

# Learning From Massive Noisy Labeled Data for Image Classification

Tong Xiao<sup>1</sup>, Tian Xia<sup>2</sup>, Yi Yang<sup>2</sup>, Chang Huang<sup>2</sup> and Xiaogang Wang<sup>1</sup>

<sup>1</sup>The Chinese University of Hong Kong

<sup>2</sup>Baidu Research

## Abstract

*Large-scale supervised datasets are crucial to train deep neural networks 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 deep nets with only limited number of clean labels and millions of easily obtained noisy labels. We model label noises with a graphical probabilistic framework and further integrate it into the deep learning system. To demonstrate the performance of our approach, we collect a large scale clothing classification dataset with both noisy and clean labels. We demonstrate the effectiveness of our approach on this dataset, showing the possibility of learning a strong deep model with only a small amount of clean labels.*

## 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 [11], attribute learning [22], scene classification [7], etc. While state-of-the-art results have been continuously reported [21, 17, 19], all these methods require reliable annotations from millions of images [5] which are often expensive and time-consuming to obtain [5], 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 [9] (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 annota-



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

tors or computing errors from extraction algorithms). Many works have shown that these noisy labels could adversely impact the classification accuracy of the induced classifiers [24, 14, 16]. Various noise-robust algorithms are developed but experiments show that performances of classifiers inferred by robust algorithms are still affected by label noise. Other noise cleaning algorithms are developed [2, 3], but these approaches are difficult in distinguishing informative hard examples from harmful mislabeled ones.

Although annotating all the noisy 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 [11], and then finetune on the smaller dataset of specific task [15, 1, 6]. 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 [23]. 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 a million clothes images from E-commerce websites. Each image is automatically assigned a noisy label according to the keywords in its surrounding text. After that, we manually refine 72,000 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 [18], which modifies a CNN by inserting a linear layer on top of the softmax layer to map clean labels to noisy labels. However, they assume 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 3, 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

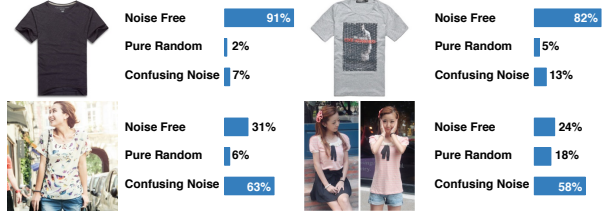


Figure 2. The probability of an image suffering from different noise types depend on the image itself. Although all the images belong to class “T-shirt”, the top two images can be easily recognized, while the bottom two tend to be confused with class “Chiffon”.



Figure 3. Images tend to be mislabeled usually share similar visual patterns.

conversion from the text to label (*e.g.*, the samples with “×” in Figure 1).

The proposed probabilistic model will 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 causes of noisy labels in real-world data and describe all the variables with a novel probabilistic model. Second, we integrate the model into existing deep learning framework to help detect and correct wrong labels. Different training strategies are investigated to make the CNNs learn from better supervisions. Finally, a large-scale clothing dataset is collected 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 [8] for a comprehensive review.

**Direct learning with noisy labels:** Among other consequences, many works have shown that noise can adversely impact the classification performances of induced classifiers [24]. Experiments in the literature show that the performances of classifiers inferred by label noise-robust algorithms are still affected by label noise. These methods seem to be adequate only for simple cases of label noise that can be safely managed by overfitting avoidance [8].

**Semi-supervised learning:** Apart from modeling label noise explicitly, some semi-supervised learning algorithms are developed to utilize weakly labeled or even unlabeled data. Label Propagation method [23] 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 data samples and cannot be applied on large-scale datasets. Chen *et al.* [4] 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.* [20] 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 [11, 15, 1, 6]. 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 [13, 12, 18]. [13] builds a simple noise model for aerial images but only considers binary classification. [12] assumes label noise is independent from the true class label which is a simple and a specific case. [18] 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 con-

fusing levels.

### 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  $\mathcal{D}_n = \{(\mathbf{x}^{(1)}, \tilde{y}^{(1)}), \dots, (\mathbf{x}^{(N)}, \tilde{y}^{(N)})\}$  with  $n$ -th image  $\mathbf{x}^{(n)}$  and its corresponding noisy label  $\tilde{y}^{(n)} \in \{1, \dots, L\}$ , where  $L$  is the number of classes. We describe how the noisy label is generated by using a probabilistic graphical model shown in Figure 4.

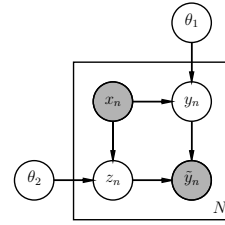


Figure 4. Probabilistic graphical model of label noise

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

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

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  $\text{tr}(\mathbf{C}) = 0$  and  $\mathbf{C}_{ij}$  denoting the confusion probability between class  $i$  and  $j$ . Then we can derive from Figure 4 the joint distribution of  $\tilde{\mathbf{y}}, \mathbf{y}$  and  $\mathbf{z}$  conditioning on  $\mathbf{x}$ :

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

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 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.*

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 [18], 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 [18], 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  $\theta_1$  and  $\theta_2$ . 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  $\text{KL}(q(\mathbf{y}, \mathbf{z}|\tilde{\mathbf{y}}, \mathbf{x})||p(\mathbf{y}, \mathbf{z}|\tilde{\mathbf{y}}, \mathbf{x}; \theta))$ , 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  $Q$ :

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

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 $\mathbf{C}$

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:

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

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:

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

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  $\tilde{\mathbf{y}}_i^{(n)} = 1$  and  $\mathbf{y}_j^{(n)} = 1$ , we set  $\mathbf{t}_n$  to 1. The resulting  $E$  is always greater than the original one, which leads to a contradiction.  $\square$

**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  $\tilde{\mathbf{y}}_k^{(i)} = \tilde{\mathbf{y}}_k^{(j)} = 1$  and  $\mathbf{y}_l^{(i)} = \mathbf{y}_l^{(j)} = 1$  but  $\mathbf{t}_i^* \neq \mathbf{t}_j^*$ . From Lemma 1 we know that elements in  $\mathbf{C}^*$  is either 0 or greater than  $\alpha$ . 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.  $\square$

**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  $\tilde{\mathbf{y}}^{(i)T} \mathbf{C}^* \mathbf{y}^{(i)} = 0$ .  $\square$

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

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

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

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

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

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

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

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

whose solution is

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

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 (SGA) with backpropagation 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. Gathering ground truth object classes is straight forward, since we can just manually label some images by expert. The resulting strongly labeled dataset  $\mathcal{D}_c$  can be directly used for training the network that predicts  $p(\mathbf{y}|\mathbf{x})$ . On the other hand, although off-the-shelf supervision for  $p(\mathbf{z}|\mathbf{x})$  is not available, we can heuristically generate some data by utilizing the images

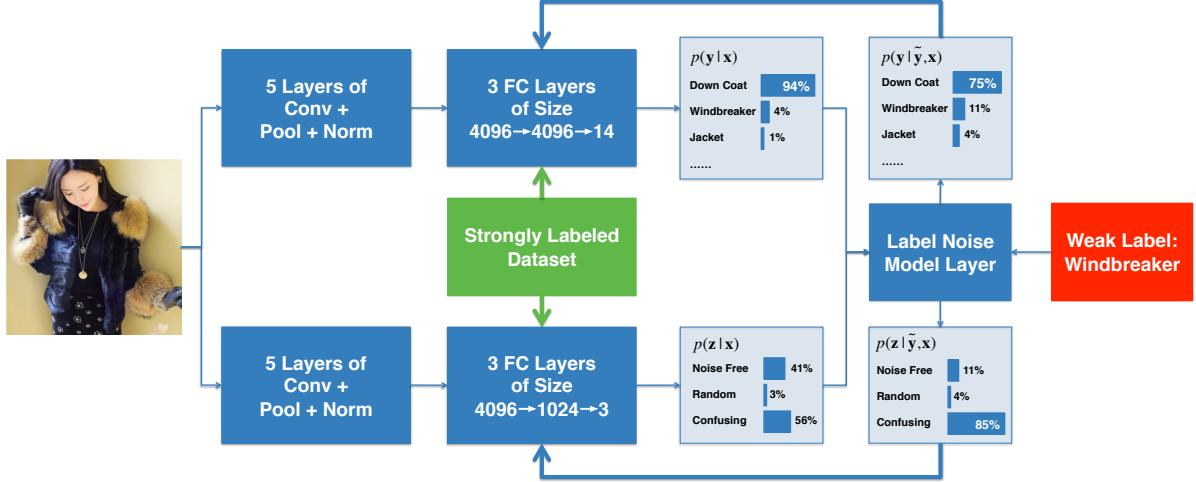


Figure 5. System diagram of our proposed method. Two CNNs are exploited to predict object class  $p(\mathbf{y}|\mathbf{x})$  and noise type  $p(\mathbf{z}|\mathbf{x})$  respectively. A label noise model layer infers the ground truths according to CNNs’ results and the observed noisy label. The ground truth is then used to supervise the CNNs. A separate strongly labeled dataset is also utilized to prevent the model from drifting away.

having both strong and noisy labels. For each sample, we choose as ground truth the  $\mathbf{z}$  that maximizes the likelihood in Eq (1).

After both the CNN components are properly pre-trained, 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 strongly label data together in to our training set, which is depicted in Figure 5 as the extra supervisions for the two CNN components. 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  $\Delta_c$  and  $\Delta_n$ , 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 ground truth 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 confusing is the input image itself. Label Propagation method [23] explicitly uses the first kind of information, while we implicitly capture it with a discriminative deep model. Meanwhile, we exploit the second kind of information to bridge the semantic gap between the image and its possible noisy labels.

## 5. Experiments

### 5.1. Dataset

We build a large-scale clothes dataset by crawling images and their surrounding texts from some online shopping websites. These surrounding texts are valuable, because they usually contain several keywords that can be further converted to visual tags. Specific to our task of clothes classification, we define 14 classes: T-shirt, Shirt, Knitwear, Chiffon, Sweater, Coat, Windbreaker, Jacket, Down Coat, Suit, Shawl, Dress, Vest, and Underwear.

In order to learn a clothes classifier and evaluate its performance, we manually label a small part of all the images and split it into training ( $\mathcal{D}_c$ ), validation and test sets. Meanwhile the remaining data construct the noisy labeled training dataset  $\mathcal{D}_n$ . A crucial data preprocessing step is to remove near duplicate images from  $\mathcal{D}_c$  and  $\mathcal{D}_n$ , which ensures the reliability of our test protocol. The size of training datasets are  $|\mathcal{D}_c| = 47570$  and  $|\mathcal{D}_n| = 1.5M$ , while validation and test set consist of 14,313 and 10,526 images respectively. The confusion matrix between clean and noisy labels are presented in Figure 6. We can see that the overall accuracy is only 61.54%, which means that the labels converted from surrounding texts are quite noisy.

### 5.2. Evaluation

The effectiveness of our model is validated based on a series of experiments. We implement our method by using Caffe [10] and exploit the AlexNet [11] as the baseline model, which consists of five convolutional lay-

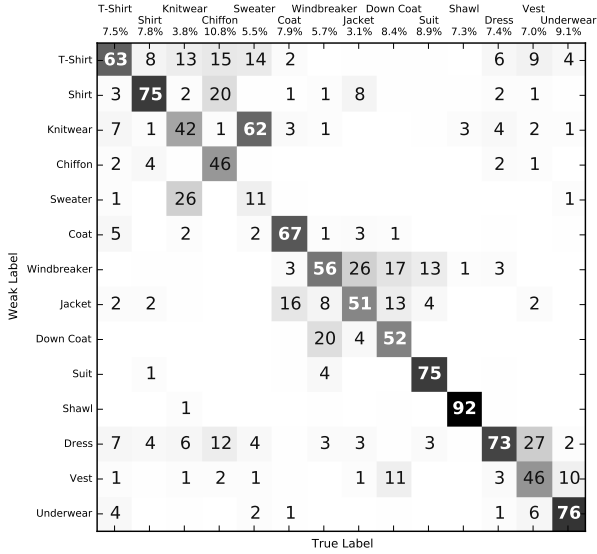


Figure 6. **Confusion matrix between clean and noisy labels.** We hide extremely small grid numbers for better demonstration. Frequency of each true label is listed at the top of each column. The overall accuracy is 61.54%, which indicates that the noisy labels are not reliable.

ers and three fully connected layers. Although recent models may have better learning capability, we choose AlexNet since it is well studied and much easier to be reimplemented (see the `bvlc_reference_caffenet`<sup>1</sup>).

We also implement the bottom up method introduced in [18]. Briefly speaking, they proposed a noise model with the assumption that a noisy label is only related to its true label. The relation is built by a confusion matrix  $Q$  whose values can be easily obtained as Figure 6 in our problem.

We list all the models and training strategies to be compared in Table 1. In general, we set the initial learning rate to be 0.001 and is multiplied by 0.1 every 50000 iterations. For each method, we keep training the model until it converges. Classification accuracies on both the validation and test set are presented in Table 2.

From row 1 we can see that when only a small amount of strongly supervised data is provided to train the deep neural network, the parameters cannot be learned properly and thus results in a bad performance. To cope with this problem, finetuning from a model pretrained on related but much larger dataset can significantly improve the accuracy, which is illustrated in the result of model #2. This is a commonly used

<sup>1</sup>[http://caffe.berkeleyvision.org/model\\_zoo.html](http://caffe.berkeleyvision.org/model_zoo.html)

#	Method	Data	Initialization
1	AlexNet	$\mathcal{D}_c$	random
2	AlexNet	$\mathcal{D}_c$	ilsvrc2012 pretrained model
3	AlexNet	$\mathcal{D}_c \cup \mathcal{D}_n$ but treat noisy labels in $\mathcal{D}_n$ as ground truth	random
4	AlexNet	$\mathcal{D}_c \cup \mathcal{D}_n$ but treat noisy labels in $\mathcal{D}_n$ as ground truth	ilsvrc2012 pretrained model
5	Bottom Up [18]	$\mathcal{D}_c \cup \mathcal{D}_n$	model #2
6	Ours	$\mathcal{D}_c \cup \mathcal{D}_n$	model #2

Table 1. Models and corresponding training strategies used in our experiments

#	Validation Accuracy	Test Accuracy
1	64.28%	64.54%
2	72.21%	72.63%
3	73.76%	74.03%
4	75.57%	75.30%
5	75.97%	76.22%
6	77.65%	78.24%

Table 2. Classification accuracies on validation and test set

technique to train a deep model with limited data for specific task.

However, this transfer learning scheme may still suffer from suboptimal model parameters if the datasets for pretraining and finetuning are loosely related, just like the clothes vs. general objects in our case. We see from row 3 that better performance can be achieved when we train the same model with random initialization on massive data with label noise, but treat noisy labels just as ground truth. Model #4 further improves the accuracy by initializing with ImageNet pretrained model.

Row 5 and 6 show the effect of handling label noise. While model #5 is only 0.4% ~ 0.9% better than the baseline model #4, our proposed method gains improvement of 2.1% ~ 2.9%. Since model #5 assumes that the noisy label only relates to the ground truth, which may not be appropriate in our problem. On the contrary, our model predicts the noise type from the image itself. We will discuss about it in the following section.

### 5.3. Effect of Noise Estimation

In order to understand the way our model handles noisy labels, we demonstrate several examples in Fig-

Layout in each block



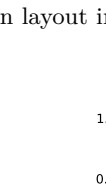
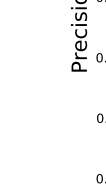
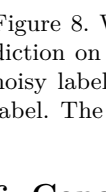
Image		$p(\mathbf{y} \mathbf{x})$	$p(\mathbf{y} \tilde{\mathbf{y}},\mathbf{x})$
		$p(\mathbf{z} \mathbf{x})$	$p(\mathbf{z} \tilde{\mathbf{y}},\mathbf{x})$
Noisy Label	True Label		
	Coat	T-Shirt 89%	T-Shirt 71%
		Chiffon 5%	Coat 24%
		Vest 2%	Chiffon 1%
		Noise Free 53%	Noise Free 24%
	Shirt	Random 12%	Random 5%
		Confusing 34%	Confusing 71%
		Sweater 44%	Shirt 91%
		Shirt 39%	Sweater 3%
	T-Shirt	Knitwear 10%	Chiffon 1%
		Noise Free 65%	Noise Free 91%
		Random 19%	Random 6%
		Confusing 16%	Confusing 3%
	Chiffon	Chiffon 86%	Chiffon 64%
		Shirt 11%	Shirt 13%
		T-Shirt 3%	T-Shirt 9%
		Noise Free 36%	Noise Free 9%
	T-Shirt	Random 4%	Random 4%
		Confusing 60%	Confusing 87%
		Chiffon 86%	Chiffon 64%
		Shirt 11%	Shirt 13%

Figure 7. Examples of model predictions. The information layout in each block is illustrated in the top-left one.

ure 7. We can see that given a noisy label, our model could exploit its current prediction to correct the noise by setting a large weight to 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 as shown in the top-right and bottom-left block respectively, our model can still figure out the correct label.

Next we explain the meaning of  $p(\mathbf{z}|\mathbf{x})$  by taking a look at our model’s prediction on samples drawn from the clean label class “Coat”. As shown in Figure 3, the noisy label is not simply related to the ground truth. On the contrary, images that have high probability of confusing noise tend to share similar visual patterns. This observation indicates that  $p(\mathbf{z}|\mathbf{x})$  is a property of an image itself representing how confusing the image is. Our model is trained to capture these information and exploit them to clarify the noisy labels. Figure 8 further shows the rank-precision curve of our model’s noise prediction. We can see that nearly 80% of the top-500 high confident “confusing” samples actually suffer from noise, which again verifies the feasibility of our proposed noise model.

## 6. Conclusion

In this paper, we raised the problem of training a deep neural network with limited clean annotations and massive noisy labeled data. A novel end-to-end deep learning framework is proposed to handle label noise in training data. We exploit a novel probabilistic model to describe how a noisy label is generated. Two latent variables — ground truth and noise type – are introduced to bridge the semantic gap between

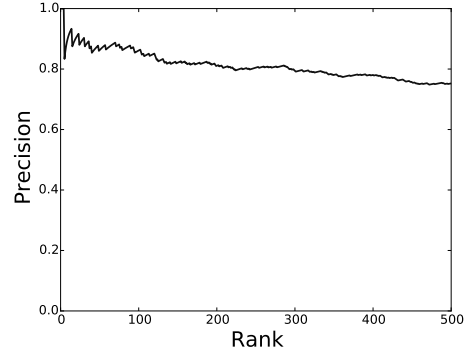


Figure 8. We first sort our model’s “confusing noise” prediction on the validation set, and then check whether the noisy label of the corresponding image mismatch its true label. The rank-precision curve is plotted.

the observed image and its corresponding noisy label. We solve the problem by EM algorithm and integrate it into the deep learning framework. Experiments on a large-scale clothes dataset show that massive noisy label data could benefit the training of deep models, and utilizing our noise handling method can 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](#)
- [3] C. E. Brodley and M. A. Friedl. Identifying mislabeled training data. *arXiv preprint arXiv:1106.0219*, 2011. [1](#)
- [4] 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](#)
- [5] 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](#)
- [6] 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](#)
- [7] 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](#)
- [8] B. Frénay and M. Verleysen. Classification in the presence of label noise: a survey. 2013. [2](#), [3](#)
- [9] 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](#)
- [10] 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. [6](#)
- [11] 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](#), [2](#), [3](#), [6](#)
- [12] 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](#)
- [13] 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](#)
- [14] 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](#)
- [15] M. Oquab, L. Bottou, I. Laptev, J. Sivic, et al. Learning and transferring mid-level image representations using convolutional neural networks. 2013. [2](#), [3](#)
- [16] 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](#)
- [17] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. [1](#)
- [18] S. Sukhbaatar and R. Fergus. Learning from noisy labels with deep neural networks. *arXiv preprint arXiv:1406.2080*, 2014. [2](#), [3](#), [4](#), [7](#)
- [19] 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](#)
- [20] 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](#)
- [21] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional neural networks. *arXiv preprint arXiv:1311.2901*, 2013. [1](#)
- [22] 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](#)
- [23] 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](#)
- [24] X. Zhu and X. Wu. Class noise vs. attribute noise: A quantitative study. *Artificial Intelligence Review*, 22(3):177–210, 2004. [1](#), [3](#)