

# Weakly Semi-Supervised Deep Learning for Multi-Label Image Annotation

Fei Wu, Zhuhaowang, Zhongfei Zhang, Yi Yang, Jiebo Luo, *Fellow, IEEE*,  
Wenwu Zhu, *Fellow, IEEE*, and Yueting Zhuang

**Abstract**—In this paper, we study leveraging both weakly labeled images and unlabeled images for multi-label image annotation. Motivated by the recent advance in deep learning, we propose an approach called weakly semi-supervised deep learning for multi-label image annotation (WeSed). In WeSed, a novel weakly weighted pairwise ranking loss is effectively utilized to handle weakly labeled images, while a triplet similarity loss is employed to harness unlabeled images. WeSed enables us to train deep convolutional neural network (CNN) with images from social networks where images are either only weakly labeled with several labels or unlabeled. We also design an efficient algorithm to sample high-quality image triplets from large image datasets to fine-tune the CNN. WeSed is evaluated on benchmark datasets for multi-label annotation. The experiments demonstrate the effectiveness of our proposed approach and show that the leverage of the weakly labeled images and unlabeled images leads to a significantly better performance.

**Index Terms**—Weakly labeled image, unlabeled image, deep learning, ranking loss

## 1 INTRODUCTION

MULTI-LABEL image annotation aims to learn the association between the visual features and the predefined concepts (labels) [1], [2], [3], [4], [5], [6], [7]. The goal is to develop methods that can annotate a new image with some relevant keywords from a vocabulary set. These results can be used as a tag suggestion that helps people label images, or used in image retrieval tasks, etc. Multi-label image annotation is becoming more and more important since the number of images people upload to social networks, e.g., Facebook and Flickr, is growing exponentially in years, and most of these images have not been deliberately annotated by the users, which makes them hard to manage and index.

Many existing efforts in the past decades focus on designing hand-crafted visual features to improve the accuracy of multi-label annotation. Very recently, deep convolutional neural networks (CNNs) have demonstrated a promising performance in feature learning. CNN is a special type of neural network that utilizes specific network structures, such as convolutional layers, spatial pooling layers, local response normalization layers and fully

connected layers. In general, a CNN designed for image understanding tasks mainly consists of two indispensable components: a multiple-layer architecture composed of several layers that gradually learns image representations from raw pixels, and a loss layer that propagates supervision cues back and fine-tunes the deep network to learn better representations for the specific tasks.

It is important to note that most of the CNNs are designed to work with single-label classification problems, where each image is labeled with only one label that describes the most significant connotation of this image, usually describing an object or a scene. However, labeling each image with only one label is not appropriate for practical applications, since a majority of the images in the real-world applications have more than one object or concept. In order to naturally describe images, it is important that we handle multi-label images with CNNs.

As shown by Szegedy et al. in [8], the larger and deeper the network is, the better the performance can be. However, as the number of parameters of CNN increases significantly with the growth of the network, it requires more supervised (labeled) training samples to prevent overfitting. Although supervised learning is effective in learning useful features, it is not always feasible to obtain sufficient labeled data. As with most machine learning problems, the quantity and quality of training data are critical to achieve a better performance. However, labeling a large set of training images accurately with a specific vocabulary set may be problematic and requires a great deal of human labor work. This process is expensive and involves a lot of possibly ambiguous decisions, and it becomes even more nontrivial for multi-label data.

Thanks to the development of social network and mobile industry, nowadays people upload tremendous amount of images everyday, and many of them are easy to obtain and free to use. For some images, users also tag

- F. Wu, Z. Wang, Z. Zhang, and Y. Zhuang are with the College of Computer Science, Zhejiang University, Hangzhou, China. E-mail: {wufei, zhuhaow, zhongfei, yzhuang}@zju.edu.cn.
- Y. Yang is with the University of Technology Sydney, Sydney, Australia. E-mail: Yi.Yang@uts.edu.au.
- J. Luo is with the Department of Computer Science, 611 Computer Studies Building, University of Rochester, Rochester, NY 14627. E-mail: jluo@cs.rochester.edu.
- W. Zhu is with the Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, Beijing, China. E-mail: wwzhu@tsinghua.edu.cn.

Manuscript received 29 June 2015; revised 27 Sept. 2015; accepted 25 Oct. 2015. Date of publication 3 Nov. 2015; date of current version 18 Dec. 2015. Recommended for acceptance by J. Wang, G.-J. Qi, N. Sebe, and C. Aggarwal. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TBDDATA.2015.2497270

them with several tags describing the objects or concepts in these images, which provides multi-label information. However, comparing to the well labeled datasets which are widely used in previous methods, the label information in these social images is highly incomplete. For example, each image may have insufficient tags describing only a small portion of this image in a few perspectives, and most of these images are even unlabeled. We call an image with a few tags describing only part of the image content as a *weakly labeled* image, and an image with no tags at all as an *unlabeled* image. These weakly labeled and unlabeled images are hard to use as training data.

In this paper, we attempt to harness such images, i.e., the weakly labelled images and the unlabeled images, in a deep learning manner. To that end, we propose an approach named *weakly semi-supervised deep learning* (WeSed) for multi-label annotation. In WeSed, we devise a pairwise-ranking loss and a triplet-ranking loss to fine-tune the convolutional neural network with weakly labeled and unlabeled images. The pairwise-ranking loss is employed to handle the weakly labeled images and the triplet-ranking loss is conducted to address the problem that images are possibly most unlabeled. Our main contributions are as follows:

- We propose an approach called weakly semi-supervised deep learning for multi-label image annotation (WeSed).
- We specifically devise a ranking loss to handle multi-label images which are weakly labeled. This loss function allows us to leverage the huge amount of weakly labeled image data readily obtained from various social networks.
- We propose a triplet CNN structure in companion with a highly efficient image triplet sampling algorithm; this algorithm samples high quality image triplets from both weakly labeled and unlabeled images for the network. This enables us to train the CNN to learn better features by exploiting the information from both weakly labeled and unlabeled images.

## 2 RELATED WORK

Recent years have witnessed the fast growth of multimedia sharing social websites such as Facebook and Flickr. These websites allow users to upload images and describe the image content with tags. However, as discussed before, the images in these websites are highly weakly labeled or even unlabeled. Qi et al. [4] proposed a model based on the distance between images in the multi-label space to deal with the noise in tag information. Later, Li et al. [5] investigated this challenging problem by exploiting labeled and unlabeled data through a semi-parametric regularization and taking advantage of the multi-label constraints into the optimization. Similar to many other efforts [9], [10], [11], [12], [13], [14], [15], [16], [17] that tried to solve the image annotation problem, these approaches all employed hand-crafted features.

Although the hand-crafted features have made a great progress in multi-label annotation, the extracted features are not always optimal. More recently, in contrast to hand-

crafted features, the learnt features with CNN have been adopted to address multi-label problems.

CNNs have outperformed existing hand-crafted features in many applications. Krizhevsky et al. [18] reported record-breaking results in image classification task on ILSVRC 2012 which consists of images from 1,000 categories. Many efforts have focused on the designation or regularization methods of the structures of CNN, such as Network in Network [19], GoogLeNet [8], Dropout [20], DropConnect [21] and Maxout [22], and achieved impressive performance on specific tasks.

However, none of the aforementioned methods can be easily extended to deal with multi-label problem with CNN. Gong et al. [3] combined a convolutional architecture with a weighted approximate ranking loss for multi-label annotation, where the label information is assumed to be *complete*, the results showed an impressive improvement over the traditional hand-crafted feature based models.

In this paper, we tend to solve the multi-label image annotation problem when the training images are from social network, which means that we have a large set of images, and their label information is incomplete while some of them are even unlabeled, we propose a model that is designed to train a CNN based on multi-labeled images that may be weakly or even unlabeled.

There are also some systems that try to learn a model from both labeled and unlabeled data as a semi-supervised problem. Fergus et al. [11] tried to model the unlabeled data by embedding the visual similarity between images into a graph and learning a model which agrees with the labeled data but is also smooth with respect to the similarity graph. A semi-supervised learning algorithm is proposed by Wang et al. [23] which is based on kernel density estimation approach for automatic video annotation. Wang et al. [24] integrated multiple graphs into a regularization framework in order to sufficiently explore the complementation of various crucial factors in video annotation with semi-supervised learning. Amiri and Jamzad [25] first constructed a generative model for each semantic class using labeled images in that class and then incorporated the unlabeled images by using a modified EM algorithm to update parameters of the constructed generative models. Li et al. [5] optimized a model which exploits both labeled and unlabeled data through a semi-parametric regularization incorporating the information of multi-label space. Wang et al. [26] proposed a hypergraph construction approach that leverages sparse representation and the semi-supervised prior knowledge for visual classification tasks. However, they are all based on hand-crafted features.

In order to solve the semi-supervised problem with CNN, we take advantage of the semantic similarity between images as a feature learning cue to fine-tune the network. As suggested by Deselaers and Ferrari in [27], this gives a better understanding of images than simple visual similarity. There are several methods [28], [29], [30] that focus on the image embedding according to their similarities. For instance, a relative similarity via triplet-ranking loss is proposed in [30] to fine-tune the CNN in order to learn a discriminative visual representation based on category-level ranking.

In this work, one of our goals is to train the CNN to extract features that represent the semantic similarities among images. Existing models for image similarity mainly focus on category-level image similarities [28] which consider images to be similar as long as they belong to the same category, or fine-grained category-level image similarities [30] which also consider the relative similarities between images in the same category. They are all based on explicitly defined category information. In our case, there is no explicit information whether given image pairs or triplets are similar or not.

### 3 ALGORITHM

Assume we have a set of training images  $\mathcal{I} = \{x_i\}$ . For the  $i$ th image  $x_i$ , we have a corresponding labeling vector  $y_i \in \{0, 1\}^m$ , where  $y_i^j = 1$  indicates that the  $j$ th label of image  $x_i$  is “present” (positive) whereas  $y_i^j = 0$  denotes that the label is “absent” (true negative) or “missing” (false negative). That is to say, in this paper,  $x_i$  is not assumed to be fully labeled (therefore weakly labeled), where there may be labels that should be present but instead unfortunately missing. In the setting of weakly labeling,  $y_i^j = 0$  denotes either image  $x_i$  does not have the  $j$ th concept at all or has the  $j$ th concept but the concept is not labeled. There may also be images in  $\mathcal{I}$  that do not have any label information, i.e.,  $\sum y_i^j = 0$ . In this case, we call  $x_i$  unlabeled. We denote the training set  $\mathcal{I}$  as two disjoint sets, *weakly labeled* images  $\mathcal{I}_w$  and *unlabeled* images  $\mathcal{I}_u$ , i.e.,  $\mathcal{I} = \mathcal{I}_w \cup \mathcal{I}_u$ .

Some of the images in  $\mathcal{I}_w$  could be *fully* labeled (i.e., there is no missing labels) rather than *weakly* labeled. However, since we hardly discern whether an image is weakly labeled or fully labeled, all of the labeled images in this paper are called weakly labeled images.

In fact, *images labeled by users* are inherently *noisy* (with both missing labels and incorrect labels<sup>1</sup>); our proposed approach can still deal with noisy labels as we will address this issue in experiments.

After given the weakly labeled images and unlabeled images in the training set, we tend to learn the prediction function  $g(\cdot)$  that outputs the label score vector  $a(x)$  of image  $x$  according to the *learnt* features  $f(x)$  via convolutional neural network  $CNN(\cdot)$ . The learnt features of  $CNN(\cdot)$  and the score vector of  $g(\cdot)$  are denoted as follows:

$$\text{CNN learnt feature : } f(x) = CNN(x), \quad (1)$$

$$\text{Annotation score : } a(x) = g(f(x)), \quad (2)$$

where  $f(x) \in \mathbb{R}^p$  and  $a(x) \in \mathbb{R}^m$  are two vectors,  $p$  is the dimension of the learnt features, and  $m$  is the size of label sets.

#### 3.1 Weakly Supervised Learning

Assume that we are given one labeled image  $x \in \mathcal{I}_w$  and its labeling vector  $y$ . We tend to devise a ranking loss which assigns a higher score to positive labels than to negative

ones, while considering the missing (false negative) labels of  $x$ .

We denote the sets of indices of positive labels and negative labels of  $x$  as:

$$\begin{aligned} C_x^+ &= \{j | y^j = 1\}, \\ C_x^- &= \{j | y^j = 0\}. \end{aligned}$$

Specifically, we devise a *weakly weighted pairwise ranking* ( $W^2PR$ ) loss to optimize the top- $k$  accuracy of multi-label image annotation for  $x \in \mathcal{I}_w$  as follows:

$$\min \sum_{x \in \mathcal{I}_w} \sum_{s \in C_x^+} \sum_{t \in C_x^-} L_w(r_s) \max(0, m_s - a^s(x) + a^t(x)), \quad (3)$$

where  $r_s$  is the rank for the positive label  $s$  of image  $x$ ,  $L_w(\cdot)$  is a weighting function for different ranks of positive labels,  $a^s$  and  $a^t$  are the output scores for the positive label  $s$  and the negative label  $t$ , respectively, and  $m_s$  is the margin.

We argue that the devised  $W^2PR$  loss is particularly attractive for the weakly labeled images. As aforementioned, there are two kinds of labeled images in  $\mathcal{I}_w$ , namely *weakly* labeled images and *fully* labeled images. Now we explain why the proposed  $W^2PR$  loss can handle these two cases in one framework. We assume that if one training sample in  $\mathcal{I}_w$  is labeled with more than  $l$  labels ( $\|y\| > l$ ), the image is fully labeled; when one training sample in  $\mathcal{I}_w$  is labeled with fewer tags ( $\|y\| \leq l$ ), the image is likely to be weakly labeled. As the value of  $l$  is a crucial part of  $W^2PR$ , we will demonstrate the multi-label annotation performance with the variation of  $l$  in the Experiments section.

For fully labeled images, all the positive labels should be ranked higher than negative ones. For images likely to be weakly labeled, we only require that the positive labels be at the top  $l$ , which allows several missing labels to be in the top since the number of labels of weakly labeled images is less than  $l$ . These requirements are reasonable for the learning scenario in the study. In order to achieve this goal, the ranking weight  $L_w(\cdot)$  in  $W^2PR$  is defined as follows:

$$L_w(r_s) = \begin{cases} 0, & r_s \leq l, \\ \sum_{n=1}^r \frac{1}{n}, & r_s > l. \end{cases} \quad (4)$$

Equation (4) means that if the positive label  $s$  is ranked at top  $l$  (i.e.,  $r_s \leq l$ ), the ranking weight is zero; otherwise, the lower the positive label ranked, the larger weight we assign to it, which pushes the positive label to the top harder. The ranking weight  $L_w(r_s)$  can deliberately deal with the weakly labeled images where some labels may be missing. Fig. 1 illustrates the ranking process via  $W^2PR$  for one weakly labeled image and one fully labeled image, respectively.

Given one image which is likely to be weakly labeled (the number of positive labels is less than  $l$ ), as long as the positive labels are all ranked higher than  $l$ , the loss of this image does not increase no matter what the exact ranking is for each positive label. This allows several negative labels to move before or around the positive ones without

1. Incorrect here means false positive, i.e.,  $y_i^j = 1$  while this label is actually not present in this image.



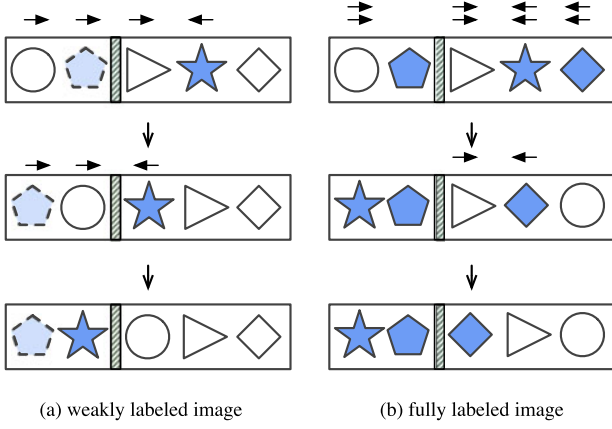


Fig. 1. The intuitive illustration of the proposed  $W^2PR$  loss to handle the weakly labeled images and fully labeled images. Each row shows the current label ranking of one image. The value of  $l$  is set to 2. The number of arrows shows the number of gradients propagated back for each label. The shapes in blue are the positive labels and white ones are negative ones. The proposed  $W^2PR$  loss pushes the positive labels of one weakly labeled image to top  $l$  but still has a chance to have the false negative (missing) label (pentagon with dash border in (a)) at the top  $l$ , while  $W^2PR$  pushes all of the positive labels of one fully labeled image at the topmost. The margin constraint is ignored in this figure for simplicity.

the margin constraint, as shown in Fig. 1a, where there is one missing label (pentagon) ranked at top  $l$ . This is very appealing to allow the network to learn to rank a *false negative* (missing) label with other image samples where this missing label is positive since this sample does not push this label down.

If there are more than  $l$  positive labels in the image (fully labeled image), the ranking loss of this image is minimized if and only if the positive labels are all ranked higher than the negative ones with margin  $m_s$ . Since there will always be some positive label(s) ranked lower than  $l$ , any positive label not ranked higher than the negative ones with margin 1 will increase the loss, as shown in Fig. 1b.

A uniform formulation allows us to deal with these two cases while there is no information indicating one image is weakly labelled or fully labelled in training set.

Next we need to compute rank  $r_s$  of positive label  $s$ . Since the label space might be large, directly sorting the label scores for each image to obtain the ranking might not be efficient; so we need to estimate ranking  $r_s$  efficiently. Following [31], for each positive label  $s$ , we sample negative labels until we find a violation  $v$ , i.e.,

$$1 - \mathbf{a}^s(\mathbf{x}) + \mathbf{a}^v(\mathbf{x}) > 0. \quad (5)$$

Then we record the number of trials  $q$  we have sampled. The rank is estimated by:

$$r_s = \left\lceil \frac{m-1}{q} \right\rceil, \quad (6)$$

where  $m$  is the number of total labels. As discussed in [3], this is a theoretical upper bound for a true value of  $r_s$ , which means that we give a slightly higher ranking to each positive label  $s$ .

$W^2PR$  is different from weighted approximate-ranking pairwise (WARP) [3], [31], since WARP is designed for

fully labeled images (no label is missing). If there is a negative label ranked higher than a positive one or the score margin is not satisfied, the loss increases. On the contrary,  $W^2PR$  deals with weakly labeled images (the label information is in fact incomplete). During the training phase, the CNN may have learned to correctly predict one positive (present) label at some point, but when training by some other images where the predicted top ranked label is actually true but missing, this label will be pushed down by WARP. This will confuse the network and thus deteriorates the performance.

### 3.2 Semi-Supervised Learning

In this section, we show how to exploit unlabeled images in  $\mathcal{I}_u$  for feature learning to boost the performance of multi-label annotation.

Traditionally, we calculate the semantic similarity of two images  $\mathbf{x}_i$  and  $\mathbf{x}_j$  according to their labeled tags  $\mathbf{y}_i$  and  $\mathbf{y}_j$  with  $\text{sim}(\cdot)$  defined as follows:

$$\text{sim}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{s=1}^m (\mathbf{y}_i^s \times \mathbf{y}_j^s). \quad (7)$$

Therefore, we may directly constrain the learnt features  $\mathbf{f}(\mathbf{x}_i)$  and  $\mathbf{f}(\mathbf{x}_j)$  to be similar *w.r.t.* their semantic similarity as follows:

$$\min w(\text{sim}(\mathbf{x}_i, \mathbf{x}_j)) \|\mathbf{f}(\mathbf{x}_i) - \mathbf{f}(\mathbf{x}_j)\|_2^2. \quad (8)$$

As we see, if  $\mathbf{x}_i \in \mathcal{I}_w$  and  $\mathbf{x}_j \in \mathcal{I}_u$ , then  $\text{sim}(\mathbf{x}_i, \mathbf{x}_j) = 0$ . However, this may not be true since the unlabeled image  $\mathbf{x}_j$  may have missing tags which are present or missing in image  $\mathbf{x}_i$ .

Suppose that  $\mathbf{x}_i$  is weakly labeled with  $d$  labels (i.e.,  $\|\mathbf{y}_i\| = d$ ) and that  $\mathbf{x}_k \in \mathcal{I}_u$  is an unlabeled image. We have  $\text{sim}(\mathbf{x}_i, \mathbf{x}_k) = 0$ . Assume that each negative label has the probability  $p$  to be false negative (missing). We denote the ground-truth (oracle) similarity as  $\widetilde{\text{sim}}(\cdot)$ . For simplicity, assume that there are no missing labels in  $\mathbf{x}_i$ . Then the probability of the oracle similarity  $\widetilde{\text{sim}} > 0$  is:

$$P(\widetilde{\text{sim}}(\mathbf{x}_i, \mathbf{x}_k) > 0) = \sum_{s=1}^d C_d^s p^s (1-p)^{(d-s)}, \quad (9)$$

where

$$C_d^s = \frac{d!}{(d-s)!s!}.$$

If  $d = 3$  and  $p = 0.1$ , the probability of  $\widetilde{\text{sim}} > 0$  is 0.271, which means that the computed  $\text{sim}$  may be incorrect with the probability 27.1 percent.

As a result, we propose to utilize relative similarities between image triplets instead of directly relying on the pairwise similarity. After given one image triplet  $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)$ , where  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are from  $\mathcal{I}_w$  with overlapping positive labels (i.e.,  $\text{sim}(\mathbf{x}_i, \mathbf{x}_j) > 0$ ), and  $\mathbf{x}_k$  is from  $\mathcal{I}_w$  or  $\mathcal{I}_u$  which is less similar to  $\mathbf{x}_i$  than  $\mathbf{x}_j$ . The relative semantic similarity  $\text{rsim}(\cdot)$  in terms of  $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)$  is defined as follows:

$$\text{rsim}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) = \text{sim}(\mathbf{x}_i, \mathbf{x}_j) - \text{sim}(\mathbf{x}_i, \mathbf{x}_k). \quad (10)$$

Similarly, the *oracle* counterpart of  $rsim(\cdot)$  is denoted as  $\widetilde{rsim}$ .

Here, we expect the learnt features of images in a triplet to meet their relative semantic similarity defined by  $rsim$ . Therefore, we optimize the following objective:

$$\min \sum_{rsim(x_i, x_j, x_k) > 0} \max(0, \|f(x_i) - f(x_j)\|_2^2 - \|f(x_i) - f(x_k)\|_2^2 + m_f), \quad (11)$$

where  $m_f$  is the margin,  $x_i, x_j \in \mathcal{I}_w$  and  $x_k \in \mathcal{I}$ . We call the objective as *triplet similarity* (TS) loss.

The triplet similarity loss expects that the learnt features of  $x_i$  and  $x_j$  are more similar than those of  $x_i$  and  $x_k$ . Now we show why (11) is more accurate than Equation (8).

Suppose that  $\|y_i\| = d$  and  $sim(x_i, x_j) = e$ .  $x_k$  is sampled from  $\mathcal{I}_u$ , and for simplicity, there are no missing labels in  $x_i$  and  $x_j$ . Then we have:

$$P(rsim(\widetilde{x}_i, x_j, x_k) \leq 0) = \sum_{s=e}^d C_d^s p^s (1-p)^{(d-s)}. \quad (12)$$

If  $d = 3$ ,  $e = 2$  and  $p = 0.1$ , we have only probability 2.8 percent that the triplet we use does not satisfy the requirement  $\widetilde{rsim} > 0$ . It is worth noting that both  $x_i$  and  $x_j$  may have some labels missing. However, the probability that a label is missing in each image is the same. The probability of  $P(rsim \leq 0)$  is not much different from that in Equation (12) (less than 1 percent when  $p = 0.1$ . We omit the proof due to the space limitation). Accordingly, we have a very high confidence that for a randomly sampled image  $x_k$  from  $\mathcal{I}_u$ , we have  $rsim(x_i, x_j, x_k) > 0$ .

Further, we give out a highly efficient sampling algorithm to boost the semantic accuracy of the training image triplets in the next section.

### 3.2.1 Generating Triplets

The number of the potential triplets satisfying

$$rsim(x_i, x_j, x_k) > 0$$

increases cubically, and some of the unsatisfying triplets may be sampled (with a little probability). Thus, training with all the potential triplets is computationally prohibitive and sub-optimal. We need an efficient sampling method to generate satisfying triplets to train the network.

To sample a triplet, we first sample an image as  $x_{base} \in \mathcal{I}_w$ ; we then sample an image  $x'_{base} \in \mathcal{I}_w$  which has overlapping positive tags with  $x_{base}$ . Finally, we sample an image  $\bar{x} \in \mathcal{I}$  which is less similar to  $x_{base}$  than  $x'_{base}$ .

Since the image label information is incomplete, what we can be more sure about the semantic similarity is when the images we have sampled have more labels (indicating that they are less likely to have missing labels) or the pairs of the similar images share more labels.

First, image  $x_i$  is sampled as  $x_{base}$  from  $\mathcal{I}_w$  with a probability:

$$p(x_i) = \frac{\sum_{t=1}^m y_i^t}{Z}, \quad (13)$$

where  $Z$  is the normalization constant. The more tags  $x_i$  has, the larger probability it is sampled as  $x_{base}$ .

Given  $x_{base}$ , image  $x_j$  is sampled as  $x'_{base}$  with the following probability depending on their semantic similarity  $sim(\cdot)$ :

$$p(x_j)_{x_j \neq x_{base}} = \frac{sim(x_{base}, x_j)}{Z}, \quad (14)$$

which means that if one image has more overlapping positive tags with  $x_{base}$ , then this image is more likely sampled as  $x'_{base}$ .

Finally,  $\bar{x}$  can be sampled from either  $\mathcal{I}_w$  or  $\mathcal{I}_u$  as we discussed before. In the experiments, we sample  $\bar{x}$  from  $\mathcal{I}_w$  or  $\mathcal{I}_u$  with a ratio  $\gamma$ .

When sampling  $\bar{x}$  from  $\mathcal{I}_w$ , we resort to  $rsim(\cdot)$ , since a larger value of  $rsim(\cdot)$  means that  $\widetilde{rsim}(\cdot)$  is larger than zero with a higher probability. As a result, image  $x_k \in \mathcal{I}_w$  is sampled as  $\bar{x}$  with a probability as follows:

$$p(x_k)_{rsim(x_{base}, x'_{base}, x_k) > 0} = \frac{rsim(x_{base}, x'_{base}, x_k)}{Z}. \quad (15)$$

When sampling  $\bar{x}$  from  $\mathcal{I}_u$ , we sample each unlabeled image as  $\bar{x}$  with an equal probability.

The aforementioned triplet sampling method is very efficient for even very large datasets and can be easily applied to online scenario.

For each label  $t_i$ , we maintain a list  $list_i$  that holds all the indices of images labeled with  $t_i$ , and we also maintain a list  $list_u$  that holds all the indices of unlabeled images. To sample  $x_{base}$ , we first uniformly sample a label  $t_{base}$  from all the  $m$  labels. Then we uniformly sample an image  $x_{base}$  from  $list_{base}$ . For the sampling of  $x'_{base}$ , we uniformly sample a tag  $t_j$  where  $y_{base}^j = 1$  for  $j = 1, \dots, m$ . Then we uniformly sample an image as  $x'_{base}$  from  $list_j$ . Now we determine whether  $\bar{x}$  is sampled from labeled or unlabeled images with the ratio  $\gamma$ . To sample  $\bar{x}$  from unlabeled images, we uniformly sample one from  $list_u$ . To sample  $\bar{x}$  from labeled images, we uniformly sample image  $x_k$  from all labeled images (excluding  $x_{base}$  and  $x'_{base}$ ), and take  $x_k$  as  $\bar{x}$  with the following probability:

$$p = \max\left(0, \frac{rsim(x_{base}, x'_{base}, x_k)}{sim(x_{base}, x_k)}\right). \quad (16)$$

It is worth noting that sampling by (16) does not match the exact distribution given in (15), as we approximate  $Z$  by

$$\sum_{rsim(x_{base}, x'_{base}, x_k) > 0} sim(x_{base}, x'_{base})$$

instead of the exact value

$$\sum_{rsim(x_{base}, x'_{base}, x_k) > 0} rsim(x_{base}, x'_{base}, x_k).$$

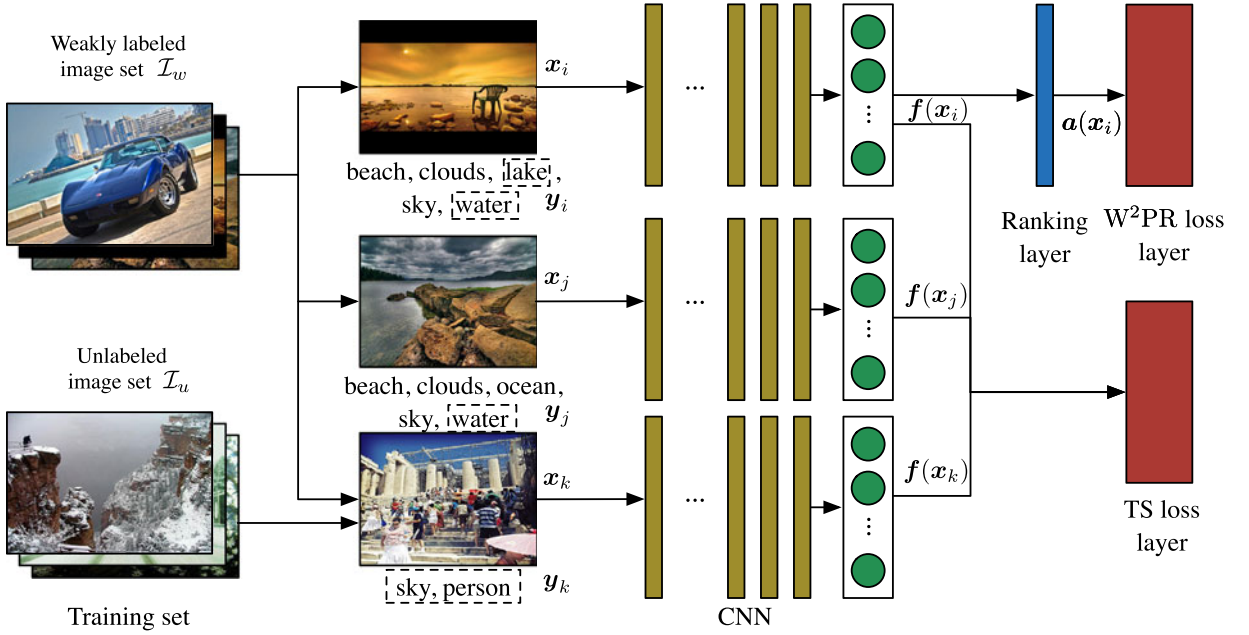


Fig. 2. The algorithmic pipeline of our proposed approach, where three deep CNNs have the same configurations and *share* the same weights to perform feature learning from an image triplet. The labels in dashed boxes are the ones that are missing (because data is not fully labeled) in the training set.  $x_k$  can be sampled from both  $\mathcal{I}_u$  and  $\mathcal{I}_w$ . In this example,  $x_k$  is an unlabeled image sampled from  $\mathcal{I}_u$ . If the label information of  $x_j$  and  $x_k$  is available, it is also possible to add ranking layer and  $W^2PR$  loss layer to  $f(x_j)$  and  $f(x_k)$  just the same as  $f(x_i)$ ; we omit them here for clarity.

Since for most images, we have  $\text{sim}(x_{\text{base}}, x_k) == 0$ , this approximation is acceptable.

In an online scenario, for every new image, we simply add its index into the corresponding list(s) *list* and then continue the sampling process.

The sampling algorithm is given in Algorithm 1.

#### Algorithm 1. Sampling of the Image Triplets

**Input:** images and their labels  $\{x_i, y_i\}$ , and image lists  $\text{list}_{1..l}$  and  $\text{list}_u$   
 Sample  $i$  from  $1 \dots l$  uniformly  
 Sample image  $x_{\text{base}}$  from list  $\text{list}_i$  uniformly  
 Sample  $j$  from set  $\{j' | y_{\text{base}}^{j'} == 1\}$  uniformly  
 Sample image  $x'_{\text{base}}$  from list  $\text{list}_j$  uniformly  
 Sample  $\text{unlabel} \sim \text{Bernoulli}(\gamma)$   
**if**  $\text{unlabel} == 1$  **then**  
   Sample  $x_k$  from list  $\text{list}_u$  uniformly  
**else**  
   **while true do**  
 Sample  $x_k$  from all labeled images uniformly  
**if**  $x_k \neq x_{\text{base}}$  and  $x_k \neq x'_{\text{base}}$  **then**  
   Sample *accept* following Equation (16)  
   **if** *accept* == 1 **then**  
     **break**  
   **end if**  
**end if**  
**end while**  
**end if**  
**Output:**  $(x_{\text{base}}, x'_{\text{base}}, x_k)$

### 3.3 Weakly Semi-Supervised Learning

The overall objective loss we optimize is given as follows (as shown in Fig. 2):

$$\sum_{(x_i, x_j, x_k)} \left\{ \underbrace{\sum_{s \in C_i^+} \sum_{t \in C_j^-} L_w(r_s) \max(0, m_s - a(x_i)^s + a(x_i)^t)}_{W^2PR \text{ loss with labeled images}} + \underbrace{\alpha \max(0, \|f(x_i) - f(x_j)\|_2^2 - \|f(x_i) - f(x_k)\|_2^2 + m_f)}_{TS \text{ loss with labeled and unlabeled images}} \right\}, \quad (17)$$

where

$$r \text{ sim}(x_i, x_j, x_k) > 0.$$

As discussed before, we do not optimize the loss on all the possible triplets, but instead train the network by the sampled triplets. It is also possible to optimize  $a(x_j)$  and  $a(x_k)$  with  $W^2PR$  loss at the same time as long as the label information of  $x_j$  and  $x_k$  is available.

## 4 EXPERIMENTS

### 4.1 Network Architecture

In our network, an image triplet is first sampled from the training image set. Before feeding the images in triplets to the CNN, each image is resized into  $256 \times 256$ . Then we crop a  $227 \times 227$  patch at random position from each image as input. This provides an augmentation of the dataset which is demonstrated to improve the generalization of the network as in [18]. Since we mainly focus on the loss layer, we use the widely used basic architecture in Alexnet [18] as the feature learning model (the CNN components in Fig. 2). The three CNNs for images in a triplet have the same architecture and *share* the same weights, which is similar to the siamese network [32] but with three networks instead of two. Then, the learnt features are fed into the triplet similarity loss layer which

TABLE 1  
The Statistics of the  $S_g$ ,  $S_g^{30}$ ,  $S_g^{50}$ ,  $S_g^{70}$  and  $S_u$  on the Training Set  $I_{train}$  of NUS-WIDE

	Ground-truth	Random dropped			User labeled
	$S_g$	$S_g^{30}$	$S_g^{50}$	$S_g^{70}$	$S_u$
Labeled images	150,000	128,632	107,969	78,017	86,048
Total labels	360,658	252,414	180,097	108,652	156,105
Average labels/image	2.40	1.96	1.67	1.39	1.81
Missing labels	0	81,653	118,596	124,675	136,507
Average missing rate (%)	0	30	50	70	33
Incorrect labels	0	0	0	0	49,710
Average incorrect rate (%)	0	0	0	0	31
Unlabeled images	0	21,368	42,031	71,983	63,952

computes the gradient of Equation (11). If the label information is available, the learnt feature is fed into a ranking layer as the activation, and the output of the ranking layer is fed into the  $W^2PR$  loss layer which computes the gradient of Equation (3).

The optimization of the entire network is achieved with stochastic gradient method, where the gradients of the connecting weights in each layer are computed by a back-propagation scheme. Following widely used parameter settings from the existing literature, the momentum is set to 0.9, and the batch size is set to 50. The learning rate for our model is set to 0.00002 at the start and we drop the learning rate after several epochs by a factor of 10.

## 4.2 Dataset

We evaluate our methods on two datasets, NUS-WIDE [33] and MS COCO 2014 [34], which we will discuss in detail.

The NUS-WIDE dataset is one of the largest datasets collected from Flickr. There are two kinds of label information for each image in NUS-WIDE described as follows:

- **$S_g$  (ground-truth information):** Each image in NUS-WIDE has been deliberately labeled with a pre-defined vocabulary set of size 81 depending on whether the concepts or objects are present in this image. By this way, NUS-WIDE provides the ground-truth label information of each image. The ground-truth label information means that every image is fully labeled according to the given vocabulary set. We refer this label information as  $S_g$ . It should be noted that the ground-truth label information is *only* utilized for performance evaluation and is not used in training for all different methods, since in this paper we are dealing with situation when the image label information is not complete.
- **$S_u$  (user labeled information):** Since each image in NUS-WIDE is from Flickr, there is label information from the user who uploaded it. However, this label information is highly noisy (i.e., many missing and incorrect tags) for each image. We refer this label information as  $S_u$ .

In this paper, the proposed method is devised to deal with missing labels (weakly labeled and unlabeled images); thus, we obtain the following label information for training as:

- **$S_g^{30}$  (missing rate 30 percent):** for the ground-truth label information of each image, we randomly set each *present* tag of this image as 0 with probability 30 percent. That is to say, for the  $i$ th image, if  $y_i^j = 1$  in  $S_g$ , there is a 30 percent probability that  $y_i^j = 0$  in the new label information. We refer the newly generated label information as  $S_g^{30}$ .
- **$S_g^{50}$  and  $S_g^{70}$  (missing rate 50 and 70 percent):** we follow the same procedure as in  $S_g^{30}$  to set each *present* tag of each image as 0 with probability 50 and 70 percent, respectively. We refer these two kinds of new label information as  $S_g^{50}$  and  $S_g^{70}$ , respectively.

As a result, images whose label information indicated by  $S_g^{30}$ ,  $S_g^{50}$  and  $S_g^{70}$  are weakly labeled or unlabeled.

For those images which are not labeled with anything in ground-truth label information  $S_g$ , we remove them from our experiments, resulting in 209,347 images in total. Then we randomly select 150,000 images as the training image set denoted as  $I_{train}$ , 19,375 images as the validation set denoted as  $I_{val}$ , and the rest 40,000 as the testing image set denoted as  $I_{test}$ . All the methods are trained based on the images in  $I_{train}$  with label information  $S_g^{30}$ ,  $S_g^{50}$ ,  $S_g^{70}$  and  $S_u$ , respectively; the ground-truth label information of each images in  $I_{train}$  is not used in training.

For images in  $I_{train}$ , Table 1 gives the statistics of different label information for training (i.e.,  $S_g^{30}$ ,  $S_g^{50}$ ,  $S_g^{70}$  and  $S_u$ ) as well as ground-truth label information  $S_g$ . As we observe from Table 1, the label information  $S_u$  is highly noisy; 136,507 tags are missing and 49,710 tags are incorrect; moreover, 63,952 images are not labeled by users. From Fig. 4, we can see that the average incorrect rate is high is mainly because the incorrect rates of less appeared labels are high.

MS COCO is another widely used large dataset in object recognition. The 2014 release contains 82,783 training and 40,504 validation images with 80 categories labeled for a total of 886,284 instances. Each instance comes with location information which we do not utilize in our experiment since our model does not resort to region information. We take the original training set as  $I_{train}$  and split the original validation set into two sets, one with 10,504 images as  $I_{val}$  and the rest as  $I_{test}$ . Following the same procedure for processing NUS-WIDE, we generate new label information  $S_g^{30}$ ,  $S_g^{50}$  and  $S_g^{70}$  for training. For images in  $I_{train}$ , Table 2 gives the



TABLE 2  
The Statistics of the  $S_g$ ,  $S_g^{30}$ ,  $S_g^{50}$  and  $S_g^{70}$  on  
the Training Set  $I_{train}$  of MS COCO

	Ground-truth	Random dropped		
	$S_g$	$S_g^{30}$	$S_g^{50}$	$S_g^{70}$
Labeled images	82,783	74,320	64,430	48,265
Total labels	241,035	168,519	120,810	72,111
Average labels/image	2.91	2.26	1.87	1.49
Missing labels	0	61,247	89,125	96,397
Incorrect labels	0	0	0	0
Unlabeled images	0	8,463	18,353	34,518

statistics of different label information for training (i.e.,  $S_g^{30}$ ,  $S_g^{50}$  and  $S_g^{70}$ ) as well as ground-truth label information  $S_g$  of MS COCO.

### 4.3 Evaluation Metrics

For each image, we assign the  $k$  highest ranked labels to the image and compare these labels with the ground-truth.

Following [1], we use the following four measures to evaluate the performance of different methods. For each label, we compute the recall and precision separately, and report the means of recall and precision on the label bases as:

$$R^+ = \frac{1}{m} \sum_{i=1}^m \frac{N_i^c}{N_i^g}, \quad P^+ = \frac{1}{m} \sum_{i=1}^m \frac{N_i^c}{N_i^p},$$

where  $N^c$  is the number of the correctly predicted images for each label,  $N^g$  is the number of the annotated images for each label in the ground-truth,  $N^p$  is the number of the images each algorithm predicts with each label. These two measures are highly biased toward infrequent labels. To make the evaluation unbiased, we also need to consider the overall recall and precision as follows:

$$R^A = \frac{\sum_{i=1}^m N_i^c}{\sum_{i=1}^m N_i^g}, \quad P^A = \frac{\sum_{i=1}^m N_i^c}{\sum_{i=1}^m N_i^p}.$$

These two metrics are dominated by frequent labels.

### 4.4 Comparison Methods

We compare our method with several other CNN-based deep learning methods as follows:

- **Softmax:** Softmax loss has been used for multi-label annotation in TagProp [2] which is also adopted in [3]. Similar to the single label problem, the target probability is given as  $y/\|y\|_1$  and cross-entropy.
- **Pairwise ranking:** Pairwise ranking loss [35] directly models the multi-label annotation problem by optimizing the ROC curve.
- **WARP:** Weighted Approximate Ranking Pairwise loss [3], [31] optimizes the top- $k$  accuracy of the annotation result. However, WARP requires that every training image be fully labeled.
- **W<sup>2</sup>PR:** The proposed approach WeSed can also be trained without the TS loss, which gives us a model with only W<sup>2</sup>PR loss without handling unlabeled images.

- $S_g$ : In this dataset, each image has different numbers of labels in ground-truth. Since we report the top ranked  $k$  tags ( $k = 3, 5$ ), the best performance is not 100 percent. As a result, we report the performance when the output is ground-truth itself in  $S_g$  which is a theoretical upper bound of our performance.

Since MS COCO only contains tens of thousands of images as training examples, which may not be sufficient to train a deep CNN from start, we train all methods on MS COCO with a network pre-trained with ImageNet [18] as the start point and we use a linear mapping following a sigmoid function as the activation of the ranking score of our model. When evaluating on NUS-WIDE, we train all methods with a random initialized network without any pre-training and use a linear mapping as the ranking score activation for our model.

It is worth noting that WeSed is the only model that is trained with both weakly labeled and unlabeled images in the experiments of this paper. Since the other methods are not able to deal with unlabeled images, they are trained with weakly labeled images only.

### 4.5 Results

We train our model and the comparison models on the training set with different label information and evaluate the performance of the trained models on the testing set with ground-truth  $S_g$ .

#### 4.5.1 Performance of Models Trained Using $S_g^{30}$ , $S_g^{50}$ and $S_g^{70}$

The results of each model trained with label information  $S_g^{30}$ ,  $S_g^{50}$  and  $S_g^{70}$  on MS COCO and NUS-WIDE are given in Tables 4, 5, 6, 7, 8, and 9.





As we see from these tables, our proposed model WeSed outperforms the other methods on both datasets when dealing with multi-label images with missing labels by handling weakly labeled images and unlabeled images at the same time.

From Table 7, we see that W<sup>2</sup>PR loss performs even worse than WARP, and in Table 4, W<sup>2</sup>PR loss is only 1 to 2 percent absolutely better than WARP, which indicates that when the missing rate is not high (only 30 percent in this case), it is very effective to allow negative labels to be ranked top, as what W<sup>2</sup>PR does. Since there are not many labels missing, W<sup>2</sup>PR allows some errors to be ranked top, which may deteriorate the performance. However, the performance of WeSed significantly outperforms the other methods, especially on NUS-WIDE, which indicates that the low missing rate gives a high semantic accuracy when sampling image triplets; this helps the network to learn better features from both weakly labeled and unlabeled images and thus improves the performance.

As the missing rate increases from 30 to 70 percent, the performance of W<sup>2</sup>PR gradually getting better comparing to other methods in all the metrics, which shows the effectiveness of our W<sup>2</sup>PR loss when dealing with weakly labeled images. The performance on MS COCO increased from 1–2 percent to 2–3 percent absolutely better compared to



TABLE 3  
Demonstration of Images in NUS-WIDE from Training Set

Image	Ground-truth	Training Data			
	$S_g$	$S_u$	$S_g^{30}$	$S_g^{50}$	$S_g^{70}$
	clouds, lake, sky, sun, sunset, water		clouds, lake, sun, sunset, water	sunset, water	lake
	animal, dog, grass, running	dog, running	animal, grass	grass, running	
	clouds, lake, ocean, sky, water	water	clouds, ocean, wa- ter	clouds, lake, ocean, sky, water	clouds, ocean
	airport, plane, sky, vehicle	<u>fire</u> , sky, <u>water</u>	airport, plane, ve- hicle	vehicle	
	frost, plants, road, tree	frost, road, <u>sun</u>	frost, plants, road	frost, road, tree	plants
	buildings, house, reflection, sky, town, window		buildings, house, window	house, reflection, window	reflection, town
	clouds, person, sky, window	<u>street</u>	clouds, person, sky, window	clouds, sky, win- dow	sky

Note the labeling information in  $S_u$  is highly missing; most images are unlabeled. There also are some labeling errors in  $S_u$ , which are underlined.

WARP, and on NUS-WIDE  $W^2PR$  outperforms WARP significantly. On the other hand, we can see the improvement of WeSed compared to  $W^2PR$  decreases gradually; this indicates that the high missing rate gives less semantically accurate image triplets and thus the effectiveness of TS loss is deteriorated.

We further demonstrate the prediction precision of some most and least appeared labels in Fig. 3. We can observe that all methods obtain comparable precision on labels with high appearance frequency, while WeSed outperforms other slightly. This is easy to see since these labels have many positive samples thus are easier to learn to predict. Some labels appearing only a few hundreds times are much harder to learn; we can observe the WeSed outperforms other methods by using a weakly ranking loss to prevent

the missing label being pushed down and using image triplets to boost the feature learning.

Overall, by leveraging both weakly labeled and unlabeled images, WeSed achieves a promising performance in all different missing conditions  $S_g^{30}$ ,  $S_g^{50}$  and  $S_g^{70}$  on both datasets.

It is also worth noting that in the above experiments, the missing rate for each label word in the vocabulary set is equal (i.e., 30, 50 or 70 percent); thus, it is demonstrated clearly that different losses benefit in different situations, as here  $W^2PR$  in the high missing rate and TS in the low missing rate. Consequently, as in practice the missing rate of each label may be different, it is noticeably beneficial to use WeSed that effectively combines  $W^2PR$  and TS loss to deal with all the situations as demonstrated below.

TABLE 4  
Multi-Label Annotation Performance: Models Trained with  $S_g^{30}$  (30 Percent Label Missing) on MS COCO

	R <sup>+</sup> @3	P <sup>+</sup> @3	R <sup>A</sup> @3	P <sup>A</sup> @3	R <sup>+</sup> @5	P <sup>+</sup> @5	R <sup>A</sup> @5	P <sup>A</sup> @5
$S_g$ (Upper bound)	79.79	67.11	78.29	75.81	93.49	45.53	93.27	54.19
Softmax	43.79	45.58	52.89	51.20	55.60	34.37	64.96	37.73
Pairwise Ranking	44.53	45.85	53.21	51.50	56.99	34.63	65.47	37.62
WARP	43.53	44.64	52.32	50.66	55.53	34.24	64.59	37.53
W <sup>2</sup> PR	44.57	46.45	53.52	51.83	57.58	34.54	66.11	38.41
WeSed	<b>44.69</b>	<b>47.50</b>	<b>53.86</b>	<b>52.16</b>	<b>57.42</b>	<b>35.26</b>	<b>66.38</b>	<b>38.57</b>

TABLE 5  
Multi-Label Annotation Performance: Models Trained with  $S_g^{50}$  (50 Percent Label Missing) on MS COCO

	R <sup>+</sup> @3	P <sup>+</sup> @3	R <sup>A</sup> @3	P <sup>A</sup> @3	R <sup>+</sup> @5	P <sup>+</sup> @5	R <sup>A</sup> @5	P <sup>A</sup> @5
$S_g$ (Upper bound)	79.79	67.11	78.29	75.81	93.49	45.53	93.27	54.19
Softmax	43.60	44.30	52.03	50.39	55.21	33.43	64.02	37.19
Pairwise Ranking	44.20	45.64	53.07	51.22	56.81	34.21	65.28	37.47
WARP	43.13	44.14	51.81	50.17	54.96	33.80	63.98	37.18
W <sup>2</sup> PR	44.30	46.03	53.37	51.69	57.03	34.08	65.70	38.17
WeSed	<b>44.38</b>	<b>46.81</b>	<b>53.60</b>	<b>51.91</b>	<b>57.04</b>	<b>34.65</b>	<b>65.89</b>	<b>38.28</b>

TABLE 6  
Multi-Label Annotation Performance: Models Trained with  $S_g^{70}$  (70 Percent Label Missing) on MS COCO

	R <sup>+</sup> @3	P <sup>+</sup> @3	R <sup>A</sup> @3	P <sup>A</sup> @3	R <sup>+</sup> @5	P <sup>+</sup> @5	R <sup>A</sup> @5	P <sup>A</sup> @5
$S_g$ (Upper bound)	79.79	67.11	78.29	75.81	93.49	45.53	93.27	54.19
Softmax	42.18	41.10	49.15	47.58	53.39	31.65	61.37	35.64
Pairwise Ranking	43.79	44.95	52.49	50.63	56.22	34.21	64.40	37.19
WARP	41.82	42.36	50.29	48.70	53.42	32.93	62.48	36.30
W <sup>2</sup> PR	<b>43.88</b>	<b>45.69</b>	52.45	50.80	56.10	<b>34.73</b>	64.49	37.47
WeSed	43.33	45.16	<b>52.63</b>	<b>50.96</b>	<b>56.11</b>	33.94	<b>64.88</b>	<b>37.69</b>

TABLE 7  
Multi-Label Annotation Performance: Models Trained with  $S_g^{30}$  (30 Percent Label Missing) on NUS-WIDE

	R <sup>+</sup> @3	P <sup>+</sup> @3	R <sup>A</sup> @3	P <sup>A</sup> @3	R <sup>+</sup> @5	P <sup>+</sup> @5	R <sup>A</sup> @5	P <sup>A</sup> @5
$S_g$ (Upper bound)	84.19	41.72	82.59	66.42	96.59	28.30	96.27	46.45
Softmax	28.48	32.31	58.24	46.93	43.97	23.60	73.75	35.43
Pairwise Ranking	26.76	30.09	56.84	45.56	41.85	22.50	71.77	34.52
WARP	29.17	31.43	58.21	46.92	45.09	25.00	73.72	35.66
W <sup>2</sup> PR	28.05	30.20	57.78	46.32	42.88	23.13	73.02	35.12
WeSed	<b>30.49</b>	<b>35.10</b>	<b>58.75</b>	<b>47.00</b>	<b>46.69</b>	<b>25.92</b>	<b>74.52</b>	<b>35.77</b>

TABLE 8  
Multi-Label Annotation Performance: Models Trained with  $S_g^{50}$  (50 Percent Label Missing) on NUS-WIDE

	R <sup>+</sup> @3	P <sup>+</sup> @3	R <sup>A</sup> @3	P <sup>A</sup> @3	R <sup>+</sup> @5	P <sup>+</sup> @5	R <sup>A</sup> @5	P <sup>A</sup> @5
$S_g$ (Upper bound)	84.19	41.72	82.59	66.42	96.59	28.30	96.27	46.45
Softmax	27.69	31.34	57.08	45.91	42.02	22.80	70.82	33.87
Pairwise Ranking	26.31	29.89	56.33	45.15	40.28	22.27	71.24	34.09
WARP	27.27	30.12	57.16	45.82	42.60	22.64	72.06	34.66
W <sup>2</sup> PR	27.58	30.35	56.91	45.62	40.70	22.69	71.62	34.45
WeSed	<b>28.95</b>	<b>33.24</b>	<b>57.37</b>	<b>45.99</b>	<b>44.29</b>	<b>24.62</b>	<b>72.67</b>	<b>34.95</b>

#### 4.5.2 Performance of Models Trained Using $S_u$

The performance results of each model trained using  $S_u$  are given in Table 10. The main differences between user label information and the synthetic label information are: the

missing rate of each label is not same and there are some incorrect information<sup>2</sup> in user label information.

2. Incorrect here means false positive, i.e.,  $y_i^j = 1$  while this label is actually not present in this image.

TABLE 9  
Multi-Label Annotation Performance: Models Trained with  $S_g^{70}$  (70 Percent Label Missing) on NUS-WIDE

	$R^+@3$	$P^+@3$	$R^A@3$	$P^A@3$	$R^+@5$	$P^+@5$	$R^A@5$	$P^A@5$
$S_g$ (Upper bound)	84.19	41.72	82.59	66.42	96.59	28.30	96.27	46.45
Softmax	24.95	28.04	55.67	44.63	38.30	21.94	70.68	33.80
Pairwise Ranking	25.33	27.40	55.11	44.18	38.84	21.63	69.69	33.52
WARP	25.28	27.87	54.95	44.38	38.94	22.34	70.07	33.95
W <sup>2</sup> PR	26.21	29.58	55.20	44.50	<b>40.86</b>	<b>22.95</b>	70.55	34.12
WeSed	<b>26.95</b>	<b>29.91</b>	<b>55.86</b>	<b>44.78</b>	40.31	22.34	<b>70.96</b>	<b>34.13</b>

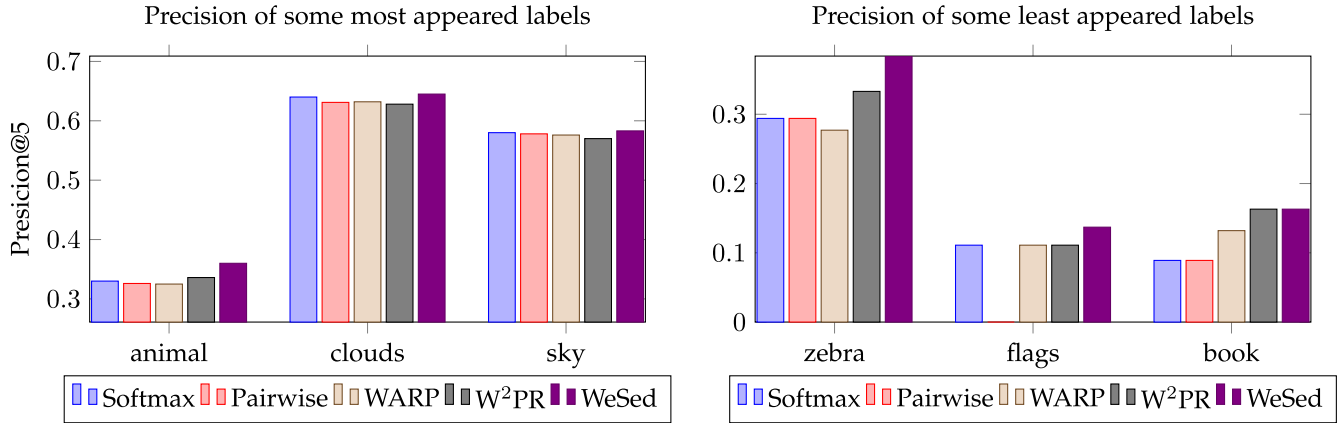


Fig. 3. Precision@5 of some most and least appeared labels. Trained with  $S_g^{30}$  (30 percent label missing) on NUS-WIDE.

TABLE 10  
Multi-Label Annotation Performance: Models Trained with  $S_u$  (User Labeled) on NUS-WIDE

	$R^+@3$	$P^+@3$	$R^A@3$	$P^A@3$	$R^+@5$	$P^+@5$	$R^A@5$	$P^A@5$
$S_g$ (Upper bound)	84.19	41.72	82.59	66.42	96.59	28.30	96.27	46.45
Softmax	27.58	22.63	36.18	29.13	38.55	17.48	48.96	23.65
Pairwise Ranking	27.55	22.62	36.58	29.46	38.83	18.10	49.87	24.09
WARP	26.27	22.23	37.00	29.79	37.53	18.20	50.15	24.23
W <sup>2</sup> PR	27.24	<b>22.76</b>	37.38	30.10	<b>38.88</b>	<b>18.57</b>	50.63	24.46
WeSed	<b>27.37</b>	22.30	<b>38.21</b>	<b>30.82</b>	38.35	17.87	<b>51.24</b>	<b>24.80</b>

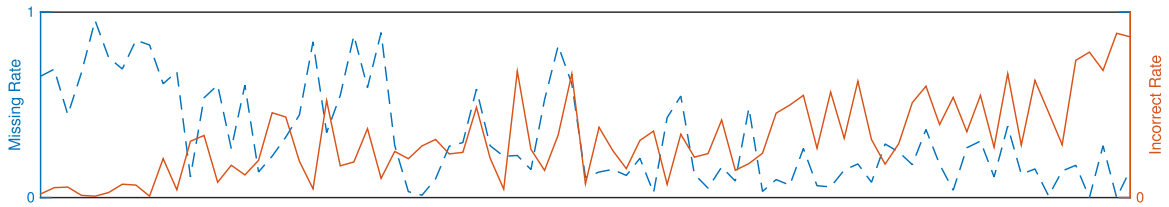


Fig. 4. The Statistics of the Missing Rate and Incorrect Rate of Labels with  $S_u$ . The labels are ordered by frequency in  $S_g$ . The missing rate is given in blue dash line and incorrect rate given in orange solid line.

The demonstration of incorrect label information is given in Table 3, where the errors are underline. The statistic of the missing rate and incorrect rate of labels is given in Fig. 4. We can observe that the most frequency labels do not have much incorrect label information while the incorrect rates of the infrequency labels are high. On the other hand, the missing rates of labels do not have any particular patterns with respect to their frequencies.

As we observe, the noise in user labeled information  $S_u$  does deteriorate the performance, especially the high missing rates in some high frequent labels. However, W<sup>2</sup>PR and WeSed outperform the other comparison methods on

all the metrics, particularly the overall recall and precision ( $R^A$  and  $P^A$ ).

#### 4.5.3 The Sensitivity of Weak Ranking Threshold $l$

To further evaluate the effectiveness of W<sup>2</sup>PR loss, we give the performance of W<sup>2</sup>PR with the variation of parameter  $l$  in Fig. 5. This experiment is performed with the same setting as that in Table 9.

As we can see from the figure, the performance of per-class recall ( $R^+@3$ ) and per-class precision ( $P^+@3$ ) is significantly improved with the growing of  $l$ , and achieves the best performance around  $l = 5$ . It is also worth noting that the

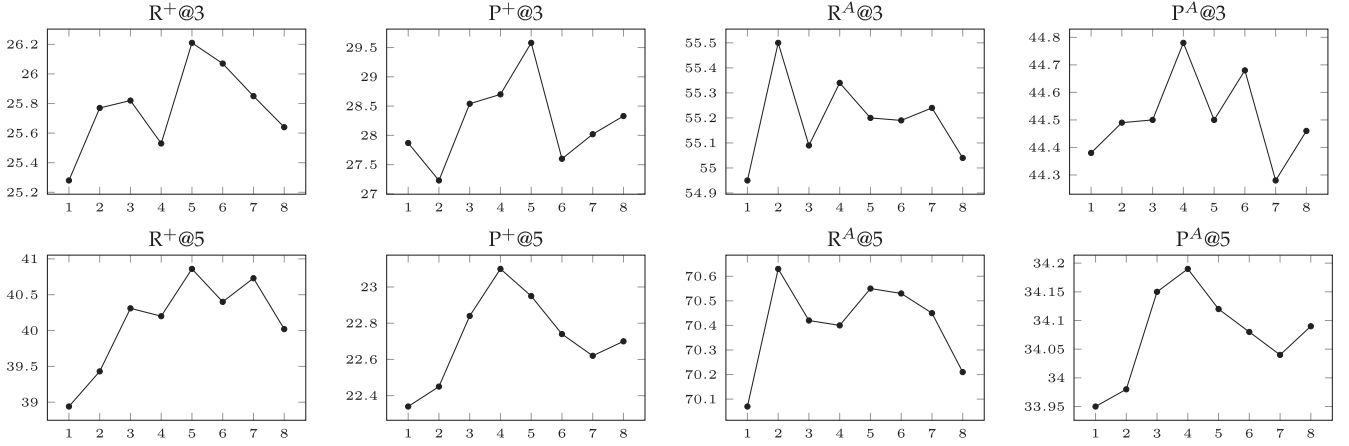


Fig. 5. The performance of model  $W^2PR$  with the variation of  $l$  on NUS-WIDE, trained with  $S_g^{70}$ .

performance of the overall recall ( $R^A@3$ ) and precision ( $P^A@3$ ) does not improve much (0.3 percent), which indicates that the main performance gain in  $W^2PR$  is from the infrequent labels. The performances evaluate with top 5 ( $R^+@5$ ,  $P^+@5$ ,  $R^A@5$ ,  $P^A@5$ ) follow a similar pattern. By setting  $l$  to a proper value,  $W^2PR$  loss allows the missing labels to be ranked top, which is of great importance for infrequent labels. Learning to annotate infrequent labels is difficult, especially when there are missing infrequent labels. Consider a model trained with WARP; at some training point when the model has learnt to annotate some infrequent label, an image sample with this label missing comes in, and the model then tries to back-propagate cues that rank this label lower; since this label appear infrequently, this information is highly informative, leading to a lower performance.  $W^2PR$ , on the contrary, allows this label to appear at the top even if this label is missing; thus these infrequent labels are better learned.

However, the best performance is achieved when  $l = 5$ , not the average number of labels per image in  $S_g$  ( $l = 2$  or  $l = 3$ ); this suggests that we allow several wrong labels to be ranked high to achieve the best performance. The reason is due to the fact that our method cannot achieve 100 percent accuracy, resulting in some mistakes ranked highly. To avoid pushing down the missing labels, we relax the weak restriction  $l$  to a larger value than the average number of labels in each image.

Overall, Fig. 5 shows the effectiveness and importance of properly handling weakly labeled images with  $W^2PR$  loss.

## 5 CONCLUSION

In this paper, we have proposed WeSed to specifically address the need of weakly semi-supervised learning for multi-label image annotation. WeSed can be trained using deep learning with a large number of weakly labeled and unlabeled images readily available from various social networks, along with the strategy of an efficient triplet sampling method. Experiment results have demonstrated its superiority over the existing systems and potential impact for practical applications with noisy data.

## ACKNOWLEDGMENTS

This work was supported in part by the National Basic Research Program of China under Grant 2015CB352300, in

part by the China Knowledge Centre for Engineering Sciences and Technology. The work of Zhongfei Zhang was supported in part by the U.S. National Science Foundation under Grant CCF-1017828 and in part by the Zhejiang Provincial Engineering Center on Media Data Cloud Processing and Analysis. Jiebo Luo would like to thank the generous support of Yahoo, Xerox, and the New York State CoE CEIS and IDS.

## REFERENCES

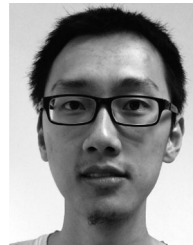
- [1] A. Makadia, V. Pavlovic, and S. Kumar, "A new baseline for image annotation," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 316–329.
- [2] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid, "Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, 2009, pp. 309–316.
- [3] Y. Gong, Y. Jia, T. Leung, A. Toshev, and S. Ioffe, "Deep convolutional ranking for multilabel image annotation," *CoRR*, vol. abs/1312.4894, 2013. Available: <http://arxiv.org/abs/1312.4894>
- [4] Z. Qi, M. Yang, Z. M. Zhang, and Z. Zhang, "Mining noisy tagging from multi-label space," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage.*, 2012, pp. 1925–1929.
- [5] Y. Li, Z. Qi, Z. M. Zhang, and M. Yang, "Learning with limited and noisy tagging," in *Proc. ACM Int. Conf. Multimedia*, 2013, pp. 957–966.
- [6] Y. Yang, F. Wu, F. Nie, H. Shen, Y. Zhuang, and A. Hauptmann, "Web and personal image annotation by mining label correlation with relaxed visual graph embedding," *IEEE Trans. Image Process.*, vol. 21, no. 3, pp. 1339–1351, Mar. 2012.
- [7] Y. Han, F. Wu, Q. Tian, and Y. Zhuang, "Image annotation by input-output structural grouping sparsity," *IEEE Trans. Image Process.*, vol. 21, no. 6, pp. 3066–3079, Jun. 2012.
- [8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. (2014). Going deeper with convolutions. *CoRR*, abs/1409.4842 [Online]. Available: <http://arxiv.org/abs/1409.4842>
- [9] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos, "Supervised learning of semantic classes for image annotation and retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 394–410, Mar. 2007.
- [10] F. Monay and D. Gatica-Perez, "PLSA-based image auto-annotation: Constraining the latent space," in *Proc. ACM Int. Conf. Multimedia*, 2004, pp. 348–351.
- [11] R. Fergus, Y. Weiss, and A. Torralba, "Semi-supervised learning in gigantic image collections," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 522–530.
- [12] Z. Ma, Y. Yang, F. Nie, J. Uijlings, and N. Sebe, "Exploiting the entire feature space with sparsity for automatic image annotation," in *Proc. ACM Int. Conf. Multimedia*, 2011, pp. 283–292.
- [13] F. Wu, Z. Yu, Y. Yang, S. Tang, Y. Zhang, and Y. Zhuang, "Sparse multi-modal hashing," *IEEE Trans. Multimedia*, vol. 16, no. 2, pp. 427–439, Feb. 2014.



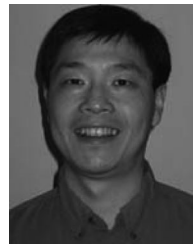
- [14] X.-S. Xu, Y. Jiang, L. Peng, X. Xue, and Z.-H. Zhou, "Ensemble approach based on conditional random field for multi-label image and video annotation," in *Proc. ACM Int. Conf. Multimedia*, 2011, pp. 1377–1380.
- [15] A. Fakeri-Tabrizi, M. R. Amini, and P. Gallinari, "Multiview semi-supervised ranking for automatic image annotation," in *Proc. ACM Int. Conf. Multimedia*, 2013, pp. 513–516.
- [16] Y. Ushiku, T. Harada, and Y. Kuniyoshi, "Efficient image annotation for automatic sentence generation," in *Proc. ACM Int. Conf. Multimedia*, 2012, pp. 549–558.
- [17] X. Tan, F. Wu, X. Li, S. Tang, W. Lu, and Y. Zhuang, "Structured visual feature learning for classification via supervised probabilistic tensor factorization," *IEEE Trans. Multimedia*, vol. 17, no. 5, pp. 660–673, May 2015.
- [18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [19] M. Lin, Q. Chen, and S. Yan, "Network in network," *CoRR*, vol. abs/1312.4400, 2013. Available: <http://arxiv.org/abs/1312.4400>
- [20] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580 [Online]. Available: <http://arxiv.org/abs/1207.0580>
- [21] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 1058–1066.
- [22] I. Goodfellow, D. Warde-farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 1319–1327.
- [23] M. Wang, X.-S. Hua, T. Mei, R. Hong, G. Qi, Y. Song, and L.-R. Dai, "Semi-supervised kernel density estimation for video annotation," *Comput. Vis. Image Understanding*, special issue on Video Analysis, vol. 113, no. 3, pp. 384–396, 2009.
- [24] M. Wang, X.-S. Hua, R. Hong, J. Tang, G.-J. Qi, and Y. Song, "Unified video annotation via multigraph learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 5, pp. 733–746, May 2009.
- [25] S. H. Amiri and M. Jamzad, "Automatic image annotation using semi-supervised generative modeling," *Pattern Recog.*, vol. 48, no. 1, pp. 174–188, 2015.
- [26] M. Wang, X. Liu, and X. Wu, "Visual classification by  $\ell_1$ -hypergraph modeling," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 9, pp. 2564–2574, Sep. 2015.
- [27] T. Deselaers and V. Ferrari, "Visual and semantic similarity in imagenet," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 1777–1784.
- [28] Y. Terada and U. V. Luxburg, "Local ordinal embedding," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 847–855.
- [29] L. Van Der Maaten and K. Weinberger, "Stochastic triplet embedding," in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process.*, 2012, pp. 1–6.
- [30] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 1386–1393.
- [31] J. Weston, S. Bengio, and N. Usunier, "Wsabie: Scaling up to large vocabulary image annotation," in *Proc. Int. Joint Conf. Artif. Intell.*, 2011, vol. 11, pp. 2764–2770.
- [32] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, 2005, vol. 1, pp. 539–546.
- [33] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng, "NUS-wide: A real-world web image database from National University of Singapore," presented at the ACM Conf. Image Video Retrieval, Santorini, Greece, 2009.
- [34] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollr, and C. Zitnick, "Microsoft Coco: Common objects in context," in *Proc. 13th Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
- [35] T. Joachims, "Optimizing search engines using clickthrough data," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2002, pp. 133–142.



**Fei Wu** received the BSc degree from Lanzhou University, Lanzhou, China, in 1996, the MSc degree from the University of Macau, Macau, China, in 1999, and the PhD degree from Zhejiang University, Hangzhou, China, in 2002, all in computer science. He is currently a full professor with the College of Computer Science and Technology, Zhejiang University. His current research interests include multimedia retrieval, sparse representation, and machine learning.



**Zhu hao Wang** received the BE degree from Zhejiang University, Hangzhou, Zhejiang, China, in 2012, where he is currently working toward the PhD degree in computer science with the Digital Media Computing and Design Laboratory. His research interests include deep learning and image understanding.



**Zhongfei Zhang** received the BS (cum laude) degree in electronics engineering and the MS degree in information science from Zhejiang University, and the PhD degree in computer science from the University of Massachusetts at Amherst. He is currently a full professor of computer science with the Binghamton University, State University of New York. He is also the director with the Multimedia Research Laboratory at Binghamton.



**Yi Yang** received the PhD degree in computer science from Zhejiang University. He was a post-doctoral research fellow with the School of Computer Science, Carnegie Mellon University, from 2011 to 2013. He is currently a senior lecturer with the Centre for Quantum Computation and Intelligent Systems, University of Technology at Sydney, Sydney. His research interests include multimedia and computer vision.



**Jiebo Luo** (S'93-M'96-SM'99-F'09) joined the University of Rochester in Fall 2011, after over 15 years at Kodak Research Laboratories, where he was a senior principal scientist leading research and advanced development. He is a fellow of the IEEE, International Society for Optics and Photonics (SPIE), and the International Association for Pattern Recognition (IAPR).



**Wenwu Zhu** (S'91-M'96-SM'01-F'10) received the PhD degree in electrical and computer engineering from the New York University Polytechnic School of Engineering, Brooklyn, NY, in 1996. He is a professor of the 1,000 People Plan of China with the Department of Computer Science, Tsinghua University, Beijing, China. His current research interests include the areas of multimedia cloud computing, social media computing, multimedia big data, and multimedia communications and networking. He is a fellow of the IEEE.



**Yueting Zhuang** received the BS, MS, and PhD degrees from Zhejiang University, Hangzhou, China, in 1986, 1989, and 1998, respectively. He is currently a full professor with the College of Computer Science and Technology, Zhejiang University. His research interests include artificial intelligence, multimedia retrieval, digital library, and video-based animation.

▷ **For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).**