Learning From Massive Noisy Labeled Data for Image Classification

Tong Xiao[1], Tian Xia[2], Yi Yang[2], Chang Huang[2] and Xiaogang Wang[1]

[1]The Chinese University of Hong Kong
[2]Baidu Research

Abstract

Large-scale supervised datasets are crucial to train convolutional neural networks (CNNs) for various computer vision problems. However, obtaining a massive amount of well labeled data is usually very expensive and time consuming. In this paper, we introduce a general framework to train CNNs with only limited number of clean labels and millions of easily obtained noisy labels. We describe label noises with a probabilistic graphical model and further integrate it into an end-to-end deep learning system. To demonstrate the effectiveness of our approach, we collect a large-scale clothing classification dataset with both noisy and clean labels. Experiments on this dataset show that our model can better detect and correct the wrong labels, which benefits the training of underlying CNNs.

1. Introduction

Deep learning with massive amount of supervised training data has recently shown very impressive improvement on multiple image recognition challenges including image classification [13], attribute learning [30], scene classification [9], *etc.* While state-of-the-art results have been continuously reported [29, 24, 26], all these methods require reliable annotations from millions of images [7] which are often expensive and time-consuming to obtain [7], preventing deep models from being quickly trained on new image recognition problems. Thus it is necessary to develop new efficient labeling and training frameworks for deep learning.

One possible solution is to automatically collect large amount of annotations from the Internet web images [11] (i.e. extracting tags from the surrounding texts or keywords from search engines) and directly use them as ground truth to train deep models. Unfortunately, these labels are extremely unreliable due to the various types of noise (i.e. labeling mistakes from annotators or computing errors from extraction algorithms). Many works have shown that these noisy labels could adversely impact the classification accuracy of the induced classifiers [32, 21, 23]. Various label noise-robust algorithms are developed but experiments show that performances of classifiers inferred by robust algorithms are still affected by label noise [3, 27]. Other data cleansing algorithms are proposed [2, 5, 18], but these approaches are difficult in distinguishing informative hard examples from harmful mislabeled ones.

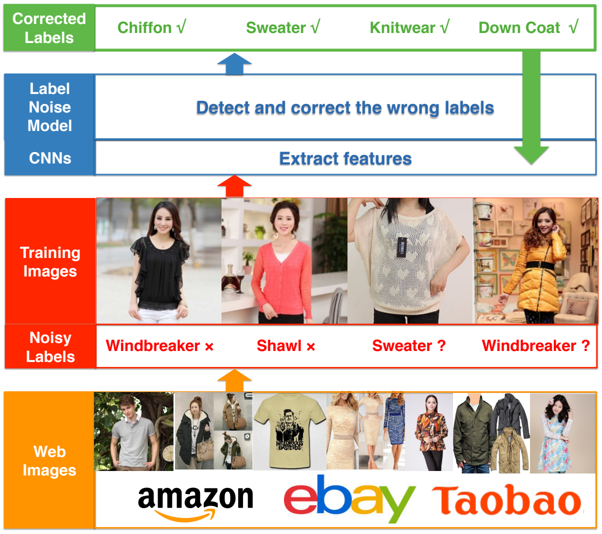


Figure 1. Overview of our approach. Labels of web images often suffer from different types of noise. A label noise model is proposed to detect and correct the wrong labels. The corrected labels are used to train underlying CNNs.

Although annotating all the data is costly, it is often easy to obtain a small amount of clean labels. Based on the observation of transferability of deep neural networks, people first initialize the parameters with a model pretrained on a larger yet related dataset [13], and then finetune on the smaller dataset of specific tasks [22, 1, 8]. Such methods may better avoid overfitting and utilize the relationships between the two datasets. However, we find that training a Convolutional Neural Network (CNN) from scratch with limited clean labels and massive noisy labels is better than finetuning it only on the clean labels. Other approaches address the problem as semi-supervised learning where noisy labels are discarded [31]. These algorithms usually suffer from model complexity thus cannot be applied on large-scale datasets. Therefore, it is inevitable to develop a better way of using huge amount of noisy labeled data.

Our goal is to build an end-to-end deep learning system that is capable of training with both limited clean labels and massive noisy labels more effectively. Figure 1 shows the framework of our approach. We collect $1,000,000$ clothing images from some online shopping websites. Each image is automatically assigned with a noisy label according to the keywords in its surrounding text. After that, we manually refine $72,409$ image labels, which constitute a clean sub-dataset. All the data are then used to train CNNs, while the major challenge is to identify and correct the wrong labels during the training process.

To cope with this challenge, we extend CNNs with a novel probabilistic model, which infers the ground truths and uses them to supervise the training of the network. Our work is inspired by [25], which modifies a CNN by inserting a linear layer on top of the softmax layer to map clean labels to noisy labels. However, [25] assumes noisy labels are conditionally independent with input images given clean labels. By examining our collected dataset, we find that this assumption is too strong to fit the real-world data well. For example, in Figure 2, all the images should belong to "Hoodie". The top five are correct while the bottom five are either mislabeled as "Windbreaker" or "Jacket". It shows that images tend to be mislabeled may share similar visual patterns. This is because different sellers have their own bias on different categories, thus they may provide wrong keywords for many similar clothes. Based on these observations, we introduce two types of label noise:

- **Confusing noise** makes the noisy label reasonably wrong. It usually occurs when the image content is confusing (*e.g.*, the samples with "?" in Figure 1).

- **Pure random noise** makes the noisy label totally wrong. It is often caused by either the mismatch between an image and its surrounding text, or false conversion from the text to label (*e.g.*, the samples with "×" in Figure 1).

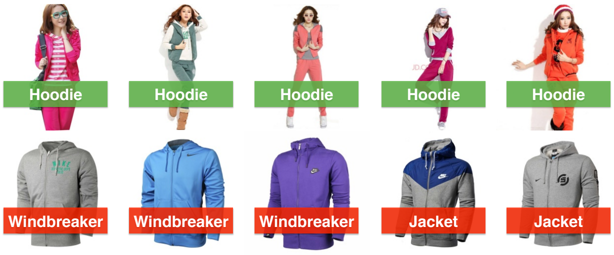


Figure 2. Images tend to be mislabeled often share similar visual patterns.

The proposed probabilistic model build the relations among the image, noisy label, ground truth label, and the noise type, where the latter two are treated as latent variables. We use the Expectation-Maximization (EM) algorithm to solve the problem and integrate it into the training process of CNNs. Experiments on the collected dataset show that our model can better detect and correct the wrong labels.

Our contribution comes from three aspects. First, we study the cause of noisy labels in real-world data and describe it with a novel probabilistic model. Second, we integrate the model into existing deep learning framework to help detect and correct wrong labels. We study different training strategies to make the CNNs learn from better supervisions. Finally, we collect a large-scale clothing dataset with both noisy and clean labels, which will be released for academic use.

2. Related Work

For most of the related works including effect of label noises, taxonomy of label noises, robust algorithms and noise cleaning algorithms for learning with noisy data, we refer to [10] for a comprehensive review.

Direct learning with noisy labels: Many works have shown that label noise can adversely impact the classification accuracy of induced classifiers [32]. To better handle label noise, some approaches rely on training classifiers with label noise-robust algorithms [4, 16]. However, [3] proves that most of the loss functions are not completely robust to label noise. Experiments in [27] show that the classifiers inferred by label noise-robust algorithms are still affected by label noise. These methods seem to be adequate only when label noise can be safely managed by overfitting avoidance [10]. On the other hand, some label noise cleansing methods are proposed to remove or correct mislabeled instances [2, 5, 18], but these approaches are difficult in distinguishing informative hard examples from harmful mislabeled ones. Thus they may remove too many instances and the overcleansing can reduce the performances of classifiers [17].

Semi-supervised learning: Apart from directly learning with label noise, some semi-supervised learning algorithms are developed to utilize weakly labeled or even unlabeled data. Label Propagation method [31] explicitly uses ground truths of well-labeled data to classify unlabeled samples. However, it suffers from computing pairwise distance, which has quadratic complexity with the number of samples and cannot be applied on large-scale datasets. Chen *et al.* [6] tackled this problem in the scenario of image tagging by a co-regularized duo classifier that can efficiently use both the image and incomplete tags. Weston *et al.* [28] proposed to embed a pairwise loss in the middle layer of a deep neural network, which benefits the learning of discriminative features. But they need extra information about whether a pair of unlabeled images belong to the same class, which cannot be obtained in our problem.

Transfer learning: The success of Convolutional Neural Networks (CNNs) lies in their capability of learning rich and hierarchical image features. However, the model parameters cannot be properly learned when training data is not enough. Researchers proposed to conquer this problem by first initializing the CNN parameters with a model pretrained on a larger yet related dataset, and then finetuning it on the smaller dataset of specific task [13, 22, 1, 8]. Nevertheless, this transfer learning scheme could be suboptimal when the two tasks are just loosely related. In our case of clothing classification, we find that training a CNN from scratch with limited clean labels and massive noisy labels is better than finetuning it from an ImageNet pretrained model only on the clean labels.

Noise modeling with deep learning: Various methods have been proposed to handle label noise in different problem settings, but there are very few works about deep learning from noisy labels [19, 14, 25]. [19] builds a simple noise model for aerial images but only considers binary classification. [14] assumes label noise is independent from the true class label which is a simple and a specific case. [25] generalizes from them by considering multi-class classification and modeling class dependent noise, but they assume the noise is conditionally independent with the image content, ignoring the hardness of labeling images of different confusing levels. Our work can be viewed as a generalization of [25, 20] and our model is flexible enough to not only class dependent noise but also input dependent noise.

3. Label Noise Model

We target on the problem of learning a classifier from a set of images with noisy labels. To be specific, we have a noisy labeled dataset $\mathcal{D}_n = \left\{\left(\mathbf{x}^{(1)}, \tilde{y}^{(1)}\right), \ldots, \left(\mathbf{x}^{(N)}, \tilde{y}^{(N)}\right)\right\}$ with n-th image $\mathbf{x}^{(n)}$ and its corresponding noisy label $\tilde{y}^{(n)} \in \{1, \ldots, L\}$, where L is the number of classes. We describe how the noisy label is generated by using a probabilistic graphical model shown in Figure 3.

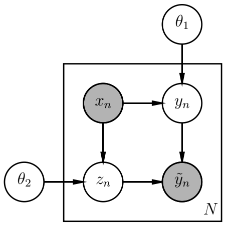


Figure 3. Probabilistic graphical model of label noise

Despite the observed image $\mathbf{x}$ and the noisy label $\tilde{\mathbf{y}}$, we exploit two discrete latent variables — $\mathbf{y}$ and $\mathbf{z}$ — to represent the ground truth and the type of label noise, respectively. Both $\tilde{\mathbf{y}}$ and $\mathbf{y}$ are L-dimensional binary random variables in 1-of-L fashion, *i.e.*, only one element is equal to 1 while others are all 0.

On the other hand, the latent variable $\mathbf{z}$ representing label noise type is an 1-of-3 binary random variable. It is associated with three different semantic meanings:

1. The label is noise free, *i.e.*, $\tilde{\mathbf{y}}$ should be equal to $\mathbf{y}$

2. The label suffers from a pure random noise, *i.e.*, $\tilde{\mathbf{y}}$ can take any possible value other than $\mathbf{y}$

3. The label suffers from a confusing noise, *i.e.*, $\tilde{\mathbf{y}}$ can take several values that are confusing with $\mathbf{y}$

Following this assignment rule, we define the conditional probability of the noisy label as

$$p(\tilde{\mathbf{y}}|\mathbf{y}, \mathbf{z}) = \begin{cases} \mathbf{I}\mathbf{y} & \text{if } \mathbf{z}_1 = 1 \\ \frac{1}{L-1}(\mathbf{U} - \mathbf{I})\mathbf{y} & \text{if } \mathbf{z}_2 = 1 \\ \mathbf{C}\mathbf{y} & \text{if } \mathbf{z}_3 = 1 \end{cases} \tag{1}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{U}$ is the unit matrix (all the elements are one), and $\mathbf{C}$ is a sparse stochastic matrix with $\mathrm{tr}(\mathbf{C}) = 0$ and $\mathbf{C}_{ij}$ denoting the confusion probability between class i and j. Then we can derive from Figure 3 the joint distribution of $\tilde{\mathbf{y}}, \mathbf{y}$ and $\mathbf{z}$ conditioning on $\mathbf{x}$:

$$p(\tilde{\mathbf{y}}, \mathbf{y}, \mathbf{z}|\mathbf{x}) = p(\tilde{\mathbf{y}}|\mathbf{y}, \mathbf{z})p(\mathbf{y}|\mathbf{x})p(\mathbf{z}|\mathbf{x}) \tag{2}$$

While the class label probability distribution $p(\mathbf{y}|\mathbf{x})$ is comprehensible, the semantic meaning of $p(\mathbf{z}|\mathbf{x})$ needs extra clarification: it represents how confusing

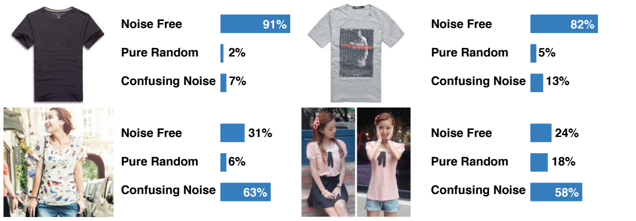


Figure 4. Predicting noise types of four different "T-shirt" images. The top two can be recognized with little ambiguity, while the bottom two are easily confusing with the class "Chiffon". Image content can affect the possibility of it to be mislabeled.

the image content is. Specific to our clothing classification problem, $p(\mathbf{z}|\mathbf{x})$ can be affected by different factors, including background clutter, image resolution, the style and material of the clothes, *etc.* We show some examples in Figure 4.

To illustrate the relations between noisy label and ground truth, we can derive their conditional probability from Eq 2

$$p(\tilde{\mathbf{y}}|\mathbf{y},\mathbf{x}) = \sum_{\mathbf{z}} p(\tilde{\mathbf{y}},\mathbf{z}|\mathbf{y},\mathbf{x}) = \sum_{\mathbf{z}} p(\tilde{\mathbf{y}}|\mathbf{y},\mathbf{z})p(\mathbf{z}|\mathbf{x}) \quad (3)$$

which can be interpreted as a mixture model. Given an input image $\mathbf{x}$, the conditional probability $p(\mathbf{z}|\mathbf{x})$ can be seen as the prior of each mixture component. This makes a key difference between our work and [25], where they assume $\tilde{\mathbf{y}}$ is conditionally independent with $\mathbf{x}$ if $\mathbf{y}$ is given. All the images share a same noise model in [25], while in our approach each data sample has its own.

3.1. Learning the Parameters

We exploit two CNNs to model $p(\mathbf{y}|\mathbf{x})$ and $p(\mathbf{z}|\mathbf{x})$ separately. Denote the parameter set of each deep model by θ_1 and θ_2. Our goal is to find the optimal $\theta = \theta_1 \cup \theta_2$ that maximize the incomplete log-likelihood $\log p(\tilde{\mathbf{y}}|\mathbf{x};\theta)$. The Expectation-Maximization algorithm is used to solve this problem iteratively.

For any probability distribution $q(\mathbf{y},\mathbf{z}|\tilde{\mathbf{y}},\mathbf{x})$, we can derive a lower bound of the incomplete log-likelihood by

$$\begin{aligned} \log p(\tilde{\mathbf{y}}|\mathbf{x};\theta) &= \log \sum_{\mathbf{y},\mathbf{z}} p(\tilde{\mathbf{y}},\mathbf{y},\mathbf{z}|\mathbf{x};\theta) \\ &\geq \sum_{\mathbf{y},\mathbf{z}} q(\mathbf{y},\mathbf{z}|\tilde{\mathbf{y}},\mathbf{x}) \log \frac{p(\tilde{\mathbf{y}},\mathbf{y},\mathbf{z}|\mathbf{x};\theta)}{q(\mathbf{y},\mathbf{z}|\tilde{\mathbf{y}},\mathbf{x})} \end{aligned} \quad (4)$$

E-Step The difference between $\log p(\tilde{\mathbf{y}}|\mathbf{x};\theta)$ and its lower bound is the Kullback-Leibler divergence $\mathrm{KL}\left(q(\mathbf{y},\mathbf{z}|\tilde{\mathbf{y}},\mathbf{x})||p(\mathbf{y},\mathbf{z}|\tilde{\mathbf{y}},\mathbf{x};\theta)\right)$, which is equal to zero if and only if $q(\mathbf{y},\mathbf{z}|\tilde{\mathbf{y}},\mathbf{x}) = p(\mathbf{y},\mathbf{z}|\tilde{\mathbf{y}},\mathbf{x};\theta)$. Therefore, in each iteration t, we first compute the posterior of latent variables using current parameters $\theta^{(t)}$:

$$\begin{aligned} p(\mathbf{y},\mathbf{z}|\tilde{\mathbf{y}},\mathbf{x};\theta^{(t)}) &= \frac{p(\tilde{\mathbf{y}},\mathbf{y},\mathbf{z}|\mathbf{x};\theta^{(t)})}{p(\tilde{\mathbf{y}}|\mathbf{x};\theta^{(t)})} \\ &= \frac{p(\tilde{\mathbf{y}}|\mathbf{y},\mathbf{z};\theta^{(t)})p(\mathbf{y}|\mathbf{x};\theta^{(t)})p(\mathbf{z}|\mathbf{x};\theta^{(t)})}{\sum_{\mathbf{y}',\mathbf{z}'} p(\tilde{\mathbf{y}}|\mathbf{y}',\mathbf{z}';\theta^{(t)})p(\mathbf{y}'|\mathbf{x};\theta^{(t)})p(\mathbf{z}'|\mathbf{x};\theta^{(t)})} \end{aligned} \quad (5)$$

Then the expected complete log-likelihood can be written as

$$Q(\theta;\theta^{(t)}) = \sum_{\mathbf{y},\mathbf{z}} p(\mathbf{y},\mathbf{z}|\tilde{\mathbf{y}},\mathbf{x};\theta^{(t)}) \log p(\tilde{\mathbf{y}},\mathbf{y},\mathbf{z}|\mathbf{x};\theta) \quad (6)$$

M-Step Since deep neural networks are exploited to model the probability $p(\mathbf{y}|\mathbf{x};\theta_1)$ and $p(\mathbf{z}|\mathbf{x};\theta_2)$, we perform gradient ascent on Q:

$$\begin{aligned} \frac{\partial Q}{\partial \theta} &= \sum_{\mathbf{y},\mathbf{z}} p(\mathbf{y},\mathbf{z}|\tilde{\mathbf{y}},\mathbf{x};\theta^{(t)}) \frac{\partial}{\partial \theta} \log p(\tilde{\mathbf{y}},\mathbf{y},\mathbf{z}|\mathbf{x};\theta) \\ &= \sum_{\mathbf{y}} p(\mathbf{y}|\tilde{\mathbf{y}},\mathbf{x};\theta^{(t)}) \frac{\partial}{\partial \theta_1} \log p(\mathbf{y}|\mathbf{x};\theta_1) + \\ &\qquad \sum_{\mathbf{z}} p(\mathbf{z}|\tilde{\mathbf{y}},\mathbf{x};\theta^{(t)}) \frac{\partial}{\partial \theta_2} \log p(\mathbf{z}|\mathbf{x};\theta_2) \end{aligned} \quad (7)$$

The M-Step above is equivalent to minimize the cross entropy between the estimated ground truth distribution and the prediction of the classifier.

3.2. Estimating the Matrix C

Notice that we do not set parameters to the conditional probability $p(\tilde{\mathbf{y}}|\mathbf{y},\mathbf{z})$ in Eq (1) and keep it unchanged during the learning process. Because without other regularizations, learning all the three parts could lead to trivial solutions. For example, the network will always predict $\mathbf{y}_1 = 1$, $\mathbf{z}_3 = 1$, and the matrix $\mathbf{C}$ is learned to make $\mathbf{C}_{1i} = 1$ for all i. To avoid this degeneration, we choose to give $\mathbf{C}$ a good estimation while let the deep model learn to predict $p(\mathbf{y}|\mathbf{x})$ and $p(\mathbf{z}|\mathbf{x})$ for each input image.

We estimate $\mathbf{C}$ on a relatively small dataset $\mathcal{D}_c = \{(\mathbf{x},\mathbf{y},\tilde{\mathbf{y}})\}_N$, where we have N images with both clean and noisy labels. As prior information about $\mathbf{z}$ is not available, we just solve the following optimization problem:

$$\max_{\mathbf{C},\mathbf{z}^{(1)},\cdots,\mathbf{z}^{(N)}} \sum_{i=1}^{N} \log p(\tilde{\mathbf{y}}^{(i)}|\mathbf{y}^{(i)},\mathbf{z}^{(i)}) \quad (8)$$

Obviously, sample i contributes nothing to the optimal $\mathbf{C}^*$ if $\mathbf{y}^{(i)}$ and $\tilde{\mathbf{y}}^{(i)}$ are equal. So that we ignore those samples and reinterpret the problem in another form by exploiting Eq 1:

$$\begin{aligned}\max_{\mathbf{C},\mathbf{t}} \quad & E = \sum_{i=1}^{N'} \log \alpha^{\mathbf{t}_i} + \log(\tilde{\mathbf{y}}^{(i)T}\mathbf{C}\mathbf{y}^{(i)})^{1-\mathbf{t}_i} \\ \text{subject to} \quad & \mathbf{C} \text{ is a stochastic matrix of size } L \times L \\ & \mathbf{t} \in \{0,1\}^{N'}\end{aligned} \tag{9}$$

where $\alpha = \frac{1}{L-1}$ and N' is the number of remaining samples. The semantic meaning of the above formulation is that we need to assign each $(\mathbf{y}, \tilde{\mathbf{y}})$ pair the optimal noise type, while finding the optimal $\mathbf{C}$ simultaneously.

Next, we will show that the problem can be solved by a simple yet efficient algorithm in $O(N' + L^2)$ time complexity. Before we introduce the algorithm itself, some theorems need to be proved. Denote the optimal solution by $\mathbf{C}^*$ and $\mathbf{t}^*$

Lemma 1. $\mathbf{C}^*_{ij} \neq 0 \Rightarrow \mathbf{C}^*_{ij} > \alpha, \forall i,j \in \{1,\dots,L\}$

Proof. Suppose there exists some i, j such that $0 < \mathbf{C}^*_{ij} \leq \alpha$. Then we conduct following operations. First, we set $\mathbf{C}^*_{ij} = 0$ while adding its original value to other elements in column j. Second, for all the samples n where $\tilde{\mathbf{y}}^{(n)}_i = 1$ and $\mathbf{y}^{(n)}_j = 1$, we set $\mathbf{t}_n$ to 1. The resulting E is always greater than the original one, which leads to a contradiction. □

Theorem 1. $(\tilde{\mathbf{y}}^{(i)}, \mathbf{y}^{(i)}) = (\tilde{\mathbf{y}}^{(j)}, \mathbf{y}^{(j)}) \Rightarrow \mathbf{t}^*_i = \mathbf{t}^*_j, \forall i,j \in \{1,\dots,N'\}$

Proof. Suppose $\tilde{\mathbf{y}}^{(i)}_k = \tilde{\mathbf{y}}^{(j)}_k = 1$ and $\mathbf{y}^{(i)}_l = \mathbf{y}^{(j)}_l = 1$ but $\mathbf{t}^*_i \neq \mathbf{t}^*_j$. From Lemma 1 we know that elements in $\mathbf{C}^*$ is either 0 or greater than α. If $\mathbf{C}^*_{kl} = 0$, we can set $\mathbf{t}^*_i = \mathbf{t}^*_j = 1$, otherwise we can set $\mathbf{t}^*_i = \mathbf{t}^*_j = 0$. In either case the objective function E is greater than the original one, which draws a contradiction. □

Theorem 2. $\tilde{\mathbf{y}}^{(i)T}\mathbf{C}^*\mathbf{y}^{(i)} > \alpha \Leftrightarrow \mathbf{t}^*_i = 0$ *and* $\tilde{\mathbf{y}}^{(i)T}\mathbf{C}^*\mathbf{y}^{(i)} = 0 \Leftrightarrow \mathbf{t}^*_i = 1, \forall i \in \{1,\dots,N'\}$

Proof. The first part is straight forward. For the second part, $\mathbf{t}^*_i = 1$ implies $\tilde{\mathbf{y}}^{(i)T}\mathbf{C}^*\mathbf{y}^{(i)} \leq \alpha$. By using Lemma 1 we know that $\tilde{\mathbf{y}}^{(i)T}\mathbf{C}^*\mathbf{y}^{(i)} = 0$. □

Notice that if the true label is class i while the noisy label is class j, then it can only affect the value of $\mathbf{C}_{ij}$. Thus each column of $\mathbf{C}$ can be optimized separately. Theorem 1 further shows that samples with same pair of $(\tilde{\mathbf{y}}, \mathbf{y})$ share a same noise type. Then what really matters is the frequency of each of the $L \times L$ pairs $(\tilde{\mathbf{y}}, \mathbf{y})$. Considering a particular column $\mathbf{c}$, suppose there are M samples affecting this column. We can count the frequency of noisy label class 1 to L as $m_1, \dots, m_L$ and might as well set $m_1 \geq m_2 \geq \dots \geq m_L$. The problem is then converted to

$$\begin{aligned}\max_{\mathbf{c},\mathbf{t}} \quad & E = \sum_{k=1}^{L} m_k \left(\log \alpha^{\mathbf{t}_k} + \log \mathbf{c}_k^{1-\mathbf{t}_k}\right) \\ \text{subject to} \quad & \mathbf{c} \in [0,1]^L, \sum_{k=1}^{L} \mathbf{c}_k = 1 \\ & \mathbf{t} \in \{0,1\}^L\end{aligned} \tag{10}$$

Due to the rearrangement inequality, we can prove that in the optimal solution,

$$\max(\alpha, \mathbf{c}^*_1) \geq \max(\alpha, \mathbf{c}^*_2) \geq \dots \geq \max(\alpha, \mathbf{c}^*_L) \tag{11}$$

Then by using Theorem 2, there must exist a $k^* \in \{1,\dots,L\}$ such that

$$\begin{aligned}\mathbf{t}^*_i &= 0, i = 1,\dots,k^* \\ \mathbf{t}^*_i &= 1, i = k^*+1,\dots,L\end{aligned} \tag{12}$$

This also implies that only the first k^* elements of $\mathbf{c}^*$ have nonzero values (greater than α actually). Furthermore, if k^* is known, finding the optimal $\mathbf{c}^*$ is to solve the following problem:

$$\begin{aligned}\max_{\mathbf{c}} \quad & E = \sum_{k=1}^{k^*} m_k \log \mathbf{c}_k \\ \text{subject to} \quad & \mathbf{c} \in [0,1]^L, \sum_{k=1}^{k^*} \mathbf{c}_k = 1\end{aligned} \tag{13}$$

whose solution is

$$\begin{aligned}\mathbf{c}^*_i &= \frac{m_i}{\sum_{k=1}^{k^*} m_k}, i = 1,\dots,k^* \\ \mathbf{c}^*_i &= 0, i = k^*+1,\dots,L\end{aligned} \tag{14}$$

The above analysis leads to a simple algorithm. We enumerate k^* from 1 to L. For each k^*, $\mathbf{t}^*$ and $\mathbf{c}^*$ are computed by using Eq (12) and (14), respectively. Then we evaluate the objective function E and record the best solution.

4. Deep Learning from Noisy Labels

We integrate the proposed label noise model into a deep learning framework. As demonstrated in Figure 5, we predict the probability $p(\mathbf{y}|\mathbf{x})$ and $p(\mathbf{z}|\mathbf{x})$ by using two independent CNNs. Moreover, we append a label noise model layer at the end, which takes as input the CNNs' prediction scores and the observed noisy label. Stochastic Gradient Ascent with backpropagation

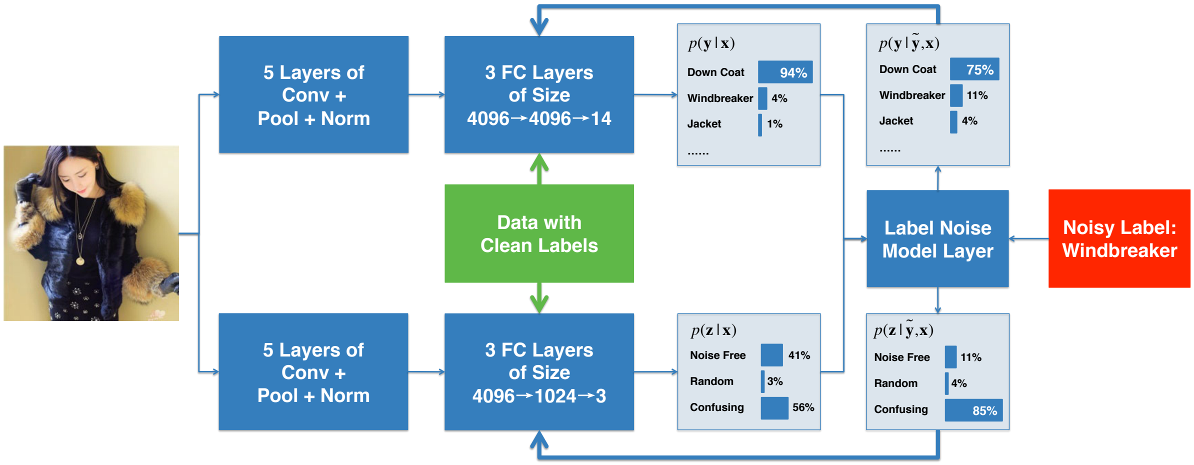

5 Layers of Conv + Pool + Norm
3 FC Layers of Size 4096→4096→14
$p(\mathbf{y}|\mathbf{x})$
Down Coat 94%
Windbreaker 4%
Jacket 1%
......
$p(\mathbf{y}|\tilde{\mathbf{y}},\mathbf{x})$
Down Coat 75%
Windbreaker 11%
Jacket 4%
......
Data with Clean Labels
Label Noise Model Layer
Noisy Label: Windbreaker
5 Layers of Conv + Pool + Norm
3 FC Layers of Size 4096→1024→3
$p(\mathbf{z}|\mathbf{x})$
Noise Free 41%
Random 3%
Confusing 56%
$p(\mathbf{z}|\tilde{\mathbf{y}},\mathbf{x})$
Noise Free 11%
Random 4%
Confusing 85%

Figure 5. System diagram of our method. Two CNNs are used to predict the class label $p(\mathbf{y}|\mathbf{x})$ and the noise type $p(\mathbf{z}|\mathbf{x})$, respectively. The label noise model layer infers the true label according to the predictions and the noisy label, which is then used to supervise the training of CNNs. Data with clean labels are also mixed in to prevent the models from drifting away.

technique is used to approximately optimize the whole network. In each forward pass, the label noise model layer computes the posterior of latent variables according to Eq (5). While in the backward pass, it computes the gradients according to Eq (7).

Directly training the whole network with random initialization is impractical, because the posterior computation could be totally wrong. Therefore, we need to pretrain each CNN component with strongly supervised data. Images and their ground truth labels in the dataset $\mathcal{D}_c$ are used to train the CNN that predicts $p(\mathbf{y}|\mathbf{x})$. On the other hand, the optimal solutions of $\mathbf{z}^{(1)}, \cdots, \mathbf{z}^{(N)}$ in Eq (8) are used to train the CNN that predicts $p(\mathbf{z}|\mathbf{x})$.

After both CNN components are properly pretrained, we can start training the whole network with massive noisy labeled data. However, some practical issues need to be further discussed. First, if we merely use noisy labels, we will lose precious knowledge that we have gained before and the model could be drifted. Therefore, we need to mix the data with clean labels into our training set, which is depicted in Figure 5 as the extra supervisions for the two CNNs. Then each CNN receives two kinds of gradients, one is from the clean labeled data and the other is from the noisy labeled data. We denote them by Δ_c and Δ_n, respectively. A potential problem is that $|\Delta_c| \ll |\Delta_n|$, because clean data is much less than the noisy data. To deal with this problem, we bootstrap the clean data $\mathcal{D}_c$ to half amount of the noisy data $\mathcal{D}_n$. This upsampling process brings another advantage — the gradients we calculated in each mini-batch are much more stable.

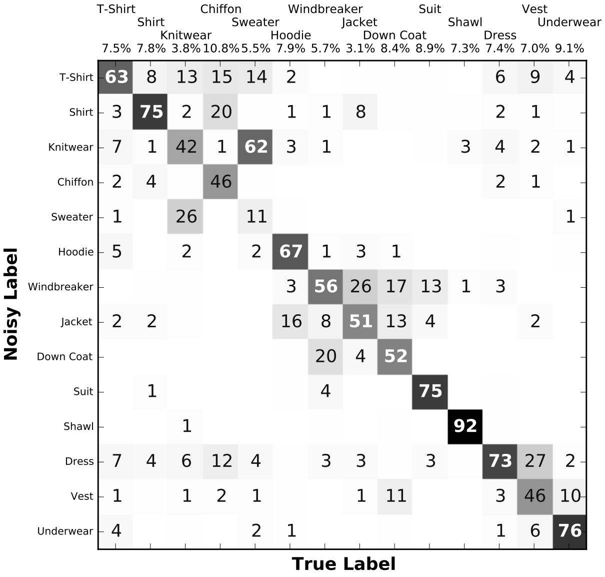
Our proposed method has the ability to figure out the true label given the image and its noisy label. From the perspective of information, our model predicts from two kinds of clues: what are the true labels for other similar images; and how easily for the image to be mislabeled. Label Propagation method [31] explicitly uses the first kind of information, while we implicitly capture it with a discriminative deep model. On the other hand, the second kind of information correlates the image content with the label noise, which can help distinguish between hard samples and mislabeled ones.

5. Experiments

5.1. Dataset

Since there is no publicly available dataset that has both clean and noisy labels, to test our method under real-world scenario, we build a large-scale clothing dataset by crawling images from some online shopping websites. More than a million images are collected together with their surrounding texts provided by the sellers. These surrounding texts usually contain rich information about the products, which can be further converted to visual tags. Specific to our task of clothing classification, we define 14 class labels: T-shirt, Shirt, Knitwear, Chiffon, Sweater, Hoodie, Windbreaker, Jacket, Down Coat, Suit, Shawl, Dress, Vest, and Underwear. We assign an image a noisy label if we find its surrounding text contains only the keywords of that label, otherwise we discard the image to reduce ambiguity.

After that we manually refine the noisy labels of a small portion of all the images and split them into training ($\mathcal{D}_c$), validation and test sets. On the other



	T-Shirt	Shirt	Knitwear	Chiffon	Sweater	Hoodie	Windbreaker	Jacket	Down Coat	Suit	Shawl	Dress	Vest	Underwear
	7.5%	7.8%	3.8%	10.8%	5.5%	7.9%	5.7%	3.1%	8.4%	8.9%	7.3%	7.4%	7.0%	9.1%
T-Shirt	63	8	13	15	14	2						6	9	4
Shirt	3	75	2	20		1	1	8				2	1	
Knitwear	7	1	42	1	62	3	1				3	4	2	1
Chiffon	2	4		46								2	1	
Sweater	1		26		11									1
Hoodie	5		2		2	67	1	3	1					
Windbreaker						3	56	26	17	13	1	3		
Jacket	2	2				16	8	51	13	4			2	
Down Coat							20	4	52					
Suit		1					4			75				
Shawl			1								92			
Dress	7	4	6	12	4		3	3		3		73	27	2
Vest	1		1	2	1			1	11			3	46	10
Underwear	4				2	1						1	6	76

Figure 6. Confusion matrix between clean and noisy labels. We hide extremely small grid numbers for better demonstration. Frequency of each true label is listed at the top of each column. The overall accuracy is 61.54%, which indicates that the noisy labels are not reliable.

hand, the remaining data construct the noisy labeled training dataset ($\mathcal{D}_n$). A crucial step here is to remove from $\mathcal{D}_c$ and $\mathcal{D}_n$ the images that are near duplicate with some image in the validation and test sets, which ensures the reliability of our test protocol. Finally, the size of training datasets are $|\mathcal{D}_c| = 47,570$ and $|\mathcal{D}_n| = 10^6$, while validation and test set have $14,313$ and $10,526$ images, respectively.

The confusion matrix between clean and noisy labels is presented in Figure 6. We can see that the overall accuracy is 61.54%, and some pairs of classes are very confusing with each other (*e.g.* Knitwear and Sweater), which means that the noisy labels are not so reliable.

5.2. Evaluation on the Collected Dataset

We validate our method through a series of experiments conducted on the collected dataset. Our implementation is based on Caffe [12], and the `bvlc_reference_caffenet`[1] is chosen as the baseline model, which approximates AlexNet [13]. Besides, we reimplement two other approaches. One is a semi-supervised learning method called Pseudo-Label [15], which exploits classifier's prediction as ground truth for unlabeled data. The other one is the Bottom-Up method introduced in [25], where the relation between noisy labels and clean labels are built by a confusion matrix Q. In the experiments, we directly use the true Q as shown in Figure 6 instead of estimating its values.

We list all the experiment settings in Table 1. Different methods require different training data. We use only the clean data $\mathcal{D}_c$ to get the baselines under strong supervisions. On the other hand, when all the data are used, we upsample the clean data as discussed in Section 4. Meanwhile, the noisy labels of $\mathcal{D}_n$ are treated as ground truths for AlexNet, and are discarded for Pseudo-Label.

In general, we use a mini-batch size of 256. The learning rate is initialized to be 0.001 and is divided by 10 after every $50,000$ iterations. We keep training each model until convergence. Classification accuracies on the test set are presented in Table 1.

We first study the effect of transfer learning and massive noisy labeled data. From row #1 we can see that training a CNN from scratch with only small amount of clean data can result in bad performance. To deal with this problem, finetuning from an ImageNet pretrained model can significantly improve the accuracy, as shown in row #2. However, by comparing row #2 and #3, we find that training with random initialization on additional massive noisy labeled data is better than finetuning only on the clean data, which demonstrates the power of using large-scale yet easily obtained noisy labeled data. The accuracy can be further improved if we finetune the model either from an ImageNet pretrained one or model #2. The latter one has slightly better performance thus is used to initialize the remaining methods.

Next, from row #6 we can see that semi-supervised learning methods may not be a good choice when massive noisy labeled data are available. Although model #6 achieves marginally better result than its base model, it is significantly inferior to model #5, which indicates that simply discarding all the noisy labels cannot make the full use of these information.

Finally, row #7 and #8 show the effect of modeling the label noise. Model #7 is only 0.9% better than the baseline model #5, while our method gains improvement of 2.9%. This result does credit to our image dependent label noise model, which fits better to the noisy labeled data crawled from the Internet.

5.3. Evaluation on Synthetic Dataset

We also conduct synthetic experiments on CIFAR-10 following the settings of [25]. We first randomly generate a confusion matrix Q between clean labels and noisy labels, and then corrupt the training labels according to it. We use Caffe's CIFAR10-quick as the base model and compare our method with [25] (Bot-

[1] http://caffe.berkeleyvision.org/model_zoo.html

#	Method	Training Data	Initialization	Test Accuracy
1	AlexNet	$\mathcal{D}_c$	random	64.54%
2	AlexNet	$\mathcal{D}_c$	ImageNet pretrained	72.63%
3	AlexNet	upsampled $\mathcal{D}_c$ and $\mathcal{D}_n$ as ground truths	random	74.03%
4	AlexNet	upsampled $\mathcal{D}_c$ and $\mathcal{D}_n$ as ground truths	ImaegNet pretrained	75.13%
5	AlexNet	upsampled $\mathcal{D}_c$ and $\mathcal{D}_n$ as ground truths	model #2	75.30%
6	Pseudo-Label[15]	upsampled $\mathcal{D}_c$ and $\mathcal{D}_n$ as unlabeled	model #2	73.04%
7	Bottom-Up [25]	$\mathcal{D}_c$ and $\mathcal{D}_n$	model #2	76.22%
8	Ours	$\mathcal{D}_c$ and $\mathcal{D}_n$	model #2	**78.24%**

Table 1. Experiment results on the collected dataset

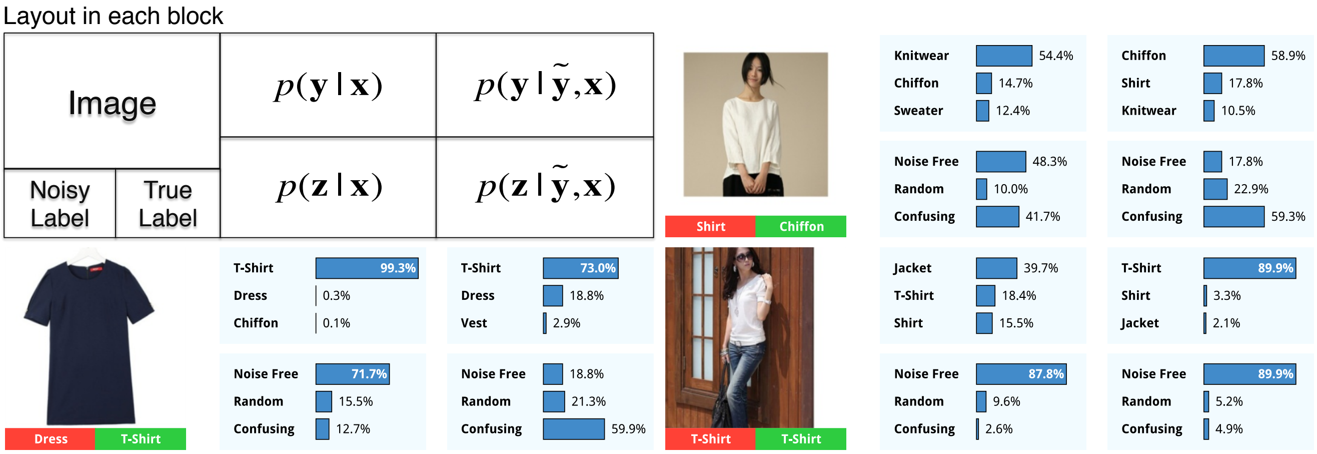


Figure 7. Examples of handling noisy labels. The information layout in each block is illustrated in the top-left one.

Noise Level	Base Model	[25]	Ours
30%	65.57%	69.73%	69.81%
40%	62.38%	66.66%	66.76%
50%	57.36%	63.39%	63.00%

Table 2. Accuracies on CIFAR-10 with synthetic label noise

tom Up with true Q) under different noise levels. The test accuracies are reported in Table 2.

It should be noticed that [25] assumed distribution of noisy labels only depends on classes, while we assume it also depends on image content. This kind of synthetic label noise exactly matches their assumption but is unfavored to our model. Thus the noise type prediction in our model could be less informative. Nevertheless, our model achieves comparable results with [25].

5.4. Effect of Noise Estimation

In order to understand how does our model handle noisy labels, we first show several examples in Figure 7. We can see that given a noisy label, our model exploits its current knowledge to estimate the probability distribution of the true label, and then use it as supervision instead of the noisy one. Another interesting observation is that if $p(\mathbf{y}|\mathbf{x})$ or $p(\mathbf{z}|\mathbf{x})$ goes wrong, our model can still figure out the correct label.

Next we demonstrate the effect of learning to predict the label noise type. We estimate $p(\mathbf{z}_2 = 1|\mathbf{x}) + p(\mathbf{z}_3 = 1|\mathbf{x})$ on the validation set and sort the images accordingly in descending order. Then for each image, we check whether its noisy label mismatches its clean label. The rank-precision curve is plotted in Figure 8. It shows that the intuitive observation — images tend to be mislabeled often share similar patterns — is reasonable, and our model is trained to recognize such patterns to help handle noisy labels.

6. Conclusion

In this paper, we raised the problem of training a classifier with limited clean annotations and massive noisy labeled data. We proposed a novel probabilistic model to describe how a noisy label is generated. Two latent variables — true label and noise type — were introduced to bridge the semantic gap between the observed image and its noisy label. We solved the

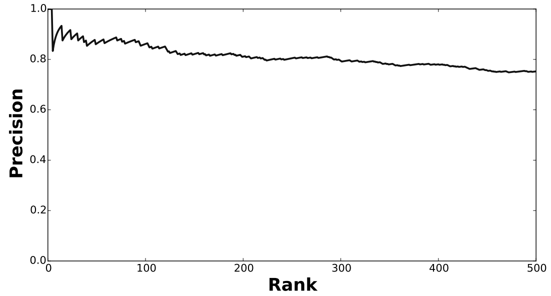


Figure 8. Rank-precision curve of label noise prediction

problem by EM algorithm and integrated it into existing deep learning framework. Experiments on a collected large-scale clothing dataset showed that massive noisy label data could benefit the training of deep models, and utilizing our method could further improve the performance.

Acknowledgements

This work is supported by the National Basic Research Program of China (973 program No. 2014CB340505).

References

[1] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson. From generic to specific deep representations for visual recognition. *arXiv preprint arXiv:1406.5774*, 2014. 2, 3

[2] R. Barandela and E. Gasca. Decontamination of training samples for supervised pattern recognition methods. In *Advances in Pattern Recognition*, pages 621–630. Springer, 2000. 1, 2

[3] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006. 1, 2

[4] E. Beigman and B. B. Klebanov. Learning with annotation noise. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 280–287. Association for Computational Linguistics, 2009. 2

[5] C. E. Brodley and M. A. Friedl. Identifying mislabeled training data. *arXiv preprint arXiv:1106.0219*, 2011. 1, 2

[6] M. Chen, A. Zheng, and K. Weinberger. Fast image tagging. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1274–1282, 2013. 3

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 1

[8] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013. 2, 3

[9] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1915–1929, 2013. 1

[10] B. Frénay and M. Verleysen. Classification in the presence of label noise: a survey. 2013. 2

[11] 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*, 106(2):210–233, 2014. 1

[12] 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 *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014. 7

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1, 3, 7

[14] J. Larsen, L. Nonboe, M. Hintz-Madsen, and L. K. Hansen. Design of robust neural network classifiers. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 2, pages 1205–1208. IEEE, 1998. 3

[15] D.-H. Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, 2013. 7, 8

[16] N. Manwani and P. Sastry. Noise tolerance under risk minimization. *Cybernetics, IEEE Transactions on*, 43(3):1146–1151, 2013. 2

[17] N. Matic, I. Guyon, L. Bottou, J. Denker, and V. Vapnik. Computer aided cleaning of large databases for character recognition. In *Pattern Recognition, 1992. Vol. II. Conference B: Pattern Recognition Methodology and Systems, Proceedings., 11th IAPR International Conference on*, pages 330–333. IEEE, 1992. 3

[18] A. L. Miranda, L. P. F. Garcia, A. C. Carvalho, and A. C. Lorena. Use of classification algorithms in noise detection and elimination. In *Hybrid Artificial Intelligence Systems*, pages 417–424. Springer, 2009. 1, 2

[19] V. Mnih and G. E. Hinton. Learning to label aerial images from noisy data. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 567–574, 2012. 3

[20] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari. Learning with noisy labels. In *Advances in Neural Information Processing Systems*, pages 1196–1204, 2013. 3

[21] D. F. Nettleton, A. Orriols-Puig, and A. Fornells. A study of the effect of different types of noise on the

precision of supervised learning techniques. *Artificial intelligence review*, 33(4):275–306, 2010. 1

[22] M. Oquab, L. Bottou, I. Laptev, J. Sivic, et al. Learning and transferring mid-level image representations using convolutional neural networks. 2013. 2, 3

[23] M. Pechenizkiy, A. Tsymbal, S. Puuronen, and O. Pechenizkiy. Class noise and supervised learning in medical domains: The effect of feature extraction. In *Computer-Based Medical Systems, 2006. CBMS 2006. 19th IEEE International Symposium on*, pages 708–713. IEEE, 2006. 1

[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[25] S. Sukhbaatar and R. Fergus. Learning from noisy labels with deep neural networks. *arXiv preprint arXiv:1406.2080*, 2014. 2, 3, 4, 7, 8

[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014. 1

[27] C.-M. Teng. A comparison of noise handling techniques. In *FLAIRS Conference*, pages 269–273, 2001. 1, 2

[28] J. Weston, F. Ratle, H. Mobahi, and R. Collobert. Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade*, pages 639–655. Springer, 2012. 3

[29] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional neural networks. *arXiv preprint arXiv:1311.2901*, 2013. 1

[30] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev. Panda: Pose aligned networks for deep attribute modeling. *arXiv preprint arXiv:1311.5591*, 2013. 1

[31] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, Technical Report CMU-CALD-02-107, Carnegie Mellon University, 2002. 2, 3, 6

[32] X. Zhu and X. Wu. Class noise vs. attribute noise: A quantitative study. *Artificial Intelligence Review*, 22(3):177–210, 2004. 1, 2