Deep Discriminative CNN with Temporal Ensembling for Ambiguously-Labeled Image Classification

Yao Yao,[1] **Jiehui Deng,**[1] **Xiuhua Chen,**[1] **Chen Gong,**[1,*] **Jianxin Wu,**[3] **Jian Yang**[1,2,*]
[1]PCA Lab, the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, School of Computer Science and Engineering, Nanjing University of Science and Technology, China
[2]Jiangsu Key Lab of Image and Video Understanding for Social Security
[3]National Key Laboratory for Novel Software Technology, Nanjing University, China
{yaoyao, jhdeng, chenxh, chen.gong, csjyang}@njust.edu.cn, wujx2001@nju.edu.cn

Abstract

In this paper, we study the problem of image classification where training images are ambiguously annotated with multiple candidate labels, among which only one is correct but is not accessible during the training phase. Due to the adopted non-deep framework and improper disambiguation strategies, traditional approaches are usually short of the representation ability and discrimination ability, so their performances are still to be improved. To remedy these two shortcomings, this paper proposes a novel approach termed "Deep Discriminative CNN" (D^{2}CNN) with temporal ensembling. Specifically, to improve the representation ability, we innovatively employ the deep convolutional neural networks for ambiguously-labeled image classification, in which the well-known ResNet is adopted as our backbone. To enhance the discrimination ability, we design an entropy-based regularizer to maximize the margin between the potentially correct label and the unlikely ones of each image. In addition, we utilize the temporally assembled predictions of different epochs to guide the training process so that the latent groundtruth label can be confidently highlighted. This is much superior to the traditional disambiguation operations which treat all candidate labels equally and identify the hidden groundtruth label via some heuristic ways. Thorough experimental results on multiple datasets firmly demonstrate the effectiveness of our proposed D^{2}CNN when compared with other existing state-of-the-art approaches.

Introduction

Image classification is a fundamental computer vision problem which has been intensively studied in the past years. In the traditional setting, the unique groundtruth label of each training image should be available when the classifiers are trained. Unfortunately, in many real-world situations, the images may lack clear labels and manually labeling them will incur unaffordable monetary or time cost. Instead of acquiring full and clear human annotations, ambiguously-labeled image classification deals with the images which are associated with multiple candidate labels, and only one of them is valid. For example, in automatic face naming (Guillaumin et al. 2008; Zeng et al. 2013; Chen et al. 2014), an image with faces is often associated with textual description, by which we can roughly know who appear in this image. However, the correspondence of the faces in the image and the names in the textual description is still unknown (see Figure 1 (a)). Another application is that in crowdsourcing area, the annotators with different levels of expertise may assign different labels (can be correct or incorrect) to the same image. Therefore, it is necessary to find out the latent groundtruth label of every annotated image (see Figure 1 (b)).

Barcelona news: Lionel Messi and Luis Suarez hold private talks over five player concerns

Annotator 1: Horse
Annotator 2: Donkey
Annotator 3: Mule

Figure 1: Applications of learning from ambiguously labeled images. (a) is a newsletter containing an image and the text caption. From the caption we know that Lionel Messi and Luis Suarez are in the image but we cannot figure out the concrete correspondence between the names and the faces. (b) shows an image of a mule for crowdsourcing. However, some annotators may mistakenly label it as a horse or a donkey due to their limited cognitive ability.

Learning from such ambiguously labeled examples is also related to "partial label learning" (Wu and Zhang 2018; Feng and An 2019a; Wang, Li, and Zhou 2019) or "superset label learning" (Liu and Dietterich 2012; 2014; Gong et al. 2018b). Formally speaking, let $\mathcal{X} \in \mathbb{R}^d$ denote the d-dimensional input space and $\mathcal{Y} = \{1, 2, ..., c\}$ denote the label space with c class labels. We denote the training set of the ambiguously labeled examples by $\mathcal{D} = \{(\mathbf{x}_i, S_i)|1 \leq i \leq n\}$, where $\mathbf{x}_i$ is the i-th input feature vector and S_i is the corresponding candidate label set of $\mathbf{x}_i$. More specifically, $S_i = A_i \cup \{\mathrm{y}_i\}$, where A_i is the set of false positive labels and y_i is the latent groundtruth label lying in S_i, which is

*Corresponding author.

not directly accessible during the training phase. Therefore, our target is to train a classifier $f : \mathcal{X} \to \mathcal{Y}$ from the training set $\mathcal{D}$ so that the correct predictions can be made on test examples. Note that ambiguously-labeled image classification differs from semi-supervised learning (Gong et al. 2015; 2016; 2018a) in that each training example in the investigated task is labeled, while massive examples in semi-supervised learning are unlabeled; and it differs from corrupted label learning (Gong et al. 2017; Yi and Wu 2019) in that each example has multiple candidate labels but only one is correct, while every example in corrupted label learning has only one label which can be corrupted or correct.

Apparently, in this task, the label information of training examples is ambiguous and cannot be directly used by the traditional supervised classifiers. To solve this problem, the common strategy is to disambiguate the set of candidate labels of each example, and there are mainly two classes of methods for such disambiguation operation, namely average-based methods and identification-based methods.

Average-based methods treat all candidate labels equally by assuming that they make equal contributions to the trained classifiers and the predictions are made by averaging their model outputs. The work (Hüllermeier and Beringer 2006) straightforwardly generalizes the k-nearest neighbor classification to resolve the ambiguous labeling problem by predicting the label of an example via the voting strategy among the candidate labels of its neighbors. Zhang *et al.* (Zhang and Yu 2015) also propose an approach where the predictions of unseen examples are made by the weighted averaging over the candidate labels of their neighbors. Average-based methods are intuitive and are easy to implement. However, these methods share a critical shortcoming that the outputs from false positive labels may overwhelm the groundtruth labels' outputs, which will severely degrade their performances.

Identification-based methods regard the unique groundtruth label as a latent variable and gradually identify it by iterative procedures. Maximum likelihood criterion and maximum margin criterion are the two most widely-used learning strategies to identify groundtruth labels. Based on EM procedure, these methods (Jin and Ghahramani 2003; Liu and Dietterich 2012) train their models by optimizing the maximum likelihood function. Nguyen *et al.* (Nguyen and Caruana 2008) maximize the margin between the outputs from candidate labels and non-candidate labels to refine groundtruth labels. One potential shortcoming of identification-based methods is that the identified label in the current iteration may turn out to be false positive and they can hardly be rectified in the subsequent iterations.

Apart from the individual shortcomings inherited by above-mentioned methods, all existing approaches are non-deep, which means that they only work on the handcrafted features and thus the performances are far from perfect in many cases. In a word, existing methods are usually short of representation ability and discrimination ability, where the former is caused by the shallow learning frameworks, and the latter is due to the imperfect disambiguation techniques. To address these shortcomings, in this paper, we propose a novel classifier to handle the ambiguously-labeled images which is named as "**D**eep **D**iscriminative **CNN** with temporal ensembling" ("D^{2}CNN" for short). Specifically, to enhance the representation ability, we employ the Deep Convolutional Neural Networks (DCNN) as the backbone of our algorithm as it is capable to learn representations of data with multiple levels of abstraction, which is much superior to the handcrafted features. To our best knowledge, this is the first work to employ DCNN for the ambiguously-labeled image classification problem. In order to improve the model's discrimination ability, two strategies are explicitly developed. Firstly, we devise a novel entropy minimization regularizer on the model predictions which can highlight the potential groundtruth label and meanwhile suppress the unlikely labels in the candidate label set. Secondly, inspired by the temporal ensembling technique which is utilized by semi-supervised learning (Laine and Aila 2016), we assemble the model outputs of the different epochs and regard them as additional supervision information for the next epoch. By assembling the predictions of different stages during training, our model is able to obtain accurate confidence levels of labels. Therefore, our method can automatically decide which candidate label is reliable and is likely to be the groundtruth. As a result, the model's discrimination ability is further strengthened. Intensive experiments on four datasets substantiate the superiority of our proposed D^{2}CNN to the state-of-the-art methodologies.

The Proposed D^{2}CNN Approach

This section presents our proposed D^{2}CNN approach. We denote $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ as the training image set with each column $\mathbf{x}_i$ ($i = 1, 2, \cdots, n$) representing the i-th image and n denoting the total number of training images. Besides, by denoting $\mathbf{y}_i$ as a c-dimensional column vector which records the candidate labels of $\mathbf{x}_i$, the ambiguous label matrix associated with $\mathbf{X}$ can be formed as $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n]^\top \in \{0, 1\}^{n \times c}$, where c is the number of the classes and the (i, j)-th element $\mathrm{y}_{ij} = 1$ means that the j-th label is a candidate label of the image $\mathbf{x}_i$, otherwise $\mathrm{y}_{ij} = 0$.

Network Establishment

As mentioned in the introduction, this is the first innovative work which formulates ambiguously-labeled image classification into the framework of DCNN. A typical DCNN is normally composed of a series of layers, such as convolutional layers, pooling layers, and fully connected layers. The previous layers in the network capture the lower-level features such as edges or corners while the deeper layers capture the higher-level features by composing the lower-level ones. As a result, DCNN is able to learn representations of data with multiple levels of abstraction, so it usually brings about better results when compared with the non-deep methods, and that is the reason why we establish our model based on DCNN. Specifically, ResNet (He et al. 2016) is employed by us as it has achieved very impressive performances in various computer vision tasks.

Loss Function Design

Cross-entropy loss is widely used in conventional supervised classification tasks. However, when dealing with am-

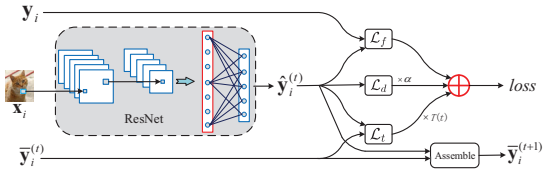

Figure 2: The pipeline of our approach. $\mathbf{y}_i$, $\hat{\mathbf{y}}_i^{(t)}$, and $\bar{\mathbf{y}}_i^{(t)}$ denote the original ambiguous label vector, predicted label vector, and the training target at the t-th epoch, respectively. $\mathcal{L}_f$, $\mathcal{L}_d$, and $\mathcal{L}_t$ represent the fidelity term, discrimination term, and temporal ensembling term correspondingly.

biguously labeled examples, we are only accessible to the candidate labels of each image. Directly minimizing the cross entropy between the original ambiguous labels and the predicted labels means that all candidate labels are treated equally and thus can be viewed as an average-based disambiguation method, of which the shortcomings have been analyzed above. Therefore, the cross-entropy loss cannot be directly utilized in our task and we further conduct experiments to demonstrate this.

To disambiguate the candidate labels effectively and train the network simultaneously, we propose a loss function for our D^{2}CNN algorithm which is defined by

$$Loss = \mathcal{L}_f(\mathbf{Y}, \hat{\mathbf{Y}}) + \alpha \mathcal{L}_d(\hat{\mathbf{Y}}) + T(t) \cdot \mathcal{L}_t(\bar{\mathbf{Y}}, \hat{\mathbf{Y}}). \quad (1)$$

In the right-hand side of equation, the first term is called *fidelity term* which computes the loss between the network prediction $\hat{\mathbf{Y}}$ and the ambiguous labels $\mathbf{Y}$. Here $\hat{\mathbf{Y}}$ denotes the output of the final c-class softmax layer of ResNet and shares the same definition with $\mathbf{Y}$ which is formatted as $\hat{\mathbf{Y}} = [\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_n]^\top \in [0,1]^{n \times c}$. Note that every element in matrix $\hat{\mathbf{Y}}$ is nonnegative and the sum of each row in $\hat{\mathbf{Y}}$ is equal to one, that is, for arbitrary i and j, we have $\hat{y}_{ij} \geq 0$ and $\sum_{j=1}^{c} \hat{y}_{ij} = 1$, therefore $\hat{y}_{ij}$ can be understood as the probability of the image $\mathbf{x}_i$ belonging to the j-th class. The second term is named *discrimination term* which drives the predictions to be discriminative by controlling the entropy of $\hat{\mathbf{Y}}$. The third term is dubbed *temporal ensembling term* which enables a consensus between the prediction of current training stage $\hat{\mathbf{Y}}$ and the outputs of the network-in-training on different epochs $\bar{\mathbf{Y}}$. Here $\bar{\mathbf{Y}} = [\bar{\mathbf{y}}_1, \ldots, \bar{\mathbf{y}}_n]^\top \in [0,1]^{n \times c}$ is the training target matrix which is generated by the assembled predictions and will be updated at every epoch. Besides, α is a trade-off parameter and $T(t)$ is a time-dependent weighting function. The structure of our proposed approach is shown in Figure 2. In the following paragraphs, we will describe the definitions of $\mathcal{L}_f$, $\mathcal{L}_d$, $\mathcal{L}_t$, and $T(t)$ in detail.

Fidelity Term $\mathcal{L}_f$: Although the groundtruth label of an image is not accessible, we can explicitly know that the labels that are outside the candidate label set S_i are certainly not the groundtruth label. In other words, the corresponding elements in the predicted label vector should be zeros. Therefore, the formulation of $\mathcal{L}_f$ is defined as:

$$\mathcal{L}_f(\mathbf{Y}, \hat{\mathbf{Y}}) = -\frac{1}{n} \sum_{i=1}^{n} (\mathbf{1} - \mathbf{y}_i)^\top \log(\mathbf{1} - \hat{\mathbf{y}}_i), \quad (2)$$

where $\mathbf{1}$ represents the all-one vector. By minimizing $\mathcal{L}_f$, the probability that the labels in non-candidate label set being the groundtruth label is constrained to zero. As a result, the predicted label can only come from the candidate label set.

Discrimination Term $\mathcal{L}_d$: To make the predictions more discriminative, we introduce the discrimination term $\mathcal{L}_d$ to make the potential groundtruth label become prominent among all labels. Considering that entropy can be used to measure the label uncertainty and it will decrease along with the lessening of the uncertainty (see Figure 3 (a)), we propose to minimize the entropy of each label vector $\hat{\mathbf{y}}_i$. As a result, the specific formulation of $\mathcal{L}_d$ is:

$$\mathcal{L}_d(\hat{\mathbf{Y}}) = -\frac{1}{n} \sum_{i=1}^{n} \hat{\mathbf{y}}_i^\top \log \hat{\mathbf{y}}_i. \quad (3)$$

Note that due to constraints $\sum_{j=1}^{c} \hat{y}_{ij} = 1$ and $\hat{y}_{ij} \geq 0$, minimizing the entropy of $\hat{\mathbf{y}}_i$ will widen the gap of values between possible labels and unlikely labels, which is beneficial to generate a discriminative prediction $\hat{\mathbf{y}}_i$ for $\mathbf{x}_i$.

Suppose we are dealing with binary classification problem, *i.e.* $c = 2$, and the candidate label set of an image $\mathbf{x}_i$ contains the labels 1 and 2, which means that both of these two labels could be the groundtruth label of $\mathbf{x}_i$. We denote $\hat{\mathbf{y}}_i = [\hat{y}_{i1}, \hat{y}_{i2}]^\top$ as the predicted label vector of $\mathbf{x}_i$. When the predicted label vector $\hat{\mathbf{y}}_i = [0.5, 0.5]^\top$, which means that the probability values that the image $\mathbf{x}_i$ belongs to class 1 and class 2 are equivalent, we consider that it is unacceptable because we cannot figure out the groundtruth label of $\mathbf{x}_i$ from $\hat{\mathbf{y}}_i$. In other words, the uncertainty of this prediction is too large and the corresponding entropy reaches the maximum value (see the point A in Figure 3 (b)). In contrast, if the predicted label vector $\hat{\mathbf{y}}_i = [0.8, 0.2]^\top$, we can clearly tell that the label 1 is very likely to be the groundtruth label of $\mathbf{x}_i$. In this case, the prediction is less uncertain and its entropy is relatively small (see the point B in Figure 3 (b)). To sum up, minimizing the entropy of label predictions will decrease the label uncertainty and make the obtained label predictions more discriminative.

Temporal Ensembling Term $\mathcal{L}_t$: Inspired by the temporal ensembling technique that has been applied to semi-supervised learning (Laine and Aila 2016), we assemble the model predictions of different epochs and regard them as the auxiliary supervision information for the next epoch. The training targets (*i.e.* $\bar{\mathbf{y}}_i, i = 1, 2, \cdots, n$) generated by the assembled predictions are likely to be closer to the correct labels when compared with the current predictions, and thus they can be used to guide the training of the network. Furthermore, by taking the reliability w_i of each training target $\bar{\mathbf{y}}_i$ into consideration, the formulation of $\mathcal{L}_t$ is defined as:

$$\mathcal{L}_t(\bar{\mathbf{Y}}, \hat{\mathbf{Y}}) = -\frac{1}{n} \sum_{i=1}^{n} w_i \cdot \bar{\mathbf{y}}_i^\top \log \hat{\mathbf{y}}_i, \quad (4)$$

where $\bar{\mathbf{Y}}$ denotes the training target matrix which is generated by the assembled predictions and $w_i \in [0,1]$ represents the weight of $\bar{\mathbf{y}}_i$ which will be detailed later. Besides, we use the Exponential Moving Average (EMA) of predictions in every epoch to generate the assembled predictions,

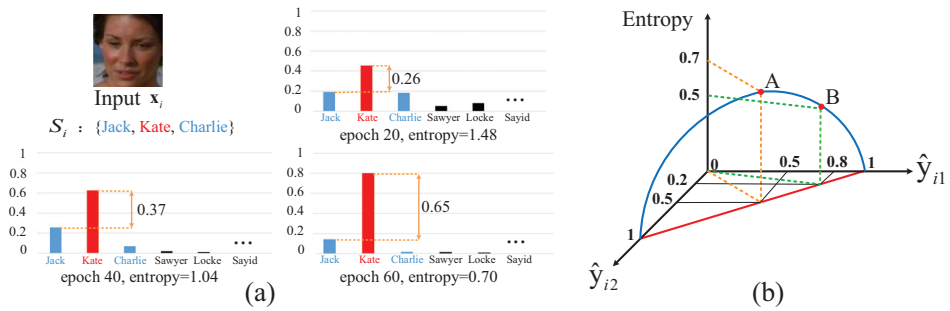

Figure 3: (a) shows a face image in *Lost* dataset, whose candidate label set contains "Jack", "Kate", and "Charlie", among which "Kate" is the groundtruth label. When the training goes on, the margin between the largest label output and the second largest label output in label vector increases, which indicates that the uncertainty of the label vector decreases. Therefore, the entropy of label vector declines too. (b) shows the curve depicting the relationship between the predicted label vector $\hat{\mathbf{y}}_i = [\hat{y}_{i1}, \hat{y}_{i2}]^\top$ and its entropy. Note that the red line is the feasible region of $\hat{\mathbf{y}}_i$ and the blue curve record the entropy of each predicted label vector lying on the red line. When $\hat{\mathbf{y}}_i = [0.5, 0.5]^\top$, the entropy reaches the maximum value and this is the most ambiguous prediction vector. Therefore, minimizing the entropy of $\hat{\mathbf{y}}_i$ will enforce it to approach to the discriminative prediction vectors such as $[1, 0]^\top$ or $[0, 1]^\top$.

which usually have higher quality than the current predictions. Consequently, the assembled predictions $\mathbf{S}$ and the training target $\bar{\mathbf{Y}}$ are updated as:

$$\mathbf{S}^{(t)} = \gamma \mathbf{S}^{(t-1)} + (1-\gamma)\hat{\mathbf{Y}}^{(t)}, \tag{5}$$

$$\bar{\mathbf{Y}}^{(t+1)} = \mathbf{S}^{(t)}/(1-\gamma^t), \tag{6}$$

where $\mathbf{S}^{(t-1)}$ and $\mathbf{S}^{(t)}$ denote the assembled predictions at the $(t-1)$-th epoch and the t-th epoch, respectively. The coefficient $\gamma \in [0, 1]$ is a momentum term that controls how far the ensemble reaches into training history, and the expression γ^t denotes the γ to the power of t. As expressed in Equations (5) and (6), EMA assigns greater weights to the recent network predictions by exponentially decreasing the weights of the early predictions. This is reasonable as the recent predictions are more reliable when compared with the earlier predictions as the training process proceeds. To rectify the startup bias in $\mathbf{S}^{(t)}$, we divide $\mathbf{S}^{(t)}$ by the factor $(1-\gamma^t)$ and then we can obtain the training target for the next epoch, *i.e.* $\bar{\mathbf{Y}}^{(t+1)}$. As no prediction is assembled before the first epoch, $\mathbf{S}$ and $\bar{\mathbf{Y}}$ are both initialized to zeros.

As mentioned above, w_i decides to what degree the ensembling result $\bar{\mathbf{y}}_i$ should be trust to guide the training process. That is to say, if $\bar{\mathbf{y}}_i$ is very likely to be the groundtruth label vector of $\mathbf{x}_i$, w_i should be relatively large, which means that the model will pay more attention to $\mathbf{x}_i$. Otherwise, w_i should be small. Inspired by this work (Nguyen and Caruana 2008), we employ the predictive margin to measure whether the training target $\bar{\mathbf{y}}_i$ is reliable. More precisely, we denote $m_i \in [-1, 1]$ as the margin between the maximum value in $\bar{\mathbf{y}}_i$ from candidate labels and that from non-candidate labels, namely $m_i = \max_{y_j \in S_i} \bar{y}_{ij} - \max_{y_k \notin S_i} \bar{y}_{ik}$. Then, the weight w_i can be defined as:

$$w_i = \begin{cases} 0 & m_i \leq 0 \\ {m_i}^2 & otherwise \end{cases}. \tag{7}$$

Equation (7) indicates that when the margin is negative, which means that the maximum value in training target $\bar{\mathbf{y}}_i$ does not come from $\mathbf{x}_i$'s candidate label set, then the weight of $\bar{\mathbf{y}}_i$ is zero (*i.e.* $w_i = 0$). This is reasonable as in this case the training target $\bar{\mathbf{y}}_i$ is completely wrong. Therefore, the model will not learn from the unreliable training target $\bar{\mathbf{y}}_i$. If the margin is positive, the weight of $\bar{\mathbf{y}}_i$ is proportional to the square of the margin. That is to say, if the training target is more likely to be correct, the model will pay more attention to it. In general, measuring the reliability of the each training target helps the model to decide to what degree they can be relied on, and thus is beneficial for the model to precisely disambiguate the ambiguously labeled images.

Formation of $T(t)$: In our proposed loss function (*i.e.* Equation (1)), $T(t)$ is a time-dependent weighting function. Roughly speaking, $T(t)$ increases with the number of epochs. At the beginning of the training phase, the network is far from well-trained and the corresponding predictions are not accurate. As a result, the assembled predictions are unreliable and the training targets generated by them can only provide very limited useful supervision information. As the training goes on, the disambiguation ability of the network is strengthened gradually and the predictions turn out to be more precise. Therefore, the training targets generated by the assembled predictions are more likely to be accurate and can better guide the training of the network. More precisely, we set $T(t) = T_{max} \exp\left[-5(1-t)^2\right]$ in our implementation, where T_{max} denotes the maximum value that $T(t)$ could reach and t increases linearly from zero to one during the rising phase. In summary, the network mainly learns from the original ambiguous labels at the initial training phase and gradually learns from the assembled predictions when the training process proceeds.

Optimization

The network parameter can be trained via back propagation algorithm. In our implementation, Adam (Kingma and Ba 2014) is adopted to optimize the network parameter Θ.

Given a training example $\mathbf{x}_i$ and the corresponding net-

work's output $\hat{\mathbf{y}}_i = [\hat{y}_{i1}, \ldots, \hat{y}_{ij}, \ldots, \hat{y}_{ic}]^\top$, the gradient at the timestep t, *i.e.* $g^{(t)}$, can be computed as follows according to the chain rule, namely

$$g^{(t)} = \frac{\partial Loss}{\partial \Theta} = (\frac{\partial \mathcal{L}_f}{\partial \hat{\mathbf{y}}_i} + \alpha \frac{\partial \mathcal{L}_d}{\partial \hat{\mathbf{y}}_i} + T(t) \frac{\partial \mathcal{L}_t}{\partial \hat{\mathbf{y}}_i}) \frac{\partial \hat{\mathbf{y}}_i}{\partial \Theta}, \quad (8)$$

where

$$\frac{\partial \mathcal{L}_f}{\partial \hat{\mathbf{y}}_i} = \frac{1}{n} \cdot [\frac{1 - \mathrm{y}_{i1}}{1 - \hat{\mathrm{y}}_{i1}}, \ldots, \frac{1 - \mathrm{y}_{ij}}{1 - \hat{\mathrm{y}}_{ij}}, \ldots, \frac{1 - \mathrm{y}_{ic}}{1 - \hat{\mathrm{y}}_{ic}}]^\top, \quad (9)$$

$$\frac{\partial \mathcal{L}_d}{\partial \hat{\mathbf{y}}_i} = -\frac{1}{n} \cdot [\log \hat{\mathrm{y}}_{i1} + 1, \ldots, \log \hat{\mathrm{y}}_{ij} + 1, \ldots, \log \hat{\mathrm{y}}_{ic} + 1]^\top, \quad (10)$$

$$\frac{\partial \mathcal{L}_t}{\partial \hat{\mathbf{y}}_i} = -\frac{w_i}{n} \cdot [\frac{\bar{\mathrm{y}}_{i1}}{\hat{\mathrm{y}}_{i1}}, \ldots, \frac{\bar{\mathrm{y}}_{ij}}{\hat{\mathrm{y}}_{ij}}, \ldots, \frac{\bar{\mathrm{y}}_{ic}}{\hat{\mathrm{y}}_{ic}}]^\top. \quad (11)$$

Given above equations, Θ can be updated as

$$\Theta^{(t)} = \Theta^{(t-1)} - \tau \cdot \hat{m}^{(t)} / (\sqrt{\hat{v}^{(t)}} + \epsilon), \quad (12)$$

where τ denotes the learning rate and ϵ is added to avoid the situation when the denominator is zero. The variables $\hat{m}^{(t)}$ and $\hat{v}^{(t)}$ are the bias-corrected estimates of the first moment estimate (*i.e.* $m^{(t)}$) and the second raw estimate (*i.e.* $v^{(t)}$) of the gradients correspondingly, and they are updated as

$$m^{(t)} = \beta_1 \cdot m^{(t-1)} + (1 - \beta_1) \cdot g^{(t)}, \quad (13)$$

$$\hat{m}^{(t)} = m^{(t)} / (1 - \beta_1^t), \quad (14)$$

and

$$v^{(t)} = \beta_2 \cdot v^{(t-1)} + (1 - \beta_2) \cdot (g^{(t)} \odot g^{(t)}), \quad (15)$$

$$\hat{v}^{(t)} = v^{(t)} / (1 - \beta_2^t), \quad (16)$$

where β_1 and β_2 are hyperparameters indicating the exponential decay rates for the moment estimates, and "$\odot$" denotes the elementwise product.

Experiments

In this section, we evaluate the performance of the proposed D^{2}CNN approach on two synthesized datasets and two real-world datasets. The compared algorithms dealing with ambiguously labeled examples include PLKNN (Hüllermeier and Beringer 2006), RegISL (Gong et al. 2018b), M3PL (Yu and Zhang 2016), IPAL (Zhang and Yu 2015), SURE (Feng and An 2019b), MCar (Chen, Patel, and Chellappa 2018), and WMCar-ICE (Chen, Patel, and Chellappa 2018). Conventional DCNN with cross-entropy loss is also incorporated as a baseline. Furthermore, the robustness and the effectiveness of the main contributions of the proposed D^{2}CNN approach are also empirically validated.

Implementation Details

For our D^{2}CNN approach, we adopt a relatively shallow network ResNet-20 (He et al. 2016) as backbone due to the small size of the input images and the limited training examples in some studied datasets. In the preprocessing step, we perform whitening and data augmentation by horizontal random flip and random crops for all training images in the adopted datasets. Adam (Kingma and Ba 2014) is utilized to optimize the networks with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Besides, we employ the weight decay of 0.0001, minibatch size of 100, and ensembling weighting degree $\gamma = 0.6$ for all experiments. We train the networks for 300 epochs and enable $T(t)$ to reach T_{max} after 200 epochs in all runs.

Figure 4: Example images from the adopted datasets. (a) *Fashion-Mnist* dataset; (b) *SVHN* dataset; (c) *Lost* dataset; (d) *Yahoo!News* dataset.

Experiments on Synthesized Datasets

This section presents the experiments on *Fashion-Mnist* dataset and *SVHN* dataset with manually added ambiguous labels.

The *Fashion-Mnist* dataset comprises of 70,000 fashion products from 10 categories with 7,000 grayscale images per category, and the size of each image is 28×28. The *SVHN* dataset contains 99,289 images of digits belonging to 10 categories and all images have been resized to a fixed resolution of 32×32. On both datasets, apart from the original groundtruth label, we also randomly choose some false positive labels for each image which are different from its groundtruth label to compose the candidate label set. Therefore, the datasets with suffix "-v1" means that one extra noisy label is added to the candidate label set for every contained image, and the suffix "-v3" means that another three incorrect labels are incorporated to the candidate label set. Table 2 shows the characteristics of the adopted datasets where "*Avg # labels*" indicates the average number of candidate labels for a single image. Figure 4 shows the example images of the adopted datasets.

For non-deep baseline approaches including PLKNN, RegISL, M3PL, IPAL, SURE, MCar and WMCar-ICE, we extract the 512-dimensional GIST (Oliva and Torralba 2001) feature to represent the images. For the conventional DCNN baseline, we also use ResNet-20 for feature extraction as the backbone network of our proposed D^{2}CNN. Specifically, the regularization parameter C_{max} in M3PL is set to 0.01 via cross validation. In PLKNN, IPAL, and RegISL, the number of nearest numbers k is chosen from the set {5,10,15,20}. For MCar and WMCar-ICE, we fix λ as $1/\sqrt{\max(c+d, n)}$ where c, d, and n respectively denote the number of classes, the dimension of the input features, and the number of the training examples as suggested. As for the proposed D^{2}CNN approach, we employ the initial learning rate 0.001 and divide it by 1.25 after 100 and 200 epochs for all experiments

Table 1: Classification accuracy (mean $\pm$ std) of every compared approach on adopted datasets. $\bullet/\circ$ indicates that D^{2}CNN is significantly superior / inferior to the baselines on the corresponding dataset (pairwise t-test with 0.05 significant level) and "-" denotes that the method is not scalable to the corresponding datasets.

	FM-v1	FM-v3	SVHN-v1	SVHN-v3	Lost	Yahoo!News
RegISL	-	-	-	-	0.761 ± 0.037 $\bullet$	0.598 ± 0.016 $\bullet$
SURE	-	-	-	-	0.794 ± 0.037 $\bullet$	0.729 ± 0.010 $\bullet$
WMCar-ICE	-	-	-	-	0.795 ± 0.020 $\bullet$	0.705 ± 0.010 $\bullet$
MCar	0.913 ± 0.003 $\bullet$	0.825 ± 0.002 $\bullet$	0.796 ± 0.004 $\bullet$	0.545 ± 0.004 $\bullet$	0.743 ± 0.011 $\bullet$	0.671 ± 0.010 $\bullet$
PLKNN	0.897 ± 0.004 $\bullet$	0.848 ± 0.004 $\bullet$	0.736 ± 0.003 $\bullet$	0.636 ± 0.002 $\bullet$	0.651 ± 0.012 $\bullet$	0.562 ± 0.017 $\bullet$
M3PL	0.884 ± 0.002 $\bullet$	0.874 ± 0.004 $\bullet$	0.827 ± 0.002 $\bullet$	0.788 ± 0.003 $\bullet$	0.678 ± 0.032 $\bullet$	0.613 ± 0.001 $\bullet$
IPAL	0.912 ± 0.005 $\bullet$	0.905 ± 0.003 $\bullet$	0.798 ± 0.003 $\bullet$	0.777 ± 0.001 $\bullet$	0.790 ± 0.034 $\bullet$	0.647 ± 0.017 $\bullet$
DCNN	0.902 ± 0.007 $\bullet$	0.890 ± 0.008 $\bullet$	0.922 ± 0.003 $\bullet$	0.908 ± 0.007 $\bullet$	0.580 ± 0.031 $\bullet$	0.740 ± 0.006 $\bullet$
D^{2}CNN	$\mathbf{0.936 \pm 0.002}$	$\mathbf{0.927 \pm 0.003}$	$\mathbf{0.937 \pm 0.003}$	$\mathbf{0.929 \pm 0.001}$	$\mathbf{0.838 \pm 0.014}$	$\mathbf{0.833 \pm 0.009}$

Table 2: Characteristics of the adopted datasets.

Datasets	# Images	# Classes	Avg # labels
FM-v1	70,000	10	2
FM-v3	70,000	10	4
SVHN-v1	99,289	10	2
SVHN-v3	99,289	10	4
Lost	1,122	16	2.23
Yahoo!News	14,322	38	1.44

on these synthesized datasets. The parameters α and T_{max} are set to 0.001 and 100, respectively. The average classification accuracies of D^{2}CNN and other baselines produced by the five-fold cross validation are shown in Table 1, where "-" means that the corresponding method is not scalable to the investigated dataset.

Table 1 clearly shows that D^{2}CNN achieves superior performances against other baselines on these synthesized datasets. Such superiority is also confirmed by the pairwise t-test with significance level 0.05. Besides, we can clearly observe that when the number of the candidate labels increases, the performances of all baselines decrease, especially on *SVHN* dataset. However, the performance drop of D^{2}CNN is far less than that of other baselines, which further demonstrates the effectiveness and robustness of the proposed D^{2}CNN approach.

Experiments on Real-world Datasets

Besides the experiments on synthesized datasets, we also conduct experiments on two real-world datasets including *Lost* dataset and *Yahoo!News* dataset.

The *Lost* dataset is collected from the TV serial "Lost" by Cour *et al.* (Cour et al. 2009), which aims to associate the faces appear in some certain frames with the correct names captured from the corresponding subtitles. This dataset contains 1,122 face images across 16 characters. The average amount of candidate labels for a single image in this dataset is 2.23. The *Yahoo!News* dataset contains the face images appeared in the news as well as the names (*i.e.* classes) in the corresponding captions. We retain the groundtruth names occurring at least 50 times and remove the remaining images whose captions do not contain these names. After the above process, we obtain the dataset which contains 14,322 face images belonging to 38 categories.

On *Lost* dataset, we set the initial learning rate to 0.01 and divide it by 10 after 200 epochs for our D^{2}CNN. The parameter α is set to 0.001 and T_{max} is set to 10. As for *Yahoo!News* dataset, we set the initial learning rate to 0.002 and also divide it by 10 after 200 epochs. α and T_{max} are tuned to 0.0001 and 20, respectively. For baseline approaches, their input features and parameter settings are the same with those on synthesized datasets.

The average classification accuracies of different approaches on these two real-world datasets are shown in Table 1. We see that the accuracy of D^{2}CNN is higher than IPAL, SURE, and WMCar-ICE on *Lost* dataset by approximately 4%. When it comes to *Yahoo!News* dataset, D^{2}CNN significantly outperforms other baselines and leads the second best method ($i.e.$ DCNN) with the margin of 9.3%. It is worth noting that the performances of conventional DCNN are satisfactory on synthesized datasets but are much worse than D^{2}CNN when it comes to real-world datasets. The reason is that DCNN with cross-entropy loss can be viewed as an average-based disambiguation strategy, of which the shortcomings have been analyzed above. That is, when training images are insufficient and complicated, the unique real label of each training image may be overwhelmed by the false positive ones in candidate label set, leading to the limited performances of DCNN. Such problem will not appear in our D^{2}CNN approach due to the specifically designed disambiguation technique.

Algorithm Validation

From the experimental results presented above, we see that our D^{2}CNN can obtain very impressive results. Therefore, this section further demonstrates the robustness of the proposed method and explores the reasons for the effectiveness of our method by analyzing the contributions of the key components in D^{2}CNN.

The robustness to proportion of ambiguously-labeled images: We conduct experiments on synthesized datasets to demonstrate that the proposed D^{2}CNN approach is robust to the proportion (*i.e.* p) of ambiguously-labeled images in the dataset. Figure 5 illustrates the classification accuracy of each algorithm as p ranges from 0.1 to 0.7, where r denotes

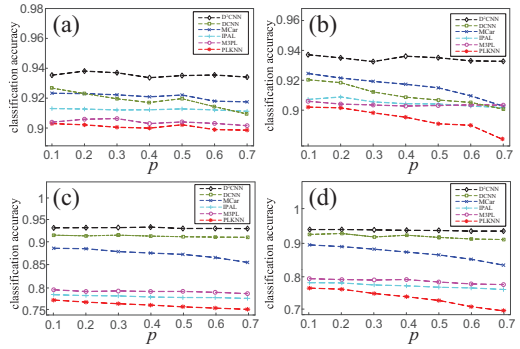

Figure 5: Sensitivity analysis of p. (a) Classification accuracies on *Fashion-Mnist* dataset when $r = 1$. (b) Classification accuracies on *Fashion-Mnist* dataset when $r = 3$. (c) Classification accuracies on *SVHN* dataset when $r = 1$. (d) Classification accuracies on *SVHN* dataset when $r = 3$.

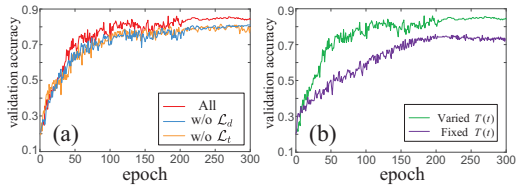

Figure 6: (a) Classification accuracy with different loss functions. The red curve denotes the loss function consisted of all three regularizers including $\mathcal{L}_f$, $\mathcal{L}_d$, and $\mathcal{L}_t$. The blue curve and orange curve indicate the classification accuracy when the discrimination term $\mathcal{L}_d$ and temporal ensembling term $\mathcal{L}_t$ are removed, respectively. (b) The accuracy comparison of fixed $T(t)$ and varied $T(t)$.

the number of false positive labels of each ambiguously-labeled image apart from the unique correct label. The experimental results indicate that our method can generally work well under different proportions of ambiguously-labeled images in the dataset.

The effectiveness of $\mathcal{L}_d$ and $\mathcal{L}_t$: We conduct ablation studies on *Lost* dataset to demonstrate the effectiveness of the proposed discrimination term $\mathcal{L}_d$ and temporal ensembling term $\mathcal{L}_t$. Figure 6 (a) shows the results, from which we can observe that the loss function with all three terms generates the highest accuracy than other settings. In contrast, the accuracy will decrease when either $\mathcal{L}_d$ or $\mathcal{L}_t$ is removed from the loss function, therefore the effectiveness and indispensability of the discrimination term $\mathcal{L}_d$ and temporal ensembling term $\mathcal{L}_t$ is validated.

The effectiveness of varied $T(t)$: In our proposed method, $T(t)$ increases with the number of epochs to gradually emphasize the effect of temporal ensembling term $\mathcal{L}_t$. In this part we illustrate that the rising $T(t)$ is better than the fixed $T(t)$. Specifically, we conduct the experiments on *Lost* dataset by comparing the accuracies under the fixed $T(t) = T_{max}$ and a varied $T(t)$ which increases from zero to T_{max}. The results shown in Figure 6 (b) clearly reflect that the varied $T(t)$ is better than the fixed $T(t)$ as it achieves a faster convergence rate as well as a higher classification accuracy.

Table 3: Classification accuracy (mean±std) of various methods with deep feature on *Lost* dataset. •/◦ indicates that D^{2}CNN is significantly superior / inferior to the baselines (pairwise t-test with 0.05 significant level).

Methods	GIST Feature	Deep Feature
RegISL	0.761 ± 0.037 •	0.786 ± 0.025 •
SURE	0.794 ± 0.037 •	0.826 ± 0.006
WMCar-ICE	0.795 ± 0.020 •	0.798 ± 0.035 •
MCar	0.743 ± 0.011 •	0.811 ± 0.038 •
PLKNN	0.651 ± 0.012 •	0.705 ± 0.017 •
M3PL	0.678 ± 0.032 •	0.774 ± 0.032 •
IPAL	0.790 ± 0.034 •	0.824 ± 0.021 •
D^{2}CNN	-	**0.838 ± 0.014**

Comparison with baselines + deep feature: As mentioned above, traditional approaches for dealing with ambiguously-labeled examples are all non-deep and only work on hand-crafted features. For fair comparison, we also utilize the deep features of the images as the input of the baselines and compare their outputs with our D^{2}CNN on *Lost* dataset. More precisely, we train the network ResNet-20 on the *Lost* dataset with the fidelity term $\mathcal{L}_f$ as loss function. After pooling and flattening, we obtain a 4096-dimensional deep feature for each input image by using the output of the last convolutional layer of the trained network. Table 3 records the classification accuracy of each baseline with deep features, from which we have two findings: 1) The deep feature consistently works better than the GIST feature for all of the baselines, so adopting deep network is beneficial to improving the performances; and 2) Our proposed D^{2}CNN still holds the highest accuracy when compared with other baselines with deep feature, so the designed loss function in Equation (1) also contributes to the success of our method.

Conclusion

In this paper, we propose a novel deep learning framework for ambiguously-labeled image classification. We are able to obtain the discriminative label predictions by introducing an entropy-based discrimination regularizer and a temporal ensembling regularizer that assembles the outputs of the network-in-training on recent epochs. As a result, the potential groundtruth label can be gradually highlighted in a reliable way. Besides, we want to mention that our method is quite general and can be equipped with other popular DCNN apart from the ResNet in this paper, such as DenseNet (Huang et al. 2017), Inception (Szegedy et al. 2017), etc. Due to the powerful representation ability and discrimination ability of the proposed D^{2}CNN, our method is able to achieve significantly better performance than other existing methods on different datasets with ambiguously-labeled images.

Acknowledgments

This research is supported by NSF of China (Nos: 61973162, 61602246, U1713208, 61772256), NSF of

Jiangsu Province (No: BK20171430), the Fundamental Research Funds for the Central Universities (No: 30918011319), the Summit of the Six Top Talents Program (No: DZXX-027), the Innovative and Entrepreneurial Doctor Program of Jiangsu Province, the Young Elite Scientists Sponsorship Program by Jiangsu Province, the Young Elite Scientists Sponsorship Program by CAST (No: 2018QNRC001), and Program for Changjiang Scholars.

References

Chen, Y.-C.; Patel, V. M.; Chellappa, R.; and Phillips, P. J. 2014. Ambiguously labeled learning using dictionaries. *IEEE Transactions on Information Forensics and Security* 9(12):2076–2088.

Chen, C.-H.; Patel, V. M.; and Chellappa, R. 2018. Learning from ambiguously labeled face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40(7):1653–1667.

Cour, T.; Sapp, B.; Jordan, C.; and Taskar, B. 2009. Learning from ambiguously labeled images. In *Proc. Computer Vision and Pattern Recognition*, 919–926.

Feng, L., and An, B. 2019a. Partial label learning by semantic difference maximization. In *Proc. International Joint Conference on Artificial Intelligence*, 2294–2300.

Feng, L., and An, B. 2019b. Partial label learning with self-guided retraining. In *Proc. AAAI Conference on Artificial Intelligence*, 3542–3549.

Gong, C.; Liu, T.; Tao, D.; Fu, K.; Tu, E.; and Yang, J. 2015. Deformed graph laplacian for semisupervised learning. *IEEE Transactions on Neural Networks and Learning Systems* 26(10):2261–2274.

Gong, C.; Tao, D.; Maybank, S. J.; Liu, W.; Kang, G.; and Yang, J. 2016. Multi-modal curriculum learning for semi-supervised image classification. *IEEE Transactions on Image Processing* 25(7):3249–3260.

Gong, C.; Zhang, H.; Yang, J.; and Tao, D. 2017. Learning with inadequate and incorrect supervision. In *Proc. International Conference on Data Mining*, 889–894.

Gong, C.; Chang, X.; Fang, M.; and Yang, J. 2018a. Teaching semi-supervised classifier via generalized distillation. In *Proc. International Joint Conference on Artificial Intelligence*, 2156–2162.

Gong, C.; Liu, T.; Tang, Y.; Yang, J.; Yang, J.; and Tao, D. 2018b. A regularization approach for instance-based superset label learning. *IEEE Transactions on Cybernetics* 48(3):967–978.

Guillaumin, M.; Mensink, T.; Verbeek, J.; and Schmid, C. 2008. Automatic face naming with caption-based supervision. In *Proc. Computer Vision and Pattern Recognition*, 1–8.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proc. Computer Vision and Pattern Recognition*, 770–778.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proc. Computer Vision and Pattern Recognition*, 4700–4708.

Hüllermeier, E., and Beringer, J. 2006. Learning from ambiguously labeled examples. *Intelligent Data Analysis* 10(5):419–439.

Jin, R., and Ghahramani, Z. 2003. Learning with multiple labels. In *Proc. Advances in Nural Information Processing Systems*, 921–928.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. In *Proc. International Conference on Learning Representations*.

Laine, S., and Aila, T. 2016. Temporal ensembling for semi-supervised learning. In *Proc. International Conference on Learning Representations*.

Liu, L., and Dietterich, T. 2012. A conditional multinomial mixture model for superset label learning. In *Proc. Advances in Neural Information Processing Systems*, 548–556.

Liu, L., and Dietterich, T. 2014. Learnability of the superset label learning problem. In *Proc. International Conference on Machine Learning*, 1629–1637.

Nguyen, N., and Caruana, R. 2008. Classification with partial labels. In *Proc. International Conference on Knowledge Discovery and Data Mining*, 551–559.

Oliva, A., and Torralba, A. 2001. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision* 42(3):145–175.

Szegedy, C.; Ioffe, S.; Vanhoucke, V.; and Alemi, A. A. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proc. AAAI Conference on Artificial Intelligence*, 4278–4284.

Wang, Q.-W.; Li, Y.-F.; and Zhou, Z.-H. 2019. Partial label learning with unlabeled data. In *Proc. International Joint Conference on Artificial Intelligence*, 3755–3761.

Wu, X., and Zhang, M.-L. 2018. Towards enabling binary decomposition for partial label learning. In *Proc. International Joint Conference on Artificial Intelligence*, 2868–2874.

Yi, K., and Wu, J. 2019. Probabilistic end-to-end noise correction for learning with noisy labels. In *Proc. Computer Vision and Pattern Recognition*, 7017–7025.

Yu, F., and Zhang, M.-L. 2016. Maximum margin partial label learning. In *Proc. Asian Conference on Machine Learning*, 96–111.

Zeng, Z.; Xiao, S.; Jia, K.; Chan, T.-H.; Gao, S.; Xu, D.; and Ma, Y. 2013. Learning by associating ambiguously labeled images. In *Proc. Computer Vision and Pattern Recognition*, 708–715.

Zhang, M.-L., and Yu, F. 2015. Solving the partial label learning problem: An instance-based approach. In *Proc. International Joint Conference on Artificial Intelligence*, 4048–4054.