TSS-Net: two-stage with sample selection and semi-supervised net for deep learning with noisy labels

X. Lyu, J. Wang, T. Zeng, X. Li, J. Chen, et al.

X. Lyu, J. Wang, T. Zeng, X. Li, J. Chen, X. Wang, Z. Xu, "TSS-Net: two-stage with sample selection and semi-supervised net for deep learning with noisy labels," Proc. SPIE 12509, Third International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI 2022), 125092F (12 January 2023); doi: 10.1117/12.2655832

SPIE.

Event: Third International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI 2022), 2022, Guangzhou, China

TSS-Net: Two-stage with Sample selection and Semi-supervised Net for deep learning with noisy labels

X Lyu, J Wang, T Zeng, X Li, J Chen, X Wang and Z Xu
College of Computer and Information, Hohai University, Nanjing, China
lvxin@hhu.edu.cn

Abstract

The significant success of Deep Neural Networks (DNNs) relies on the availability of annotated large-scale datasets. However, it is time-consuming and expensive to obtain the available annotated datasets of huge size, which hinders the development of DNNs. In this paper, a novel two-stage framework is proposed for learning with noisy labels, called Two-Stage Sample selection and Semi-supervised learning Network (TSS-Net). It combines sample selection with semi-supervised learning. The first stage divides the noisy samples from the clean samples using cyclic training. The second stage uses the noisy samples as unlabelled data and the clean samples as labelled data for semi-supervised learning. Unlike previous approaches, TSS-Net does not require specifically designed robust loss functions and complex networks. It achieves decoupling of the two stages, which means that each stage can be replaced with a superior method to achieve better results, and this improves the inclusiveness of the network. Our experiments are conducted on several benchmark datasets in different settings. The experimental results demonstrate that TSS-Net outperforms many state-of-the-art methods.

Keywords- deep learning, noisy labels, sample selection, semi-supervised learning

1. Introduction

Although DNNs have achieved tremendous success in computer vision tasks, such as classification [1- 2], object detection [3-4], and remote sensing[5-6]. However, it is extremely sensitive to noise, and its performance is degraded significantly with noisy datasets. Unfortunately, it is expensive and time-consuming to obtain high-quality data without noisy labels in practical applications, which hinders the application of deep learning.

Currently, there are several solutions for Learning with Noisy Labels (LNL): (1) robust loss functions, (2) sample label correction and (3) noisy sample selection.

Robust loss function. The approach concentrates on designing loss functions which are robust to noise [7-9]. Loss functions with robustness to noise are proposed in [7-9]. It is demonstrated that the methods improve the noise robustness of the DNN. However, they only perform well for the simple case. In addition, modifying the loss function leads to an increase in convergence time [10].

Label correction. In the solution, predictions of DNNs are used as a replacement or weighing of the original labels of the samples [11-13]. A bootstrapping loss function is proposed in [11] to generate training objectives using a combination of training labels and model predictions. The approach avoids directly modelling training datasets containing noise. A joint optimization framework of network parameters and labels is proposed in [12] to improve the robustness of the model to noisy labels. However, the predictions of DNNs are characterized by randomness and are not always accurate. As a result, the label correction methods perform weakly on datasets with a high proportion of noise [14].

Sample selection. The approach concentrates on selecting noisy data during training and preventing noise from participating in model optimization [15-17]. In the literature [15], an influence function is proposed to select samples which negatively affect the training during the training process. A small proportion of reliable samples are selected by using a priori knowledge in the literature [16-17], and the noisy labels are corrected to the real labels by inferring them using clean labels.

Sample selection methods are currently the most effective methods in LNL [14], such as CJC-Net [14], Co-teaching [18], JoCoR [19], O2U-net [20], CurriculumNet [21], MentorNet [22]. However, the method is affected by the cumulative

Third International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI 2022),
dited by Kannimuthu Subramanian, Proc. of SPIE Vol. 12509, 125092F
 · 0277-786X · doi: 10.1117/12.2655832

error of wrong selection noise. [16]. Moreover, the distribution of the dataset would be corrupted by deleting the noisy samples directly [8].

Sample selection and semi-supervised learning (SSL) are combined for addressing the above problems of sample selection, in the study. The labelled data and the unlabelled data are used as training data for the SSL. The properties of the labelled data are consistent with the clean samples in LNL. The properties of the unlabelled data match those of the noisy samples with label uncertainty, thus semi-supervised learning can be embedded well in LNL. In the study, we propose a Two stage Sample selection and Semi-supervised learning network, called TSS-Net. After our experiments, we show that the combination of sample selection and semi-supervised learning achieves excellent results with multiple datasets and different noise ratios. The main contributions of this study are as follows:

- A new framework for learning with noisy labels is proposed, named Two-stage Sample selection and Semi-supervised learning Network (TSS-Net), which combines sample selection and semi-supervised learning. The selected noisy samples are used as unlabelled data for semi-supervised learning. As an end-to-end noise learning network, the two stages of TSS-Net are independent of each other. It means that a more advanced network can be substituted for either of them, resulting in better performance, which improves the inclusiveness of our network.

- This study means a deeper work based on sample selection, further explores the link between sample selection and semi-supervised learning and combines two advanced methods of noisy learning. Different from previous sample selection methods, the noisy samples selected in the study are not discarded directly. It avoids the issue of sample selection destroying the distribution of the dataset.

- No additional auxiliary clean subsets are required for TSS-Net, and no specialized loss functions need to be designed which are robust to noise. Extensive experiments in Section 4 demonstrate its insensitivity to hyperparameters as well as to dataset types. The TSS-Net achieves better performance compared to previous studies, despite omitting the hyperparameter tuning process. Moreover, it is validated to achieve high accuracy on both synthetic datasets with noisy labels on real datasets, and numerous experimental results show that TSS-Net outperforms the state-of-the-art baseline.

2. Related works

As mentioned in the previous section, the existing methods are mainly to learn with noisy labels. In this section, we briefly introduce relevant studies on noise-selection methods, cyclic training, and semi-supervised learning.

2.1 Sample selection

Designing reasonable and reliable evaluation criteria to distinguish between clean and noisy samples is the essence of most sample selection methods. Loss, which is generated by the sample during training, is the most popular criterion. It has been shown that DNNs first learn about clean samples and then gradually fit noisy samples [23]. Thus, the loss of clean samples, in the earlier training epochs, is generally less than that of the noisy samples. Thus, the loss of a clean sample is, in general, less than that of a noisy sample in the early training epochs, and it is only in the later epochs that their loss values become indistinguishable. In general, the bigger the loss of a sample, the higher the probability of it being a noisy sample.

Co-teaching is proposed in [18], which employs two classifiers with identical structures and different initializations. Samples with small loss are selected by each model as clean samples and provided to the other network for learning. On this basis, JoCoR is proposed by [19] which adds a regularization term for the same loss function. Thus, the difference between the two base classifiers is reduced and thus reliable data is selected. CurriculumNet, proposed by [21], ranks the samples of the training set from easy to hard using image complexity. CurriculumNet, proposed by [21], ranks the samples of the training set from easy to hard by using image complexity and trains the DNNs according to the strategy of curriculum learning. MentorNet was proposed by [22] using a teacher-student network, with the teacher network selecting clean samples to provide for learning by the student network.

2.2 Cyclical training

The objective optimization function of DNNs may be multi-peaked, where multiple local minima are available in addition to the global minimum. Global minima, or better still local minima, result in better DNNs performance on a training dataset with only clean samples. However as shown in figure 1, when noisy samples are contained on the

training dataset, global minima instead imply that the model overfits the noisy samples, which reduces the accuracy of DNNs on clean samples.

DNNs is not susceptible to noisy labels on training when a bigger learning rate is used, and vice versa [12]. The reason for this phenomenon is that a large learning rate can, to some extent, skip the minimums fitted to noisy samples. When the DNNs drops into a local minimum, it can "jump out" of the current minimum and search for other minimums by suddenly increasing the learning rate.

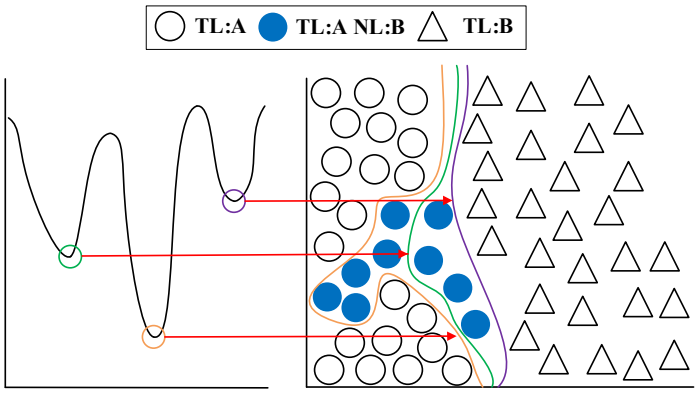

Figure 1. The left picture simulates the case where the DNNs objective optimization function is multi-peaked. The right picture indicates the fit of the model corresponding to the three local minima. The white circles indicate clean samples of category A, the white triangles indicate clean samples of category B, the blue circles indicate noisy samples of category A mislabelled as B.

The “overfitting to underfitting” (O2U) noise detection strategy is proposed based on this phenomenon. The DNNs is switched between overfitting and underfitting by cyclically adjusting the learning rate, and finally removing the top-k% samples by ranking the average loss of each sample from largest to smallest. CJC-Net combines the Cyclic training method with the Joint loss and Co-teaching strategies, simultaneously trains both networks and then executes the cyclic training strategy under a modified joint teaching approach according to these two pre-trained networks.

2.3 Semi-supervised learning

While collecting data is easy, the cost of collecting label data is very expensive. Semi-supervised learning aims to reduce the cost of manually annotating labels, using a small proportion of labelled data and a large proportion of unlabelled data to improve DNNs performance. SSL focuses on improving the performance of DNNs on unlabelled samples through consistency regularization and entropy minimization.

Consistency regularization [24-26] is based on the idea that for an input, the network may produce an output that is consistent with the original even if it is perturbed. Entropy minimization [27] encourages DNNs to produce confident predictions on unlabelled data, and predictions are expected to have low entropy, i.e. predictions should be close to one-hot. MixMatch is proposed in [28], which simultaneously utilizes entropy minimization, consistency regularization and Mixup data augmentation [29] to achieve state-of-the-art results. Moreover, the performance is much better than that of the sub-optimal algorithm.

3. Method

In this section, we introduce TSS-Net for solving DNNs for learning from noisy samples. The method is shown in figure 2. A two-stage framework is used in TSS-Net, with the first stage proceeding with sample selection work and the second stage proceeding with a semi-supervised learning stage. To improve the applicability and inclusiveness of TSS-Net, the two stages are decoupled in the study. The full algorithm flow is described in Algorithm 1.

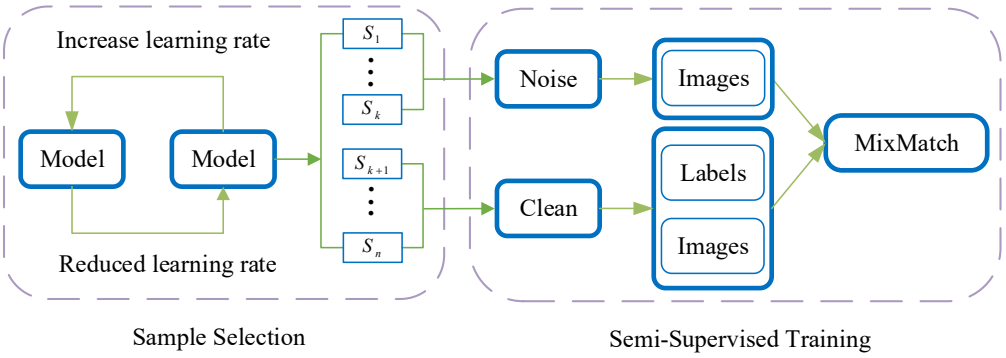

Figure 2. TSS-Net is divided into two stages for learning with noisy labels. The first stage is the sample selection stage and the second stage is the semi-supervised learning stage.

3.1 Sample Selection Stage

The primary task of the sample selection stage is to divide the dataset containing noisy samples into a clean sample subset $\boldsymbol{\mathcal{C}}$ and a noisy sample subset $\boldsymbol{\mathcal{N}}$. Existing methods, multi-round and multi-network methods [10], are primarily used to split noisy datasets directly into clean and noisy subsets. Co-training of multiple models is typically employed by multiple networks to reduce the probability of incorrect predictions by DNNs. However, this requires network structures specifically designed to be robust to noise [18-19], which increases the computational complexity, and endangers the inclusiveness of LNL. In addition, the assumption of a specific loss distribution is also used for sample selection. The method assumes that the loss distribution of clean and noisy samples obeys certain models [8], e.g. BMM (beta mixture model), and Gaussian Mixture Model (GMM). Nevertheless, the loss distribution differs for various datasets and noise ratios, thus this approach is limited in some way.

A multi-round approach, which is more universal and inclusive, is used in the study to perform sample selection. The sample selection stage performs supervised cyclic training on the entire training dataset containing noisy data to detect noisy data more accurately. The cyclic cosine annealing learning rate schedule is used in the study to decay the learning rate. The strategy is first proposed by [30] with the aim of using a hot restart approach to skip out of the local minimum and find a path to the global minimum. Unlike the purpose of the article, the goal of using the learning rate strategy is to make the DNNs jump out of the overfitting state for noisy samples to more precisely distinguish noisy samples from clean ones.

Algorithm 1:Training of TSS-Net

Input: the dataset with noise labels D^{noise}; the network parameters $\mathcal{W}$; noise rate k; sharpening temperature $\mathcal{T}$.

Step 1: Cyclical Training	**Step2: Semi-supervised learning**
Initialization: the network parameters $\mathcal{W}$; training epoch N;	**Initialization:** the network parameters $\mathcal{W}$;
for t =1,2,⋯,N:	$\mathcal{X}, \mathcal{U} = \mathcal{C}, (u_b; u_b \in \mathcal{N})$
compute learning rate η using Eq.(1);	fetch mini batch set $\mathcal{X}_m$ from $\mathcal{X}$, $\mathcal{U}_m$ from $\mathcal{U}$;
fetch mini batch set D_m from D^{noise};	**for** each input $\hat{x}_b$ in $\mathcal{X}_m$, $\hat{u}_b$ in $\mathcal{U}$:
for each input(x_i, y_i) in D_m:	$\hat{x}_b, \hat{u}_{b,k} = \text{Augment}(x_b), \text{Augment}(u_b), k \in (1, \dots, \text{K})$;
compute and record loss l_i;	$q_b = \text{Sharpen}(\bar{q}_b, \mathcal{T})$;
$\mathcal{W}^t = \text{SGD}(\mathcal{L}, \mathcal{W}^t)$;	**end for**
end for	$\mathcal{X}', \mathcal{U}' = \text{MixUp}((\hat{x}_b, p_b), \hat{u}_{b,k}, q_b)$;
record normalized average loss $\bar{l}_n$ of each sample;	$\mathcal{L}_{\mathcal{X}}, \mathcal{L}_{\mathcal{U}} = \text{MixMatch}(\hat{\mathcal{X}}, \hat{\mathcal{U}})$
end for	$\mathcal{L} = \mathcal{L}_{\mathcal{X}} + \lambda_u \mathcal{L}_{\mathcal{U}}$
rank all samples in descending order by $\bar{l}_n$;	
select top k% samples as noisy dataset $\mathcal{N}$,	
the rest as clean dataset $\mathcal{C}$;	

During the training process, the learning rate is decayed cyclically from the maximal value to the minimal value. At the beginning of each epoch, the learning rate is determined based on the value of the current epoch, the total epoch value, and the maximum and minimum learning rate. The learning rate adjustment formula is as follows:

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})\left(1 + cos\left(\frac{T_{cur}}{T}\pi\right)\right) \tag{1}$$

where $\boldsymbol{\eta_{max}}$ and $\boldsymbol{\eta_{min}}$ denote the maximum and minimum values of the learning rate, respectively, defining the range of learning rates, which are the same in each loop in the study. $\boldsymbol{T_{cur}}$ denotes the number of epochs currently executed. $\boldsymbol{T}$ indicates the total number of epochs.

The average loss of each sample is recorded during the training process. As shown in equation (2) to (3) the training loss of sample $\boldsymbol{i}$ is normalized when training to the *k*th epoch to reduce the negative impact caused by odd sample data. The loss of sample $\boldsymbol{i}$ is subtracted from the average loss of all samples in the current epoch, and the normalized loss is then accumulated with the loss values of the previous *k-1*epochs of sample $\boldsymbol{i}$.

$$\bar{l}_i = l_i - \frac{1}{N}\sum_{n=1}^{N} l_n \tag{2}$$

$$l_i = \bar{l}_i + \sum_{j=1}^{k-1} \bar{l}_j \tag{3}$$

After the completion of the sample selection stage, the entire training dataset is divided into a noisy set $\boldsymbol{\mathcal{N}} = \{(\boldsymbol{x_i}, \boldsymbol{y_i}); \boldsymbol{i} \in (\boldsymbol{1}, \ldots, \boldsymbol{k})\}$ and a clean set $\boldsymbol{\mathcal{C}} = \{(\boldsymbol{x_i}, \boldsymbol{y_i}); \boldsymbol{i} \in (\boldsymbol{k+1}, \ldots, \boldsymbol{N})\}$ by sorting the samples in descending order according to the cumulative loss. Where k denotes the noise rate of the dataset, which [31] shows that the noise rate is easy to estimate.

3.2 Semi-Supervised Training Stage

The primary task of the stage is to perform unsupervised learning with both noisy sets and clean sets. The labels of the samples from the noisy dataset are discarded and are treated as unlabelled data $\boldsymbol{\mathcal{U}} = \{(\boldsymbol{y_i}); \boldsymbol{y_i} \in \boldsymbol{\mathcal{N}}\}$ so that the divided dataset containing noisy data can be used for semi-supervised learning. The clean dataset is used as the labelled data $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{C}}$. Lastly, they are jointly subjected to semi-supervised learning.

The MixMatch algorithm for semi-supervised learning is used in this study. MixMatch integrates a variety of semi-supervised algorithms, including consistency regularisation, entropy minimisation and conventional regularisation. For any $\boldsymbol{x_b}$ of labelled data $\boldsymbol{\mathcal{X}}$, one data augmentation operation is performed on it. For any $\boldsymbol{u_b}$ of unlabelled data $\boldsymbol{\mathcal{U}}$, *K* data augmentations are performed on it.

What we found is that if the model is consistent across multiple data augmentations of the sample, its predictions are generally correct. Thus, the model predicts *K* data augmentations for unlabelled samples, averaging p over the distribution of categories predicted by the model over *K* augmentations of $\boldsymbol{u_b}$. Further, a sharpening function is used to reduce the entropy of the pseudo-label distribution of the unlabelled samples and to obtain the pseudo-label $\boldsymbol{q_b}$ of the unlabelled samples. A common technique for adjusting the "temperature" of the classification distribution is to use the following:

$$\text{Sharpen}(\text{p}, \text{T})_i \coloneqq \frac{p_i^{\frac{1}{T}}}{\sum_{j=1}^{L} p_i^{\frac{1}{T}}} \tag{4}$$

For each average category prediction *p*, the temperature hyperparameter *T* is set to regulate the categorical entropy, and the sharpen output converges to the one-hot distribution when $\boldsymbol{T \to 0}$.

Mixup is used to generate data augmentation to obtain $\boldsymbol{\mathcal{X}'}$ and $\boldsymbol{\mathcal{U}'}$ after obtaining pseudo-labels for $\boldsymbol{\mathcal{U}}$. Mixup is a simple data augmentation strategy by applying a simple linear transformation to the input data. It can increase the generalisation ability of the model and can improve the robustness of the model to noise.

After obtaining $\boldsymbol{X}'$ and $\boldsymbol{U}'$, we use the standard semi-supervised losses shown in equation (5) to (7). The cross-entropy loss function (equation (5)) is used for the labelled samples. The pseudo-labelling of noisy samples has uncertainty and requires a more stringent loss function. Therefore, the L2 loss function (equation (6)) is used in this study to calculate the loss of unlabelled samples. The total loss is a weighted sum of the losses of the clean and noisy samples (equation (7)).

$$\mathcal{L}_{\mathcal{X}} = -\frac{1}{|\mathcal{X}'|}\sum_{x,p\in\mathcal{X}'}\sum_{c} p_c \log\left(p^c_{model}(x;\theta)\right) \tag{5}$$

$$\mathcal{L}_{\mathcal{U}} = -\frac{1}{|\mathcal{U}'|}\sum_{x,p\in\mathcal{U}'} \left|\left|p - P_{model}(x;\theta)\right|\right|_2^2 \tag{6}$$

$$\mathcal{L} = \mathcal{L}_{\mathcal{X}} + \lambda_u \mathcal{L}_{\mathcal{U}} \tag{7}$$

4. Experiments

4.1 Style and spacing

Three benchmark datasets, CIFAR-10, CIFAR-100 [32] and Clothing1M [33], which are the most commonly used datasets in noisy label learning, are used to validate the effectiveness of the proposed method. Among these, Clothing 1M is the large real-world dataset, which is approximately 38% noisy. It contains nearly 1 million noisy samples and 48K clean samples primarily, and its test dataset only contains clean samples.

Based on previous research [8][12][20], various proportions of symmetric noise [34] and asymmetric noise [35] are added to the training datasets of CIFAR-10 and CIFAR100. Symmetric noise randomly flips the label of the data in each class to a mislabel with probability p, with p={0.1,0.2,0.4,0.8} in our experiments. Asymmetric noise is designed to mimic real-world noise by modifying the labels of the samples to labels of categories similar to theirs, e.g. TRUCK→AUTOMOBILE, BIRD→AIRPLANE. A representation of the two types for noise is shown in figure 3.

60%	10%	10%	10%	10%
10%	60%	10%	10%	10%
10%	10%	60%	10%	10%
10%	10%	10%	60%	10%
10%	10%	10%	10%	60%

(a)

60%	0%	40%	0%	0%
0%	60%	0%	0%	40%
0%	40%	60%	0%	0%
0%	40%	0%	60%	0%
0%	0%	0%	40%	60%

(b)

Figure 3. Example of the noise excess matrix Q (assuming that only 5 classes are present in the dataset and the noise rate is 0.4). (a) represents symmetric noise. (b) represents asymmetric noise.

The proposed model is compared to recently superior approaches for noisy label learning, and the experiment environment is implemented on an NVIDIA 3080 GPU using Pytorch. Resnet18 [36]is the network model used for both stages, and both use SGD with a momentum of 0.9 and a weight decay of 0.0005. During the sample selection stage, the number of hot restart epochs is set to 50 for 5 cycles, with a maximum learning rate of 0.01 and a minimum learning rate of 0.0002. In the semi-supervised stage, we set the noise sample data augmentation K=2, the batch size to 16, and set the constant learning rate to 0.002. The TSS-Net is compared with the following baselines: Bootstrapping[11], Co-teaching[18], CurriculumNet[21], MentorNet[22], O2U-Net[20] and CJC-Net[14].

4.2 Comparison with the state-of-the-art methods

The accuracy of TSS-Net on the CIFAR-10 dataset with different proportions of noise is shown in table 1. The experiments show the accuracy as an average of the last 10 epochs tested. It can be seen that the TSS-Net network obtains the best results at 10%, 20%, 40% and 80% symmetrical noise. In most of the cases with 10% asymmetric noise, it outperforms the previous methods, which demonstrates its robustness to noisy labels. By analyzing different noisy ratios, it is found that the TSS-Net method still performs much better than the previous method, although, at an 80% noise ratio, the accuracy of the TSS-Net method is lower than the previous one. It demonstrates the wealth of knowledge that can be learned from noisy samples. Fortunately, it is rare to achieve such a high ratio of noisy labels from even large-size datasets obtained in the real world. Research has demonstrated that datasets from the real world contain between approximately 8.0% and 38.5% of noisy labels [17][33]. In order to illustrate the superiority of TSS-Net as compared to previous approaches, the same parameters are used for training overall ratios without any fine-tuning.

Table 1. Average accuracy (%) of the last 10 epochs of TSS-Net on the training CIFAR-10 containing noisy data. Symmetry-20% indicates the existence of 20% symmetric labels on the training dataset, by asymmetry-10% indicates the existence of 10% asymmetric noise on the training dataset.

Methods	Symmetry-10%	Symmetry-20%	Symmetry-40%	Symmetry-80%	Asymmetry-10%
Standard	82.67	76.42	56.08	17.67	88.17
Soft Bootstrapping	82.68	75.21	54.55	17.65	90.08
Hard Bootstrapping	89.69	84.88	68.90	15.59	89.17
Co-teaching	90.36	87.26	82.80	26.23	90.77
CurriculumNet	90.59	84.65	69.45	17.95	90.45
MentorNet	92.80	91.23	88.64	46.31	91.02
O2U-net	93.58	92.57	90.33	37.76	94.14
CJC-net	93.41	92.38	90.13	36.85	92.87
Ours	95.67	94.95	93.85	75.25	95.32

Table 2 presents the accuracy of TSS-Net on the CIFAR-100 dataset for different ratios of noise. As with CIFAR-10, the best accuracy is achieved in most cases compared to previous work. It is the same on the asymmetric 10% noise. In order to demonstrate the superiority of TSS-Net, when training on the CIFAR-100 dataset, all settings are the same, except for the dataset being different from CIFAR-10. It indicates that TSS-Net is insensitive to hyperparameters and networks. Accuracy in table 2 is generally lower than that in table 1 which is because CIFAR-100 is much more complex than CIFAR-10.

Table 2. Average accuracy (%) of the last 10 epochs of TSS-Net on CIFAR-100 with noise.

Methods	Symmetry-10%	Symmetry-20%	Symmetry-40%	Symmetry-80%	Asymmetry-10%
Standard	68.89	62.73	48.87	9.21	69.10
Soft Bootstrapping	69.87	62.71	48.01	9.05	71.30
Hard Bootstrapping	70.31	63.36	48.55	8.88	70.77
Co-teaching	68.81	64.40	57.42	15.16	70.02
CurriculumNet	73.23	67.09	51.68	9.63	73.30
MentorNet DD	73.14	72.64	67.51	30.12	71.96
O2U-net	75.43	74.12	69.21	39.39	62.32
CJC-net	72.30	70.13	66.26	12.71	71.67
Our	77.75	76.05	74.32	51.69	76.94

There is a difference between artificially created noise and real-world noise, and to demonstrate the effectiveness of TSS-Net in labelling noise in the real world, we evaluate our method on Clothing1M. All parameters are the same as for

training CIFAR-10 and CIFAR-100. Again the average test accuracy for the last 10 epochs is chosen. The results are displayed in table 3, and the accuracy of TSS-Net on this dataset is also superior to the other methods, indicating that our method is effective on real-world datasets with noisy labels as well. This demonstrates the superiority of our method.

Table 3. Accuracy of TSS-Net in Clothing1M.

Methods	best	last
Standard	67.22	64.68
Co-teaching	69.21	68.51
MentorNet	69.85	70.33
O2U-net	71.95	71.95
CJC-net	72.71	72.71
Our	73.75	73.75

5. Conclusion

TSS-Net is proposed in the study, which combines sample selection and semi-supervised learning with noisy labels. A two-stage network is used in our approach, with samples first selected and then the labels of the selected noisy samples discarded as unlabelled data for semi-supervised learning. There is no need to design complex loss functions and network models for our approach. The two stages of the training strategy are independent of each other, which means that it is possible to replace either stage of the network with a more advanced one to achieve better results.

Furthermore, TSS-Net requires no excessive hyperparameters and is extremely easy to implement. Both synthetic and real-world datasets have demonstrated excellent performance, with substantial performance improvements achieved. It can be observed that the average loss of the noisy samples in each epoch is bigger than the average loss of the clean samples. Therefore, when the cyclic training is completed, most of the noisy samples can be selected based on the accumulated normalized loss of the samples during the cyclic training stage.

Acknowledgments

This research was supported in part by The Fundamental Research Funds for the Central Universities (Grant No. B210202080), Project of Water Science & Technology of Jiangsu Province (Grant No. 2021080).

References

[1] Krizhevsky A, Sutskever I and Hinton G E. 2012 Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.
[2] C. Zhang, J. Cheng and Q. Tian. 2019 Unsupervised and semi-supervised image classification with weak semantic consistency, IEEE Trans. 21: 2482–2491.
[3] Girshick R, Donahue J, Darrell T and Malik J. 2014 Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Columbus, OH. 580-587.
[4] S. Ren, K. He, R. Girshick, J. Sun 2015 Faster r-cnn: Towards real-time object detection with region proposal networks. In: Proc. IEEE Conf. Adv. Neural Inf. Process. Sys. 91–99.
[5] Li X, Xu F, Lyu X, Gao H, Tong Y, Cai S , Li S and Liu D. 2021 Dual attention deep fusion semantic segmentation networks of large-scale satellite remote-sensing images. Remote Sensing,42: 3583-3610.
[6] Li X, Xu F, Xia R, Lyu X, Gao H and Tong, Y (Tong, Yao). 2021 Hybridizing Cross-Level Contextual and Attentive Representations for Remote Sensing Imagery Semantic Segmentation. Remote Sensing, 13: 2986.
[7] Zhang Z and Sabuncu M. 2018 Generalized cross entropy loss for training deep neural networks with noisy labels. Advances in neural information processing systems, 31.

[8] Arazo E, Ortego D, Albert P, O'Connor N and McGuinness K. 2019 Unsupervised label noise modeling and loss correction. In: International conference on machine learning. Long Beach, CA.312-321.
[9] Wang Y, Ma X, Chen Z, Luo Y, Yi J and Bailey J. 2019 Symmetric cross entropy for robust learning with noisy labels. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. Seoul, SOUTH KOREA. 322-330.
[10] Song H, Kim M, Park D, Shin Y and Lee J. 2022 Learning from noisy labels with deep neural networks: A survey. IEEE Transactions on Neural Networks and Learning Systems.
[11] Reed S, Lee H, Anguelov D, Szegedy C, Erhan D and Rabinovich A. 2015 Training deep neural networks on noisy labels with bootstrapping: International Conference on Learning Representations, 3.
[12] Tanaka D, Ikami D, Yamasaki T and Aizawa K. 2018 Joint optimization framework for learning with noisy labels. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Salt Lake City, UT. 5552-5560.
[13] Yi K and Wu J. 2019 Probabilistic end-to-end noise correction for learning with noisy labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach, CA. 7017-7025.
[14] Zhang Q, Lee F, Wang Y, Ding D, Yang S, Lin C and Chen Q. 2021 CJC-net: a cyclical training method with joint loss and Co-teaching strategy net for deep learning under noisy labels. Information Sciences, 579: 186-198.
[15] Koh P and Liang P. 2017 Understanding black-box predictions via influence functions. In: International conference on machine learning. Sydney, AUSTRALIA. 1885-1894.
[16] Zhang X, Zhu X and Wright S. 2018 Training set debugging using trusted items. In: Proceedings of the AAAI conference on artificial intelligence. New Orleans, LA. 4482-4489.
[17] Lee K H, He X, Zhang L and 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. Salt Lake City, UT. 5447-5456.
[18] Han B, Yao Q, Yu X, Niu G, Xu M, Hu W, Tsang I and Sugiyama M. 2018 Co-teaching: Robust training of deep neural networks with extremely noisy labels. Advances in neural information processing systems, 31.
[19] Wei H, Feng L, Chen X and An B. 2020 Combating noisy labels by agreement: A joint training method with co-regularization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 13726-13735.
[20] Huang J, Qu L, Jia R and Zhao B. 2019 O2u-net: A simple noisy label detection approach for deep neural networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. Seoul, SOUTH KOREA. 3326-3334.
[21] Guo S, Huang W, Zhang H, Zhuang C, Dong D, Scott M and Huang D. 2018 Curriculumnet: Weakly supervised learning from large-scale web images. In: Proceedings of the European Conference on Computer Vision (ECCV). Munich, GERMANY. 135-150.
[22] Jiang L, Zhou Z, Leung T, Li L and Li F. 2018 Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In: International Conference on Machine Learning. Stockholm, SWEDEN. 2304-2313.
[23] Laine S and Aila T. 2017 Temporal ensembling for semi-supervised learning. International Conference on Learning Representations, 5.
[24] Tarvainen A and Valpola H. 2017 Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. Advances in neural information processing systems, 30.
[25] Miyato T, Maeda S, Koyama M and Ishii S. 2018 Virtual adversarial training: a regularization method for supervised and semi-supervised learning. IEEE transactions on pattern analysis and machine intelligence, 41: 1979-1993.
[26] Grandvalet Y and Bengio Y. 2004 Semi-supervised learning by entropy minimization. Advances in neural information processing systems, 17.
[27] Lee D H. 2013 Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. Workshop on challenges in representation learning, 3: 896.
[28] Berthelot D, Carlini N, Goodfellow I, Oliver A, Papernot N and Raffel C. 2019 Mixmatch: A holistic approach to semi-supervised learning. Advances in neural information processing systems, 32.
[29] Zhang H, Cisse M, Dauphin Y and Lopez-Paz D. 2018 MixUp: Beyond empirical risk minimization. International Conference on Learning Representations, 6.
[30] Loshchilov I, Hutter F. 2016 Sgdr: Stochastic gradient descent with warm restarts. International Conference on Learning Representations, 5.

[31] Yu X, Liu T, Gong M, Batmanghelich K and Tao D. 2018 An efficient and provable approach for mixture proportion estimation using linear independence assumption. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT. 4480-4489.
[32] Krizhevsky A and Hinton G. 2019 Learning multiple layers of features from tiny images. Handbook of Systemic Autoimmune Diseases.
[33] Xiao T, Xia T, Yang Y, Huang C and Wang X. 2015 Learning from massive noisy labeled data for image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Boston, MA. 2691-2699.
[34] Van Rooyen B, Menon A and Williamson R C. 2015 Learning with symmetric label noise: The importance of being unhinged. Advances in neural information processing systems, 28: 10-18.
[35] Patrini G, Rozza A, Menon AK, Nock R and 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. Honolulu, HI. 1944-1952.
[36] He K, Zhang X, Ren S, et al. 2016 Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 770-778.