Semi-Supervised Robust Deep Neural Networks for Multi-Label Classification

Hakan Cevikalp[1], Burak Benligiray[2], Omer Nezih Gerek[2], Hasan Saribas[2]
[1]Eskisehir Osmangazi University, [2]Eskisehir Technical University
Electrical and Electronics Engineering Department
hakan.cevikalp@gmail.com, {burakbenligiray,ongerek,hasansaribas}@eskisehir.edu.tr

Abstract

In this paper, we propose a robust method for semi-supervised training of deep neural networks for multi-label image classification. To this end, we use ramp loss, which is more robust against noisy and incomplete image labels compared to the classical hinge loss. The proposed method allows for learning from both labeled and unlabeled data in a semi-supervised learning setting. This is achieved by propagating labels from the labeled images to their unlabeled neighbors. Using a robust loss function becomes crucial here, as the initial label propagations may include many errors, which degrades the performance of non-robust loss functions. In contrast, the proposed robust ramp loss restricts extreme penalties for the samples with incorrect labels, and the label assignment improves in each iteration and contributes to the learning process. The proposed method achieves state-of-the-art results in semi-supervised learning experiments on the CIFAR-10 and STL-10 datasets, and comparable results to the state-of the-art in supervised learning experiments on the NUS-WIDE and MS-COCO datasets.

1. Introduction

Multi-label image classification is a challenging task, and it has recently attracted great attention from the research community. The majority of the studies on image classification focus on single-label classification, and very successful methods have been proposed for this task [18, 25]. However, using a single label for an image is generally not appropriate for real-world applications, as the majority of the images can be associated with multiple labels to describe its semantic contents, such as objects, scenes, actions and attributes. A successful multi-label classification framework should be able to learn the association between visual features and these complex tags.

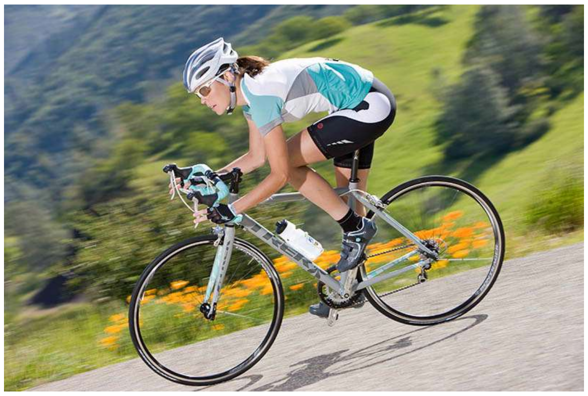
Figure 1: Although a human annotator is going to use labels such as *people*, *bicycle*, *road* for this image, additional labels such as *wheel*, *saddle*, *pedal*, *cycling* can also be used for labeling.

Popularization of digital cameras, web-based services and social networks let users to upload, share and tag tremendous amount of images everyday. Clearly, there is an emerging need to correctly label images, learn from them and retrieve relevant ones when needed. Manually labeling all images is too costly and difficult in practice. As a consequence, image labels are mostly collected by (semi) automatic tools using image file names, the text surrounding the image, or tags provided by the users. This labeling procedure causes several problems. One of these is the fact that the labels are going to be incomplete for most cases. Consider the image in Fig. 1 as an example: A human annotator may label this image with *people*, *bicycle*, *road*, but in addition to these, *wheel*, *saddle*, *pedal*, *cycling* are also correct. Similarly, in a hierarchical class model like ImageNet, labeling an object to be a *poodle* logically implies it is a *dog* and a *mammal* as well. Those labels will often be absent because they are assumed to be obvious. Furthermore, a poodle may still have additional attributes, such as *female* and *pink*. Even in a user-model-free pure attribute model, there are usually statistical image relationships. For example, *zebra* coincides with *striped*, and *car* typically implies *wheel*. ImageNet nominally has only one labeled class per image, but in reality, several others are often present, and even visually dominant. Therefore, a successful classification system must not consider all absent labels as negatives. Another property of image labels is that they are unlikely to be independent or uncorrelated. In fact, recent studies [16,32,14,37] show that there is a strong correlation among the labels of the multi-labeled images. For example, *car* and *pedestrian* usually appear together, but *pet* and *whale* are rarely seen together. Therefore, there may be a need for modeling label dependencies and adopting a learning algorithm that utilizes this information.

The final critical problem with the labels is that some of them are incorrect since they are mostly collected by (semi) automatic tools. For instance, Sun et al. [29] recently collected one of the largest visual object classification dataset, JFT-300M, which has more than 375M noisy labels for 300M images. The labels are collected automatically, which resulted in many missing and noisy labels. Although the authors use sophisticated algorithms to clean the noisy labels, approximately 20% of the labels are still estimated to be erroneous. These kinds of examples show the urgent need for a robust multi-label learning algorithm.

Our Contributions: In this paper, we introduce a semi-supervised multi-label image classification method that can learn from images that have noisy and incomplete labels, and even unlabeled images. In order to achieve this, we interchange the hinge loss used in the weighted approximate ranking method [9] with the more robust ramp loss. To utilize unlabeled image data, we use label propagation by assigning labels to the unlabeled data based on their nearest labeled neighbors, resulting in bootstrapping. Our proposed classifier can both be used with hand-crafted features, or can be jointly trained with the feature extractor. To achieve the latter, we integrated the proposed classifier into a deep convolutional neural network as the loss function of the final classification stage.

Related Work: Various methods have been proposed for multi-label image classification over the past few years. One of the most common approaches is based on label ranking, which has been applied to powerful deep convolutional neural networks successfully. Pair-wise ranking loss was first used in [30], and it was extended to weighted approximate ranking (WARP) in [9,34]. Wu et al. [36] used WARP for weakly labeled image classification problems. Kanehira and Harada [15] also used ranking and proposed multi-label PU (positive and unlabeled) method, which mostly focused on positive labels and took into consideration the fact that absent labels are not necessarily negative as we have described earlier. However, they assume that all assigned positive labels are definitely correct, thus their method is not very robust to noisy labels. Lapin et al. [19,21,20] introduced methods that optimize top-k error loss functions, such as smooth top-k hinge loss and top-k softmax loss, and they report significant improvements on some multi-label image data sets.

Several methods treat the multi-label classification problem as a deep-neural network based object detection problem, and they follow a similar strategy to the R-CNN method of Girshick et al. [8], which returns promising candidate regions from an image and classifies them with a CNN. To achieve this, Wei et al. [33] use BING to return candidate regions, and a hypothesis extraction method to reduce and refine these regions using NCuts clustering. The resulting regions are then fed into a CNN classifier as in [8]. Differently from these, Zhang et al. [37] and Karpathy and Fei-Fei [16] both use a CNN to extract candidate regions and these regions are fed to an RNN to model label dependencies. However, both methods require bounding box annotations, which can be seen as a major limitation, since bounding box annotation is even more costly than labeling itself. Similar to these, [32] and [14] also use CNN and RNN together, but they feed the entire image to CNN and use the resulting output in an RNN to produce multiple dependent labels. Therefore, these methods do not require bounding box information. More recently, Zhu et al. [40] have introduced a multi-label image classification method that exploits both semantic and spatial relations between labels with only image-level supervision. In addition to these methods, there are also more general multi-label classification methods using SVMs (Support Vector Machines) or kernel methods [6], nearest neighbors [38], and decision trees [3]. A more comprehensive survey of multi-label classification methods can be found in [39].

Recently, there have been efforts [24,10,22,35,26,17] to train deep neural networks in a semi-supervised way to utilize both labeled and unlabeled data, as neural nets require a very large amount of data due to their size. Haeusser et al. [10] proposed a sophisticated semi-supervised deep neural network method that utilizes learning by association. Their proposed method encourages correct association cycles between the embeddings of labeled and unlabeled samples. In [22], a simple method is introduced, which assigns pseudo-labels to unlabeled data based on maximum predicted class membership probabilities, and trains a supervised neural network by using both supervised labels and predicted labels. In a recent work, Wang et al. [31] have proposed a semi-supervised deep neural network method that alternates iteratively between growing convolutional layers and selecting confidently pseudo-labeled data. There are also other semi-supervised learning methods that use auto-encoder structures [24] or generative adversarial networks (GANs) [26,17,5].

2. Method

In this work, we consider the case where we have l multi-labeled images $\mathcal{L} = \{(I_1, \mathbf{y}_1), \ldots, (I_l, \mathbf{y}_l)\}$ and u unlabeled images $\mathcal{U} = \{I_{l+1}, \ldots, I_{l+u}\}$, where I_i represents the image, and $\mathbf{y}_i \in \{-1, 1\}^m$ is the corresponding label vector with -1 indicating a negative and 1 indicating a positive class. As mentioned earlier, labels of images are usually collected from image file names and nearby text, or tags provided by users. Therefore, some labels may be incorrect or missing. Let us denote the visual feature vector of an image as $\mathbf{x}_i \in \mathbb{R}^d$. Supervised and semi-supervised learning process cases are as described below.

2.1. Supervised Learning

Assuming that $C^+_{\mathbf{x}_i}$ and $C^-_{\mathbf{x}_i}$ denote the sets of indices with positive and negative labels for $\mathbf{x}_i$, our proposed classifier solves the following optimization problem

$$\min_{\mathbf{W}} \frac{\lambda}{2} \text{trace}(\mathbf{W}^\top \mathbf{W}) + \sum_{i=1}^{l} \sum_{j=1}^{C^+_{\mathbf{x}_i}} \sum_{k=1}^{C^-_{\mathbf{x}_i}} L(r_j) R_s(\mathbf{w}_j^\top \mathbf{x}_i - \mathbf{w}_k^\top \mathbf{x}_i) + \kappa \sum_{i=1}^{l} \sum_{j=1}^{m} R_s(y_{ij}(\mathbf{w}_j^\top \mathbf{x}_i)), \quad (1)$$

where $L(.)$ is a weighting function for different ranks, r_j is the rank of the i-th image sample for the j-th class, $R_s(t) = \min(1-s, \max(0, 1-t))$ is the ramp loss function, and $-1 < s \leq 0$ is a parameter that must be fixed by the user. Note that $\mathbf{w}_j$ corresponds to the j-th column of $d \times m$ weight matrix $\mathbf{W}$, λ is the regularization parameter that must be set by the user, and κ is also a user defined parameter that controls the weight of the ramp loss. The ramp loss can be written as the difference of two convex hinge losses, $R_s(t) = H_1(t) - H_s(t)$, where $H_a(t) = \max(0, a-t)$ is the classical hinge loss. In our experiments, we set $s = -0.80$.

The first two terms in our optimization problem (Eq. 1) are similar to WARP [9,34], in the sense that $L(.)$ is used to control the ranking. The first term in the optimization problem is a regularization term to ensure that the method can also be used with hand-crafted features. When the proposed method is integrated into the deep neural nets, this regularization term is omitted, as described below. Differently from WARP, we use the robust ramp loss function instead of the classical hinge loss in the second term of the optimization problem. In addition, WARP does not include the last term in the optimization problem and this term ensures that the resulting classifier returns positive scores (> 1) for the positive class samples and negative scores (< -1) for the negative class samples.

We use the same weighting function as in [9,36], which is defined as

$$L(r) = \sum_{j=1}^{r} \alpha_j, \quad (2)$$

where α_j is set to $1/j$. To compute rank r_j, we sample negative labels until we find a violation, i.e. $(1 - \mathbf{w}_j^\top \mathbf{x}_i + \mathbf{w}_k^\top \mathbf{x}_i) \geq 0$. Then, the rank is calculated according to the following formulation

$$r_j = \left\lfloor \frac{m-1}{q} \right\rfloor, \quad (3)$$

where q is the number of sampling trials. It should be noted that the function $L(.)$ controls the ranking of labels during optimization. More precisely, $L(.)$ function assigns a small weight to the loss if a positive label is ranked at the top in the label list, whereas it assigns a large weight to push the positive label to the top if the positive label is not ranked at the top of the list.

Since there is no bound on the hinge loss, it produces very large losses for incorrect labels far from the margin. As a result, these mislabeled samples play a dominant role in determining the weight matrix, which deteriorates the overall classification performance. Alternatively, using ramp loss instead of hinge loss is more advantageous: When ramp loss is used, a mislabeled image sample can introduce a limited amount of loss, regardless of its position with respect to the margin. Therefore, the mislabeled samples cannot dominate the optimization. Similarly, in the case of missing labels, the score of a missing class can be higher than a positive labeled class. A good learning algorithm should not penalize such cases heavily as they are not actual mistakes. As in the previous case, using ramp loss is better suited for such cases, because it restricts extreme penalties.

Superiority of ramp loss over hinge loss is also well-proved and demonstrated in the literature for both supervised and transductive learning [1,7], but hinge loss became more popular as it is convex, whereas ramp loss is not. However, Plessis et al. [4] demonstrate the necessity of using the non-convex ramp loss instead of the convex hinge loss when learning from positive and unlabeled data both theoretically and empirically. Therefore, the use of non-convex functions for multi-labeled semi-supervised classification tasks is essential. Lastly, it should be noted that the second term and the last term of the optimization problem (Eq. 1) work in a complementary way, in the sense that they both aim to maximize the margin between the positive and negative classes. Besides, adding the last term improves the results especially on classification problems in which the images have a single positive class.

To solve the described optimization problem, we use the stochastic gradient (SG) method. The proposed method is fast and it scales well with large scale data, as it was proved that run time of SG does not depend directly on the size of the training set for these types of quadratic loss functions [28].

2.2. Semi-Supervised Learning

Semi-supervised learning algorithms try to improve the generalization performance by using unlabeled data. Majority of the semi-supervised and transductive learning methods are built on the cluster assumption, which states that the decision boundary should lie in low-density regions to improve the generalization performance. Until now, almost all binary transductive learning algorithms used the hinge loss cost $H_1(|\mathbf{w}^\top \mathbf{x}_i + b|)$ and its variants with unlabeled data. This cost function pushes the decision boundaries to less dense areas. It must be noted that, to use this cost, the classifier must return positive scores for positive class samples and negative scores for negative class samples. WARP does not necessarily satisfy this condition, which is why we introduced the last term in our optimization function (Eq. 1). In our multi-label classification setting, we can enforce this loss as

$$\min_{\mathbf{W}} \frac{\lambda}{2} \text{trace}(\mathbf{W}^\top \mathbf{W}) + \sum_{i=1}^{l} \sum_{j=1}^{C^+_{\mathbf{x}_i}} \sum_{k=1}^{C^-_{\mathbf{x}_i}} L(r_j) R_s(\mathbf{w}_j^\top \mathbf{x}_i - \mathbf{w}_k^\top \mathbf{x}_i) + \kappa \sum_{i=1}^{l} \sum_{j=1}^{m} R_s(y_{ij}(\mathbf{w}_j^\top \mathbf{x}_i)) + \eta \sum_{i=l+1}^{l+u} \sum_{j=1}^{m} H_1(|\mathbf{w}_j^\top \mathbf{x}_i|), \quad (4)$$

where η is a user defined parameter that controls the weight of the loss associated to the unlabeled samples with respect to the labeled samples. The classifier trained by solving this optimization problem was shown to be more accurate experimentally. However, this improvement was not very significant even though the majority of the training data was unlabeled. Furthermore, we observed a slight decrease in the accuracy as the number of unlabeled data samples is increased. In contrast, the semi-supervised methods that use label propagation report significant improvements as the number of unlabeled data samples is increased [2, 27]. These methods use the assumption that the nearest neighbors in the feature space tend to have the same labels by transitivity, thus we can pass the labels of labeled samples to their nearest unlabeled neighbors. By following this idea, we found the nearest labeled sample of each unlabeled data sample, and computed a similarity score by using the heat kernel function

$$s_i = \exp(-d(\mathbf{x}_i, \tilde{\mathbf{x}}_i)/t), \quad i = l+1, \ldots, l+u, \quad (5)$$

where $d(\mathbf{x}_i, \tilde{\mathbf{x}}_i)$ is the Euclidean distance between the unlabeled data sample $\mathbf{x}_i$ and its labeled nearest neighbor $\tilde{\mathbf{x}}_i$. Note that this function is widely used for measuring the similarity between samples in graph-based methods, including spectral clustering. The heat kernel function returns similarity scores between 0 and 1, and scores closer to 1 indicate that the samples are very close to each other, whereas scores closer to 0 mean that the samples are far from each other. Using the heat kernel function for computing similarity is important, as there may be image samples among unlabeled data that do not come from any of the classes used for training. The similarities returned for those irrelevant image samples will be low, thus their effect to the learning process will be diminished. In the proposed semi-supervised learning method, we pass the labels of labeled data samples to their nearest unlabeled samples, and the similarity score is used for controlling how much we can rely on this label assignment. By setting $s_i = 1, i = 1, ..., l$, for labeled samples, our final semi-supervised learning objective function can be written as

$$\min_{\mathbf{W}} \frac{\lambda}{2} \text{trace}(\mathbf{W}^\top \mathbf{W}) + \sum_{i=1}^{l+u} \sum_{j=1}^{C^+_{\mathbf{x}_i}} \sum_{k=1}^{C^-_{\mathbf{x}_i}} s_i L(r_j) R_s(\mathbf{w}_j^\top \mathbf{x}_i - \mathbf{w}_k^\top \mathbf{x}_i) + \kappa \sum_{i=1}^{l+u} \sum_{j=1}^{m} s_i R_s(y_{ij}(\mathbf{w}_j^\top \mathbf{x}_i)). \quad (6)$$

Here, using a robust ramp loss is very important since there may be many mislabeled samples among the unlabeled data, which are labeled by using the label propagation procedure. In contrast, a non-robust hinge loss can be adversely affected by these mislabeled image samples.

Our proposed semi-supervised multi-label classifier uses the optimization problem given in (Eq. 6), and this optimization problem can be solved using the SG algorithm as in the supervised case. For hand-crafted features, we compute the similarities at the beginning of the algorithm and pass the labels to the unlabeled data. These assignments stay permanent until the end of the algorithm. However, while we are training deep neural nets, the visual features change over time, thus we update the similarities and label assignments as described in the next section.

2.3. Robust Semi-Supervised Multi-Label Deep Neural Networks

To train the proposed method for feature extraction and classification jointly, we integrate the proposed classifier to the deep convolutional neural networks and call the resulting method as Robust Multi-Label Convolutional Neural Network (RML-CNN). In order to achieve this, we omit the first regularization term of the optimization problem (Eq. 6), and optimize only for the remaining terms, as various regularization methods are already implemented in deep learning frameworks. For instance, Caffe [13] allows users to select L1 or L2-norm based regularizers controlled by the weight decay parameter. In the proposed method, we take the gradients of the second and the last term separately, as the former focuses on maximizing the difference between positive and negative class scores, and requires updating two weight vectors at the same time. As indicated earlier, the ramp loss can be written as the difference of two convex hinge losses, thus the gradient of the loss J for the second term with respect to $\mathbf{w}_j$ and $\mathbf{w}_k$ can be found by using

$\psi \equiv (1 - \mathbf{w}_j^\top \mathbf{x}_i + \mathbf{w}_k^\top \mathbf{x}_i > 0)\ \&\ (s - \mathbf{w}_j^\top \mathbf{x}_i + \mathbf{w}_k^\top \mathbf{x}_i < 0)$

$$\frac{\partial J}{\partial \mathbf{w}_j} = \begin{cases} \frac{-s_i L(r_j)\mathbf{x}_i}{l+u}, & \text{if } \psi \text{ is True} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

$$\frac{\partial J}{\partial \mathbf{w}_k} = \begin{cases} \frac{s_i L(r_j)\mathbf{x}_i}{l+u}, & \text{if } \psi \text{ is True} \\ 0, & \text{otherwise} \end{cases}$$

In a similar manner, the gradient of the last term with respect to $\mathbf{w}_j$ can be computed by using

$$\frac{\partial J}{\partial \mathbf{w}_j} = \begin{cases} \frac{-s_i \kappa \mathbf{x}_i}{l+u}, & \text{if } (1 - \mathbf{w}_j^\top \mathbf{x}_i > 0)\ \&\ (s - \mathbf{w}_j^\top \mathbf{x}_i < 0) \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

These gradients are then propagated back by using the chain rule. For the semi-supervised case, we update the similarity scores and the labels of unlabeled data at every 20 epochs, where each epoch sees all training data (both labeled and unlabeled). To initialize the scores and labels of the unlabeled data at the beginning of the algorithm, we use the CNN model trained only by using labeled data.

2.4. Setting Design Parameters and Implementation Details

There are two parameters of the proposed method that must be set by the user: the regularization parameter λ, and the weight κ of the ramp loss that enforces to have positive scores for positive class samples and negative scores for negative class samples, as seen in Eq. 6. For deep neural nets, the regularization parameter is commonly called as the weight decay. The common practice is to set the weight decay to 0.00001 for most deep neural network architectures, thus we tried values closer to this number. We obtained the best results for 0.0005 on validation data for semi-supervised learning experiments, whereas the best validation accuracy is obtained by 0.00005 for the supervised learning experiments. For κ, we set it to 5 for all experiments, and the results show that values between 8 and 2 produce similar and good results. Thus, we set κ to the mean value of these two values.

We used the ResNet-101 architecture [11], and trained it with the SGD algorithm. The mini-batch size is set to 96 in our experiments. The momentum parameter is set to 0.9 and we initialized the weights both randomly and by using the weights of a pre-trained network. For semi-supervised learning experiments, the learning rate is set to 0.001 for randomly initialized models, and 0.0001 for fine-tuned models. The learning rate is kept fixed for these experiments. For supervised learning experiments, the learning rate starts from 0.00005, and is divided by 10 at the 10K-th and 16K-th iterations. Semi-supervised learning experiments were conducted using 4 NVIDIA GTX 1080 (8 GB memory) GPUs, whereas supervised learning experiments were conducted using a Tesla K40 (12 GB memory) GPU. Lastly, we used data augmentation in all experiments.

3. Experiments

We tested the proposed methods[1] on four visual object classification datasets: CIFAR-10, STL-10, NUS-WIDE, and MS-COCO. CIFAR-10 and STL-10 are single-label datasets. We experimented on these to compare our semi-supervised learning method to the state-of-the art, as the recent semi-supervised methods are generally tested on these. NUS-WIDE and MS-COCO are multi-labeled image datasets that do not contain unlabeled data. As the network architecture, we used the recently proposed ResNet-101 [11], which performs very well for many classification tasks. To compare our cost functions to the others, we also implemented WARP in the same network.

To assess the performance, we used the classical classification accuracy for single-label datasets whereas we used precision, recall and F1 measures computed based on the top-3 highest ranked labels as in [9,40] for the multi-label datasets. In addition to these, we also report mean average precision (mAP) scores obtained from per-class average precisions for multi-label image classification experiments.

3.1. Experiments on Semi-Supervised Learning

3.1.1 CIFAR-10 Dataset

The CIFAR-10 dataset includes 60K 32$\times$32 images of 10 objects: *airplane, automobile, bird, cat, deer, dog, frog, horse, ship* and *truck*. 50K samples are used as training data and they are split into 5 batches, whereas the remaining 10K samples are used for testing. We randomly selected 4000 labeled samples from each batch as in [24], and the remaining 46K samples are used as unlabeled data. This procedure is repeated 5 times for each batch, and the final accuracies are the averages over these trials. We resized all images to 256$\times$256 to be able to use them with the ResNet-101 network. A mini-batch size of 96 was used. We conducted two sets of experiments. For the first set of experiments, we start the training from scratch by random initialization for all tested cost functions. In the second case, we apply fine-tuning by using the weights of a model trained on the ILSVRC 2013.

Results are given in Table 1. It should be noted that WARP, softmax, and the proposed methods RML-CNN and SS-RML-CNN are all integrated into the same ResNet-101 network, and therefore they are directly comparable. The other methods given in the table use different type of network structures, but they are trained and tested by using the same experimental settings we have used here. The best accuracy is achieved by the proposed semi-supervised method

[1]Our code is available at https://github.com/bbenligiray/rml-cnn.

Table 1: Average classification accuracies on the CIFAR-10 dataset.

Method	Accuracy (%)
RML-CNN (ILSVRC init.)	89.24 ± 0.3
SS-RML-CNN (ILSVRC init.)	$\mathbf{91.46} \pm 0.2$
WARP (ILSVRC init.)	85.41 ± 0.4
Softmax (ILSVRC init.)	89.32 ± 0.4
RML-CNN (random init.)	52.37 ± 2.4
SS-RML-CNN (random init.)	58.86 ± 1.7
WARP (random init.)	50.52 ± 0.5
Softmax (random init.)	51.46 ± 1.1
Haeusser et al. [10]	$--$
DGL [31]	82.44 ± 0.3
Improved GAN [26]	81.37 ± 2.3
ALI [5]	82.01 ± 1.6
Ladder Network [24]	79.60 ± 0.5

using ILSVRC initialization, which is followed by the networks using softmax and RML-CNN. The proposed method significantly outperforms all other recent state-of-the-art methods, as well as WARP. The accuracies obtained by random initialization are quite low for all tested losses since there are very few examples for such a big deep network structure. Using unlabeled data significantly improves the results over supervised RML-CNN. The improvement is especially very high for the deep networks using random initialization. More precisely, the proposed semi-supervised method introduces approximately 6.5% improvement over the supervised one. WARP does not perform well compared to softmax, as the dataset is composed of single-label examples. However, the proposed supervised method produces comparable or better results than softmax.

3.1.2 STL-10 Dataset

The STL-10 dataset also includes images from 10 classes as in CIFAR-10. However, the resolution of images is higher (96×96), and the dataset is especially developed for unsupervised and semi-supervised learning. This dataset includes 5K labeled samples and 100K unlabeled samples. Unlabeled data includes images from the 10 classes, as well as some additional classes that are not present in the labeled set. In contrast, the unlabeled data from the CIFAR-10 dataset only include images from the 10 classes. Some images from the STL-10 dataset are illustrated in Fig. 2. This dataset is split into 10 pre-defined folds, where each fold includes 100 labeled images per class. The test set includes 8K images. We used 1000 labeled samples provided in each fold and all unlabeled data for learning and repeat this procedure for each fold. The final accuracy is again the averages of accuracies obtained from 10 folds.

Figure 2: Example images from the STL-10 dataset. The images shown in the first five rows are from the 10 classes, whereas the last row shows unlabeled examples that do not belong to the 10 classes in the labeled data.

Table 2: Average classification accuracies on the STL-10 dataset.

Method	Accuracy (%)
RML-CNN (ILSVRC init.)	91.11 ± 0.3
SS-RML-CNN (ILSVRC init.)	$\mathbf{94.1} \pm 0.2$
WARP (ILSVRC init.)	85.63 ± 0.6
Softmax (ILSVRC init.)	91.34 ± 0.3
RML-CNN (random init.)	40.00 ± 0.9
SS-RML-CNN (random init.)	44.25 ± 0.8
WARP (random init.)	39.15 ± 0.7
Softmax (random init.)	39.27 ± 1.7
Haeusser et al. [10]	$81.00 \pm --$
Huang et al. [12]	76.80 ± 0.3

Accuracies are presented in Table 2. The results are similar to the results of the CIFAR-10 dataset, in the sense that the proposed semi-supervised method again achieves the best accuracy, followed by softmax and RML-CNN. ILSVRC initializations again produce much higher accuracies compared to random initialization for all tested methods. WARP performs worse than both softmax and RML-CNN for ILSVRC initialization, but its accuracy is similar to these methods for random initialization. Our reported results are based on 10 repetitions, but Haeusser et al. [10] performs only 1 experiment. It should be noted that the pro-

Table 3: Accuracies on the NUS-WIDE dataset on 81 classes for top-3 highest ranked labels.

Method	P-C	R-C	F1-C	P-O	R-O	F1-O	mAP
RML-CNN	44.3	54.8	45.3	55.7	69.0	61.7	57.4
WARP	40.2	51.8	42.2	53.9	66.7	59.7	48.5
Softmax	31.7	31.2	31.4	47.8	59.5	53.0	–
WARP [9]	31.7	35.6	33.5	48.6	60.5	53.9	–
CNN+RNN [32]	40.5	30.4	34.7	49.9	61.7	55.2	–
RLSD [37]	44.4	49.6	46.9	54.4	67.6	60.3	54.1
ResNet-107 (binary relevance) [40]	46.7	**56.8**	46.9	**55.9**	**69.2**	**61.8**	59.5
ResNet-101 (binary relevance) [40]	46.4	55.3	47.0	**55.9**	**69.2**	**61.8**	**60.1**
CNN+RMLC (ours)	**48.2**	56.0	**48.8**	**55.9**	69.0	**61.8**	58.8

posed methods significantly outperform both the method of Haeusser et al. [10] and the previous state-of-the-art method of [12], which achieves 76.8% accuracy.

3.2. Experiments on Multi-Label Learning

3.2.1 NUS-WIDE Dataset

This dataset contains 269,648 images from Flickr that are annotated manually. There are 81 categories, with 2.4 category labels per image on the average. In addition to the objects classes, there are event/activity names among the labels such as *swimming* and *running*, as well as scene/location names such as *airport* or *ocean*. The officially provided train/test split is used in our experiments. More precisely we use 161,789 images for training and validation and 107,859 images for testing.

Results are given in Table 3. It should be noted that RML-CNN and WARP given at the top of the table are integrated into the same network again, thus they are directly comparable. The proposed method, RML-CNN, yielded lower results than the ResNet-101 architecture with binary relevance cost for multi-label learning [40]. We trained the same network with the provided code and obtained slightly lower results compared to the proposed method RML-CNN[2]. Therefore, we also conducted new experiments by using the CNN features of the trained model provided by the authors of [40]. In this setting, we treated the CNN features as hand-crafted features and then trained our classifier using the cost function (Eq. 1). The resulting classifier (CNN+RMLC) achieves the best P-C, F1-C, P-O, and F1-O scores. Especially, its F1-C score is 1.8% better compared to the results of [40]. Both of the proposed methods, RML-CNN and CNN+RMLC, significantly outperform WARP. It should be noted that RLSD method uses bounding box information for multi-label image classification, yet the proposed methods still outperform it with a small margin.

[2]We trained the network using the code and parameters provided by the authors of [40]. The authors of [40] use 4 GPUs with a batch size of 96 (i.e., 24 images per GPU, due to memory limitations). Since we had a single GPU, we collected gradients for 96 images in 4 mini-batches, and then made the updates. The difference may be due to this procedure.

3.2.2 MS-COCO Dataset

The Microsoft COCO (MS-COCO) dataset [23] is a large-scale multi-label benchmark dataset collected for several vision tasks such as recognition, segmentation, and captioning. The training set includes 82,783 images whereas the test set includes 40,504 images. The objects are categorized into 80 classes and there are approximately 2.95 object labels per image.

The accuracies based on the top-3 highest ranked labels are given in Table 4. As in the previous experiment, in addition to RML-CNN, we also conducted new experiments by using the CNN features of the ResNet-101 trained model provided by the authors of [40]. We again treated the CNN features as hand-crafted features and then trained our classifier using the cost function (Eq. 1). The best F1-C scores are achieved by the proposed CNN+RMLC and RSLD methods whereas the best mAP and R-O scores are obtained by the proposed CNN+RMLC and ResNet-101 method of [40]. The CNN+RNN achieves the best P-O and F1-O accuracies, and RSLD also obtains the best P-C accuracy. However, it should be kept in mind that RSLD needs bounding box annotations, which are not available for most visual recognition datasets.

3.3. Experiments with Noisy Data

To demonstrate the robustness of the proposed method, we designed an experiment where image labels are deliberately corrupted. In this experiment, we used 5 batches of 4000 labeled examples from the CIFAR-10 dataset, similar to the supervised experiments in Section 3.1. In the same vein, we subsampled the MS-COCO and NUS-WIDE datasets while maintaining the class frequencies, which allowed us to run many experiments across different parameters.

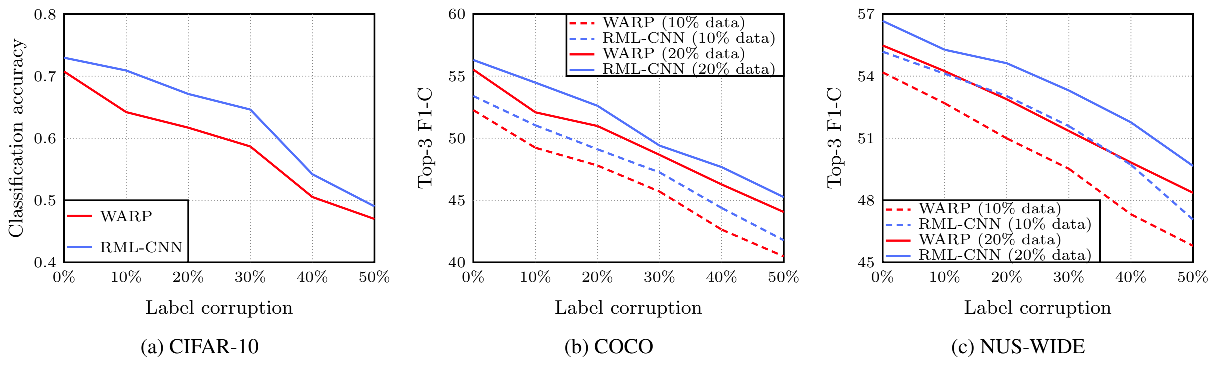

Figure 3: Classification performance on the CIFAR-10, MS-COCO and NUS-WIDE datasets with varying density of corrupted labels.

Table 4: Accuracies on the MS-COCO dataset on 80 classes for top-3 highest ranked labels.

Method	P-C	R-C	F1-C	P-O	R-O	F1-O	mAP
RML-CNN	62.4	61.1	59.8	62.8	65.4	64.1	71.5
WARP	60.7	59.8	58.5	61.5	64.1	62.8	63.2
WARP [9]	59.3	52.5	55.7	59.8	61.4	60.7	–
CNN+RNN [32]	66.0	55.6	60.4	**69.2**	66.4	**67.8**	–
RLSD [37]	**67.7**	56.4	**61.5**	70.5	59.9	64.8	67.4
ResNet-101 (binary relevance) [40]	65.3	**62.6**	61.3	64.1	**66.8**	65.4	**75.2**
CNN+RMLC (ours)	64.9	62.0	**61.5**	64.1	**66.8**	65.5	**75.2**

To corrupt the dataset, a subset is chosen randomly, and the corresponding labels are altered stochastically, while maintaining the class frequencies of the original dataset. The same subset of labels are gradually corrupted between 10% and 50% for all parameters to reach consistent results. The models are initialized with ImageNet pre-trained models, and the hyperparameters are optimized for the illustrated metrics.

The results are illustrated in Fig. 3. The most apparent result is that RML-CNN outperforms WARP across all datasets and parameters. The experiments with the MS-COCO and NUS-WIDE datasets are done with both 10% and 20% of the dataset, which gave consistent results. Both RML-CNN and WARP performances degrade linearly as the ratio of degraded labels increases. We can observe that using more data (i.e., 20%, instead of 10% of the dataset) provides a constant improvement across all parameters. Finally, it is surprising to see that both methods manage to converge even when half of the dataset is mislabeled.

4. Conclusion

This study argues that image labels in large-scale datasets are typically noisy and incomplete, and traditional methods using non-robust loss functions may fail to classify them accurately. For such cases, it is advantageous to use loss functions that are more robust against noisy and missing labels. In this paper, we proposed a robust semi-supervised method for multi-label image classification. To this end, we interchanged the classical non-robust hinge loss from the WARP method with the robust ramp loss, and added another loss term that encourages margin maximization. To incorporate the unlabeled image data in the learning process, we also used the label propagation, by assigning labels to the unlabeled data based on their closest labeled neighbors. Our proposed classifiers achieved state-of-the-art results in semi-supervised classification on CIFAR-10 and STL-10 datasets. The proposed classifiers also significantly outperformed WARP on noisy datasets. Moreover, we obtained comparable results to the state-of-the-art on multi-labeled NUS-WIDE and MS-COCO datasets for multi-label classification. The proposed classifiers are robust against noisy and missing image labels, but they do not explicitly model label dependencies. As a future work, we are planning to add another RNN network to the current CNN network, and build a complete end-to-end deep network structure that explicitly models such correlations between the labels.

Acknowledgments: The authors would like to thank NVIDIA for Tesla K40 GPU donation used in this study.

References

[1] H. Cevikalp, M. Elmas, and S. Ozkan. Towards category based large-scale image retrieval using transductive support vector machines. In *ECCV Workshops*, 2016. 3

[2] H. Cevikalp, J. Verbeek, F. Jurie, and A. Klaser. Semi-supervised dimensionality reduction using pairwise equivalence constraints. In *International Conference on Computer Vision Theory and Applications*, 2008. 4

[3] A. Clare and R. D. King. Knowledge discovery in multi-label phenotype data. In *Lecture Notes in Computer Science*, 2001. 2

[4] M. C. du Plessis, G. Niu, and M. Sugiyama. Analysis of learning from positive and unlabeled data. In *NIPS*, 2014. 3

[5] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. L. abd M. Arjovsky, and A. Courville. Adversarially learned inference. In *ICLR*, 2017. 2, 6

[6] A. Elisseeff and J. Weston. A kernel method for multi-labeled classification. In *NIPS*, 2001. 2

[7] S. Ertekin, L. Bottou, and C. Giles. Nonconvex online support vector machines. *IEEE Transactions on PAMI*, 33:368–381, 2011. 3

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 2

[9] Y. Gong, Y. Jia, T. Leung, A. Toshev, and S. Ioffe. Deep convolutional ranking for multilabel image annotation. In *arXiv:1312.4894*, 2014. 2, 3, 5, 7, 8

[10] P. Haeusser, A. Mordvintsev, and D. Cremers. Learning by association a versatile semi-supervised training method for neural networks. In *CVPR*, 2017. 2, 6, 7

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[12] C. Huang, C. C. Loy, and X. Tang. Unsupervised learning of discriminative attributes and visual representations. In *CVPR*, 2016. 6, 7

[13] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 4

[14] J. Jin and H. Nakayama. Annotation order matters: Recurrent image annotator for arbitrary length image tagging. In *ICPR*, 2016. 2

[15] A. Kanehira and T. Harada. Multi-label ranking from positive and unlabeled data. In *CVPR*, 2016. 2

[16] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015. 2

[17] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling. Semi-supervised learning with deep generative models. In *arXiv:1406.5298*, 2014. 2

[18] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep neural networks. In *NIPS*, 2012. 1

[19] M. Lapin, M. Hein, and B. Schiele. Top-k multiclass svm. In *NIPS*, 2015. 2

[20] M. Lapin, M. Hein, and B. Schiele. Analysis and optimization of loss functions for multiclass, top-k, and multilabel classification. In *arXiv:1612.03663*, 2016. 2

[21] M. Lapin, M. Hein, and B. Schiele. Loss functions for top k-error: analysis and insights. In *CVPR*, 2016. 2

[22] D.-H. Lee. The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshops*, 2013. 2

[23] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 7

[24] A. Rasmus, H. Valpoa, M. Honkala, M. Berglund, and T. Raiko. Semi-supervised learning with ladder networks. In *NIPS*, 2015. 2, 5, 6

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2015. 1

[26] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *arXiv:1606.03498*, 2016. 2, 6

[27] B. Scholkopf, D. Zhou, and J. Huang. Learning from labeled and unlabeled data on a directed graphs. In *ICML*, 2005. 4

[28] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated subgradient solver for svm. In *NIPS*, 2007. 3

[29] C. Sun, A. Shrivastava, S. Singh, and A. Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 2017. 2

[30] N. Usunier, D. Buffoni, and P. Gallinari. Ranking with ordered weighted pairwise classification. In *ICML*, 2009. 2

[31] G. Wang, X. Xie, J. Lai, and J. Zhuo. Deep growing learning. In *ICCV*, 2017. 2, 6

[32] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu. Cnn-rnn: A unified framework for multi-label image classification. In *CVPR*, 2016. 2, 7, 8

[33] Y. Wei, W. Xia, M. Lin, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan. Hcp: A flexible cnn framework for multi-label image classification. *IEEE Transactions on PAMI*, 39:1901–1907, 2016. 2

[34] J. Weston, S. Bengio, and N. Usunier. Wsabie: Scaling up to large vocabulary image annotation. In *IJCAI*, 2011. 2, 3

[35] J. Weston, F. Ratle, and R. Collobert. Deep learning via semi-supervised embedding. In *ICML*, 2008. 2

[36] F. Wu, Z. Wang, Z. Zhang, Y. Yang, J. Luo, W. Zhu, and Y. Zhuang. Weakly semi-supervised deep learning for multi-label image annotation. *IEEE Transactions on Big Data*, 1:109–122, 2015. 2, 3

[37] J. Zhang, Q. Wu, C. Shen, J. Zhang, and J. Lu. Multi-label image classification with regional latent semantic dependencies. In *arXiv:1612.01082*, 2016. 2, 7, 8

[38] M.-L. Zhang and Z. Zhou. Ml-knn: a lazy learning approach to multi-label learning. *Pattern Recognition*, 40:2038–2048, 2007. 2

[39] M.-L. Zhang and Z.-H. Zhou. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26:1819–1837, 2014. 2

[40] F. Zhu, H. Li, W. Ouyang, N. Yu, and X. Wang. Learning spatial regularization with image-level supervisions for multi-label image classification. In *CVPR*, 2017. 2, 5, 7, 8