IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 6, 2019, accepted February 20, 2019, date of publication March 11, 2019, date of current version April 2, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2904403

A Recursive Ensemble Learning Approach With Noisy Labels or Unlabeled Data

YUCHEN WANG[1], YANG YANG[1], (Member, IEEE), YUN-XIA LIU[2], (Member, IEEE), AND ANIL ANTHONY BHARATH[3], (Member, IEEE)

[1]School of Information Science and Engineering, Shandong University, Qingdao 266237, China
[2]School of Information Science and Engineering, University of Jinan, Jinan 250022, China
[3]Department of Biomedical Engineering, Imperial College London, London SW7 2AZ, U.K.

Corresponding author: Yang Yang (yyang@sdu.edu.cn)

This work was supported by the National Key Research and Development Program of China under Grant 2017YFC0803404, Grant 2017YFC0803405, Grant 2018YFC0831103, and Grant 2018YFC0831105.

ABSTRACT For many tasks, the successful application of deep learning relies on having large amounts of training data, labeled to a high standard. But much of the data in real-world applications suffer from label noise. Data annotation is much more expensive and resource-consuming than data collection, somewhat restricting the successful deployment of deep learning to applications where there are very large and well-labeled datasets. To address this problem, we propose a recursive ensemble learning approach in order to maximize the utilization of data. A disagreement-based annotation method and different voting strategies are the core ideas of the proposed method. Meanwhile, we provide guidelines for how to choose the most suitable among many candidate neural networks, with a pruning strategy that provides convenience. The approach is effective especially when the original dataset contains a significant label noise. We conducted experiments on the datasets of Cats versus Dogs, in which significant amounts of label noise were present, and on the CIFAR-10 dataset, achieving promising results.

INDEX TERMS Noisy labels, pruning strategy, semi-supervised learning, ensemble learning, deep learning, neural networks.

I. INTRODUCTION

Deep neural networks (DNNs) have achieved outstanding results in several benchmarks for image recognition. However, these achievements have typically been attained on large quantities of accurately-labeled data; in real-world applications of machine learning, there is often little or only a small amount of labeled data. It is not difficult to collect large amounts of data, but significant time and manpower is required to annotate it to high quality. It would be better to be able to make use of both labeled and unlabeled data in training deep networks. For example, even unlabeled clinical medical data can be a source of information to train deep models about the variability of measurements within and between patients; if one were required to label entire patient databases to be able to learn, the process would require annotators to have significant and patient-specific medical knowledge, and would consume signicant time. Further, data acquired for specific studies might provide valuable information for other types of inference problem. In short, it is wasteful not to make use of all available data, because it has very important implications for building deep models which might have applications beyond the initial purpose of data collection.

Errors are inevitable during the labeling process: human errors in interpretation, measurement errors, and subjective biases are all contributing factors to label noise. Many large-scale datasets are trawled from websites, another reason for noisy labels. Moreover, even high-quality datasets will contain label noise, often due to ambiguity in the data itself. Noisy labels can have adverse effects on training [1], although in standard curated datasets, it is generally recognized that artificially introduced inconsistencies tend to dilute the effect of mislabeling. It is not clear whether synthetically generated label corruption fully reflects the nature of mislabeling in practice. For example, when visually similar items are mis-categorised by human annotators, there are likely to be consistent effects on gradients during learning that would perturb training.

The associate editor coordinating the review of this manuscript and approving it for publication was Juan Liu.

Research on noisy labels has attracted attention in recent years [2], [3]. Zhang *et al.* [4], [5] researched the problem of noisy labeling, making some progress towards improving performance in imbalanced datasets. Meanwhile, deep neural networks are increasingly applied to new sources of image data. A single neural network may be unable to achieve its intended function because it is limited by factors such as network depth. To solve this problem, we propose a recursive ensemble learning approach, attractive because it places no strict requirements on the dataset or the network. Our goal is to improve network performance accurately and efficiently without changing any original data, whether the original dataset contains corrupted labels or not. Our approach has different strategies for different raw datasets; Fig. 1 shows the concept. The second column of the table in the picture first divides the output of four neural networks into six groups (any two results are grouped together) and then votes, so we get six different scores. The main contributions of the work are as follows:

1) We propose a recursive ensemble learning approach to deal with the partially labeled/noisy label problem. Through comparison, we found that performance can be improved more effectively by adopting plurality voting than by using weighted voting. Meanwhile, we make use of pseudo-labeling [6], [7] to make the best use of available data. We can reasonably speculate that if the volume of data increases, the ensemble learning method we propose will increase in performance accordingly.
2) Through the combination strategy between networks, we can use a smaller amount of data to achieve higher precision, which improves data use efficiency. We also propose a pruning strategy which removes certain networks from the ensemble.
3) We evaluated performance in Cats vs. Dogs and CIFAR-10 datasets. Experimental results suggest that our method compares favorably with other approaches, and can improve on the performance of them by a minimum of 10%; indeed, in one setting, the improvement achieved against naive network training is as high as 50%.

II. RELATED WORK

A. ENSEMBLE LEARNING

Ensemble learning [8] works by building and combining multiple learners; the principle is variously referred to as multi-classifier systems, committee-based systems, and so on. First, a group of individual learners is generated, and once trained, their decisions are combined in some way. Individual learners are usually generated from training data by an existing learning algorithm. Ensemble learning, by combining multiple learners, usually achieves a generalization capability that is significantly better than that of a single learning device. It is worth noting that although the use of weak learners is theoretically sufficient to achieve good performance, in practice, for various reasons, such as the desire to use fewer individual learners, there is rather more focus on stronger learners [9]. The theory of ensemble learning is also applicable to neural networks, generally regarded as being within the class of strong learners.

B. LEARNING ON SMALL OR NOISY DATASETS

1) SMALL DATASET: ALSO VALUABLE

a: IS TRANSFER LEARNING SUITABLE FOR USE HERE?

Deep learning restricts its development in terms of massive training data, flexible model and sufficient computing capacity. For example, the AlexNet network with 8-layers of neurons achieved a 16% error rate on the ImageNet dataset in 2012, and an iterative operation of the network would require approximately 1.4 GFLOP of computation. The residual network (ResNet), proposed by Microsoft, and using 152 layers of neurons, achieved a 3.5% error rate on the same dataset in 2015. One iteration consumes around 22.6 GFLOP, 16 times that of AlexNet. In today's production environments, network models for the processing of images, speech, and natural language processing, such as speech-to-text, machine translation, etc., even if given a considerable amount of computing resources, many still take several weeks to complete training, and are relatively expensive to deploy at scale, particularly as data changes.

In many situations, the amount of data is insufficient to perform end-to-end learning. When the amount of data is insufficient, one of the solutions is to use a model that is pre-trained on the large dataset, then to fine-tune it on a new dataset [10]–[14]. This type of transfer learning, typified by the MDNet architecture [15], which won the 2015 VOT challenge, is now commonly used.

Nevertheless, transfer learning of this form requires the source and target datasets to have a certain similarity. At the same time, for the noisy labels problem, we only know that the labels of the dataset have errors, but we do not know which are wrong. This makes the process of fine-tuning affected by these errors. Indeed, Xiao *et al.* [16] found it is better to use noisy data to train a new model from scratch than to only use the clean data to fine-tune. Given the above considerations, transfer learning is not the optimal solution to our problem.

b: SMALL BUT NEED TO BE ENOUGH: A DISCUSSION

In some cases, even if we have the time and effort, we simply cannot collect sufficient numbers of image examples to train a classification network. Examples might be training a deep model to recognize a rare biological specimen, training a network to interpret data from a new form of sensing device, or diagnosis of unusual biomarkers for unusual clinical cases: the problem of small data is sometimes unavoidable. But the word ''small'' has different definitions for datasets of different complexity. Despite the theoretically correct but unsatisfying mantra that ''more data is better'', Warden [17] suggested a useful rule of thumb. And Du *et al.* [18] also explored the question of ''how many samples are needed to

FIGURE 1. Process of training and dataset division. The neural networks are retrained at every stage. The hold-out set, denoted data_0 is the test set. The performance of the networks is always judged based on the score on data_0.

learn a convolutional neural network?'', detailing the relationship between the number of datasets, the size of the training stride, the filter size and model performance.

2) NOISY DATASETS: HOW TO USE THEM

a: FIND OR CORRECT THE NOISY LABELS

Techniques based on attempting to correct labels are one possible approach. Nicholson *et al.* [19], [20] proposed two algorithms to deal with the problem. One utilizes self-training to re-label data, called Self-Training Correction (STC). Another is a clustering-based method, which groups instances together to infer their ground-truth labels, is called Cluster-based Correction (CC). After a series of meticulous experiments, they concluded that CC performs the best. And Zhang *et al.* [21] proposed a novel algorithm termed ''adaptive voting noise correction (AVNC)'', used to precisely identify and correct noisy labels. Another approach makes use of the DNN; Reed *et al.* [22] trained the DNN with bootstrapping, whilst Daiki Tanaka *et al.* [23] used a joint optimization framework, which can find and correct the noisy labels. However, firstly, this method still requires the availability of large amounts of data as a premise, leading to relatively poor performance with small datasets. At the same time, modifying the labels is unsatisfactory: for many difficult samples, it is hard to determine whether the label [24] is correct or erroneous.

b: USE LOSS FUNCTIONS THAT ARE MORE APPROPRIATE FOR DATASETS

For a small number of studies, the loss function that is more suitable for datasets is used. For datasets with noise, ramp loss [25] and unhinged loss [26] can be adopted [27]. In the field of deep learning, Ghosh *et al.* [28] used mean square error and absolute error for noise-tolerance, but both modified loss functions and the method of correcting the labels are limited by the size of the original dataset.

c: OTHER IDEAS

Dropout regularization [29] and lifelong learning [30] are also promising strategies to deal with label noise. Generally speaking, lifelong learning is the accumulation of knowledge that can be used in future tasks. Lifelong learning should include the ability to apply knowledge and adapt to new situations. There are many challenges in the field of lifelong learning: (1) a lifelong learning algorithm might not have any indication of whether knowledge being acquired is right or not; (2) knowledge of the past is not necessarily applicable as a domain changes; (3) it is uncertain how accumulated knowledge should be represented, or (4) how to combine long-term learning with other aspects of artificial intelligence (e.g. search-based) to make a relatively complete system.

C. SEMI-SUPERVISED LEARNING

In the traditional supervised learning, the learner trains a large number of marked samples to build a model to predict the marking of unobserved cases. However, with the rapid development of data collection and storage technology, it has become easy to collect large amounts of unlabeled data; in contrast significant resource and manpower and is required to accurately annotate data. Semi-supervised learning [31] provides a way to take advantage of unlabeled samples. Semi-supervised learning has a wide range of applications in object detection [32], biomedical signal processing [33] and image processing.

1) PSEUDO-LABEL: THE SIMPLE AND EFFICIENT SEMI-SUPERVISED LEARNING METHOD FOR DEEP NEURAL NETWORKS

Pseudo labels [34], [35] are a simple and effective semi-supervised learning method. The pseudo-labels are not the original labels. They represent the class which has the maximum predicted probability at every weight update, and are used as if they were true labels. Experiments in this paper suggest that neural networks can process imperfectly labeled data quite well, so pseudo-labels can be exploited even if they are not always entirely correct. In particular, in the face of datasets with incorrect labeling – often met in real application environments – the quality of pseudo-labels may be higher than the quality of the original dataset itself, so the final recognition accuracy can be greatly improved.

2) GENERATE ANTAGONISTIC NETWORK: AN ESAY WAY TO GET DATA

In recent studies, there has been increased use of Generative Adversarial Networks (GANs) [36] for semi-supervised learning. The generator of a GAN produces new data samples which are similar to the original ones through learning the sample features. The new sample is fed into the discriminator to determine whether the new sample is sufficiently similar to the training set sample. Radford *et al.* [37] proposed the Deep Convolutional GAN (DCGAN), using it to synthesize many image samples of plausible bedrooms. Despite the possibility of producing new class-conditional samples, this approach has high requirements on the accuracy of the original dataset. If there are errors in the original sample labels in the dataset, errors will not be corrected. In addition, the newly generated samples do not actually exist, and in some contexts (e.g. medical image data synthesis) using synthetic samples as a form of data augmentation to help build an inference model might even be considered dangerous.

3) SYNTHETIC DATA: GUIDE THE NEURAL NETWORK TO LEARN THE KEY KNOWLEDGE

Similar to the GAN network, synthetic data [38] can also be used as a source of new samples. While generating new visual samples, for example, backgrounds, texture, illumination, angle and other elements that can affect the identification results can be added. For example, a technique referred to as domain randomization has been used to better enable networks to learn to ignore aspects of a scene that are not of interest to a specific task.

III. RECURSIVE ENSEMBLE LEARNING APPROACH

We will discuss how to use the combination of semi-supervised learning and ensemble learning to make neural networks achieve better performance on small numbers of datasets with precision and datasets with noise, respectively.

A. BAGGING STRATEGY AMONG NEURAL NETWORKS

Bagging [39] is perhaps the best known example of ensemble learning. It uses the bootstrapping method to extract n training samples from the original samples in each round, and uses one training set to develop a model each time. For classification problems, voting is used to obtain classification results for k models obtained in the previous step. For regression problems, the mean value of the above model is calculated as the final result. From the perspective of bias and variance, bagging is mainly concerned with reducing variance, so it is more effective in non-pruning decision trees, neural networks and other learning devices susceptible to sample disturbance. The algorithm description of Bagging is shown as follows:

Algorithm 1 Algorithm 1 for Bagging

Input:

Training set, $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$
Number of iterations, T
Learning rule, ς

1: **for** $t = 1, 2, 3, \ldots, T$ **do**
2: $h_t = \varsigma(D, D_{bs})$
3: **end for**

Output: $H(x) = \arg\max_{y \in Y} \sum_{t=1}^{T} (h_t(x) == y)$

Algorithm 2 The Improved Collaborative Training Algorithm With Two Networks

Input:

Labeled sets, $\text{Data}_1(x_1, y_1)$, $\text{Data}_0(x_0, y_0)$
Unlabeled sets, $\text{Data}_2(x_2) \ldots \text{Data}_m(x_m) \ldots$
Neural networks, η_1,η_2
Number of unlabeled sets T
Voting strategy ς

1: **for** $t = 1, 2, 3, \ldots, T$ **do**
2: $\eta_1(Data_1, \ldots Data_t; Data_{t+1}) -> y_{t+1\_1}$,
$\eta_2(Data_1, \ldots Data_t; Data_{t+1}) -> y_{t+1\_2}$,
3: $\varsigma(y_{t+1\_1}, y_{t+1\_2}) -> y_{t+1}$
4: Test accuracy a_t in $\text{Data}_0(x_0, y_0)$
5: **end for**

Output:

Labels, $y_2, \ldots y_m, \ldots$
Accuracy, $a_1, \ldots a_{m-1}, \ldots$

B. DISAGREEMENT-BASED LABELING METHODS

Disagreement-based methods use multiple learners, and the disagreement between the learners is critical to the utilization of unlabeled data. Although the process of collaborative training is simple, a theoretical analysis surprisingly shows that if the two views are sufficient and the conditions are independent, then the generalization ability of the weak classifier can be improved to any level by the unlabeled data through collaborative training [40]. For different neural networks, it is difficult to find two or several neural networks that are completely independent for collaborative training, because of the implicit dependence on (usually) the same source of training data. However, the experiment in Section 4 shows that even if our neural network is not completely independent, we can still find a strategy to improve the performance of a neural network. The improved collaborative training algorithm with two networks is shown in Algorithm 2:

There are two forms of data, corresponding to labeled and unlabeled samples. The total number of labeled samples is N_labeled, and the total amount of unlabeled samples is N_unlabeled. Unlabeled data are divided into several parts, data_2, data_3, …, data_m.… The labeled data are also divided into two parts, one (data_1) for training and the other (data_0) for testing the accuracy of the neural network; data_0 will never participate in training. First, we train the networks with the small training set (data_1). The structures of the networks and parameters of last 10 epochs are saved before the end of training, and we use these to predict the labels of data_2. The outputs of the networks are the probability value that each sample belongs to each of the possible classes. We take the most probable class as the label of the sample. For example, we have the FCN and VGG network. For the same sample, we will get 10 predicted labels from each network under different parameters; for a total of 20 predicted labels. Then, a voting process is applied with these labels, providing the final pseudo-labels of data_2. The performance of the network is measured on data_0. Next, data_2 (with its pseudo-labels) are combined with the training set for retraining the networks from scratch. The new training set is data_2 with its pseudo-labels and data_1 with its labels. This procedure is repeated many times until all the unlabeled data are labeled. At the end of the process, we obtain pseudo-labels of all unlabeled data (data_2, data_3, …, data_m…), and note the accuracy of the neural network after each round of this process.

C. VOTING STRATEGY COMPARISON

The combination of learners may benefit from the following three aspects [41] :

1. From a statistical point of view, as the hypothesis space of learning tasks is often large, there may be multiple hypotheses that achieve the same performance on the training set. At this point, using a single learner may lead to poor generalization ability.

2. From the perspective of calculation, the learning algorithm tends to fall into local minima, and the risk of falling into bad local minima can be reduced by combining the learners after multiple operations.

3. From a presentation point of view, the actual assumption of some learning tasks may not be in the hypothetical space considered by the current learning algorithm, at which point the use of a single learner is guaranteed to be invalid. By combining multiple learners, it is possible to learn better approximations due to the expansion of the corresponding hypothesis space.

The following will introduce several different combination strategies, which will be used in the experiment in section 4 to achieve better results.

1) PLURALITY VOTING: NONDISCRIMINATORY VOTING

The plurality voting takes the class label which receives the largest number of votes as the final winner. It can be represented as

$$H(x) = c_{\arg\max_j \sum_{i=1}^{T} h_i^j(x)} \tag{1}$$

where $h_i^j(x)$ is the output of the same object from different classifiers, and $H(x)$ is the final output for this object. In this article, we use this method to determine the predictive labels. We chose the label with relatively more occurrences as the final prediction label of the network. If there are multiple labels to get the highest number of votes at the same time, then one of them is selected randomly.

2) WEIGHTED VOTING: THE OPINIONS OF GOOD STUDENTS SHOULD BE VALUED

In practical problems, the performance of each classifier is uneven. In other words, we can properly consider the weight of the good classifier and the weight of the weak classifier to balance the final output. The output class label of the ensemble is

$$H(x) = c_{\arg\max_j \sum_{i=1}^{T} w_i h_i^j(x)} \tag{2}$$

where w_i is the weight assigned to the classifier h_i, $H(x)$ is the final output. In practical applications, the weights are often normalized and constrained by $w_i \geq 0$ and $\sum_{i=1}^{T} w_i = 1$ [8].

3) COMPARISON OF PLURALITY VOTING AND WEIGHTED VOTING: HOW TO CHOOSE THE MOST APPROPRIATE STRATEGY

Let $\ell = (\ell_1, \ldots, \ell_T)^T$ denote the outputs of the individual classifiers, where ℓ_i is the class label predicted for the instance x by the classifier h_i, and let p_i denote the accuracy of h_i. There is a Bayesian optimal discriminant function for the combined output on class label c_j, i.e.,

$$H^j(x) = \log(P(c_j)P(\ell|c_j)) \tag{3}$$

Assuming that the outputs of the individual classifier are conditionally independent, then it flows that

$$\begin{aligned} H^j(x) &= \log P(c_j) + \sum_{i=1}^{T} \log(P(c_j)P(\ell_i|c_j)) \\ &= \log P(c_j) + \log\Big(\prod_{i=1,\ell_i=c_j}^{T} P(\ell_i|c_j) \prod_{i=1,\ell_i\neq c_j}^{T} P(\ell_i|c_j)\Big) \\ &= \log P(c_j) + \log\Big(\prod_{i=1,\ell_i=c_j}^{T} p_i \prod_{i=1,\ell_i\neq c_j}^{T} (1-p_i)\Big) \\ &= \log P(c_j) + \sum_{i=1,\ell_i=c_j}^{T} \log\frac{p_i}{1-p_i} + \log\sum_{i=1}^{T} \log(1-p_i) \end{aligned} \tag{4}$$

Since $\sum_{i=1}^{T} \log(1-p_i)$ does not depend on the class label c_j, and $\ell_i = c_j$ can be expressed by the result of $h_i^j(x)$,

the discriminant function can be reduced to

$$H^j(x) = \log P(c_j) + \sum_{i=1}^{T} h_i^j(x) \log \frac{p_i}{1-p_i} \tag{5}$$

whereas the optimal weights for weighted voting satisfy

$$w_i \propto \log \frac{p_i}{1-p_i} \tag{6}$$

This means that the weight we obtain should be proportional to the score of each classifier [8].

In practice, however, each classifier cannot be completely independent, so this assumption is not appropriate to some problems. For example, for the samples that are difficult to identify, every neural network will generate ambiguity on this sample; for simple samples, all the neural network will not make mistake. This leads to situations where classifiers are highly correlated. Therefore, the above problems should be fully considered in dealing with specific problems. Sometimes the plurality voting is better than the weighted voting. In the subsequent experiments, we will see that the weighting method should be selected according to the network characteristics.

D. NETWORK EVALUATION CRITERIA AND PRUNING

Since the added labels are generated by recursive ensemble learning approach, noise in them can also affect the next step of the network learning. The continuous addition of noisy labels will reduce the rate of network performance improvement, and eventually the network will no longer improve in performance. The measure of improvement in a network is

$$L = \max(l_1, l_2, \ldots l_n) \tag{7}$$

l_i can be expressed as

$$l_i = l_{i-1} + |k_i|\, k_{i+1}, \quad l_1 = k_1 k_2 \tag{8}$$

where k_i is the rate of progress, and at least three points should be used to avoid the impact of network fluctuations. The network should be early stopped in the moment L, L is the global maximal point of each network's recognition rate. As can be clearly seen in Fig. 7, for the green line, L is the moment when the amount of data reaches 40,000; for the blue line, L is the moment when the amount of data reaches 50,000. Finally, we compare the highest accuracy rate of different networks and select the most suitable networks.

IV. EXPERIMENTS

A. DATASETS

Cats vs. Dogs: the dataset consists of training data and test data. The training data contains pictures of cats and dogs. We took 10,000 images for training and 2,500 images for testing to determine how the noise rate influences the recognition of the neural network.

CIFAR-10: the original dataset has 50,000 training data and 10,000 test data(data_0). In order to measure performance in datasets with partial labels in a meaningful way, we removed the labels of 40,000 data from the training set of CIFAR-10. This mimics the setting in which one has partially labeled data, but allows performance assessment against that of the fully-labeled dataset. The 40,000 removed label data are divided into data_2(10,000 samples), data_3(20,000 samples) and data_4(10,000 samples). The 10,000 samples (or data) with labels preserved are referred to as data_1.

B. TYPES OF NOISE AND HOW IT IS INTRODUCED

Noisy labels are mainly divided into two types, one is symmetric noise. Symmetric label noise is as follows:

$$y_i = \begin{cases} y_i^{GT} & \textit{with the probability of } 1-r \\ \textit{random one} - \textit{hot vector with the probability of } r \end{cases} \tag{9}$$

And the other is the asymmetric noise. Discussion of this type of noise is in [42]. This type of label noise is primarily for confusing types. TRUCK $\rightarrow$ AUTOMOBILE, BIRD $\rightarrow$ AIRPLANE, DEER $\rightarrow$ HORSE, CAT $\leftrightarrow$ DOG. Transitions are parameterized by $r \in [0, 1]$ such that the probabilities of ground-truth and inaccurate class correspond to $1-r$ and r, respectively [23].

It should be noticed that asymmetric noise is used in Section IV.E.5, and the symmetric noise is use in Sections IV.E.2, IV.E.4, IV.E.6, IV.E.7 and IV.F.

C. IMPLEMENTATION

We did all of the following experiments using the Keras framework eiminated. We used a Titan Xp, and an i7-7700k CPU.

We use the FCN (Full Convolutional Neural Network). We used RMSProp (lr=1e-4, decay=1e-6, clipnorm=0.1) as the optimizer, and a batch size of 128.

The use of data in CIFAR-10 is in this way: we first use 10,000 data(data_1) to apply pseudo-labels to another 10,000 data(data_2), and then used this 20,000 data samples to label 20,000 data (data_3), resulting in 40,000 labeled samples, of which 30,000 have pseudo-labels. Next, we used 40,000 samples to label the remaining 10,000 samples (data_4). At the end, we obtain 40,000 samples with pseudo-labels and 10,000 data(data_1) with the original labels intact. We evaluated the performance of the neural networks on data_0, and finally we also use the 50,000 data (40,000 data with pseudo-labels) to evaluate the performance of the neural networks on data_0. During the experiment, the data with labels removed are not used; performance evaluation is done on purely on data_0 to assess the final recognition rate.

In the Table 4, 5, 7, 8 and 9, there are two variants to consider for the recognition results: one is ''FCN+VGG'', and the other is ''FCN and VGG''. The data used are the same. The difference is that the former refers to the recognition result of the network in front of the symbol ''+'', whereas the latter is the final result of voting after the two networks get their own results, respectively. It means that we first get the result of FCN and the result of VGG, the outputs of the

TABLE 1. Error tolerance rate with different label noise levels.

Noise Rate(%)	0	20	40	50
Test Accuracy(%)	82.00	76.4	66.16	50

network are the labels, and then we use plurality voting to get the final result. Both outcomes are significant.

In addition to the above FCN network, we also used the three networks of VGGNet [43], RES-20 and RES-29 [44], [45]. The batch sizes are both 128 and the optimizer is Adam.

D. RECOGNITION RATE ON CATS VS. DOGS

In order to determine the influence of the proportion of noisy labels on network recognition rate, different proportions of label noise are set in this two-class dataset, namely 0%, 20%, 40% and 50% respectively. The recognition rate is shown in Table 1.

It can be concluded that noisy labels do have a small effect to the final recognition rate, but as long as the correct labels are still in a relative majority (for example, in the 10 classification problem, the proportion of noise in each category is less than 10%), the network will eventually perform significantly better than chance. This preliminary conclusion provides the basis for the following experiments.

E. RECOGNITION RATE ON CIFAR-10

1) DATA AMOUNTS: THE MORE, THE BETTER

We compare the test accuracy for the 10,000 data and 50,000 data in CIFAR-10 data. With only 10,000 labeled data, the average test accuracy of the network 10 times is 71.73%. With 50,000 labeled data, the recognition rate can reach nearly 86%. Both of these use VGGNet with the same structure. This indicates that an increase in data volume improves the recognition rate of the neural network.

2) DILEMMA: MORE NOISE DATA OR LESS CORRECT DATA?

Here, we use 10,000 correct data of CIFAR-10 to compare with 30,000 data containing noise (the noise rate is 30%). The latter is also trained by VGGNet, and the recognition rate is 74.61% on average, which is still significantly higher than 71.73% of 10,000 clean data. Therefore, we can draw a conclusion: when the data volume increases significantly, the quality requirements for the data can be reduced. Although it is not possible to achieve a recognition rate attainable with 30,000 samples with correct labels, the conclusion is still of significance for some of the situations mentioned in Section I.

3) RECOGNITION IN A DATASET WITHOUT NOISY LABELS: WILL THE STRATEGY WORK?

As a result of IV.E.1 and IV.E.2, we can assume that there are some strategies that can be adopted to mark unlabeled data through neural networks (even if not completely correctly),

TABLE 2. The results of a single neural network.

Data amount	10,000	20,000	40,000	50,000
Test Accuracy(%)	71.73	73.20	74.11	74.98

TABLE 3. The original accuracy.

Neural	Test Accuracy(%)
FCN	73.11
VGG	71.73
RES-20	70.34
RES-29	69.15

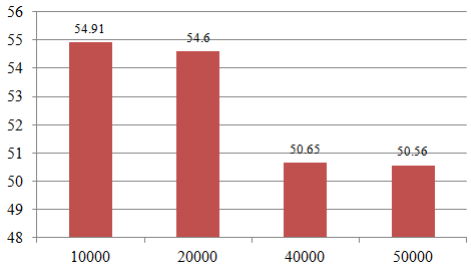


FIGURE 2. The accuracy with weighted voting strategy (%).

so as to achieve better results than by using only a small amount of data. There are two options available.

Scheme 1: the results of a single neural network are used to label the remaining data. At the same time, the original data and the data labeled by the neural network are used together as training sets in the next run to train the neural network. The results are shown in Table 2.

Therefore, we can conclude that self-labeling network can improve the final recognition rate with increasing quantities of data, but the improvement is not obvious.

Scheme 2: multi-network collaborative labeling is adopted, and plurality majority voting is adopted for the final results. The results are applied to perform continuous training, together with the original data. At the beginning of the training process, there are only 10,000 correctly marked data, and the recognition rate is as shown in the Table 3.

In Section III, we mentioned that choosing different strategies will affect the final results of the network. The key is the degree of similarity between networks. We know that the network is similar through experiments. If the method in III.C.2 is adopted, the final results are shown in Fig. 2. Here we use two relatively simple convolutional neural networks. The recognitions rate of the two networks are also close(both about 55%).

It can be seen that the recognition rate of the network decreases with increasing quantities of data, which is obviously not the result we want to get. The subjective reason for the declining accuracy is the declining quality of the labels as the quantity of data increases. Due to the high degree of similarity between networks, difficult samples are harder to

TABLE 4. The single recognitions of networks after using the new labeled data generated by plurality voting.

Method	Accuracy (%)		
Data Size	20,000	40,000	50,000
FCN+VGG	73.48	76.68	78.55
FCN+RES-20	74.63	76.59	77.73
FCN+RES-29	74.38	76.65	78.05
VGG+FCN	75.10	76.76	77.52
VGG+RES-20	74.96	77.12	78.02
VGG+RES-29	75.46	77.23	78.20
RES-20+FCN	70.94	72.95	75.80
RES-20+VGG	73.28	75.26	75.53
RES-20+RES-29	72.44	73.64	75.37
RES-29+FCN	72.17	75.07	75.29
RES-29+VGG	71.85	73.85	75.57
RES-29+RES-20	72.08	72.73	74.75

TABLE 5. The accuracy of collaboration between two networks (by voting).

Method	Accuracy(%)			
Data Size	10,000	20,000	40,000	50,000
FCN and VGG	77.80	78.13	80.69	80.50
FCN and RES-20	78.10	78.90	79.47	80.31
FCN and RES-29	79.20	79.00	79.31	80.20
VGG and RES-20	78.53	79.94	80.87	81.00
VGG and RES-29	79.05	80.57	81.02	80.91
RES-20 and RES-29	79.04	79.99	79.88	80.35

correctly classify, whereas the weight of simple samples is bound to be larger due to the better performance in them. For example, the outputs of the last 10 times of neural network for the same sample is:

$$\ell = (3, 3, 3, 3, 3, 3, 3, 8, 8, 8)$$

We assume the recognition rate of sample 3 is poor, and for sample 8, the recognition rate is good. So, the output is likely to be 8. With increasing amounts of data, the data which should be labeled as category 3 will be less well labeled. The number of data of each type in the test set is same, and the unbalanced sample in the training and the noisy labels leads to the reasons for the line graph.

If the method in III.C.1 is adopted, the results are shown in Table 4.

Table 5 shows the new results obtained by the network in pairs after each training. It is worth noting that the recognition rate is significantly improved compared with the single network.

From Table 5, it can be clearly seen that the cooperation between two networks achieves a significant improvement in performance. At the same time, the new labels generated by this cooperation are used for the original single network, which is also significantly better than the performance on its own (Scheme 1). This is because the errors made by the neural network will accumulate, to a certain extent, during the labeling process, whereas the cooperative training will tend to avoid this situation. As can be seen from Fig. 3 the recognition

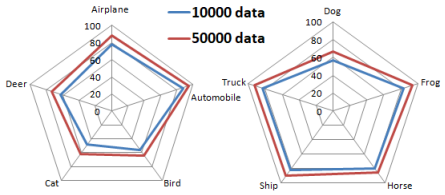


FIGURE 3. The accuracy of each category with correct labels (%).

TABLE 6. Recognition rate for dataset with noisy labels.

Network	Accuracy(%)
FCN	54.21
VGG	48.57
RES-20	37.63
RES-29	40.61

TABLE 7. The recognition of single networks after using the pseudo-labels data generated by plurality voting (with noisy labels).

Method	Accuracy(%)		
Data Size	20,000	4 0,000	50,000
FCN+VGG	58.05	58.90	60.15
FCN+RES-20	57.87	59.11	59.59
FCN+RES-29	57.68	58.08	58.80
VGG+FCN	57.83	58.82	59.23
VGG+RES-20	57.80	58.65	58.31
VGG+RES-29	57.55	59.07	58.67
RES-20+FCN	56.21	57.10	56.96
RES-20+VGG	53.95	56.03	57.63
RES-20+RES-29	51.85	56.27	54.31
RES-29+FCN	54.98	57.77	57.82
RES-29+VGG	53.71	56.10	57.94
RES-29+RES-20	51.62	52.61	55.23

rate of each type of data will be improved as the quantity of labeled data increases.

4) THE RECOGNITION IN SYMMETRIC NOISE DATASET: NOISY LABELS ARE ALSO USEFUL

It has been mentioned in IV.D that if the proportion of noisy labels does not account for the majority, the network can still achieve better recognition results. Here, we choose datasets with a proportion of 50% of incorrect labels, and the initial number of samples is still 10,000. Table 6 shows the recognition rate of four networks on this dataset.

For the above cases, only the FCN network exceeds the artificially marked recognition rate (higher than 50%); the other three networks cannot exceed the artificially marked recognition rate, but still exceed 10% (In the CIFAR-10 classification problem, the probability of a uniform randomly guess picking the correct answer is 10%.), so we can use the above output results. We used the same method as IV.E.3. Table 7 presents the recognition of a single network with noisy labels.

After cooperative labeling, the results are shown in Table 8. The performance improvements for individual networks

TABLE 8. The accuracy of collaboration between two networks with noisy labels (by voting).

Method	Accuracy (%)			
Data Size	10,000	20,000	40,000	50,000
FCN and VGG	60.19	60.86	61.14	61.58
FCN and RES-20	60.59	60.94	61.08	60.45
FCN and RES-29	60.47	60.67	60.46	60.27
VGG and RES-20	58.66	61.37	60.79	60.57
VGG and RES-29	59.09	61.12	61.24	60.99
RES-20 and RES-29	57.80	58.36	60.76	57.37

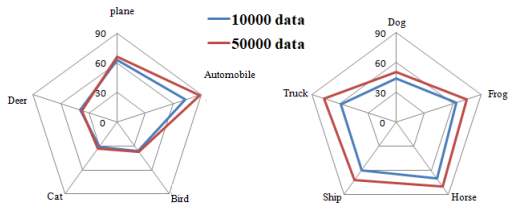


FIGURE 4. The accuracy of each category with noisy labels (%).

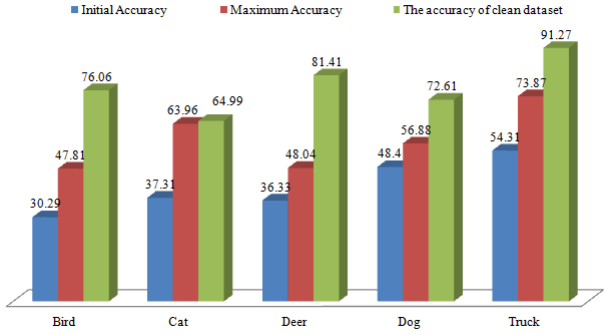


FIGURE 5. The accuracy of five categories with asymmetric Noise (%).

range from 11% to 53%. For a network with poor performance, the collaborative training can greatly improve performance. For a network with good performance, the results of cooperative training suggest the potential to be further improved.

For a single object, the same conclusion can be obtained as shown in Fig. 4. Among the 10 objects in the dataset, the recognition rate of nine objects has been improved, and there is no significant decline in deer category. It proves that the network cooperation proposed by us can effectively improve recognition ability.

5) ASYMMETRIC NOISE: NOT A SIGNIFICANT PROBLEM

Regarding asymmetric noise, our main concern is whether the recognition rate of the wrong type of data can be improved with the implementation of the proposed method. We will compare the initial accuracy (10,000 samples) and the maximum accuracy, and compare it with the recognition rate on a clean dataset (50,000 samples). The noise ratio is 30%.

It can be clearly seen from Fig. 5 that although the five types of data(containing asymmetric noisy labels) do not reach the recognition accuracy of 50,000 clean data, after using our method, the recognition rate of each type of data has some degree of improvement compared with the original.

TABLE 9. Recognitions of multi-network cooperation.

Networks	Accuracy (%)
FCN and VGG	61.85
FCN and RES-20	60.45
FCN and RES-29	60.27
VGG and RES-20	60.57
VGG and RES-29	60.99
RES20 and RES-29	57.37
FCN, VGG and RES-20	62.20
FCN, VGG and RES-29	61.88
FCN, RES-20 and RES-29	60.86
VGG, RES-20 and RES-29	61.39
All four networks	62.19

6) MORE NETWORK: THERE IS STRENGTH IN NUMBERS?

Through the experiment in IV.E.4, we showed that the result of cooperation of any two networks is better than results of a single network; we postulated that the result of cooperation among any three networks, or any four networks, should be better than some results of the cooperation between two networks. For this final verification set, the results are in Table 9.

What seems to arise from the above table is that more networks may give better results, but too many networks may lead to results that are counterproductive. In the experiments of this paper, the results of cooperation amongst four neural networks is not as good as the cooperation of three. If there are more neural networks involved in the collaboration, a network-level pruning strategy will help us save time and computing resources.

7) COMPARISON: IS OUR METHOD REALLY EFFECTIVE?

To ensure that our experiments are reliable, we first experiment with the same neural network model as Xiao *et al.* [16] to highlight the promising results of this approach. It is worth noting that since our method relies on disagreement-based labeling methods, we need to use at least two neural networks. So, we chose another neural network (which is different from the CIFAR10-quick model; we use N1 to represent it) to help our experiments go smoothly. This neural network has similar overall recognition rate with CIFAR10-quick model (we use N2 to represent it) in [16]. We use C_1 to represent the results of network cooperation between N1 and N2 with the method in III.C.1. The results indicate that the proposed method represents an improvement. The results of the last column is that we used the FCN and VGG network and still used the same dataset. We used C_2 to represent these results. By comparison, as the depth of the network increases, the effect will also be better. The results are in Table 10.

F. COMPETITION AND DECISION

We believe that there is a tradeoff in the choice of networks. The criteria of how to choose the neural networks are the

TABLE 10. Comparison with other methods under symmetric noise.

Noise Level	CIFAR10-quick	[16]	[46]	C_1	C_2
30%	65.57%	69.81%	69.73%	74.5%	83.34%
40%	62.38%	66.76%	66.66%	72.66%	80.34%
50%	57.36%	63.00%	63.39%	70.99%	78.26%

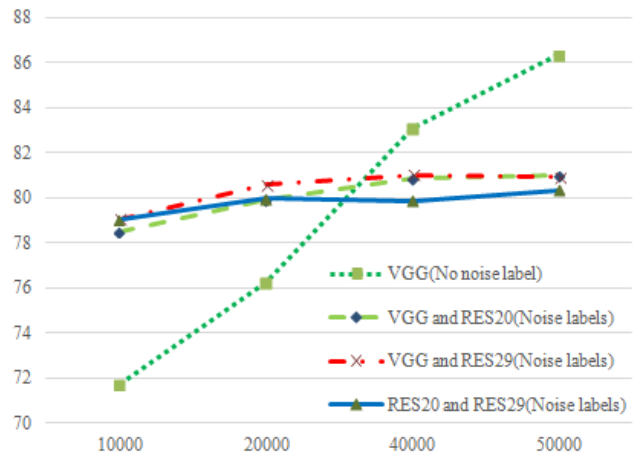


FIGURE 6. Comparisons of networks.

TABLE 11. Comparison of the time it takes to achieve the same accuracy.

Network	Time consumption
FCN or VGG	200s
RES20	550s
RES29	713s
Our approach (10,000 data)	**150s**

accuracy and the efficiency of the networks. We choose the results of cooperation among several networks with good results. With this approach, we are able to use 10,000 labeled data samples and other unlabeled data to achieve the same effect with about 30,000 samples with correct labels. Meanwhile, it can be seen from Fig. 6 that after we use the proposed cooperative method, we can need only 10,000 data samples to achieve a level of performance that originally used 25,000 samples. Using the hardware resource mentioned above, for achieve an accuracy of around 79%, and to achieve the same effect, the FCN or VGG network needed 200 seconds, RES-20 needed 550 seconds, RES-29 needed 713 seconds; the latter four also needed about 25,000 samples. Here, the residual network takes longer. One of the reasons is that it uses more data, and the other reason is that it is deeper. Under the premise of the same network structure, since we can achieve the same accuracy with less data, the time consumption is also significantly smaller. This can be seen by comparing the second and fifth rows of Table 11. This shows that the collaborative approach also provides an advantage in terms of efficiency.

At the same time, the performance of the network is affected by noisy labels, which sometimes produces a downward trend. We selected typical cases to be shown in Fig. 7, suggesting how to choose the most appropriate networks. The network represented by the blue line should be preserved, and the others should be pruned.

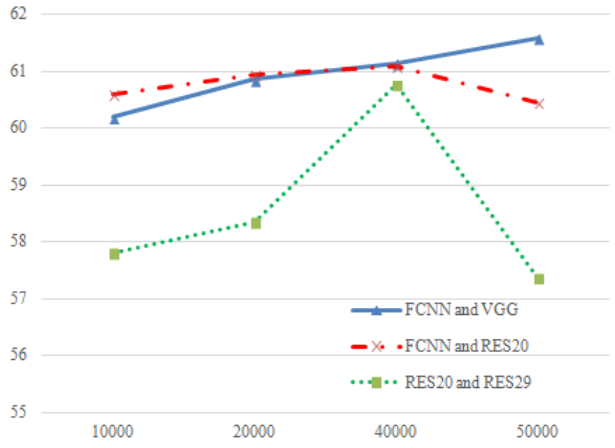


FIGURE 7. Several typical cases in which pruning strategies should be applied.

V. CONCLUSIONS

In this paper, we mainly discussed how to improve network performance for the case that only a small amount of labeled data and a large amount of unlabeled data is available, and in particular under the premise that the labeled data also contains noisy labels. The recursive ensemble learning approach performs well in the face of the problem that datasets contain noise, and greatly improves the recognition ability of the network, particular for one with weak performance on its own. At the same time, our experiment shows that the choice of neural networks and training strategy is also crucial.

REFERENCES

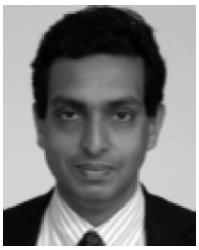
[1] D. F. Nettleton, A. Orriols-Puig, and A. Fornells, "A study of the effect of different types of noise on the precision of supervised learning techniques," *Artificial Intelligence Review*, vol. 33, no. 4, pp. 275–306, 2010.
[2] D. Guan *et al.*, "Improving label noise filtering by exploiting unlabeled data," *IEEE Access*, vol. 6, pp. 11154–11165, 2018.
[3] Y. Wang *et al.*, "Iterative learning with open-set noisy labels," in *Proc. CVPR*, Jun. 2018, pp. 8688–8696. [Online]. Available: https://arxiv.org/abs/1804.00092
[4] J. Zhang, X. Wu, and V. S. Sheng, "Active learning with imbalanced multiple noisy labeling," *IEEE Trans. Cybern.*, vol. 45, no. 5, pp. 1095–1107, May 2015.
[5] J. Zhang, X. Wu, and V. S. Sheng, "Imbalanced multiple noisy labeling," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 2, pp. 489–503, Feb. 2015.
[6] G. R. Haffari and A. Sarkar. (2012). "Analysis of semi-supervised learning with the Yarowsky algorithm." [Online]. Available: https://arxiv.org/abs/1206.5240
[7] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. Workshop Challenges Represent. Learn. (ICML)*, vol. 3, 2013, p. 2.
[8] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. London, U.K.: Chapman & Hall, 2012, ch. 4, sec. 3, pp. 71–75

[9] Z.-H. Zhou, *Machine Learning*. Beijing, China: Tsinghua Univ., 2016, ch. 8, sec. 1, pp. 171–173.
[10] H.-W. Ng, V. D. Nguyen, V. Vonikakis, and S. Winkler, "Deep learning for emotion recognition on small datasets using transfer learning," in *Proc. ACM Int. Conf. Multimodal Interact.*, 2015, pp. 443–449.
[11] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "From generic to specific deep representations for visual recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2015, pp. 36–45.
[12] J. Donahue *et al.* (2013). "DeCAF: A deep convolutional activation feature for generic visual recognition." [Online]. Available: https://arxiv.org/abs/1310.1531
[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[14] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1717–1724.
[15] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4293–4302.
[16] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, "Learning from massive noisy labeled data for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2691–2699.
[17] P. Warden. *How Many Images Do You Need to Train a Neural Network?* Accessed: Dec. 14, 2017. [Online]. Available: https://petewarden.com/2017/12/14/how-many-images-doyou-need-to-train-a-neural-network/
[18] S. S. Du, Y. Wang, X. Zhai, S. Balakrishnan, R. R. Salakhutdinov, and A. Singh, "How many samples are needed to estimate a convolutional neural network?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 371–381.
[19] B. Nicholson, J. Zhang, V. S. Sheng, and Z. Wang, "Label noise correction methods," in *Proc. IEEE Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Oct. 2015, pp. 1–9.
[20] B. Nicholson, V. S. Sheng, and J. Zhang, "Label noise correction and application in crowdsourcing," *Expert Syst. Appl.*, vol. 66, pp. 149–162, Dec. 2016.
[21] J. Zhang, V. S. Sheng, T. Li, and X. Wu, "Improving crowdsourced label quality using noise correction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1675–1688, May 2018.
[22] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, "Training deep neural networks on noisy labels with bootstrapping," in *Proc. Workshop Int. Conf. Learn. Representation*, 2015. [Online]. Available: https://arxiv.org/pdf/1412.6596v3.pdf
[23] D. Tanaka, D. Ikami, T. Yamasaki, and K. Aizawa, "Joint optimization framework for learning with noisy labels," in *Proc. CVPR*, Jun. 2018, pp. 5552–5560. [Online]. Available: https://arxiv.org/abs/1803.11364
[24] I. Guyon, N. Matic, and V. Vapnik, "Discovering informative patterns and data cleaning," in *Proc. AAAI-94 Workshop Knowl. Discovery Databases*, 1996, pp. 181–203.
[25] J. P. Brooks, "Support vector machines with the ramp loss and the hard margin loss," *Oper. Res.*, vol. 59, no. 2, pp. 467–479, 2011.
[26] B. van Rooyen, A. Menon, and R. C. Williamson, "Learning with symmetric label noise: The importance of being unhinged," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 10–18.
[27] A. Ghosh, N. Manwani, and P. S. Sastry, "Making risk minimization tolerant to label noise," *Neurocomputing*, vol. 160, pp. 93–107, Jul. 2015.
[28] A. Ghosh, H. Kumar, and P. S. Sastry, "Robust loss functions under label noise for deep neural networks," in *Proc. AAAI*, 2017, pp. 1919–1925.
[29] I. Jindal, M. Nokleby, and X. Chen, "Learning deep networks from noisy labels with dropout regularization," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 967–972.
[30] B. LIU, "Lifelong machine learning: A paradigm for continuous learning," *Frontiers Comput. Sci.*, vol. 11, no. 3, pp. 359–361, 2017.
[31] X. J. Zhu, "Semi-supervised learning literature survey," *Comput. Sci., Univ. Wisconsin-Madison* vol. 2, no. 3, p. 4, 2006.
[32] D. K. Shin, M. U. Ahmed, and P. K. Rhee, "Incremental deep learning for robust object detection in unknown cluttered environments," *IEEE Access*, vol. 6, pp. 61748–61760, 2018.
[33] Q. She *et al.*, "Safe semi-supervised extreme learning machine for EEG signal classification," *IEEE Access*, vol. 6, pp. 49399–49407, 2018.
[34] D. H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. Workshop Challenges Represent. Learn. (ICML)*, vol. 3, Jun. 2013, p. 2.
[35] P. Bachman, O. Alsharif, and D. Precup, "Learning with pseudo-ensembles," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 4, 2014, pp. 3365–3373.
[36] Z. Dai, Z. Yang, F. Yang, W. W. Cohen, and R. R. Salakhutdinov, "Good semi-supervised learning that requires a bad GAN," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6510–6520.
[37] A. Radford, L. Metz, and S. Chintala. (2015). "Unsupervised representation learning with deep convolutional generative adversarial networks." [Online]. Available: https://arxiv.org/abs/1511.06434
[38] J. Tremblay *et al.*, "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," in *Proc. CVPR*, Jun. 2018, pp. 969–977. [Online]. Available: https://arxiv.org/pdf/1804.06516.pdf
[39] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
[40] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. 11th Annu. Conf. Comput. Learn. Theory*, 1998, pp. 92–100.
[41] T. G. Dietterich, "Ensemble methods in machine learning," in *Proc. Int. Workshop Multiple Classifier Syst.*, G. Patrini, A. Rozza, A. K. Menon, R. Nock, and L. Qu, Ed., 2000, vol. 1857, no. 1, pp. 1–15.
[42] G. Patrini, A. Rozza, A. K. Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2233–2241.
[43] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556
[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[45] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 630–645.
[46] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus. (2014). "Training convolutional networks with noisy labels." [Online]. Available: https://arxiv.org/abs/1406.2080

YUCHEN WANG received the B.S. degree in communication engineering from Shandong University, Shandong, China, in 2016, where he is currently pursuing the master's degree. His research interests include deep learning, pattern recognition, and computer vision.

YANG YANG received the Ph.D. degree from Shandong University, Shandong, China, in 2009, where he is currently with the School of Information Science and Engineering and became an Associate Professor, in 2017. His research interests include image analysis, video surveillance, and pattern recognition.

YUN-XIA LIU received the Ph.D. degree from Shandong University, in 2012. She is currently with the School of Information Science and Engineering, University of Jinan, Shandong, China, where became an Associate Professor, in 2015. Her research interests include multi-scale geometry analysis, wavelet analysis, and pattern recognition.



ANIL ANTHONY BHARATH (M'98) received the B.Eng. degree (Hons.) from the Department of Electronic and Electrical Engineering, University College London, London, U.K., in 1988, and the Ph.D. degree from the Department of Electrical and Electronic Engineering, Imperial College London, London, in 1993.

In 2005, he was appointed as the Reader in image analysis. In 2006, he was an Academic Visitor with the Signal Processing Group, Department of Engineering, University of Cambridge, Cambridge, U.K. Since 1991, he has been a Lecturer and then a Senior Lecturer in medical imaging with the Department of Bioengineering, Imperial College London. He has worked and published in the areas of medical image analysis, computer vision, and signal processing, and has worked as a Consultant to the imaging sciences industry, primarily in the fields of medical imaging and medical informatics. His current research interests include visual pattern recognition, signal and image representations, statistical signal models, and biological vision.

Dr. Bharath jointly presented the 2003 Rosen Lecture at the Royal Institution on Art and Imaging with the visual artist D. Fern.