# Learning From Massive Weakly Labeled Data for Visual Classification

Anonymous CVPR submission

Paper ID 407

## Abstract

*Large-scale strongly supervised datasets are crucial to train deep neural networks for various computer vision problems. Obtaining such well labeled data is usually expensive, but collecting weakly labeled data is much more effortless. In this paper we study how to train a deep neural network to classify objects with limited strong supervisions and massive weakly labeled data. A probabilistic model is proposed to describe the relationship between image data, noisy label and ground truth. We further integrate the model into a deep learning framework. Experiments are conducted on a collected large-scale weakly labeled dataset. The results show that weakly labeled data benefit the training of the deep model, and our approach can further improve the performance.*

## 1. Introduction

Visual classification, which aims at categorizing objects into predefined semantic groups, is a traditional yet fundamental problem in computer vision. Starting from the pioneering work in [3], researchers have developed various deep learning approaches to solve object classification problem. While state-of-the-art results have been continuously reported [11, 6, 8], all the methods rely on large-scale strongly supervised training dataset [2], which consumes a lot of manpower to label millions of images. In many real-world applications, we can only get limited well labeled data. Training a deep neural network is thus difficult even if we finetune it from a pretrained ImageNet model. However, we can easily get substantial weakly supervised data by crawling images from the Internet and extracting keywords from their surrounding text. For example, product photos on *Amazon* and *Taobao* usually have relatively reasonable title and tags, which can be converted to weak object class and attributes. Thus it drives us to find some effective ways to do visual recognition by utilizing these massive information.

Without losing generality, we focus on the task of clothes recognition, *i.e.*, determining whether an image is about a T-shirt, a sweater, or a dress, *etc.* This specific object classification problem draws much attention to many e-commerce companies, since it is a basic building block of products retrieval and recommendation systems. The major challenge of using weakly labeled data in training classifier is to identify bad samples and keep them from drifting the model. Figure 1 shows some examples of label noise. Existing methods try to solve this problem from two aspects. One way is to model the relationship between true label and noisy label, but the model is too simple to fit in real-world applications. The other one exploits semi-supervised learning algorithms (such as label propagation), but their complexities are usually quadratic with the number of samples, thus cannot be applied on large-scale datasets.

To conquer this problem, we first study why a weak label goes wrong and conclude two types of label noise:

- **Pointless noise** occurs when a weak label has no semantic relationship with its image, for example, the samples in the upper red box of Figure 1. This kind of noise is often caused by either the mismatch between web image and surrounding text, or false conversion from text to label. The noisy label in this case is not acceptable.

- **Confusing noise** occurs when a weak label is a reasonable object class for its image other than the ground truth one. This kind of noise happens regularly in some sort of images. For example, the knitwear in the lower orange box of Figure 1 can be often misclassified as a sweater. However, in real applications, confusion among several related labels is generally acceptable to users.

Our goal is to identify the ground truth label given an image as well as its weak label, and then use the ground truth label as supervision instead of the weak one. Intuitively, we have two kinds of evidence to help us make decision. One comes from the ground


Figure 1. Overview of our method. Weak labels often suffer from pointless (red) or confusing noise (orange). Our model infers true label (green) based on observed image and weak label, as well as CNN predictions about object class and noise type.

truth labels of other similar images, and the other one comes from how confusing is the input image itself. In this paper, we model these two kinds of information as latent variables and use them to bridge the semantic gap between our observed image and weak label. A novel probabilistic model is proposed to build relations among all these factors. We solve the problem by Expectation-Maximization (EM) algorithm and integrate it into a deep learning framework. To validate our approach, we collected a clothes image dataset consisting of about 72,000 strongly labeled and one million weakly labeled data. Experiments on this dataset show that massive weakly labeled data benefit the training of deep neural networks, and our approach can further improve the performance.

Our contribution comes from two aspects. First, we investigate the cause of weak label in real-world data, and formulate the label noise in a probabilistic model. Second, we integrate the proposed model into a deep learning framework to help it resist bad supervision in training with weakly labeled data.

## 2. Related Work

Various methods have been proposed to handle label noise in different problem settings. Our work is inspired by [7], where they linearly map the probability distribution of the true label to that of the noisy label. But their assumption — noisy label and input image are conditionally independent given the true label — is too strong. It cannot figure out whether an observed label is correct or not. Our proposed method overcomes this problem by first estimating how likely an image suffers from pointless or confusing noise. And then predicting the true label based on this information and our observed noisy label.

Other methods model noisy labels in different context, *e.g.*, visual attributes annotation [9], or roads recognition in aerial images [4]. It is non-trivial to generalize their approaches to solver our problem, *i.e.*, multi-class object classification.

Apart from modeling label noise explicitly, some semi-supervised learning algorithms are developed to utilize weakly labeled or even unlabeled data. Label Propagation method [12] 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. Weston *et al.* [10] 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 same class, which cannot be obtained in our problem.

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 the CNN on the smaller dataset of specific task [3, 5, 1]. Nevertheless, this kind of transfer learning scheme could be suboptimal when the two tasks are just loosely related. In our case of clothes recognition, we show that training a deep model directly with massive weakly label data achieves better performance compared with finetuning from a ImageNet pretrained model.

## 3. Label Noise Model

We target on the problem of learning a classifier from a set of weakly labeled image data. To be specific, we have a weakly labeled dataset $\mathcal{D}_w = \left\{\left(\mathbf{x}^{(1)}, \tilde{y}^{(1)}\right), \ldots, \left(\mathbf{x}^{(N)}, \tilde{y}^{(N)}\right)\right\}$ 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. Based on previous analysis of the noise types, we describe how the weak label is generated by using a probabilistic graphical

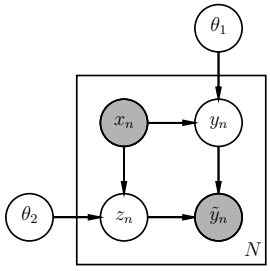


Figure 2. Probabilistic graphical model of label noise

model shown in Figure 2. Despite the observed image $\mathbf{x}$ and the weak 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 also an 1-of-3 binary random variable. We assign three semantic meanings to each possible state of $\mathbf{z}$:

1. The weak label is noise free, *i.e.*, $\tilde{\mathbf{y}}$ should be equal to $\mathbf{y}$
2. The weak label suffers from a pointless noise, *i.e.*, $\tilde{\mathbf{y}}$ can take any possible value other than $\mathbf{y}$
3. The weak label suffers from a confusing noise, *i.e.*, $\tilde{\mathbf{y}}$ can take several values that are easily confused with $\mathbf{y}$

Following this assignment rule, we define the conditional probability of the weak label by

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

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 2 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 object class probability distribution $p(\mathbf{y}|\mathbf{x})$ is comprehensible, the semantic meaning of $p(\mathbf{z}|\mathbf{x})$ needs extra clarification: it can be seen as a property of an image, which represents how confusing

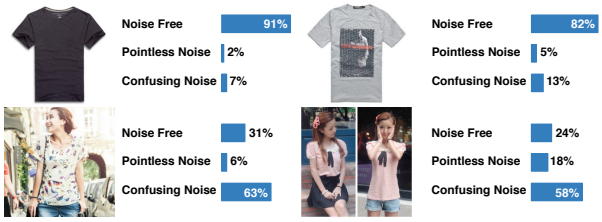


Figure 3. The probability of an image suffering from different noise types depend on the image itself. Although all the images belong to class "T-shirt", the top two images can be easily recognized, while the bottom two tend to be confused with class "Chiffon".

the image is. Specific to our clothes 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.* See Figure 3 for some examples.

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

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

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 [7], 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 [7], while in our approach each data sample has its own.

### 3.1. Learning the Parameters

We exploit two deep neural networks 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 $\log p(\tilde{\mathbf{y}}|\mathbf{x};\theta)$. 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 $\text{KL}\left(q(\mathbf{y},\mathbf{z}|\tilde{\mathbf{y}},\mathbf{x})||p(\mathbf{y},\mathbf{z}|\tilde{\mathbf{y}},\mathbf{x};\theta)\right)$, 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 $t$ of the E-Step, we first compute the posterior of latent variables using current parameters $\theta^{(t)}$:

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

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 $Q$:

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

where $w(\mathbf{y},\mathbf{z}) = p(\mathbf{y},\mathbf{z}|\tilde{\mathbf{y}},\mathbf{x};\theta^{(t)})$ can be treated as the weights of each pair of latent variables.

The semantic meaning of this optimization process is straight forward. We enumerate all the possible true labels, and weight each of them by their posterior probability based on our observation of the image and corresponding noisy label. Then we use these weighted labels to supervise the training of our deep neural networks.

### 3.2. Confusion Estimation

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 $\mathcal{D}_s = \{(\mathbf{x},\mathbf{y},\tilde{\mathbf{y}})\}_N$, where we have $N$ images with both strong label and weak label. As prior information about $\mathbf{z}$ is not available, we just solve the following optimization problem:

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

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:

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

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 $O(N'+K^2)$ 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.** $\mathbf{C}^*_{ij} \neq 0 \Rightarrow \mathbf{C}^*_{ij} > \alpha, \forall i,j \in \{1,\ldots,K\}$

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

*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 $\alpha$. 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. □

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

*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 of an image 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 $K \times K$ 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 $K$ as $m_1, \ldots, m_K$ and might as well set $m_1 \geq m_2 \geq \cdots \geq m_K$. The problem is then converted to

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

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

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

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

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

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:

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

whose solution is

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

The above analysis leads to a simple algorithm. We enumerate $k^*$ from 1 to $K$. 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 Weak Labels

We integrate the proposed label noise model into a deep learning framework. As demonstrated in Figure 4, we predict the probability $p(\mathbf{y}|\mathbf{x})$ and $p(\mathbf{z}|\mathbf{x})$ by using two independent CNNs. Moreover, we append a weak-label-loss layer at the end, which takes as input the CNNs' prediction scores and the observed weak label. Stochastic Gradient Descent (SGD) with backpropagation technique is used to approximately optimize the whole network. In each forward pass, the weak-label-loss layer computes the posterior of latent variables according to Eq (5). While in the backward pass, it computes the weighted sum of gradients with respect to $p(\mathbf{y}|\mathbf{x})$ and $p(\mathbf{z}|\mathbf{x})$, thus the errors can be further propagated back. The forward procedure is summarized in Algorithm 1, and the backward pass for computing gradient with respect to $p(\mathbf{y}|\mathbf{x})$ is illustrated in Algorithm 2.

**Algorithm 1:** Forward pass

**input** : Observed weak label vector $\tilde{\mathbf{y}}$ of size $N$
**input** : Object class scores matrix $Y$ of size $N \times L$
**input** : Noise type scores matrix $Z$ of size $N \times 3$
**output**: Posterior probability matrix $P$ of size $L \times 3$

$P_y \leftarrow$ `Softmax(`$Y$`)`;
$P_z \leftarrow$ `Softmax(`$Z$`)`;
`Q` *is the function that implements Eq* (1);
**for** $i \leftarrow 1$ **to** $N$ **do**
  sum $\leftarrow 0$;
  **for** $j \leftarrow 1$ **to** $L$ **do**
    **for** $k \leftarrow 1$ **to** 3 **do**
      $P[i,j,k] \leftarrow P_y[i,j] \times P_z[i,k] \times$ `Q(`$\tilde{\mathbf{y}}$`,`$j$`,`$k$`)`;
      sum $\leftarrow$ sum $+P[i,j,k]$;
    **end**
  **end**
  $P[i,:,:] \leftarrow P[i,:,:]/$ sum;
**end**

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 $\mathcal{D}_s$ 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 having both strong and weak 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 pretrained, we can start training the whole network with massive weakly labeled data. However, some practical

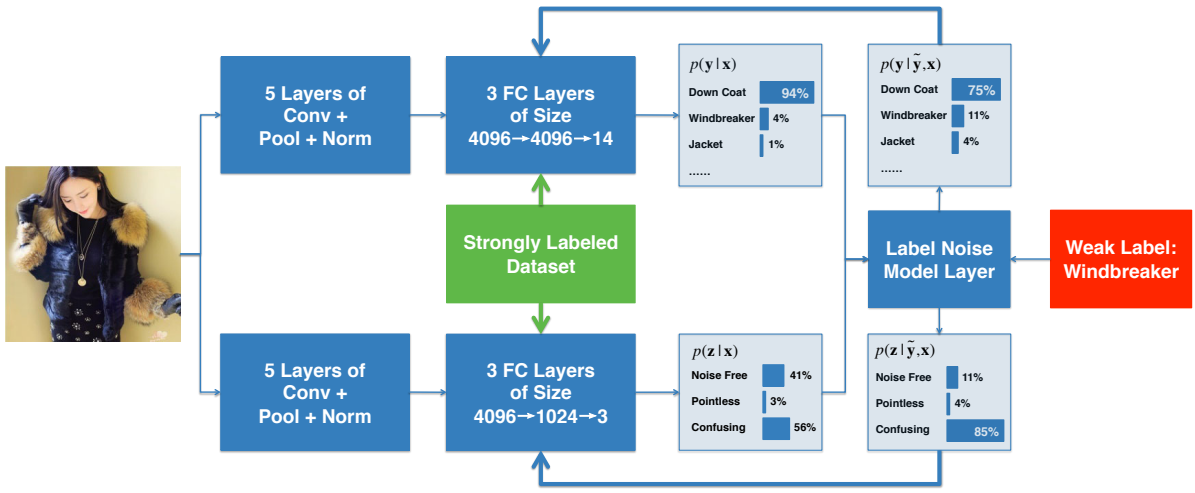


Figure 4. System diagram of our proposed method. Two CNNs are exploited to predict object class $p(\mathbf{y}|\mathbf{x})$ and noise type $p(\mathbf{z}|\mathbf{x})$ respectively. A label noise model layer infers the ground truths according to CNNs' results and the observed weak label. The ground truth is then used to supervise the CNNs. A separate strongly labeled dataset is also utilized to prevent the model from drifting away.

**Algorithm 2:** Backward pass for computing gradient w.r.t. object class scores

**input** : Object class scores matrix $Y$ of size $N \times L$
**input** : Posterior probability matrix $P$ of size $L \times 3$
**output**: Gradient matrix $\Delta_Y$ of size $N \times L$

```
for i ← 1 to N do
    for j ← 1 to L do
        sum ← 0;
        for k ← 1 to 3 do
            sum ← sum +P[i, j, k];
        end
        Δ_Y[i, :] ← Δ_Y[i, :]+ sum ×Y[i, :];
        Δ_Y[i, j] ← Δ_Y[i, j]− sum;
    end
end
```

issues need to be further discussed. First, if we merely use weak data, 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 4 as the extra supervisions for the two CNN components. Then each CNN receives two kinds of gradients, one is from the strongly supervised data and the other is from the weakly supervised data. We denote them by $\Delta_s$ and $\Delta_w$, respectively. A potential problem is that $|\Delta_s| \ll |\Delta_w|$, because strongly labeled data is much less than the weak data. To deal with this problem, we bootstrap the strongly labeled data $\mathcal{D}_s$ to half amount of the weak data $\mathcal{D}_w$. This upsampling process brings another advantage — the gradients we calculated in each mini-batch are much more stable.

Our proposed model has the ability to figure out the ground truth label given the image and its weak 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 [12] 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 text from some online shopping websites. The surrounding text is valuable, because it usually contains 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 ($\mathcal{D}_s$), validation and

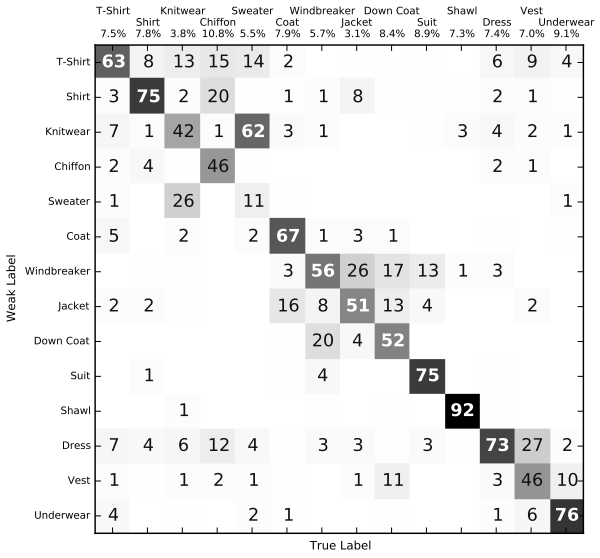


Figure 5. Confusion matrix between strong and weak labels. Grid numbers represent the percentage of confusion between true labels and weak labels. We ignore the elements whose values are less than 1% for better demonstration. Frequency of each true label is listed at the top of each column.

testing sets. Meanwhile remaining data construct the weakly labeled training dataset $\mathcal{D}_w$. Notice that near duplicate images are removed from training datasets $\mathcal{D}_s$ and $\mathcal{D}_w$ to ensure that our testing protocol is reliable. The size of training datasets are $|\mathcal{D}_s| = 47570$ and $|\mathcal{D}_w| = 1.5M$, while for validation and testing set, there are 14,313 and 10,526 images respectively. The confusion matrix between strong and weak labels are presented in Figure 5.

### 5.2. Evaluation

The effectiveness of our model is validated based on a series of experiments. We exploit the AlexNet [3] as our baseline model, which consists of five convolutional layers 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`[1]).

We also implement the bottom up method introduced in [7]. Briefly speaking, they proposed a noise model with the assumption that a weak label is only related to its true label. The relation is modeled by a confusion matrix $Q$ whose true value can be easily obtained as Figure 5 in our problem.

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

| # | Method | Data | Initialization |
|---|---|---|---|
| 1 | AlexNet | $\mathcal{D}_1$ | random |
| 2 | AlexNet | $\mathcal{D}_1$ | `ilsvrc2012` pretrained model |
| 3 | AlexNet | $\mathcal{D}_1 \cup \mathcal{D}_2$ but treat weak labels in $\mathcal{D}_2$ as ground truth | random |
| 4 | AlexNet | $\mathcal{D}_1 \cup \mathcal{D}_2$ but treat weak labels in $\mathcal{D}_2$ as ground truth | `ilsvrc2012` pretrained model |
| 5 | Bottom Up [7] | $\mathcal{D}_1 \cup \mathcal{D}_2$ | model #2 |
| 6 | Ours | $\mathcal{D}_1 \cup \mathcal{D}_2$ | model #2 |

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

| # | Validation Accuracy | Testing 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 testing set

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 testing set are presented in Table 2.

From the table we can see that when only a small amount of 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. However, this transfer learning scheme may still suffer from suboptimal model parameters if the two tasks are loosely related, just like the clothes vs. general object classification in our case. We see that sufficiently better performance can be achieved if we train the same model with random initialization on the massive weakly labeled data, but treat weak labels just as ground truth.

Among all the methods listed above, our approach

leads to the best performance. In order to understand the way our model handles weak labels, we demonstrate several examples in Figure 6. We can see that given a noisy label, our model could exploit its current knowledge to correct the noise by setting a large weight to the true label and use it as supervision instead of the noisy one. Another interesting observation is that if one of the probability $p(\mathbf{y}|\mathbf{x})$ and $p(\mathbf{z}|\mathbf{x})$ goes wrong, the 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 weak label class "Windbreaker". As shown in Figure 7, images that have high probability of confusing noise often share similar visual patterns. This observation indicates that $p(\mathbf{z}|\mathbf{x})$ is a property of an image itself, which represents how the image tends to be confused. 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 our feasibility of our proposed noise model.

## 6. Conclusion

In this paper, we raised the problem of training a deep neural network with limited strongly labeled data and massive weakly labeled data. By investigating the source of weak labels, a novel probabilistic model is proposed to describe how a noisy label is generated. Two latent variables — ground truth and noise type – are introduced to bridge the semantic gap between the observed image and its corresponding weak label. We exploit EM algorithm to learn the parameters and integrate it into a deep learning framework. Experiments on a large-scale clothes dataset show that learning from massive weakly labeled data is significantly better than using merely the limited strongly supervised data. Utilizing our proposed label noise model can further improve the performance.

## 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

[2] 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

[3] 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, 7

[4] 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. 2

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

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

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

[8] 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

[9] A. Vahdat and G. Mori. Handling uncertain tags in visual recognition. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 737–744. IEEE, 2013. 2

[10] 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. 2

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

[12] 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, 6

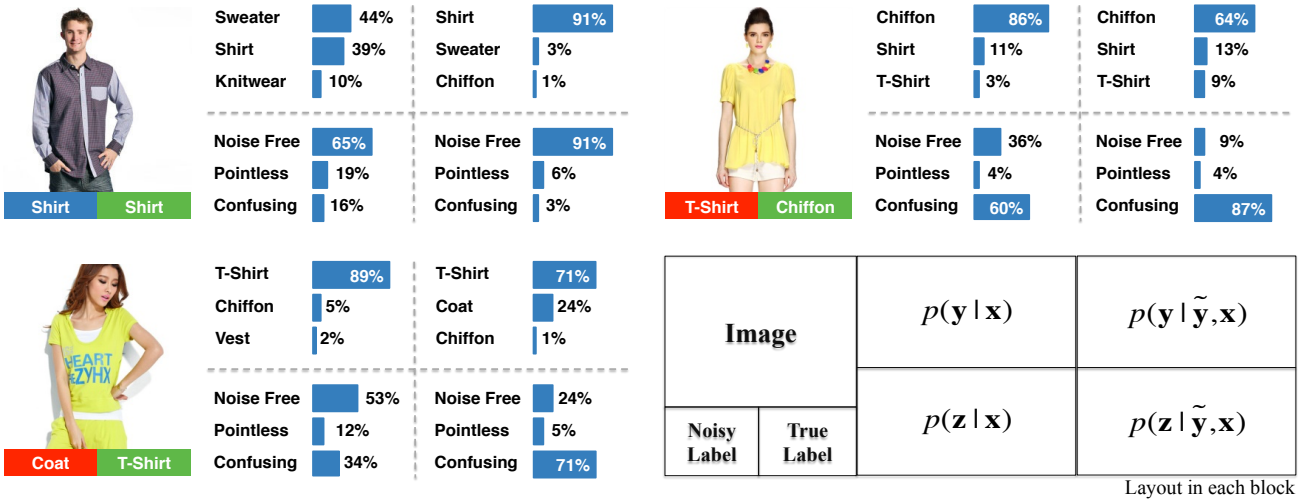


Figure 6. Examples of ground truth labels inferred by our model

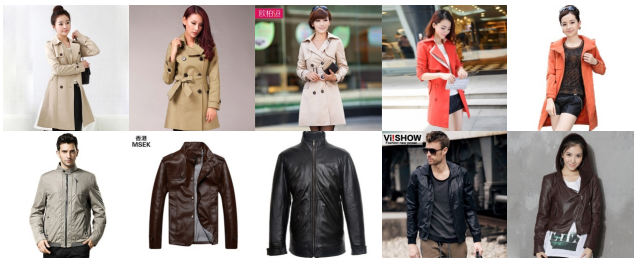

Figure 7. Images tend to suffer from confusing noise usually share similar visual patterns. After training, our model predicts high probability of "noise free" for the five images on the top, while predicting high probability of "confusing noise" for the bottom five. All the images share same the weak label "Windbreaker", but the ground truth of the bottom five images should be "Jacket".

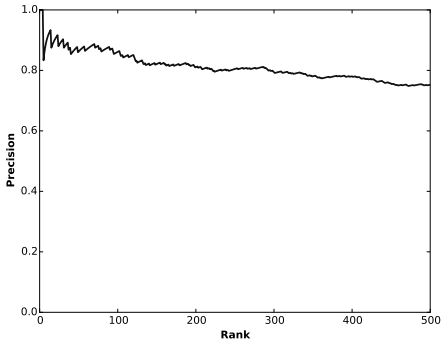


Figure 8. We first sort our model's "confusing noise" prediction on the validation set, and then check whether the weak label of the corresponding image mismatch its true label. The rank-precision curve is plotted.