is the coefficient of weight decay. We can see that the derivatives to $\mathbf{w}_j^{(M)}$ in the output layer is:

$$\frac{\partial \mathcal{L}(\mathbf{W}; \mathbf{X}, \mathbf{Y})}{\partial \mathbf{w}_j^{(M)}} = -\frac{1}{N}\sum_{i=1}^{N} \mathbf{z}^{(M-1)}(\mathbf{x}_i)[\mathbf{1}_{Y_{ij}}(j) - p(Y_{ij} = 1|\mathbf{z}^{(M-1)}(\mathbf{x}_i); \mathbf{w}_j^{(M)})] + \beta \mathbf{w}_j^{(M)}. \quad (2)$$

Parameters in other layers can be calculated by the back propagation algorithm (BP) [20].

According to the gradients, we can see that if the training data has noises, the indicator function $\mathbf{1}_{Y_{ij}}(j)$ will produce a wrong value, resulting in a wrong optimization direction or even making this optimization diverge. The reason is that traditional models completely believe the label of each image, and all the images are treated equally. As a result, the model will suffer from low accuracy if it is trained on the noisy web images.

3.2. CNN Model with Feature Relevance Feedback

The proposed model is based on the basic assumption that features of correct samples in a category tend to be similar with each other, while there is a big variance in the representation of the noise samples. As a result, the relationship among features can be utilized as a feedback to make different samples contribute differently to achieve better accuracy.

Specifically, we transform the sample features in the output layer into a new affinity representation that embeds the mutual relationship of sample features. We model this relationship as a nearest neighbor system as in [1]. We define a similarity metric $\mathbf{S} \in R^{N\times N}$ as follows:

$$S_{ij} = \begin{cases} \exp\{-\frac{||\mathbf{z}^{(M)}(\mathbf{x}_i) - \mathbf{z}^{(M)}(\mathbf{x}_j)||^2}{\gamma^2}\} & \mathbf{y}_i = \mathbf{y}_j \\ 0 & otherwise, \end{cases} \quad (3)$$

where γ is a scale factor. To better reflect the local structure, the similarity metric is normalized with a diagonal matrix $\mathbf{D}$, where $D_{ii} = \sum_{j=1}^{N} S_{ij}$. We define $\mathbf{\Psi}(\mathbf{X}, \mathbf{W}) = [\psi(\mathbf{x}_1), \ldots, \psi(\mathbf{x}_N)] = \mathbf{D}^{-1}\mathbf{S}$ as the new feature representation. Each column of the matrix $\mathbf{\Psi}(\mathbf{X}, \mathbf{W})$ embeds the relationship of an image $\mathbf{x}_i$ to other images.

Given the input images $\mathbf{X}$, we assume that the ideal model parameters are $\mathbf{W}^*$. A noise-robust learning algorithm should optimize $\mathbf{W}$ to approach to $\mathbf{W}^*$ as much as possible. The objective can be achieved by minimizing the differences $\mathbf{E_n}$ between the learned features $\mathbf{\Psi}(\mathbf{X}, \mathbf{W})$ and $\mathbf{\Psi}(\mathbf{X}, \mathbf{W}^*)$. $\mathbf{E_n}$ is the error in feature representation caused by the noisy images. In other words, we can regard $\mathbf{\Psi}(\mathbf{X}, \mathbf{W})$ as the ideal feature map plus an additive error $\mathbf{E_n}$ as:

$$\mathbf{\Psi}(\mathbf{X}; \mathbf{W}) = \mathbf{\Psi}(\mathbf{X}; \mathbf{W}^*) + \mathbf{E_n}. \quad (4)$$

According to Eqn. (4) and the low-rank representation theory [3], we consider $\mathbf{\Psi}(\mathbf{X}; \mathbf{W}^*)$ to be a low-rank matrix and we have:

$$rank(\mathbf{\Psi}(\mathbf{X}; \mathbf{W})) > rank(\mathbf{\Psi}(\mathbf{X}; \mathbf{W}^*)) \quad (5)$$

When the vocabulary size is large enough, the categories are fine-grained and images in each category are very similar to each other. Besides, the noises in one category are actually those from other categories with wrong labels. Consequently, we can assume that all the features can present at most K types of patterns and the rank of $\mathbf{\Psi}(\mathbf{X}; \mathbf{W}^*)$ equals the category number K. As a result, $\mathbf{\Psi}(\mathbf{X}; \mathbf{W}^*)$ can be calculated by the following optimization problem:

$$\begin{aligned} \min_{\mathbf{\Psi}(\mathbf{X};\mathbf{W}^*)} \quad & \|\mathbf{\Psi}(\mathbf{X}; \mathbf{W}) - \mathbf{\Psi}(\mathbf{X}; \mathbf{W}^*)\|_F, \\ s.t. \quad & rank(\mathbf{\Psi}(\mathbf{X}; \mathbf{W}^*)) = K. \end{aligned} \quad (6)$$

Since the labels are noisy, we should use the obtained ideal feature map to reduce the noise effect in the learning process. We use the ideal feature map as an input to generate the ideal prediction over different category labels by softmax function. In this way, we make the prediction as accurate as possible and thus reduce the risk that errors of the network are reinforced in each iteration. However, we find that this scheme greatly increases the time-cost in the optimization, because it involves additional computational burden in Eqn. (6). Instead of this step-by-step method, in the following, we propose an alternative solution that essentially calculates the ideal feature map and generates the ideal prediction over category labels at the same time. The proposed algorithm is based on the following proposition:

Proposition 1. *Let* $\mathbf{L} = \mathbf{D} - \mathbf{S}$, $\mathbf{H}^* \in R^{N\times K}$ *is comprised of the eigenvectors of the largest K eigenvalues of* $\mathbf{\Psi}(\mathbf{X}, \mathbf{W})$*, we have: 1) the solution of Eqn.* (6)*, i.e., the best rank-K approximation of* $\mathbf{\Psi}(\mathbf{X}, \mathbf{W})$*, is uniquely determined by the eigenvector* $\mathbf{H}^*$ *; 2)* $\mathbf{H}^*$ *is also the solution of the following optimization problem:*

$$\min_{\mathbf{H}} tr[\mathbf{H}^T\mathbf{L}\mathbf{H}] \quad s.t.\ \mathbf{H}^T\mathbf{H} = \mathbf{I}. \tag{7}$$

As both Eqn. (6) *and Eqn.* (7) *achieve the optimum at* $\mathbf{H}^*$*, Eqn.* (6) *is equivalent to Eqn.* (7)*.*

The proof of Proposition 1 is presented in the supplementary material A1. The above proposition uncovers that the optimal solution in Eqn. (6) can be obtained by solving the trace minimization problem in Eqn. (7). Therefore, we combine the softmax regression of a traditional CNN with the trace optimization. The final objective function for the noise-robust deep learning is designed as:

$$\tilde{\mathcal{L}}(\mathbf{W}; \mathbf{X}, \mathbf{Y}) = \min_{\mathbf{W}} \mathcal{L}(\mathbf{W}; \mathbf{X}, \mathbf{Y}) + \alpha\, tr[\mathbf{H}^T\mathbf{L}\mathbf{H}]. \tag{8}$$

Since the label matrix $\mathbf{Y}$ is given, $\mathbf{H}$ can be calculated by minimizing the gap between the subspace spanned by $\mathbf{H}$ and $\mathbf{Y}$ [26, 27], i.e., $\min_{\mathbf{H}} ||\mathbf{H}\mathbf{H}^T - \mathbf{Y}\mathbf{Y}^T||_F^2$. To satisfy the orthogonality, $\mathbf{Y}$ is further scaled to $\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}$. Although the solution to the above problem is not unique, $\mathbf{H} = \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-\frac{1}{2}}$ is a feasible one. It avoids the heavy computational costs for solving the eigen-decomposition problem in Eqn. (7). Besides, we find that this approximation can make the network training efficient and robust.

3.3. Analysis of Relevance Feedback

We analyze the relevance feedback from the gradient perspective to show the noise-resistance ability of the proposed objective function. Based on the definition of similarity metric $\mathbf{S}$ in Eqn. (3), the mutual relationship of features is described by the discrepancy in the output layer features. Furthermore, we define:

$$\Delta_d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{z}_d^{(M-1)}(\mathbf{x}_i) - \mathbf{z}_d^{(M-1)}(\mathbf{x}_j)\|^2, \tag{9}$$

is the discrepancy between two images in the d^{th} dimension of the $(M-1)^{th}$ layer, where $d = 1, 2, ..., d_{M-1}$. Therefore, if we use a linear activation function, the discrepancy in the M^{th} layer (output layer) can be represented by the accumulated products of the weight in the M^{th} layer and the discrepancy in each dimension of the $(M-1)^{th}$ layer.

For clarity, we use the notation $\mathbf{u}^{ij}$. $\mathbf{u}^{ij}$ is a column vector with two nonzero elements, where i^{th} and j^{th} element equals to 1 and -1, respectively. Therefore, for each element in $\mathbf{W}^{(M)} \in R^{K\times d_{M-1}}$, we have:

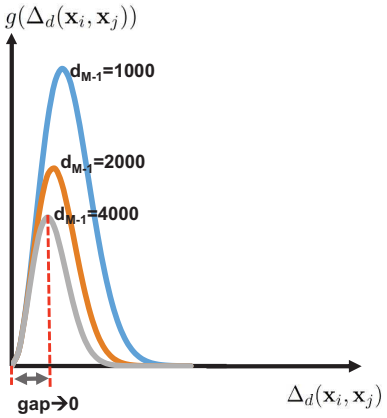

Figure 2: The curve shows the contribution of an image sample to gradients, with its distance to other images. With the increasing of d_{M-1}, only the monotone decreasing part can be reflected. Hence, we can observe that the larger the distance, the less the contribution.

$$\begin{aligned}
&\frac{\partial tr[\mathbf{H}^T\mathbf{L}\mathbf{H}]}{\partial W_{kd}^{(M)}} \\
&= tr[\mathbf{H}\mathbf{H}^T \frac{\partial \mathbf{L}}{\partial W_{kd}^{(M)}}] = tr[\mathbf{H}\mathbf{H}^T \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} \frac{\partial S_{ij}}{\partial W_{kd}^{(M)}} \mathbf{u}^{ij}(\mathbf{u}^{ij})^T] \\
&= tr[\mathbf{H}\mathbf{H}^T \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} -W_{kd}^{(M)} \frac{[\Delta_d(\mathbf{x}_i, \mathbf{x}_j)]^2}{\gamma^2} C_{ijk}\, \mathbf{u}^{ij}\, (\mathbf{u}^{ij})^T] \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N} \xi_{ij} g(\Delta_d(\mathbf{x}_j, \mathbf{x}_i))
\end{aligned} \tag{10}$$

where $C_{ijk} = \exp\{-\frac{\sum_{d=1}^{d_{M-1}} (W_{kd}^{(M)})^2 [\Delta_d(\mathbf{x}_i, \mathbf{x}_j)]^2}{2\gamma^2}\}$, ξ_{ij} is the $(i,j)^{th}$ entry of $\mathbf{H}\mathbf{H}^T$. $g(\Delta_d(\mathbf{x}_i, \mathbf{x}_j))$ represents $\frac{\partial \mathbf{L}}{\partial W_{kd}^{(M)}}$, and $g(\Delta_d(\mathbf{x}_i, \mathbf{x}_j)) \propto [\Delta_d(\mathbf{x}_i, \mathbf{x}_j)]^2 \exp\{-\sum_{d=1}^{d_{M-1}} [\Delta_d(\mathbf{x}_i, \mathbf{x}_j)]^2\}$.

Discussions: For an image $\mathbf{x}_i$, its contribution to the gradient in Eqn. (10) can be measured by $\sum_{j=1}^{N} \xi_{ij} g(\Delta_d(\mathbf{x}_i, \mathbf{x}_j))$. Obviously, this term is non-zero if and only if $i \neq j$ and $\xi_{ij} \neq 0$ (i.e., $\hat{\mathbf{y}}_i = \hat{\mathbf{y}}_j$). As ξ_{ij} plays a role of an indicator, the quantized value of the contribution mainly depends on the the value of $g(\Delta_d(\mathbf{x}_i, \mathbf{x}_j))$. The curve of $g(\Delta_d(\mathbf{x}_i, \mathbf{x}_j))$ with the changes of $\Delta_d(\mathbf{x}_i, \mathbf{x}_j)$ is show in Fig. 2. We can observe that with the increasing of d_{M-1}, the extreme point is very close to the coordinate origin and only the monotone decreasing part in the curve can be reflected. Therefore, if $\mathbf{x}_i$ is a noise sample, that is, it is very far from other images in the same category. Then the $\Delta_d(\mathbf{x}_i, \mathbf{x}_j)$ is large, and therefore its contribution $g(\Delta_d(\mathbf{x}_i, \mathbf{x}_j))$ in the gradient will be small.

Algorithm 1 Weakly-Supervised CNN model

Input: Noisy Web Training Images: $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N]$
Initial Parameters $\mathbf{W} = [\mathbf{W}^{(1)}, ..., \mathbf{W}^{(M)}]$
Rectified Linear Activation Function: $f(\cdot)$
Procedure:
Repeat:
Forward Propagation:
Implementing as the traditional CNN
Backward Propagation:
1. For $m = M$, calculate
$\frac{\partial(\tilde{\mathcal{L}})}{\partial W_{kd}^{(M)}} = \frac{\partial(\mathcal{L})}{\partial W_{kd}^{(M)}} + \alpha \sum_{i=1} \sum_{j=1} \xi_{ij} g(\Delta_d(\mathbf{x}_j, \mathbf{x}_i))$
$\delta_k^{(M)} = -\frac{\partial(\tilde{\mathcal{L}})}{\partial \mathbf{z}_k^{(M)}}$
2. For $m = M-1$ to $m = 2$, set
$\frac{\partial(\tilde{\mathcal{L}})}{\partial \mathbf{W}^{(m)}} = \delta^{(m+1)}(f(\mathbf{Z}^{(m)}))^T$
$\delta^{(m)} = [(\mathbf{W}^{(m)})^T \delta^{(m+1)}] \cdot f'(\mathbf{Z}^{(m)})$
Until The max iteration number
Output: $\mathbf{W} = [\mathbf{W}^{(1)}, ..., \mathbf{W}^{(M)}]$

Note that this suppression will be back-propagated to the first several layers by the "error term" $\delta^{(M)}$ defined in BP [20], thereby the contribution of the noise samples will be limited in each layer. The complete algorithm of our weakly-supervised CNN model is in Algorithm 1.

4. Experiments

Our experiments consist of two parts. First, we show the noise-robust performance of the proposed approach in image categorization tasks. Second, we show the vocabulary-free tagging performance with the vocabulary of personal photos and WordNet.

4.1. Image Categorization

Datasets: We conducted experiments on two widely-used datasets in image categorization. One is **CIFAR-10**, which consists of $60,000$ 32×32 color images of 10 classes. $50,000$ images are for training and $10,000$ for testing. To generate the noisy training data with different percentages in **CIFAR-10**, the training images of a certain percentage in a certain category were randomly replaced by the training images in other categories. The total number of images in a category remained unchanged. We set the percentages of the noise data from 10% to 90%.

The other dataset is **PASCAL VOC2007**, which consists of 9963 images of 20 classes, with the split of 50% for training/validation and 50% for testing. We trained a classification model of 20 categories using Web training images, and compared with the state-of-the-art methods.

Baselines: We denote the proposed method as noise-robust CNN (**NRCNN**). We compared the proposed method

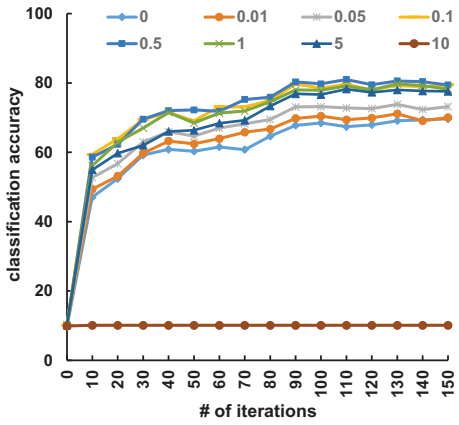

Figure 3: The performances with different α. We can see that the performances can keep in a stable range, except for one too large value ($\alpha = 10$).

with four baselines in CIFAR-10 and six baselines in VOC2007. The common four baselines are:

- **CNN**: the state-of-the-art **CNN** network with convolutional layers and fully-connected layers. We will specify the network structure in each task.
- **RPCA+CNN**: before the CNN training, we reconstruct each training sample by **RPCA** [3] and remove those samples with large reconstruction error. The removal ratio is set as the same as the noise percentage.
- **CAE+CNN**: we pre-train the convolutional layers of CNN by the convolutional autoencoder (**CAE**) in a layer-wise way and fine-tune the entire network, which is reported in [15] to reduce the noise effect.
- **NL+CNN**: we reproduce the additional bottom-up **n**oise-adaption **l**ayer in [21], and combine this layer with CNN network.

We also compared with another two methods in VOC2007.

- **Best_VOC**: pre-training using ImageNet, and fine-tuning in VOC2007, which has achieved the state-of-the-art performance [18].
- **Web_HOG**: training concept representations by the part-based model and human-crafted features with Web training images [22], which is the most recent work in this topic.

Results: First of all, we adjusted the weight decay value of the basic CNN model, i.e., β in Eqn. (2), on the two datasets. For different noise percentages (from 10% to 90%), this value is 0.004 for 10%, 0.008 for 20%, and 0.04 for the rest. We found that the above parameters can make the basic CNN model achieve the best result on both datasets. In addition, we empirically set γ to 0.1 in Eqn. (3) so that the similarity value can be in an appropriate scale. Besides, there is only one adjustable parameter α in our model. Fig. 3 shows the effect of α to the classification

accuracy on the CIFAR-10 training data with 20% noises. We found that only when α is too large (e.g., 10), the model lost the classification ability and the accuracy remained at random values. For other values, the performance maintains at a stable range and achieves the best at 0.5. Besides, we found that the value of 0.5 can also ensure the best results for other noise percentages. Therefore, α is set to 0.5 in the following experiments.

Tab. 1 shows the classification accuracy on different noise percentages in CIFAR-10[3]. We can see that our model achieves the best accuracy for all cases. Our approach even achieved a slight improvement on the clean training data, compared with the traditional CNN. We found that the traditional CNN dropped by nearly 20% in CIFAR-10 with 30% noises. In contrast, our method only dropped about 10%, showing a strong robustness to noisy training data. In addition, we found an interesting fact about the data preprocessing method RPCA+CNN. When the noise percentage is less than 50%, this method shows performance improvement over traditional CNN. As the noises increase, the performance of RPCA+CNN gets lower than that of traditional CNN. The reason is that the risk of removing the correct samples by mistake will significantly increase with the increase of the noise percentage, which leads to the increase of noises in the final training data. The performance of CAE+CNN and NL+CNN are substantially similar. In the case of 30% noises, they drop by 17.0% and 15.9%, respectively. It indicates that although CAE+CNN can solve the problem where the noises are region-level [15] (e.g., background noises), its performance will greatly drop when the noises are sample-level, i.e., some images are totally noises to their categories. For NL+CNN, our experiments also demonstrated that it is insufficient to enhance the noise immunity simply by adding the last noise-adaption layer. In contrast, our method can limit the noise effect in all layers by the role of back propagation. Therefore we achieved the best classification results. We drew Fig. 4 to clearly reflect the declines of the classification accuracy on different noise percentages.

Furthermore, we evaluated the image categorization performance on PASCAL VOC2007 dataset[4]. We pre-trained our network by ImageNet 1,000 categories as in [18], and fine-tuned the network by Web training data. Training images were crawled from commercial image search engines by using each category in VOC2007 as a query, where duplicate images were removed. We followed the splits of positive/negative samples provided by VOC2007 to construct the Web training dataset for each category. Note that we have two training sets. First, we kept the positive/negative samples with the same number of that in VOC2007 and denoted methods on this training set as CNN (Web) and NRCNN (Web). Second, we increased the positive samples to 4 times of that in VOC2007 for each category, and denoted methods on this training set as CNN (Web$\times$4) and NRCNN (Web$\times$4). From our statistics, the noise percentages for the setting of (Web) and (Web$\times$4) are about 20% and 40%, respectively. The average precision in VOC2007 test set is shown in Tab. 2. We can draw the following conclusions:

- CNN (Web) can surpass over Web_HOG with a significant gain, which demonstrates a stronger noise-robust ability of deep learning methods than the method using human-crafted features on noisy training data.
- NRCNN (Web$\times$4) is better than the state-of-the-art performance in [18]. Since the Web data is readily available, the cost of our model is small. We demonstrate that effectiveness of training a neural network by noise-robust model with noisy Web training images. However, the traditional CNN model cannot achiever the comparable result with ours.

Besides, we found that the proposed model took 1.34 times time-cost of the standard CNN model with the 128 batch size. We also found that the performances dropped about 5.0% and 3.1% when using the features in the first and second fully-connected layers for similarity computation respectively, compared to the last layer. The reason is the lack of the high-level semantics in other layers.

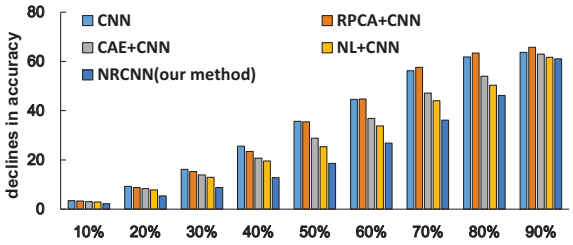

Figure 4: Compared to the performance on clean training data, the declined performance in terms of image classification accuracy on different noise percentages. The less the declined number, the better the method.

4.2. Tagging with the Vocabulary of Personal Photos

One of the most attractive features in the proposed method is that we can quickly obtain a deep learning model to describe any tags (categories) by leveraging the unlimited tags and training data on the Web. For example, categories in personal photos are typically biased toward the tags related to "landscape," "family," for which we do not have a human-labeled training set.

After collecting a set of 200 frequent categories from user-contributed tags from $10,000$ active users (who had uploaded more than 500 photos in the recent six months

[3]We used the "cifar10_quick_train_test" network in Caffe (caffe.berkeleyvision.org) as the baseline CNN model in this task.

[4]We used the "alexnet_train_val" network in Caffe as the baseline CNN model in this task

Table 1: Accuracy of image classification on the clean training data and the training data with different noise percentages.

Method	CIFAR-10									
	clean	10%	20%	30%	40%	50%	60%	70%	80%	90%
CNN	81.24	77.79	71.97	65.09	55.65	45.60	36.65	25.02	19.46	17.55
RPCA[3]+CNN	81.24	77.94	72.44	65.94	57.82	45.77	36.55	23.68	17.85	15.49
CAE[16]+CNN	81.55	78.54	73.19	67.69	60.83	52.71	44.71	34.39	27.54	18.61
NL+CNN[21]	81.16	78.28	73.36	68.26	61.63	55.83	47.33	37.12	30.81	19.49
NRCNN(our method)	**81.60**	**79.39**	**76.21**	**72.81**	**68.79**	**63.01**	**54.78**	**45.48**	**35.43**	**20.56**

Table 2: Average precision per class on the VOC2007 test set. The words in brackets indicate: "Web," this method uses the positive/negative Web training images of the same number as the standard setting in VOC2007; "Web×4," compared to "Web," the number of positive images used in this setting is increased to 4 times.

class / method	plane	bike	bird	boat	btl	bus	car	cat	chr	cow	tab	dog	horse	moto	pers	plnt	shp	sfa	train	tv	**mAP**
Best_VOC [18]	88.5	**81.5**	**87.9**	**82.0**	47.5	75.5	**90.1**	87.2	61.6	75.7	**67.3**	85.5	83.5	80.0	**95.6**	60.8	76.8	**58.0**	**90.4**	**77.9**	77.7
Web_HOG [22]	68.5	48.2	47.3	55.7	40.0	56.3	60.1	64.1	43.6	59.2	32.9	46.5	56.2	62.4	41.3	29.6	41.4	35.6	68.9	35.5	49.6
CNN(Web)	84.1	68.8	77.1	73.0	63.0	74.2	74.3	79.2	61.8	73.8	48.9	79.5	81.0	82.1	48.4	57.9	72.0	31.6	83.4	64.7	68.9
CNN(Web×4)	85.4	69.4	77.1	74.5	63.7	74.7	75.0	81.6	62.3	75.7	53.3	80.2	83.8	84.6	50.7	58.9	75.9	41.0	84.5	69.1	71.1
NRCNN(Web)	85.8	69.7	77.4	75.1	63.8	75.8	75.6	82.7	62.7	76.9	53.5	80.6	84.7	84.9	49.2	59.1	76.0	50.8	84.8	69.2	71.9
NRCNN(Web×4)	**91.3**	75.2	83.3	81.5	**70.2**	**81.3**	80.6	**88.3**	**67.0**	**82.5**	60.0	**86.3**	**90.0**	**90.3**	75.8	**64.8**	**81.0**	57.8	89.9	74.9	**78.6**

with registration time more than two years) in Flickr, we found that 50 categories, e.g., "sunset," "sightseeing," and "birthday" cannot be found even in the category list in ImageNet. For the all 200 categories, we can only use the ImageNet dataset to train a CNN model on the 150 existing categories, with $1,000$ clean ImageNet training images for each category. We denote this method as **CNN (ImageNet)**. To train the complete 200 categories, we crawled $1,000$ images from a commercial image search engine for each category, removed duplicate images and trained deep learning models. Note that all methods were conducted with their best parameters, respectively. Besides, an alternative way to predict new categories is by zero-shot learning. We therefore implemented DeViSE [7] as an additional baseline, which is trained on the 150 existing categories as CNN (ImageNet) and tested on the complete 200 categories by semantic extension.

We used the same network as in PASCAL VOC2007, and trained the network without pre-training scheme. A randomly-selected $1,000$ photos from **MIT-Adobe FiveK** Dataset [2] were used as the test set. Each method produces top five categories with the highest prediction scores as a tagging list. 25 human-labelers were employed to evaluate each tag with three levels: 2–Highly Relevant; 1–Relevant; 0–Non Relevant. We adopted the Normalized Discounted Cumulative Gain (NDCG) as the metric to evaluate the tagging performance. The NDCG measures multi-level relevance and assumes that the relevant tags are more useful when appearing higher in a ranked list. This metric at the position of p in the ranked list is defined by:

$$NDCG@p = Z_p \sum_{i=1}^{p} \frac{2^{r^i} - 1}{log(1+i)}, \tag{11}$$

where 2^{r^i} is the relevance level of the i^{th} tag and Z_p is a normalization constant such that $NDCG@p = 1$ for the perfect ranking.

The result is shown in Tab. 3. We can see that our proposed method achieves a consistently better result than other noise-resistant methods. Besides, CNN (ImageNet) is inferior to our method, because of its limited vocabulary. The result also demonstrates that by leveraging Web training images on new categories, we can obtain a superior re-

Figure 5: Tagging results produced by the proposed method. Note that the underlined tags are missing in the ImageNet categories, but are important for personal photos.

Table 3: Tagging performance in terms of NDCG for the 1,000 testing photos in MIT-Adobe FiveK dataset.

	CNN (Web)	RPCA+CNN (Web)	CAE+CNN (Web)	NL+CNN (Web)	CNN (ImageNet)	DeViSE (ImageNet)[7]	NRCNN (Web)
NDCG@1	0.08	0.23	0.11	0.24	0.20	0.28	**0.32**
NDCG@3	0.18	0.32	0.25	0.33	0.29	0.36	**0.41**
NDCG@5	0.26	0.39	0.34	0.41	0.39	0.43	**0.46**

Table 4: The tagging performance in terms of Similarity@K trained by different models and different vocabulary sets. ImageNet-1K is the vocabulary set of 1,000 categories in ImageNet competition. WordNet-63K is the largest vocabulary set used in this paper.

Vocabulary Set	Model	Similarity@1	Similarity@2	Similarity@5	Similarity@10	Similarity@20
ImageNet-1K	CNN	**0.88**	**0.85**	**0.51**	0.43	0.22
WordNet-63K	CNN	0.57	0.47	0.38	0.31	0.26
WordNet-63K	NRCNN (our method)	0.58	0.56	**0.51**	**0.45**	**0.36**

sult than the semantic-embedded method DeViSE. Fig. 5 further illustrates some exemplary tagging results. We can observe that our approach can provide users with accurate tags where some are even excluded in the category list in ImageNet.

4.3. Tagging with the Vocabulary of WordNet

We further train a tagging model with a larger vocabulary set from WordNet. WordNet covers about 82,000 pairs of the item ID and tag list[5]. Since the tags in a tag list refer to a synset, we keep the first tag as the representative of the tag list. For each tag, we crawled about 50 images from a commercial image search engine as the training data. We removed invalid images or images whose width or height is smaller than 200 pixels. Then after this processing, we further removed tags which contained less than 30 images from the vocabulary set. Finally, we collected 63,043 tags and about 2.4 million training images in total. To the best of our knowledge, this is the largest vocabulary set in the image tagging area. We kept the same network as above, fine-tuned the network by Web training data with the released Alex's network parameters [12] in Caffe as the pre-trained parameters. Although the number of training data for each category is limited, we will show good image tagging results with the help of the proposed noise-robust model and the largest vocabulary set.

We randomly selected 20,000 images from the ImageNet validation set as the testing images. To compare the tagging performance of different approaches, we calculated the cosine similarity between the word vector of the category name of each testing image and the word vector of each tag produced by different models. The word vectors can be calculated by this tool[6]. We defined an average similarity as Similarity@K by averaging the similarity scores among the top-ranked K tags.

We show the results in Tab. 4. The results of the second row are achieved by the released Alex's network in Caffe. The results of the third and forth row are achieved by fine-tuning the network by Web training data on about 63,000 tags with the released Alex's network as the pre-trained parameters. We observe that our model can achieve better results from Similarity@5 to Similarity@20, than the traditional CNN model, which is implemented by the released Alex's network on the 1,000 vocabulary set. Our model can predict a wide range of tags, and achieve a significant improvement with the gain of 14.0% in terms of Similarity@20, against the CNN model on the 1,000 vocabulary set. We show the exemplar tagging results in the supplementary material A2. The lower results on Similarity@1 and Similarity@2 are derived from the variety of tags and the limited number of training images. We will solve this problem by using more powerful GPUs that can involve more training samples in each category within a reasonable time-cost (e.g, one week as we need currently).

5. Conclusions

In this paper, we propose a noise-robust deep learning model on noisy training data. The merit is that we can quickly train a deep learning model for any categories without human-labeled training data and apply the model to real-applications. By leveraging the mutual relationships of features in the output layer, the contribution of noise images are weakened in the back propagation. Experiments demonstrate the superior performance. In the future, we will apply the weakly-supervised model to more image domains.

6. Acknowledgements

This work was supported by the 863 Program 2014AA015104, and the National Natural Science Foundation of China (61273034 and 61332016).

[5] image-net.org/archive/words.txt
[6] nlp.stanford.edu/projects/glove/

References

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, pages 585–591, 2001.

[2] V. Bychkovsky, S. Paris, E. Chan, and F. Durand. Learning photographic global tonal adjustment with a database of input / output image pairs. In *CVPR*, 2011.

[3] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *J. ACM*, 58(3):11:1–11:37, 2011.

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

[5] J. Deng, A. C. Berg, K. Li, and L. Fei-Fei. What does classifying more than 10,000 image categories tell us? In *ECCV*, pages 71–84, 2010.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[7] A. Frome, G. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013.

[8] J. Fu, T. Mei, K. Yang, H. Lu, and Y. Rui. Tagging personal photos with transfer deep learning. In *WWW*, pages 344–354, 2015.

[9] J. Fu, J. Wang, Y. Rui, X.-J. Wang, T. Mei, and H. Lu. Image tag refinment with view-dependent concept representations. In *IEEE Transactions on CSVT*, volume 25, pages 1409–1422, 2015.

[10] B. Graham. Spatially-sparse convolutional neural networks. *arXiv preprint arXiv:1409.6070*, 2014.

[11] J. Kim and C. D. Scott. Robust kernel density estimation. In *JMLR*, pages 2529–2565, 2012.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012.

[13] J. Larsen, L. N. Andersen, M. Hintz-madsen, and L. K. Hansen. Design of robust neural network classifiers. In *ICASSP*, pages 1205–1208, 1998.

[14] W. Liu, G. Hua, and J. R. Smith. Unsupervised one-class learning for automatic outlier removal. In *CVPR*, 2014.

[15] P. Luo, X. Wang, and X. Tang. Hierarchical face parsing via deep learning. In *CVPR*, pages 2480–2487, 2012.

[16] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *ICANN*, pages 52–59, 2011.

[17] V. Mnih and G. E. Hinton. Learning to label aerial images from noisy data. In *ICML*, 2012.

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

[19] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Is object localization for free? - weakly-supervised learning with convolutional neural networks. In *CVPR*, pages 685–694, 2015.

[20] D. Rumelhart, G. Hintont, and R. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[21] R. F. Sainbayar Sukhbaatar. Learning from noisy labels with deep neural networks. *arXiv preprint arXiv:1406.2080*, 2014.

[22] C. S.K. Divvala, A.Farhadi. Learning everything about anything: Webly-supervised visual concept learning. In *CVPR*, 2014.

[23] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *JMLR*, 11:3371–3408, 2010.

[24] J. Wu, Y. Yu, C. Huang, and K. Yu. Deep multiple instance learning for image classification and auto-annotation. In *CVPR*, pages 3460–3469, 2015.

[25] H. Xu, C. Caramanis, and S. Mannor. Outlier-robust pca: The high-dimensional case. *IEEE Transactions on Information Theory*, 59(1):546–572, 2013.

[26] Y. Yang, H. T. Shen, Z. Ma, Z. Huang, and X. Zhou. $l_{2,1}$-norm regularized discriminative feature selection for unsupervised learning. In *IJCAI*, pages 1589–1594, 2011.

[27] J. Ye, Z. Zhao, and M. Wu. Discriminative k-means for clustering. In *NIPS*, 2007.