Multimedia Tools and Applications
https://doi.org/10.1007/s11042-021-10760-z

Ensemble diversified learning for image classification with noisy labels

Ahmed Ahmed[1] · Hayder Yousif[2] · Zhihai He[1]

Received: 10 April 2020 / Revised: 8 January 2021 / Accepted: 25 February 2021
Published online: 09 March 2021

Abstract
In this work, we develop a new approach for learning a deep neural network for image classification with noisy labels using ensemble diversified learning. We first partition the training set into multiple subsets with diversified image characteristics. For each subset, we train a separate deep neural network image classifier. These networks are then used to encode the input image into different feature vectors, providing diversified observations of the input image. The encoded features are then fused together and further analyzed by a decision network to produce the final classification output. We study image classification on noisy labels with and without the access to clean samples. Our extensive experimental results on the CIFAR-10 and MNIST datasets demonstrate that our proposed method outperforms existing methods by a large margin.

Keywords Noisy labels · Deep neural networks · Encoder networks · K-means clustering

1 Introduction

Deep neural networks have achieved excellent performance in various visual recognition tasks such as image classification [41, 40], object detection [37], and speech recognition [15]. These achievements rely on large-scale labeled datasets, such as the ImageNet [7]. It is often expensive and time-consuming to obtain large datasets with high-quality annotations. An alternative solution is to the annotation data available on the Internet, for example, image tags from social networks or keywords from search engines. We can also resort to crowdsourcing by non-professional volunteers [10, 34]. It should be noted that these labels are often unreliable

✉ Zhihai He
hezhi@missouri.edu

[1] Department of Electrical Engineering and Computer Science, University of Missouri, Columbia, MO 65211, USA

[2] Department of Electrical Power Engineering, Basra Engineering Technical College, Southern Technical University, Basra, Iraq

with significant amounts of incorrect labels. These noisy labels will affect the classification performance of the trained classifiers [33]. Therefore, it is important to develop robust learning methods for image classification in the presence of label noise.

Approaches for learning effective image classifiers from noisy labels can be categorized into two major groups. The first group learns from noisy labels directly. They focus on noise-robust learning algorithms [30, 4] and label noise cleansing to remove or correct the mislabeled samples [5, 3, 32]. These approaches have the difficulty in identifying informative hard training samples [6] from mislabeled samples. The second group assumes that an extra set of clean samples with correct labels verified by human are available for guiding the training process [48, 45]. For instance, Lee et al. [21] used the clean samples to train an additional network to verify whether the label is correct. Clean samples have been used by Jiang et al. [16], Ren et al. [38], and Han et al. [12] to determine weights on noisy samples. Sun et al. [43] proposed a weakly supervised learning approach to learn a deep convolutional neural network (DCNN) from unlabeled data using a small set of labeled data. Methods have also been developed to estimate the noise transition matrix and adjust the loss function [35].

We observe that an important component in learning deep neural networks with noisy labels is to exploit the consistency between the predictions by different networks and use this consistency constraint to correct label errors. This requires that these networks participating the joint error correction should have diversified observations. Otherwise, fusing their predictions together will not be helpful for label error correction. To this end, we propose to partition the training set into multiple subsets with diversified image characteristics. For each subset, we train a separate deep neural network image classifier. These networks are then used to encode the input image into different feature vectors, providing diversified observations of the input image. The encoded features are then fused together and further analyzed by a decision network to produce the final classification output. We study two different scenarios: learning the deep neural network classifier with and without clean samples. Our extensive experimental results on the CIFAR-10 and MNIST datasets demonstrate that our proposed method outperforms existing methods by a large margin.

The rest of the paper is organized as follows. Section 2 reviews the related work. Section 3 presents our method. Experimental results are provided in Section 4. Further discussions and concluding remarks are provided in Section 5.

2 Related work and major contributions

Learning with noisy labels has emerged as an important research topic in machine learning and computer vision. A number of methods have been developed to train convolutional neural networks with noisy labels [2, 28, 39, 51]. Han et al. [12] proposed a co-teaching method which learns two networks simultaneously. Each network teaches the other network using the most useful samples for parameter updating. An improved coteaching method was developed by Yu et al. [52] which explores the prediction disagreement between two networks to update parameters. Yuan et al. [53] proposed a method to improve the performance of neural networks with and without a monitoring set. They used two independent networks to update the label during the training process. Han et al. [13] presented a so-called pump-out method to squeeze out the impact of noisy samples from the training model. Zhang and Sabuncu [54] proposed new loss functions for training deep neural networks with noise. A noise-tolerant learning algorithm was proposed by Li et al. [29], where meta-learning update is performed before the conventional gradient update. The meta-learning method simulates actual training with label noise.

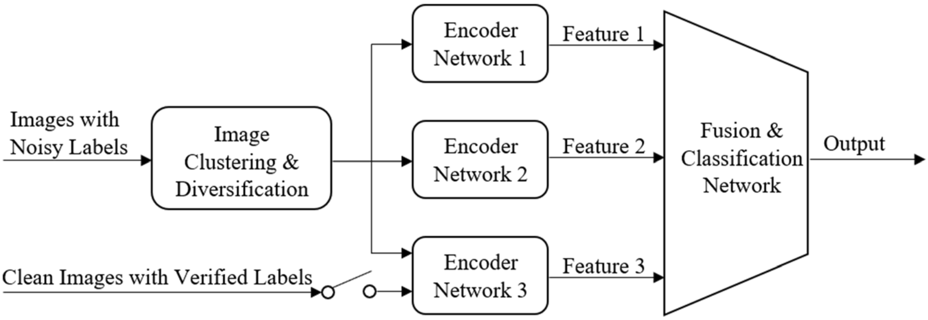

Fig. 1 Overview of the proposed method with two phases

Another set of approaches combine noisy labels with a small set of clean samples for training convolutional neural networks (CNNs). Xiao et al. [51] used two independents CNNs to predict the class label and the noise type. Azadi et al. [2] proposed an auxiliary image regularizer to actively select reliable images to improve noise robustness. The clean samples are used to train a CNN feature extraction network. Vahdat [45] built an undirected graphical model to model label noise in training datasets. Li et al. [28] proposed a unified distillation method to learn from noisy labels using a small amount of clean samples. Hendrycks et al. [14] presented a loss correlation method, which learns a label corruption matrix using trusted data and then uses the corruption matrix to train a corrected classifier. Deep networks are able to accommodate a certain level of label noise [49]. Fawzi et al. [9] studied the robustness of neural networks against adversarial perturbations, geometric transformations, etc. Van et al. [46] found that learning algorithms are robust to annotation errors and training data corruption if the training set has sufficient samples. Rolnick et al. [39] demonstrated that convolutional neural networks trained on large datasets are sufficiently robust to label noise. They demonstrated that larger batch sizes with lower learning rates are beneficial for learning with label noise.

This work is also related to feature fusion networks [24]. The methods developed in [25] and [27] explored distinctive patterns of various descriptors to improve the fusion performance. Multi-orientation score-level fusion has been developed in [23] and [22]. The multi-directional fusion method developed in [26] is able to fuse the matching score to improve fusion performance. Tang et al. [44] have shown that the classification performance based on fusion of electroencephalography (EEG) and electromyography (EMG) data is better than that using the EEG features only.

Table 1 Performance (classification accuracy) evaluation with different groups of datasets on the CIFAR-10 dataset at different noise rates with clean network

Number of groups	Clusters per group	Noise levels 30%	50%	70%
2	6	**77.94**	**73.82**	**69.16**
3	4	77.60	72.56	68.40
4	3	77.10	72.06	67.54
5	2	76.48	71.32	66.02

The bold numbers represent the highest classification accuracies

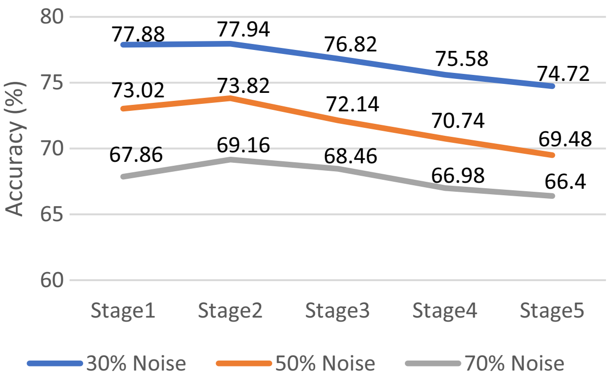

Fig. 2 Accuracy of five stages on CIFAR-10 dataset in different levels of noise with clean samples

Compared to the existing work, the ***main contributions*** of this work can be summarized as follows: (1) we develop a new approach to learn deep neural network image classification from noisy labels using ensemble diversified observations. (2) We learn a set of networks on diversified training sets to encode the input image to provide diversified descriptions of the image. These features will be fused together and further analyzed by a network to generate the final classification output. (3) Our experimental results on benchmark datasets demonstrate that our method outperforms existing methods by a large margin.

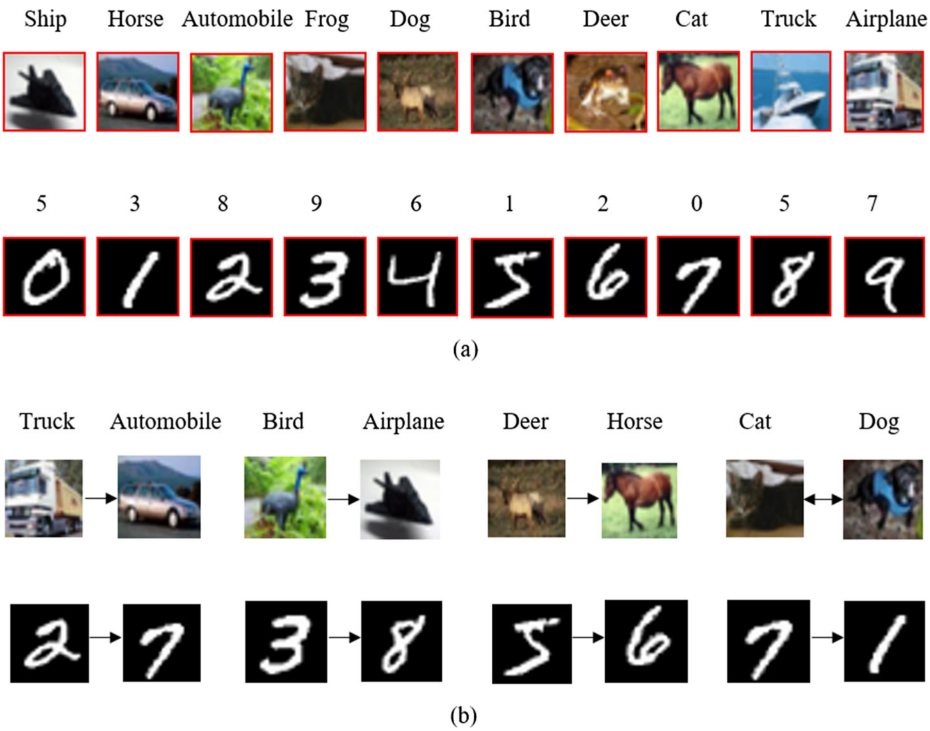

Fig. 3 Samples of CIFAR-10 and MNIST training sets diluted with noisy labels. **a** symmetric noise; **b** asymmetric noise. The red box represents noisy label

Table 2 Performance (classification accuracy) comparison on the CIFAR-10 dataset at different noise rates with clean network

Method	30%	50%	70%
Base CNN	74.46	70.38	63.30
Base CNN Iterative cross learning (ICL) [53]	74.90	70.10	62.90
True noise [42]	75.20	70.40	63.80
Dropout [17]	75.90	72.30	65.60
Base CNN+ICL [53]	75.1 0	72.40	68.00
Dropout+ICL [53]	76.10	74.00	70.00
Ours (Phase1) [1]	77.94	73.82	69.16
Ours (Phase2)	**79.72**	**75.02**	**70.40**

The bold numbers represent the highest classification accuracies

3 Method

In this section, we present our proposed method to learn a deep neural network for image classification with noisy labels.

3.1 Overview

The overall framework of the proposed method is illustrated in Fig. 1. We first use a pre-trained deep convolution neural network (DCNN) to extract features for all images in the training set. We then apply k-means clustering to their features and partition them into $M \cdot L$ clusters. We then split these $M \cdot L$ clusters into M groups with each group having L clusters. In this way, these M subsets of training images will have diversified image characteristics since they consist of different clusters. Each group is then used to train the Encoder Network. We concatenate the features extracted by the Encoder Networks and then train the fusion network to generate the classification output. In the following sections, we explain these major components in more details.

3.2 Image clustering and diversification

The objective of this task is to partition the input training samples based on the similarity of extracted features into clusters with different characteristics using k-means clustering. We randomly partition these clusters into Mgroups, which are then used to train different networks.

Table 3 Performance (classification accuracy) comparison on the MNIST dataset at different noise rates with clean network

Method	30%	50%	70%
Base CNN	97.93	97.03	95.60
Base CNN ICL [53]	97.83	96.8	94.25
True noise [42]	98.70	97.94	96.69
Dropout [17]	98.75	98.20	96.99
Base CNN+ICL [53]	98.34	97.88	97.46
Dropout+ICL [53]	98.93	98.68	98.22
Ours (Phase1) [1]	98.38	98.18	98.08
Ours (Phase2)	**99.18**	**98.82**	**98.64**

The bold numbers represent the highest classification accuracies

Table 4 Performance (classification accuracy) comparison on the CIFAR-10 dataset at different noise without clean network

Method	30%	50%	70%
Base CNN	74.46	70.38	63.30
Base CNN ICL [53]	74.30	69.40	59.60
True noise [42]	75.20	70.40	63.80
Dropout [17]	75.40	71.40	64.70
Base CNN+ICL [53]	74.50	71.90	66.20
Dropout+ICL [53]	75.50	73.20	**69.40**
Ours (Phase 1) [1]	76.00	72.12	66.38
Ours (Phase 2)	**78.72**	**74.68**	67.30

The bold numbers represent the highest classification accuracies

In this case, each network is trained with different sets of samples. In this paper, we use the pre-trained clean network to extract the features from the training set if clean samples are available. Otherwise, we use the pretrained base model to extract the features. This approach is more effective than random splitting of the training set since those groups obtained by the clustering approach will have much more diversified image characteristics than those obtained by random partitions [1]. The number of groups M is an important parameter in our algorithm. Table 1 shows that two groups with 6 clusters for each achieve the best performance since each group has a sufficient number of samples to train the Encoder Networks. Based on this observation, in our experiments, we set the default value of M to be 2.

3.3 CNN-Based image encoding

We use the groups of training images obtained from k-means clustering to train the Encoder Networks 1 and 2. The Encoder Network 3 is trained on a set of clean samples with verified labels if it is available. Otherwise, if the clean samples are not available, we randomly select 90 % of the whole training set to train this network. We use these three Encoder Networks to predict new labels in the next stage based on majority voting. The predicted labels will be used to update the network in the next stage. If two or more networks predict the same label, the original label will be updated to this label. Otherwise, we keep the original label. Once labels are updated, we then come back to refine those two networks. This label update and network refinement process is repeated for multiple times.

Table 5 Performance (classification accuracy) comparison on the MNIST dataset at different noise without clean network

Method	30%	50%	70%
Base CNN	97.93	97.03	95.60
Base CNN ICL [53]	97.71	96.24	93.40
True noise [42]	98.70	97.94	96.69
Dropout [17]	98.71(98.80)	97.88(98.08)	96.04(96.88)
Base CNN+ICL [53]	98.00	97.46	96.42
Dropout+ICL [53]	98.83	98.50	97.66
Ours (Phase 1) [1]	98.58	98.06	97.68
Ours (Phase 2)	**99.14**	**99.04**	**98.56**

The bold numbers represent the highest classification accuracies

3.4 Image feature fusion and classification

We concatenate the features obtained from the Encoder Networks. These networks have the same structure, where Encoder Networks 1 and 2 have been trained on different subsets of input training data and Encoder Network 3 is trained with and without clean samples. The fusion network takes the input from the last fully connected layer of each Encoder Network and concatenates them together. Finally, the fusion network is trained on the input training set to evaluate the performance.

3.5 Algorithm summary and complexity analysis

In this section, we summarize the two phases of our algorithm in Algorithm 1 and Algorithm 2. In Algorithm 1, we use the groups of samples obtained by the k-means clustering to train the Encoder Networks 1 and 2. The Encoder Network 3 is trained on clean samples. If no clean samples are available, it is trained with 90 % samples randomly selected from the whole training set. The original label will be updated at each training stage. In Algorithm 2, we connect these Encoder Networks with the fusion network which predicts the classification output.

In this work, we use a multi-stage training process. At each stage, the fusion network will update the label of each sample. Once the labels are updated, we will come back to fine-tune the encoder and fusion networks. With a number of stages, more and more error labels will be corrected. Figure 2 shows the best classification accuracy is achieved at stage 2 for three different label noise levels 30 %, 50 %, and 70 %. When the number of stages is more than two, the continuous label updating becomes less and less accurate and introduces additional noise into the labels, which degrades the overall performance.

Algorithm 1 Training Encoder Networks with and without clean samples

Input: Noisy dataset D, Pretrained clean CNN or Pretrained base CNN
Output: Encoder Networks

```
Initialize: n ← 1, N ← 5, Batch size, and learning rate.
Extract features from Noisy dataset using pretrained clean CNN or pretrained base CNN
Use k-means clustering to partition noisy dataset into G's groups
repeat
    Train net1 on G1
    Train net2 on G2
    Train net3 on 90% of G1G2
    Predict labels L1 by net1
    Predict labels L2 by net2
    Predict labels L3 by clean net or by net3
    for sample x in D do
        if x's label is equal in L1 and L2 or x's label is equal in L1 and L3 then
            Update x's label to L1
        elsif x's label is equal in L2 and L3 then
            Update x's label to L2
        else keep the same label of sample x
        end if
    end for
    n ← n+1
until n < N
```

Algorithm 2 Training fusion network

Input: Noisy dataset D and Encoder Networks
Output: Learned model
1: **Initialize:** m ← 1, Max epoch M, Batch size, and learning rate.
2: concatenate Encoder Networks into net4
3: **repeat**
4: train net4 on noisy dataset D
5: predict noisy labels L by net4
6: **Until** $m < M$

4 Experimental results

In this section, we conduct extensive experiments on benchmark datasets to evaluate the performance of our proposed image classification method with noisy labels.

4.1 Datasets

In this work, we use the CIFAR-10, MNIST, and 10-class ImageNet (ILSVRC2012) datasets to evaluate our method. The MNIST dataset is a set of images of handwritten digits [20]. It has 10 classes with 60,000 training images and 10,000 test images. Each image has a size of 28×28 pixels. The CIFAR-10 dataset consists of 60,000 color images with resolution 32×32 [18]. We use 50,000 training images and 10,000 images for testing. In the ImageNet dataset [40], we use 10 classes of animals (*Bulbul, Goose, Gorilla, Hen, Ice bear, King penguin, Mud turtle, Ox, Red fox, and Sea Snake)* with size 227×227 to evaluate our method. For the training dataset, we apply data augmentation methods, including random reflection, random translation, and center cropping. We corrupt the training datasets with two types of label noise: symmetric and asymmetric label noise. In symmetric noise, we randomly generate different labels for one class based on a uniform random distribution. In asymmetric noise, we introduce label errors between similar classes. For example, in the CIFAR-10 dataset, sample labels may be corrupted as follows: TRUCK → AUTOMOBILE, BIRD → AIRPLANE, DEER → HORSE, CAT ↔ DOG. In the MNIST dataset, the sample labels may be corrupted as: 2 → 7, 3 → 8, 5 → 6, 7 → 1. Figure 3 shows some examples of training images with noisy labels.

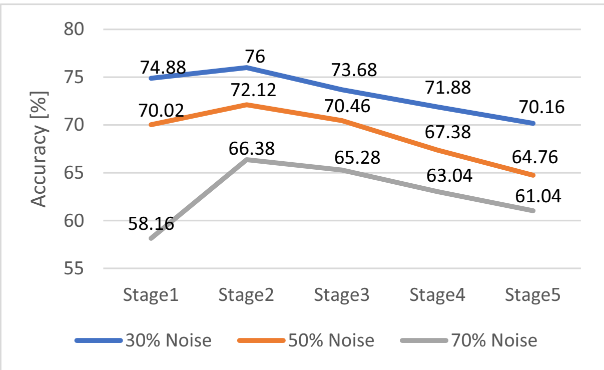

Fig. 4 Accuracy of five stages on CIFAR-10 dataset in different levels of noise without clean samples

Table 6 Classification accuracy of sampling different percentages of whole data on CIFAR-10 dataset at different noise without clean network

	Noise levels		
Percentage	30%	50%	70%
50%	74.90	70.62	63.00
70%	75.36	71.74	64.92
90%	**76.00**	**72.12**	**66.38**
100%	75.94	71.82	64.78

The bold numbers represent the highest classification accuracies

4.2 Results

In this section, we evaluate the performance of our method on the CIFAR-10, MNIST, and ImageNet (ILSVRC2012) datasets with different levels of noise. We compare the performance of our method with the baseline CNN and the methods developed by Yuan et al. [53], Ding et al. [8], and Wang et al. [50]. We use the same model architecture as in these three methods during performance comparisons.

4.2.1 Results on the CIFAR-10 and MNIST datasets with clean labels

In the following experiments, we use a set of clean samples to train the clean network. Following the procedure in Yuan et al. [53], we divide the 10,000 images into 5,000 images for testing and 5,000 images for training the clean network. We use the same network architecture as in Yuan et al. [53] on the CIFAR-10 and MNIST datasets with two convolutional layers and two fully connected layers. Each convolutional layer is followed by a ReLU (Rectified Linear Unit) layer and a max-pooling layer. The first fully connected layer (fc1) followed by ReLU activation is added to the second max-

Table 7 Performance (classification accuracy) comparison on CIFAR-10 and MNIST datasets at different noise rates without clean network

	CIFAR-10		MNIST	
Method	20% Symmetric	20% Asymmetric	20% Symmetric	20% Asymmetric
Base CNN	83.66	84.92	97.68	98.20
cross-entropy [35]	83.70	85.00	96.9±0.1	97.5±0.0
unhinged (BN) [47]	84.10	83.80	96.9±0.1	97.0±0.1
sigmoid (BN) [11]	66.60	71.80	93.1±0.1	96.7±0.1
Savage [31]	77.40	76.00	96.9±0.0	97.0±0.1
bootstrap soft [36]	84.30	84.60	96.9±0.0	97.5±0.0
bootstrap hard [36]	83.60	84.70	96.8±0.0	97.4±0.0
Backward [35]	80.40	83.80	96.9±0.0	96.7±0.1
Forward [35]	83.40	87.00	96.9±0.0	97.7±0.0
cross-entropy [8]	82.40	85.50	97.1±0.1	97.6±0.2
improved baseline [8]	83.60	85.20	97.1±0.1	97.7±0.1
Semi-Supervised learning [8]	84.50	85.60	97.70±0.4	97.80±0.1
Ours (Phase 1) [1]	85.11	86.47	97.99	98.40
Ours (Phase 2)	**87.16**	**87.54**	**99.06**	**99.01**

The bold numbers represent the highest classification accuracies

Table 8 Performance (classification accuracy) comparison on CIFAR-10 at different noise rates without clean network

	Noise levels	
Method	20 %	40 %
Base CNN	81.23	77.00
Cross-entropy [50]	74.17	62.38
Backward [35]	76.27	75.18
Forward [35]	79.25	77.81
Bootstrapping [36]	74.39	69.50
CNN-CRF [45]	80.15	78.69
Iterative learning [50]	81.36	78.15
Ours (Phase 1) [1]	82.38	78.94
Ours (Phase 2)	**83.86**	**81.19**

The bold numbers represent the highest classification accuracies

pooling layer. Finally, a softmax layer is added after the second fully connected layer (fc2) for classification.

In Tables 2 and 3, we can see that our method outperforms the baseline CNN by 5.26 %, 4.64 %, and 7.10 % for label noise levels of 30 %, 50 %, and 70 %, respectively on the CIFAR-10 dataset, and by 1.25 %, 1.79 %, and 2.88 %, respectively on the MNIST dataset. Compared to Yuan's method, it has improved the classification accuracy by 3.62 %, 1.02 % and 0.4 %, respectively on the CIFAR-10 dataset, and by 0.25 %, 0.14 %, and 0.26 % on MNIST. Our method achieves the best performance when compared to the other two methods, the true noise [42] and dropout regularization [17] methods, at all noise levels.

4.2.2 Results on the CIFAR-10 and MNIST datasets without clean labels

Here, we evaluate our method without clean samples. Tables 4 and 5 summarize the performance comparison results at different noise levels. We can see that our method outperforms the base CNN by 4.26 %, 4.30 %, and 4.00 % for noise levels of 30 %, 50 %, and 70 %, respectively on the CIFAR-10 dataset and by 1.31 %, 2.01 %, and 2.96 %, respectively on the MNIST. Compared to Yuan's method, it has improved the classification accuracy by 3.22 % and 1.48 % at noisy levels 30 % and 50 %, respectively. But its accuracy dropped by 2.10 % at noisy level 70 % on the CIFAR-10. On the MNIST dataset, our method improved the accuracy by 0.41 %, 0.54 %, and 0.90 %, respectively. Without clean samples, our method also outperforms the true noise [42] and dropout regularization [17] methods at all noise levels.

Figure 4 shows the highest accuracy obtained at stage 2 for three different label noise levels 30 %, 50 %, and 70 %. The classification accuracy drops after stage 2. In Table 6,

Table 9 Performance evaluation on 10 classes ImageNet dataset with clean network

	Noise levels		
Method	30 %	50 %	70 %
Base CNN	71.37	63.52	51.78
Ours (Phase 1) [1]	73.37	67.30	**59.44**
Ours (Phase 2)	**75.07**	**67.89**	57.67

The bold numbers represent the highest classification accuracies

Table 10 Performance evaluation on 10 classes ImageNet dataset without clean network

	Noise levels		
Method	30 %	50 %	70 %
Base CNN	71.37	63.52	51.78
Ours (Phase 1) [1]	72.67	66.52	55.93
Ours (Phase 2)	**74.89**	**67.37**	**56.52**

The bold numbers represent the highest classification accuracies

we show the impact of different percentages of the training set used for training on the final classification performance. We can see that the performance is optimized when we randomly sample 90 % of the whole training set for training. In this work, we compare our method with Ding [8] with symmetric and asymmetric label noise. For symmetric label noise, we randomly generate different labels to replace the true label for each class with 20 % of noise level. For asymmetric labels noise, we introduce label errors between similar classes, for example TRUCK → AUTOMOBILE, BIRD → AIRPLANE, DEER → HORSE, CAT ↔ DOG and 2 → 7, 3 → 8, 5 → 6, 7 → 1. In Table 7, we can see that our proposed method outperforms other methods and achieves a high accuracy at noise level of 20 % with symmetric and asymmetric of noise. We also compare our method with Wang [50] on the CIFAR-10 dataset. We use the same network architecture. Our method outperforms the baseline CNN and Wang [50] with 20 % and 40 % noise levels as shown in Table 8.

4.2.3 Results on the 10-class ImageNet (ILSVRC2012) dataset

To further evaluate the performance of our method, we choose a very large dataset, the 10-class ImageNet. We randomly divide the entire dataset of 13,500 images into 70 % for training and 20 % for testing, and the remaining 10 % as cleaning samples. We use the AlexNet architecture [19] to train the Encoder Networks. We train the Encoder Network 3 with clean samples and then we use this pretrained network with Encoder Networks 1 and 2 to predict the label. The results are summarized in Table 9. If clean samples are not available, we randomly select 90 % of the training set to train Encoder Network 3 and then use this network with Encoder Networks 1 and 2 to predict the label in the next stage. The results are summarized in Table 10. We can see that, in both cases, our method outperforms the base CNN method. We could not compare our method against other methods since they did not provide results on this dataset.

5 Conclusions

We have successfully developed a new method to learn a deep neural network image classifier from noisy labels. We have demonstrated that, by partitioning the training set into subset of training image with diversified image characteristics, we can train separate networks to capture different descriptions of the input images. Once these diversified descriptions are fused together, they can correct error labels. We considered the network training process with and without clean samples. Our experimental results show the proposed method is robust to label noise and outperforms existing methods.

References

1. Ahmed A, Yousif H, Kays R, He Z (2020) Animal species classification using deep neural networks with noise labels. Eco Inform 57:101063
2. Azadi S, Feng J, Jegelka S, Darrell T (2015) Auxiliary image regularization for deep cnns with noisy labels. arXiv preprint arXiv:1511.07069
3. Barandela R, Gasca E (2000) Decontamination of training samples for supervised pattern recognition methods. In: Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR). Springer, Berlin, Heidelberg, pp 621–630
4. Beigman E, Klebanov BB (2009) Learning with annotation noise. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, pp 280–287
5. Brodley CE, Friedl MA (1999) Identifying mislabeled training data. J Artif Intell Res 11:131–167
6. Chu J, Guo Z, Leng L (2018) Object detection based on multi-layer convolution feature fusion and online hard example mining. IEEE Access 6:19959–19967
7. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L (2009) Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, pp 248–255. IEEE, Piscataway
8. Ding Y, Wang L, Fan D, Gong B (2018) A semi-supervised two-stage approach to learning from noisy labels. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, Piscataway, pp 1215–1224
9. Fawzi A, Moosavi-Dezfooli SM, Frossard P (2017) The robustness of deep networks: A geometrical perspective. IEEE Signal Process Mag 34(6):50–62
10. Fergus R, Fei-Fei L, Perona P, Zisserman A (2010) Learning object categories from internet image searches. Proc IEEE 98(8):1453-66
11. Ghosh A, Manwani N, Sastry PS (2015) Making risk minimization tolerant to label noise. Neurocomputing 160:93–107
12. Han B, Yao Q, Yu X, Niu G, Xu M, Hu W, Tsang I, Sugiyama M (2018) Co-teaching: Robust training of deep neural networks with extremely noisy labels. Adv Neural Inf Proces Syst 31:8527–8537
13. Han B, Niu G, Yao J, Yu X, Xu M, Tsang I, Sugiyama M (2018) Pumpout: A meta approach to robust deep learning with noisy labels. arXiv preprint arXiv:1809.11008.
14. Hendrycks D, Mazeika M, Wilson D, Gimpel K (2018) Using trusted data to train deep networks on labels corrupted by severe noise. In: Advances in neural information processing systems, pp 10456–10465
15. Hinton G, Deng L, Yu D, Dahl GE, Mohamed AR, Jaitly N, Senior A, Vanhoucke V, Nguyen P, Sainath TN, Kingsbury B (2012) Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Process Mag 29(6):82–97
16. Jiang L, Zhou Z, Leung T, Li LJ, Fei-Fei L (2018) Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In: International Conference on Machine Learning, pp 2304–2313. PMLR.
17. Jindal I, Nokleby M, Chen X (2016) Learning deep networks from noisy labels with dropout regularization. In: 2016 IEEE 16th International Conference on Data Mining (ICDM). IEEE, Piscataway, pp 967–972
18. Krizhevsky A, Hinton G (2009) Learning multiple layers of features from tiny images
19. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems
20. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–324
21. Lee KH, He X, Zhang L, Yang L (2018) Cleannet: Transfer learning for scalable image classifier training with label noise. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 5447–5456
22. Leng L, Zhang J (2011) Dual-key-binding cancelable palmprint cryptosystem for palmprint protection and information security. J Netw Comput Appl 34(6):1979–1989
23. Leng L, Zhang J (2013) Palmhash code vs. palmphasor code. Neurocomputing 108:1–12
24. Leng L, Zhang J, Chen G, Khan MK, Bai P (2011) Two dimensional PalmPhasor enhanced by multi-orientation score level fusion. In: FTRA International Conference on Secure and Trust Computing, Data Management, and Application. Springer, Berlin, Heidelberg, pp 122–129

25. Leng L, Li M, Teoh AB (2013) Conjugate 2DPalmHash code for secure palm-print-vein verification. In: 2013 6th International Congress on Image and Signal Processing (CISP), vol 3. IEEE, Piscataway, pp 1705–1710
26. Leng L, Teoh AB, Li M, Khan MK (2014) A remote cancelable palmprint authentication protocol based on multi-directional two-dimensional PalmPhasor-fusion. Secur Commun Netw 7(11):1860–1871
27. Leng L, Li M, Kim C, Bi X (2017) Dual-source discrimination power analysis for multi-instance contactless palmprint recognition. Multimed Tools Appl 76(1):333–354
28. Li Y, Yang J, Song Y, Cao L, Luo J, Li LJ (2017) Learning from noisy labels with distillation. In: Proceedings of the IEEE International Conference on Computer Vision, pp 1910–1918
29. Li J, Wong Y, Zhao Q, Kankanhalli MS (2019) Learning to learn from noisy labeled data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 5051–5059
30. Manwani N, Sastry PS (2013) Noise tolerance under risk minimization. IEEE Trans Cybern 43(3):1146–1151
31. Masnadi-Shirazi H, Vasconcelos N (2008) On the design of loss functions for classification: theory, robustness to outliers, and savageboost. Adv Neural Inf Proces Syst 21:1049–1056
32. Miranda AL, Garcia LP, Carvalho AC, Lorena AC (2009) Use of classification algorithms in noise detection and elimination. In: International Conference on Hybrid Artificial Intelligence Systems. Springer, Berlin, Heidelberg, pp 417–424
33. Nettleton DF, Orriols-Puig A, Fornells A (2010) A study of the effect of different types of noise on the precision of supervised learning techniques. Artif Intell Rev 33(4):275–306
34. Niu L, Li W, Xu D (2015) Visual recognition by learning from web data: A weakly supervised domain generalization approach. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2774–2783
35. Patrini G, Rozza A, Krishna Menon A, Nock R, Qu L (2017) Making deep neural networks robust to label noise: A loss correction approach. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1944–1952
36. Reed S, Lee H, Anguelov D, Szegedy C, Erhan D, Rabinovich A (2014) Training deep neural networks on noisy labels with bootstrapping. arXiv preprint arXiv:1412.6596
37. Ren S, He K, Girshick R, Sun J (2016) Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39(6):1137–1149
38. Ren M, Zeng W, Yang B, Urtasun R (2018) Learning to reweight examples for robust deep learning. arXiv preprint arXiv:1803.09050
39. Rolnick D, Veit A, Belongie S, Shavit N (2017) Deep learning is robust to massive label noise. arXiv preprint arXiv:1705.10694
40. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC (2015) Imagenet large scale visual recognition challenge. Int J Comput Vision 115(3):211–252
41. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv: 1409.1556
42. Sukhbaatar S, Bruna J, Paluri M, Bourdev L, Fergus R (2014) Training convolutional networks with noisy labels. arXiv preprint arXiv:1406.2080
43. Sun L, Zhao C, Yan Z, Liu P, Duckett T, Stolkin R (2018) A novel weakly-supervised approach for RGB-D-based nuclear waste object detection. IEEE Sens J 19(9):3487–3500
44. Tang Z, Yu H, Lu C, Liu P, Jin X (2019) Single-trial classification of different movements on one arm based on ERD/ERS and corticomuscular coherence. IEEE Access 7:128185–128197
45. Vahdat A (2017) Toward robustness against label noise in training deep discriminative neural networks. In: Advances in Neural Information Processing Systems, pp 5596–5605
46. Van Horn G, Branson S, Farrell R, Haber S, Barry J, Ipeirotis P, Perona P, Belongie S (2015) Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 595–604
47. Van Rooyen B, Menon A, Williamson RC (2015) Learning with symmetric label noise: The importance of being unhinged. Adv Neural Inf Process Syst 28:10–18
48. Veit A, Alldrin N, Chechik G, Krasin I, Gupta A, Belongie S (2017) Learning from noisy large-scale datasets with minimal supervision. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 839–847
49. Wang F, Jiang M, Qian C, Yang S, Li C, Zhang H, Wang X, Tang X (2017) Residual attention network for image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3156–3164

50. Wang Y, Liu W, Ma X, Bailey J, Zha H, Song L, Xia ST (2018) Iterative learning with open-set noisy labels. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 8688–8696
51. Xiao T, Xia T, Yang Y, Huang C, Wang X (2015) Learning from massive noisy labeled data for image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2691–2699
52. Yu X, Han B, Yao J, Niu G, Tsang IW, Sugiyama M (2019) How does disagreement help generalization against label corruption? arXiv preprint arXiv:1901.04215
53. Yuan B, Chen J, Zhang W, Tai HS, McMains S (2018) Iterative cross learning on noisy labels. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, Piscataway, pp 757–765
54. Zhang Z, Sabuncu M (2018) Generalized cross entropy loss for training deep neural networks with noisy labels. Adv Neural Inf Proces Syst 31:8778–8788