Received: 26 July 2021 | Accepted: 25 December 2021

DOI: 10.1002/int.22814

RESEARCH ARTICLE

WILEY

Semisupervised image classification by mutual learning of multiple self-supervised models

Jian Zhang[1] | Jianing Yang[1] | Jun Yu[2] | Jianping Fan[3]

[1]School of Science and Technology, Zhejiang International Studies University, Hangzhou, Zhejiang, China

[2]Computer and Software School, Hangzhou Dianzi University, Hangzhou, Zhejiang, China

[3]Department of Computer Science, University of North Carolina at Charlotte, Charlotte, North Carolina, USA

Correspondence

Jian Zhang, School of Science and Technology, Zhejiang International Studies University, 310023 Hangzhou, Zhejiang, China.
Email: jeyzhang@outlook.com

Jun Yu, Computer and Software School, Hangzhou Dianzi University, 310018 Hangzhou, Zhejiang, China.
Email: yujun@hdu.edu.cn

Funding information

National Natural Science Foundation of China, Grant/Award Numbers: 61972361, 61772161, 61836002, 62020106007; National Outstanding Youth Foundation of China, Grant/Award Number: 62125201

Abstract

Image classification has been widely adopted by current social media applications. Compared with fully supervised classification, semisupervised classification attracts more attention because it is commonly observed that category labels are only available for a small portion of images while most images on social media platforms do not have labels. To this end, we propose a two-stage semisupervised learning framework. In the first stage, we train two Self-supervised Models (SSMs). One model is initialized by predicting the rotation angles of pretransformed training images and then further trained by the labeled images. The other model is initialized by making consistent predictions for the transformed images in color, shape, and quality from the same sample image, and then further trained by the labeled images. In the second stage, we fuse the two SSMs through deep mutual learning, which enhances each of the two SSMs with the complementary information provided by the other such that the correct prediction could be shared. Experimental results on CIFAR and Caltech-256 data sets demonstrate the effect of the proposed framework.

KEYWORDS

contrastive learning, image classification, mutual learning, self-supervised model, semisupervised learning

1 | INTRODUCTION

Image classification problem is very popular to social media applications.[1–3] Machine learning techniques have been successfully used to train the classifiers. Ideally, all the training samples have labels reflecting the category semantic, and the pairwise data and labels can be used to train the model. This learning scheme is fully supervised learning. However, more common situation is that only a small portion of data have category labels while the rest majority data have not, or the labels are weak and not directly related to the category information.[4–7] In this case, the model training process should simultaneously exploit the labeled data and unlabeled data such that both groups of data can contribute to the classification. This is referred to as the semisupervised learning (SSL). The key to SSL is how to discover useful information from the unlabeled data. For example, Yang et al. built pseudopairs from unlabeled data and merged them with labeled data pairs to learn a discriminative projection for Person reidentification.[8]

The SSL approaches can be organized into four groups, that is, pseudolabeling methods, label propagation methods, consistency regularization methods, and self-supervision methods. Pseudolabeling methods[9,10] train an initial model with labeled data in a supervised manner, then annotate the unlabeled data through the model and update the model using the data with ground truth and predicted labels. The annotation is conducted in a progressive way such that the data with most confident prediction results have priority in model updating. Label propagation methods leverage the similarity between data samples to derive the labels of the unlabeled data from the labeled data.[11,12] The representative methods are graph-embedding SSL methods,[13] which convey the geometric structure of the data set to feature embeddings through a graph that covers both labeled and unlabeled data and enables the label propagation from the labeled data to unlabeled ones. In past years, the graph-based methods were further extended to deep learning.[14,15] Consistency regularization methods deem that the model should make the same prediction for given data and the perturbed data. The perturbation could be exerted on the input data,[16] the network parameters,[17] or the way we use the intermediate output of network.[18]

The aforementioned SSL approaches still have their weaknesses. The pseudolabeling methods depend heavily on the small group of labeled data while rarely exploit the task-irrelevant semantic contained in the unlabeled data. Label propagation methods need to encode the whole graph into the network for label propagation, which is impracticable in case of the vast amount of training data, especially for those deep learning-based methods. Under this situation, the network has to exploit only the local geometry reflected by the current batch of data, which degrades the propagation effect. Consistency regularization methods organize the SSL into a multitask scheme. One task forces the predictions of labeled data to approach the ground truth labels and the other imposes certain consistency regularization on the unlabeled data. Therefore, they may suffer from the imbalance between the losses of the two subtasks. Weighting factor alleviates the problem, but the optimal weight computation could be complicated according to Reference [19].

Self-supervised models (SSMs) successfully address these problems. An SSM follows a two-stage pipeline. First, all the data are used to train a model aiming to solve a carefully designed pretext task that is independent of data annotations.[20,21] Second, the first few layers of the model are fixed as feature extractors and linked to a task module that is subsequently trained with the feature representations and labels of those labeled data. SSMs share similarities with knowledge distillation[22] and external knowledge incorporation[23] in that they all exploit prior information derived from existed models or relevant domains. SSMs can discover task-irrelevant semantic from the vast amount of unlabeled data and encode this semantic into the learning of the task model. In the meanwhile, SSMs do not need to build a global graph representing the geometric relationships between data for label propagation. Owing to the two-stage pipeline, SSMs also avoid the imbalance between the losses of multiple subtasks. Therefore, self-supervision methods yield state-of-the-art performance in SSL.

In spite of the success of SSMs, we seek for further performance improvement on this basis. Ensemble learning has proved that a stronger learner can be obtained by integrating multiple base learners.[24,25] Different ensemble strategies vary in how the base learners are integrated. We believe different base models may make different predictions for the same data sample even after being well trained through labeled data, and the probability that none of the base models makes the right prediction is low. Therefore, the key to model integration is achieving mutual complement of the base models, which means the base models should be able to guide each other such that the right prediction could be shared. To this end, deep mutual learning (DML)[26] emerges, which allows each base model to generate approximately the same classification probability as the other model while keeping the prediction close to the labels.

In this paper, we propose a novel SSL framework named Mutual Learning of Multiple Self-supervised Models (MLMSM) for social media image classification. This framework comprises two stages. In the first stage, we train multiple SSMs initialized by other label-independent tasks. In the second stage, we fuse the SSMs through DML. Specifically, two SSMs are trained. One model perturbs the training set by rotating each of the images by certain angles, and learns the mapping from the rotated images to the designated angles. Then we fix several initial layers of this model as feature extractor, and feed the feature representations to a classifier that is subsequently trained using the labeled data (features). To improve performance, the fixed feature extractor can be fine-tuned by the labeled data during the training of classifier. This model has been described in Reference [21]. The other model augments the training set by exerting transformations in geometry, color, and image quality on the images, and learns to make consistent predictions for those transformed images from the same image. We can fix the feature extractor and train the following classifier in the similar way to the model we discussed before. This model has been described in Reference [20]. More importantly, we further fine-tune the two SSMs simultaneously using the labeled data. For each of the models, we force the predictions to be close to both the labels and the predictions outputted by the other model such that they can learn from each other. We believe the correct predictions can be shared between the two models owing to the mutual learning.

The contribution of this paper is twofold:

1. We propose a novel framework for SSL, which follows a two-stage pipeline. First, we train multiple SSMs initialized by other label-independent tasks. This is to encode the structural characteristics hiding among the data from different perspectives into the classifiers. Second, we fuse the pretrained models through DML, and this is to take advantage of the mutual complementary information between the models to further improve the performance.
2. We implement this framework with two state-of-the-art SSMs. One SSM starts from predicting the predefined rotation angles of sample images, and the other SSM starts from making consistent predictions for the augmented images from one same image. They achieve feature representation from different views and could be complementary to each other.

The rest of this paper is organized as follows. Section 2 reviews the relevant bibliography. Section 3 discusses the proposed method in detail. We report the experimental results in Section 4 and conclude this paper in Section 5.

2 | RELATED WORKS

2.1 | Pseudolabeling methods

Pseudolabeling methods are a group of methods that train a classifier with labeled data in a supervised way and predict target distribution for unlabeled data, which are assumed to be the labels and used to update the classifier. The pioneering work was discussed in Reference [9]. The authors simply predicted pseudolabels for the unlabeled data and merged the data with highest classification confidence into the labeled data to update the model. During this procedure, the balancing coefficient between the labeled and unlabeled parts of the loss is progressively changed to avoid classification bias. The noisy student method proposes a teacher–student scheme to implement the pseudolabeling method.[27] A teacher network is first trained with the labeled data, and predicts pseudolabels for the unsupervised data. Then, the data with true and pseudolabels are used to train a student network, during the course of which data augmentation, model dropout and stochastic depth are adopted to enhance the generalization power of the student. The authors redefined the student as a new teacher to repeat this process and observed performance improvement. This scheme is further enhanced by meta pseudolabeling.[28] It differs from previous pseudolabeling methods in that the feedback of student network on validation set is adopted to compute the gradient of the teacher network and update the teacher's parameters. Then the teacher can generate better pseudolabels for the student recursively. Another group of pseudolabeling methods train several classifiers and employ the disagreement during the learning process. Deep cotraining method[10] simultaneously trains two networks in a multitask paradigm. One subtask minimizes the cross-entropy loss between the predictions and labels for both networks. One subtask forces the target distributions of the two networks to be close to each other for unlabeled data. To encourage the view difference, another subtask generates adversarial examples from original samples and forces the predictions of the original samples by one network to approach that of the adversarial examples by the other network. Tri-net method[29] builds three subnetworks and trains each network with output smearing trick based on the labeled data. Then the networks can be updated through a voting strategy, that is, if two of the subnetworks make consistent prediction for an unlabeled sample, the sample and its pseudolabel can be merged into the training set of the rest subnetwork. Pseudolabeling methods suffer from the unawareness of the optimal pseudolabel. Using data with suboptimal pseudolabels to update the model may cause obvious confirmation bias and degrade the performance of classifier. In addition, pseudolabeling methods rarely consider the task-irrelevant semantics hiding behind the data.

2.2 | Label propagation methods

Label propagation methods leverage the similarity between the data samples and spread the labels to the unlabeled data such that similar samples to the labeled ones should have the same labels. The pairwise similarity between samples is usually represented via a graph whose nodes denote the data and edges depict the similarity. Graph and its variant have been widely used in image searching[30] and visual classification.[31] Zhu and Ghahramani[11] name the graph as probabilistic transition matrix, which is then multiplied to the label matrix to achieve the label propagation. This operation is conducted iteratively, and the ground truth labels should be reset to their original values in each iteration to guarantee the accuracy. Similar strategies are adopted by References [32,33]. These methods apply the graph only to the label matrix without explicitly computing the feature representations of each data sample, which is conducted in graph-embedding methods.[13,34] Following manifold assumption, Zhang et al.[13] deem that each data sample is more likely to be influenced by the most similar samples instead of the dissimilar ones, therefore the graph does not need to reflect all the pairwise similarities between data. The authors constructed a Laplacian graph that encodes the local similarity between each sample and its neighbors (several most similar samples), and achieved feature embedding and label propagation using this graph. Tang et al.[34] combine the first-order and second-order similarity to build the graph. The first-order similarity between two samples refers to their joint probabilistic distribution, and the second-order similarity is the Kullback–Leibler divergence (KL divergence) between the two samples' respective local contextual information, which is represented by a conditional distribution over one sample and its neighboring samples. The aforementioned methods belong to transductive learning that estimates the labels for unlabeled data during the course of model optimization. Therefore, these methods suffer from huge computation burden in case of a large amount of data. Recently, the graph-embedding methods have been extended to deep learning community. Weston et al.[14] formulate the graph into a weighting matrix, which defines the similar and dissimilar samples to each training sample. Then a margin-based metric loss is linked to the network to congregate the embeddings of the similar samples and disperse the dissimilar ones. Gilmer et al.[35] feed the embeddings of each sample as well as its neighboring samples on the graph together into each layer of the network to preserve the local geometry at each data point during training. Weston et al.[14] and Gilmer et al.[35] exploit only the local part of the graph in training such that the label propagation could be suboptimal. Kipf and Welling[15] encode the graph into each network layer to exert similarity constraint on feature embeddings. However, the graph encoding could be difficult when the data amount is huge.

2.3 | Consistency regularization methods

The basic concern of consistency regularization is that the model should be robust to the perturbation which means the perturbation of the data or the model parameters should not change the prediction results of the model. To this end, a consistency regularization term is jointly used with the classification loss to achieve SSL. The Π-model may be the simplest consistency regularization method.[17] During each training epoch, the network makes predictions twice for one same unlabeled sample. Through data augmentation and model dropout, stochasticity and noise are injected into the training such that the two predictions can be different from each other. However, the Π-model forces the two predictions to be the same. The authors further improved the Π-model to obtain temporal ensembling SSL,[17] which only makes one prediction with data augmentation and model dropout, and the other prediction is replaced with an exponentially moving average of the predictions obtained in several previous epoches. Temporal ensembling method computes the exponentially moving average only once in each epoch, and this influences the learning efficiency. Instead of the moving average of predictions, mean teacher[18] method computes the exponentially moving average of the model parameters in several previous training steps. Specifically, mean teacher method trains a student network through the labeled data, and averages the parameters of the student model in temporal domain to obtain mean teacher network. For each unlabeled data sample, the prediction of the teacher and student should be consistent with each other. Since the consistency regularization term works in a stepwise way rather than epochwise, mean teacher method improves the learning efficiency. Xie et al.[16] apply RandAugment including image shearing, transformation, and color adjustment to unlabeled images, and forces the predictions for the images before and after augmentation to be the same. The consistency regularization methods are formulated in a multitask learning scheme. The loss value of the supervised part and unsupervised part usually differs from each other apparently and we need to keep balance between the two losses. Commonly used strategy is adding a weighting coefficient to one of the losses, but how to obtain an optimal coefficient is still an open problem. Some researchers deem the coefficient to be learnable, and this increases the problem complexity.[19]

2.4 | Self-supervision methods

Self-supervision methods learn feature representations with task-irrelevant loss and feed the representations to some downstream tasks. Because the learning of feature representations is actually an unsupervised process, and the task-irrelevant loss can be carefully designed to endow the representations various and abundant semantics that could be helpful to the downstream tasks, self-supervision methods have received much attention in recent years. In self-supervised learning domain, the task-irrelevant loss is also known as the pretext task. Various pretext tasks have been proposed. Doersch et al. adopted a Jigsaw problem as the pretext task.[36] They partitioned an image into nonoverlapping patches, and for an arbitrary patch, the model learns to predict its correct positional relationships to other patches. The model is trained using each patch and its neighboring eight patches. In Reference [37], the model learns to solve an image colorization problem, that is, predicting the a and b channels of each pixel based on the inputting L channel. The color value is transformed into a one-hot vector and the network is trained through minimizing the cross-entropy loss between the ground truth one-hot color vector and weighted combination of five predicted vectors that are closest to the ground truth one. Gidaris et al.[21] augmented the unlabeled images via rotating each of them by 90°, 180°, and 270°, then trained a model to predict the rotation angles for the augmented images. This method is simple yet surprisingly effective. Contrastive learning refers to a group of methods that exploit contrastive loss to narrow the distance between similar samples and let dissimilar samples far from each other. Contrastive learning usually applies data augmentation including image cropping, resizing, cutout, flipping, rotating, color distortion, and quality degeneration to original sample images, and defines the augmented versions of the same image as similar samples.[20] The loss function of contrastive learning is similar to metric learning that minimizes the distances between similar data.[38,39] It is believed that the pretext tasks can encode certain types of structural characteristics into the feature representations and the structural characteristic will be useful for downstream tasks, like, image classification and clustering.

3 | MUTUAL LEARNING OF MULTIPLE SELF-SUPERVISED MODELS

3.1 | The framework

In this paper, we propose a novel framework termed MLMSMs for social media image classification. This framework is formulated into a two-stage pipeline. In the first stage, we train two SSMs with respective pretext task using all the data. Then feed the output feature representations of each model to a classifier that is subsequently trained using the labeled data. To avoid underfitting, we allow the feature extraction parts of both models to be fine-tuned slightly during training of the classifiers. In the second stage, we further improve the prediction accuracy by DML of the two classifiers.[26] The feature extractor of each SSM is a convolutional neural network. This framework can be interpreted in Figure 1 where the green arrows stand for the unsupervised information flow and the red arrows represent the supervised flow. The arrow width means the data amount. Specifically, one SSM adopts a rotation angle prediction task as the pretext task. The other SSM adopts a contrastive loss as the pretext task. In addition, we extend this framework to incorporate more than two SSMs in the mutual learning procedure. Detailed descriptions of the SSMs can be found below.

3.2 | Rotation angle prediction task

This pretext task was first proposed in Reference [21]. To enable this pretext task, we rotate each training image by 90°, 180°, and 270°, and collect the original images and their respective three rotated versions to construct the task-irrelevant training image set with the rotation angles as the image labels. Then, we train a convolutional neural network to predict the rotation angles from the inputting images. In order that this network adapts to the

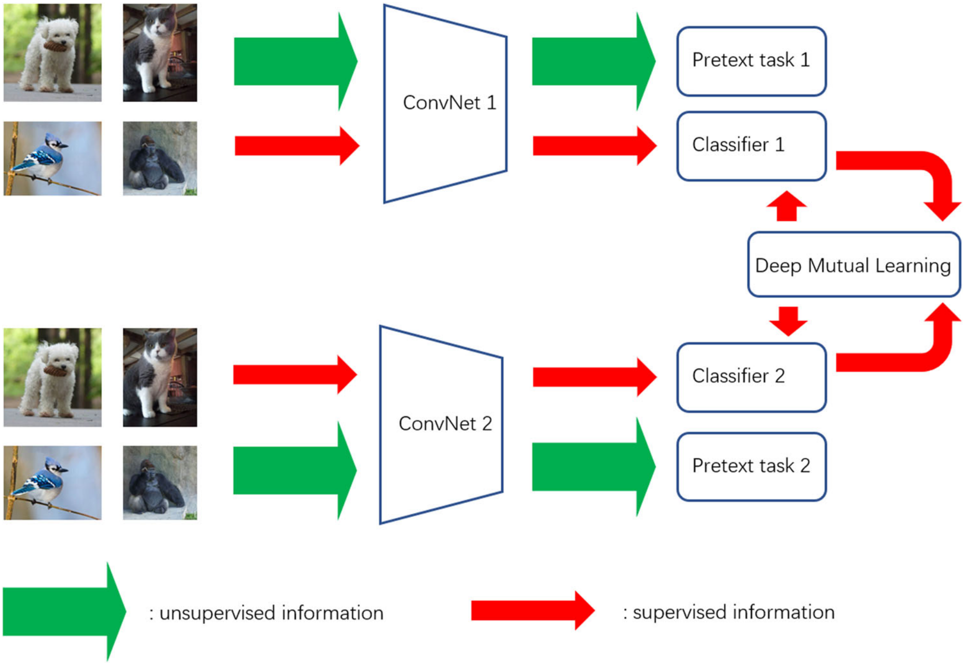

FIGURE 1 The framework of the proposed Mutual Learning of Multiple Self-supervised Models (MLMSM). Two Self-supervised Models (SSM) are first trained with respective pretext tasks using all the data, then the feature representations are fed to two classifiers that are subsequently learned using the labeled data. Deep mutual learning is adopted to further improve the prediction performance. The green arrows stand for the unsupervised information flow while the red arrows denote the supervised information flow. The width of arrows represents the data amount

semisupervised classification task, we fix several early convolutional layers and feed the feature representations to a classifier that is subsequently trained with the labeled images.

Suppose X is a sample image, $d \in \{1, 2, 3, 4\}$ denotes the rotation degrees 0, 90, 180, and 270, respectively, that will be applied to X, then the image rotation preprocessing can be depicted as $\{R(X, d)\}_{d=1}^{4}$. Let $F(\cdot)$ be the convolutional neural network, it accepts $\{R(X, d)\}_{d=1}^{4}$ as input and predicts the probability distribution over all designated rotation angles:

$$F(R(X, d), \theta) = \{F^{d}(R(X, d), \theta)\}_{d=1}^{4}, \tag{1}$$

where θ represents the network parameters.

Given N training images $T = \{X_i\}_{i=1}^{N}$, the pretext task is formulated into the following problem:

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} L(X_i, \theta), \tag{2}$$

where the loss function L is

$$L(X_i, \theta) = -\frac{1}{4} \sum_{d=1}^{4} \log(F^{d}(R(X, d), \theta)). \tag{3}$$

Once the training converges, the network can be used to predict the rotation angle from given image with unknown angel $\tilde{X}$ as

$$F(\tilde{X}, \theta). \tag{4}$$

Specifically, we adopt AlexNet[40] and ResNet-50,[41] respectively, as the backbone network. For AlexNet, we fix the convolution layers as the feature extractor. For ResNet-50, we fix the first five convolution blocks and the following avgPooling layer as the feature extractor. The classifier of our method is made up of a fully connected layer and a softmax function followed by a cross-entropy loss. Figure 2 demonstrates the working pipeline of this SSM with ResNet backbone, where FC layer means the fully connected layer, the green arrows denote the unlabeled information flow, and the red arrows denote the labeled flow.

The rotation angle prediction task has been proved effective to the downstream image classification task. The reason is that to know the image rotation angle, the model has to learn the content of the image, recognize the type and orientation of salient objects and discover the dominant orientation of the objects. This information is also useful to the classification task. Actually, other networks could also be used to achieve the angle prediction pretext task, which will be proved in Section 4.

3.3 | Contrastive learning task

This pretext task has been adopted in Reference [20] for unsupervised feature learning. It is based on the assumption that the features of different views about the same data sample should be identical to each other. Following this assumption, we apply random augmentations such as random cropping, resizing, flipping, cutout, random color distortions, and random Gaussian blur to the original images, and the derived images from the same image can be regarded as

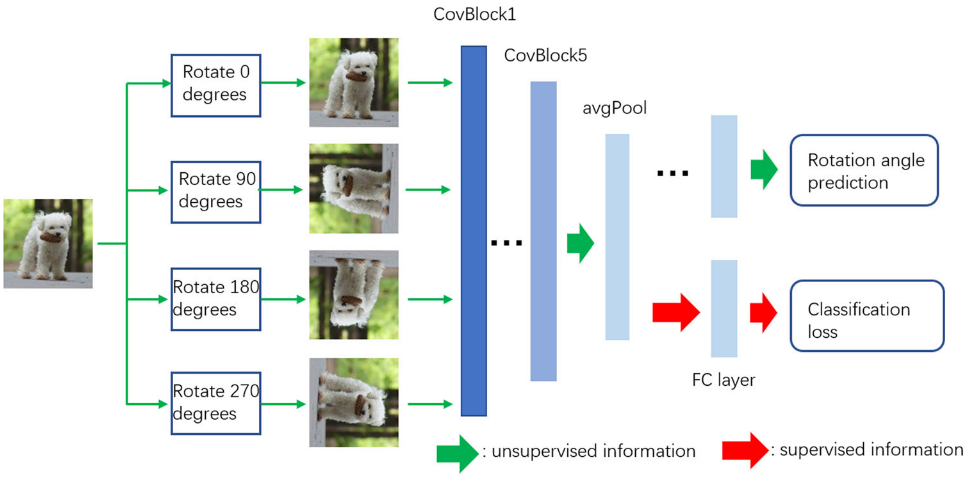

FIGURE 2 The pipeline of the self-supervised model towards rotation angle prediction. The unsupervised data are used to train the angle prediction model from which the first five convolution blocks and the following avgPooling layer are extracted and linked to a classifier that is trained using the labeled data. The green arrows denote the unlabeled information flow and the red arrows denote the labeled flow

different views of the image. The examples of the random augmentation can be found in Figure 3. Specifically, we select N unlabeled training samples and apply two types of random augmentations to each of them to obtain a pair of positive samples, and collect the $2N$ generated images as the training set. Then, we train a backbone neural network to minimize the divergence between the positive samples with a contrastive loss. It is noteworthy that we insert a nonlinear projection head (a Multilayer Perceptron with Rectified Linear Unit [ReLU] activation) right before the contrastive loss into the network to improve the performance. Once the network converges, we fix its feature extraction part and feed the feature representations to a classifier, which is subsequently trained using the labeled data. Theoretically, any backbone network could work for this pretext task. We also use ResNet-50 as the backbone model, and regard the first five convolution blocks with the following avgPooling layer as the feature extractor. The feature extractor can also be fine-tuned to avoid underfitting. The classifier is also constructed by a fully connected layer followed by a cross-entropy loss. The working pipeline of the contrastive learning task is depicted in Figure 4 where the green arrows denote the unsupervised information flow and the red arrows mean the supervised information flow.

Suppose X is an unlabeled image, $\tilde{X}_i$ and $\tilde{X}_j$ represent the pair of positive samples with respect to X, the contrastive learning aims to minimize the divergence between $\tilde{X}_i$ and $\tilde{X}_j$. To this end, we define the similarity between two vectors p and q as

$$\mathrm{sim}(p, q) = \frac{p^T q}{\|p\|\|q\|}, \tag{5}$$

then the loss for a positive sample pair can be represented as

$$L_{i,j} = -\log \frac{\exp(\mathrm{sim}(y_i, y_j)/\tau)}{\sum_{k \in \{1, \ldots, 2N\}, k \neq i} \exp(\mathrm{sim}(y_i, y_k)/\tau)}, \tag{6}$$

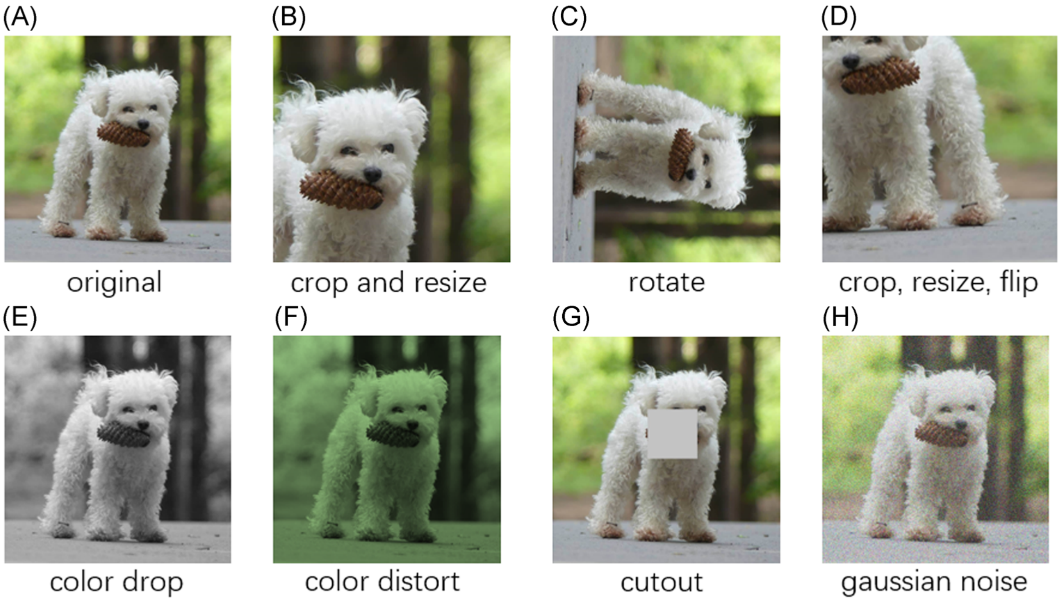

FIGURE 3 The examples of the random augmentation. (A) Original image, (B) random cropping and resizing, (C) image rotation, (D) cropping and resizing with flipping, (E) color drop, (F) color distort, (G) image cutout, and (F) adding Gaussian noise

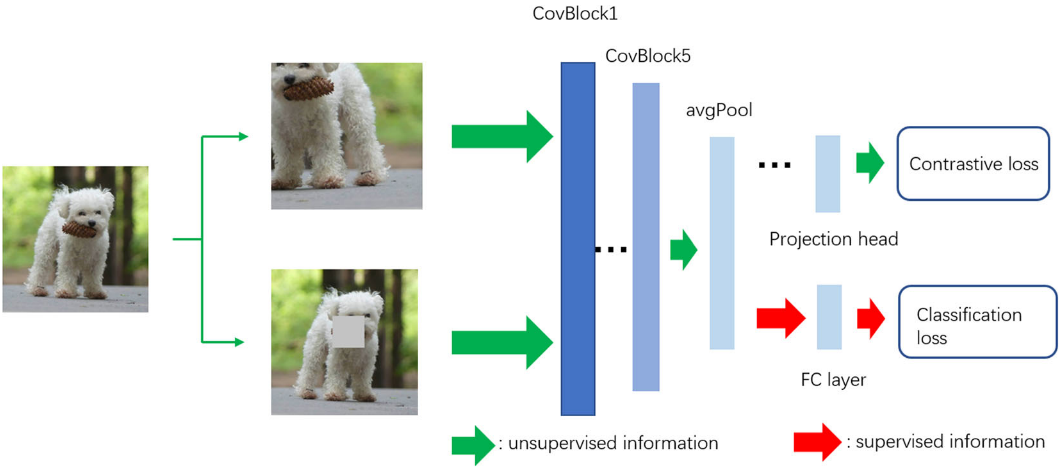

FIGURE 4 The pipeline of the self-supervised model using contrastive loss. The unsupervised data are used to train the network from which the first five convolution blocks and the following avgPooling layer are extracted and linked to a classifier that is trained using the labeled data. The green arrows denote the unlabeled information flow and the red arrows denote the labeled flow

where y_i and y_j are the feature representations of $\tilde{X}_i$ and $\tilde{X}_j$, τ is a temperature parameter often adopted by network distillation,[42] and we raise τ higher in training and reset it to 1 in prediction. We sum up the losses generated by all the positive training pairs to obtain the final contrastive loss function. It can be observed in (6) that for a given sample y_i, not only the positive sample y_j, but also the samples augmented from other images are used which actually play the role of negative samples. The loss leverages the probability distribution of the similarity between a sample to its positive counterpart over the similarities between the sample to all of the others except itself. The subsequent classifier to be trained for SSL is also made up of a fully connected layer and a softmax function followed by a cross-entropy loss.

Contrastive learning task enables the network to discover common characteristics from all the positive samples with respect to the same image, and we believe these common characteristics imply certain high-level semantic independent of the transformations, like, rotation, cropping, and color distortion. Therefore, the feature representations yielded by contrastive learning can be helpful to downstream image classification task.

3.4 | Mutual learning of multiple SSMs

We adopt DML approach[26] to ensemble the two SSMs for semisupervised image classification. It is commonly accepted that different neural networks should make the same prediction for the same input data, which is however often disobeyed by the actual observations. In the meanwhile, the probability that all of the networks make the wrong prediction is low, which means different networks could learn complementary information to each other. Therefore, the basic concern of DML is that on the one hand, the output of these networks should approach the ground truth labels; on the other hand, each network should take other networks' predictions as reference.

Specifically, we softmax the penultimate layer output of each SSM to obtain the prediction probabilities that the input belongs to a specific class. Then, KL divergence is leveraged to match the probability generated by one network to that by the other. Simultaneously, a cross-entropy loss function is adopted to match the probability to the ground truth labels. The mutual learning paradigm can be depicted in Figure 5 where the orange arrows denote the dataflow in the well-trained SSMs, the red arrows denote the dataflow in the mutual learning part. Apparently, the mutual learning further updates the parameters of the two SSMs' feature extractors, but this update is accompanied with the mutual complement between the two models, and therefore can further improve the performance.

Let $X = \{x_i\}_{i=1}^N$ and $B = \{b_i\}_{i=1}^N, b_i \in \{1, ..., C\}$ be the labeled images and their labels, the prediction probability by each of the SSMs that sample x_i belongs to class c can be depicted as

$$p^c(x_i) = \frac{\exp(v^c/\tau)}{\sum_{c=1}^C \exp(v^c/\tau)}, \tag{7}$$

where v is the feature representation of x_i outputted by the feature extractor of an SSM, and τ is the temperature parameter to manipulate the prediction probability. Following the principle of network distillation, we raise τ during the mutual learning process and set it to one in testing.

We use the following objective function with a cross-entropy loss to guarantee the prediction of each SSM approaches the true labels:

$$L_{\text{class}} = -\sum_{i=1}^N \sum_{c=1}^C A(b_i, c)\log(p^c(x_i)), \tag{8}$$

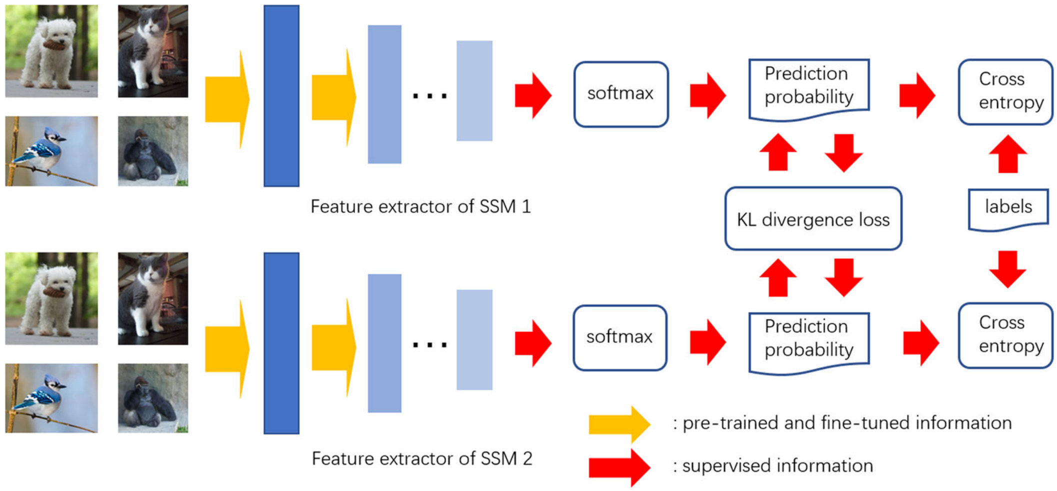

FIGURE 5 The mutual learning of two well-trained SSMs. First, the probabilities generated by the two SSMs should match each other by a KL divergence loss. Second, the two SSMs' probabilities should approach the true label via a cross-entropy loss. The orange arrows denote the dataflow in the well-trained SSMs, and the red arrows denote the dataflow in the mutual learning part. KL, Kullback–Leibler; SSM, Self-supervised Model

where

$$A(b_i, c) = \begin{cases} 1, & b_i = c, \\ 0, & b_i \neq c \end{cases} \tag{9}$$

are indicators.

The KL divergence to match the probability produced by network 1 to that by network 2 can be computed as

$$D_{\mathrm{KL}}(p_2 \| p_1) = \sum_{i=1}^{N} \sum_{c=1}^{C} p_2^c(x_i) \log \frac{p_2^c(x_i)}{p_1^c(x_i)}, \tag{10}$$

where p_1 is the prediction of network 1 and p_2 is the prediction of network 2. Similarly, the KL divergence to match p_2 to p_1 is

$$D_{\mathrm{KL}}(p_1 \| p_2) = \sum_{i=1}^{N} \sum_{c=1}^{C} p_1^c(x_i) \log \frac{p_1^c(x_i)}{p_2^c(x_i)}. \tag{11}$$

Therefore, the total loss function to enhance the two well-trained SSMs is:

$$\begin{cases} L_{\mathrm{SSM}_1} = L_{\mathrm{class}_1} + \alpha_1 D_{\mathrm{KL}}(p_2 \| p_1), \\ L_{\mathrm{SSM}_2} = L_{\mathrm{class}_2} + \alpha_2 D_{\mathrm{KL}}(p_1 \| p_2), \end{cases} \tag{12}$$

where the subscripts 1 and 2 stand for the ID of SSMs, and α is a hyperparameter to balance L_{class} and the KL divergence.

We can easily extend the mutual learning module to incorporate more than two SSMs. Suppose there are M SSMs, the KL divergence to match the probability produced by network p to that by network $q \in \{1, ..., M-1\}$ is

$$D_{\mathrm{KL}}(p_q \| p_p) = \sum_{i=1}^{N} \sum_{c=1}^{C} p_q^c(x_i) \log \frac{p_q^c(x_i)}{p_p^c(x_i)}, \tag{13}$$

and the loss function to enhance network p can be written as

$$L_{\mathrm{SSM}_p} = L_{\mathrm{class}_p} + \alpha_p \sum_{q=1}^{M-1} D_{\mathrm{KL}}(p_q \| p_p). \tag{14}$$

4 | EXPERIMENT

We conduct a group of experiments on CIFAR-10 data set and Caltech-256 data set. First of all, we briefly introduce the two data sets and the training and testing data configuration. Second, we indicate the evaluation metrics used in the experiments. Then, we provide the ablation study to verify the significance of the mutual learning. At last, a group of comparative experimental results is demonstrated to show the superiority of this method.

4.1 | Data set

CIFAR-10: This data set covers 10 classes of images with size 32 × 32, and each class has 6000 images, among which 5000 images are designated as training samples and the rest 1000 images are designated as testing samples. In training the semisupervised models, we randomly choose 300 images with labels from each class of the training set as the labeled data, and use the rest of 47,000 training images as the unlabeled training data. We adopt the 10,000 testing images to evaluate the model performance. CIFAR-10 data set is used in the ablation and comparative experiments.

Caltech-256: Actually, Caltech-256 image set covers 257 classes. A total of 256 of them describe various classes of objects and one of them denotes the background images. We also regard the background images as a special class. For each class, we randomly select 20 images with labels as the labeled training data, and use another collection of 60 images as the unlabeled training data. For the classes with over 100 sample images, we use extra 20 images per class for testing.

The detailed configuration of the two data sets can be seen in Table 1.

4.2 | Evaluation metrics

We focus on the classification performance of the Semisupervised Models. To this end, we evaluate the Classification Accuracy (*ACC*), Classification Precision (*PREC*), and Classification Recall (*RECALL*) of the algorithm.

TABLE 1 The configuration of the CIFAR-10 and Caltech-256 data sets in the experiments

Data set	Classes	Labeled training images per class	Unlabeled training images per class	Testing images per class
CIFAR-10	10	300	4700	1000
Caltech-256	257	20	60	20

ACC: *ACC* indicates the probability that the testing images are classified correctly. It is computed as the ratio of correctly classified samples against all the testing samples. Specifically, the classification algorithms predict a probability over each of the classes for a testing sample, and categorize the sample into the class identified by the leading probability. Then, the accuracy can be computed based on the prediction results. In practice, commonly adopted evaluation metric is *top-KACC*, which means that a sample is correctly classified if the leading *top-K* probabilities comprise the right class. This paper adopts *top*-1, *top*-3, and *top*-5 *ACC*.

PRECSION: For each class, the *PRECSION* is the ratio of correctly predicted images against all the images that are categorized into this class. Let the number of correctly predicted images be *TP*, and the number of images that are categorized into this class but actually should not be *FP*, *PRECSION* with respect to this class can be computed as

$$PREC = \frac{TP}{TP + FP}.$$

RECALL: For each class, the *RECALL* is the ratio of correctly predicted images against all the images that actually belong to this class. Let the number of correctly predicted images be *TP*, and the number of images that are categorized into other classes but actually belong to this class be *FN*, *RECALL* with respect to this class can be computed as

$$RECALL = \frac{TP}{TP + FN}.$$

4.3 | Ablation study

In the ablation study, we test different training pipelines and different parameters of the method. First of all, we train a ResNet-50 in a supervised way using only the labeled data, and regard this ResNet as a baseline. Then, we train the self-supervised rotation prediction network and the self-supervised contrastive learning network, respectively, using ResNet-50 backbone, fix the feature extraction part and use the feature representations to train classifiers based on the supervised data. During the training of classifiers, we allow the feature extractors to be further fine-tuned slightly to avoid the underfitting problem. At last, we fuse the two SSMs using DML and show the results of both models after mutual learning. The ablation experiments compute the *top*-1 *ACC* as the evaluation metric.

The experimental results on CIFAR-10 data set are shown in Table 2. The supervised ResNet-50 yields the lowest *top*-1 *ACC* of 63.76%. This is predictable because there are only 3000 labeled training data, and this data amount is insufficient to build good classifier. The classifier trained through the features of a self-supervised rotation prediction network achieves slightly higher *top*-1 *ACC* of 66.73%, but it is improved dramatically by over 15% after fine-tuning the extractor using 3000 labeled data. The fine-tune improvement can be also observed for the classifier trained through the features of a self-supervised contrastive learning network. However, the improvement is not so obvious as that for rotation prediction network. This is because contrastive learning has been strong enough to capture discriminative information for image classification. It yields an 85.5% *top*-1 *ACC* without fine-tune. Therefore, we believe the effect of the fine-tune operation varies in different SSMs. After mutual learning of the two classifiers, the *top*-1 *ACC*s of both networks are further improved. The classifier based on rotation prediction yields 0.9%′ performance improvement while the classifier based on contrastive learning yields 1.29%′ performance improvement. The reason is that mutual learning learns complementary information from the two classifiers, which enables the correct prediction result to be shared between one network and the other.

Also, ablation experiments are conducted on Caltech-256 data set, and the results are shown in Table 3. Similarly, the supervised ResNet-50 has the lowest *top*-1 *ACC*. The classifiers based on rotation prediction network and contrastive learning network yield much higher *top*-1 *ACC*s of 36.75% and 38.68%, respectively. In general, the *top*-1 *ACC*s are not as high as those obtained on CIFAR-10. To our surprise, the two classifiers achieve little or even no performance improvement after fine-tuning. We think the reason is that Caltech-256 data set covers too many categories while each category does not contain enough training images. This may lead to underfitting of the SSMs. Owing to the lack of training images, fine-tuning could not obviously improve the classification results. In spite of this problem, the mutual learning of the two classifiers still yields at least 2.28% improvement in *top*-1 *ACC*. We think this justifies the effect of the proposed MLMSM framework.

To justify the generalization ability of the proposed framework, we additionally conduct an ablation study using SSMs based on the VGG-11 backbone network on CIFAR-10. The experimental results are shown in Table 4. The results based on VGG-11 are generally not so good as ResNet-50 owing to its simpler network structure, but the two SSMs after mutual learning gain solid improvement in *top*-1 *ACC*. It is noteworthy that the performance of the contrast learning network is improved by 7.87% after mutual learning. This is because its initial *top*-1

TABLE 2 The *top*-1 *ACC* of different method configurations on CIFAR-10 based on ResNet-50 backbone

Methods	Without fine-tune (%)	Fine-tune (%)
Supervised	63.76	\
Rotation prediction	66.73	82.22
Contrastive learning	85.50	86.71
Rotation prediction after mutual learning	\	83.12
Contrastive learning after mutual learning	\	88

TABLE 3 The *top*-1 *ACC* of different method configurations on Caltech-256 based on ResNet-50 backbone

Methods	Without fine-tune (%)	Fine-tune (%)
Supervised	17.59	\
Rotation prediction	36.75	36.47
Contrastive learning	38.68	39.17
Result after mutual learning	\	41.45

TABLE 4 The *top*-1 *ACC* of different method configurations on CIFAR-10 based on VGG-11 backbone

Methods	Fine-tune (%)
Rotation prediction	69.95
Contrastive learning	62.35
Rotation prediction after mutual learning	70.28
Contrastive learning after mutual learning	70.22

TABLE 5 The *top*-1 *ACC* obtained by different mutual learning parameters on Caltech-256 (ResNet-50 backbone)

Testing set				Training set			
τ	$\alpha = 0.1(\%)$	$\alpha = 0.2(\%)$	$\alpha = 0.3(\%)$	τ	$\alpha = 0.1(\%)$	$\alpha = 0.2(\%)$	$\alpha = 0.3(\%)$
1	39.04	39.38	38.92	1	93.75	93.68	94.49
10	41.45	40.26	39.52	10	96.36	96.23	90.78
20	40.97	40.36	39.77	20	96.11	94.57	95.54
30	40.75	40.03	39.63	30	95.60	93.79	94.36

ACC is obviously lower than the rotation prediction network, and the results of mutual learning will theoretically surpass each of the SSMs.

One of the key issues to be addressed in the mutual learning is the selection of the temperature τ and hyperparameter α. We test the mutual learning module (ResNet-50 backbone) with different τs and αs on Caltech-256 data set, and demonstrate the results in Table 5 where the left part shows the *top*-1 *ACC* on the testing set and the right part shows that on the training set. It can be observed that $\tau = 10$ and $\alpha = 0.1$ lead the training of mutual learning module to better result. Therefore, our framework adopts this parameter setup in all the experiments.

4.4 | Comparative experiments

We compare the proposed framework with several state-of-the-art works by demonstrating the *ACC*, *RECALL*, and *PRECISION* obtained by these methods on CIFAR-10 and Caltech-256 data sets. The involved comparative methods are:

Pseudo-labeling[9]: This method is based on self-training. First of all, a neural network is trained using only the labeled images, then the network predicts pseudolabels for the unlabeled training images. The images with maximum classification confidence are selected and added into the training set together with their pseudolabels as if they were true labeled data, and then the classifier is retrained based on the enlarged training set. This procedure will be repeated until the performance does not change much. During the model update, a weight is adaptively computed to balance the influence of images with pseudolabels to the classifier. In this experiment, we adopt ResNet-50 as the classifier.

Deep Co-training[10]: This method is proposed to learn the complementary information of two networks such that the classification performance can be promoted. Each network learns from respective view, and view difference should be encouraged. Therefore, the adversarial examples for each network are generated from the original images such that these examples could be difficult for the current network but easy for the other one to classify. Theoretically, the original image and the adversarial examples should have the same class label. To this end, the prediction of the original image by the current network is forced to be the same as the prediction of the adversarial examples by the other network. In this experiment, the two classifiers are ResNet-50 and VGG-13, respectively.

Self-supervised Rotation Prediction (SSRP)[21]: This is a self-supervised method, which augments the sample set by rotating each image by 90°, 180°, and 270°, and learns to predict the rotation angles. The representations of feature extraction part of the network are used to train a classifier based on the labeled images. In method implementation, we use ResNet-50 and AlexNet as the backbone network, respectively, and fine-tune the feature extraction part via the labeled images during the training of classifier.

Self-supervised Contrastive Learning (SSCL)[20]: This self-supervised method enhances the sample set through random augmentation, and learns to make consistent predictions for the augmented samples that are derived from the same image. Then, similar to Reference [21], the representations of feature extraction part of the network are used to train a classifier based on the labeled images. We also use ResNet-50 as the backbone, and fine-tune the feature extractor via the labeled images.

We implement the aforementioned methods as well as the proposed framework (MLMSM) on CIFAR-10. The *top*-1, *top*-3, and *top*-5 *ACC*s are demonstrated in Table 6, where the method abbreviations are explained under the table. Since the proposed framework exploits mutual learning to promote two or more SSMs, we term the proposed algorithms as MLMSM(SSRP–VGG), MLMSM(SSCL–VGG), MLMSM(SSRP–ResNet), MLMSM(SSCL–ResNet), and MLMSM(m1–m2–m3–m4) wherein SSRP–VGG and SSCL–VGG mutually learn from each other to obtain MLMSM (SSRP–VGG) and MLMSM(SSCL–VGG), SSRP–ResNet and SSCL–ResNet (actually SSCL) mutually learn from each other to obtain MLMSM(SSRP–ResNet) and MLMSM(SSCL–ResNet), MLMSM (m1–m2–m3–m4) is the best model after mutual learning of SSRP–VGG, SSCL–VGG, SSRP–ResNet, and SSCL–ResNet. SSRP–VGG and SSCL–VGG use VGG-11 as backbone while SSRP–ResNet and SSCL–ResNet use ResNet-50 as backbone.

Classification through pseudolabels fails to achieve good results because the pseudolabels are derived from the model trained only on labeled data and this process does not exploit the semantic contained in the unlabeled data. Deep cotraining improves the generated target labels through joint learning of two networks that are designed to make predictions from different views. Therefore the result is much higher than pseudolabel method. However, this method still cannot avoid the suboptimal label estimation. In addition, it follows a multitask paradigm, and the balancing between the supervised and unsupervised part in the loss function is not seriously considered. SSRP–AlexNet is inferior to deep cotraining owing to the simple structure of the

TABLE 6 The *ACC* of different methods on CIFAR-10

Methods	*top*-1 *ACC* (%)	*top*-3 *ACC* (%)	*top*-5 *ACC* (%)
Psuedolabeling[9]	75.36	92.76	97.35
Deep cotraining[10]	86.24	96.93	99.08
SSRP–AlexNet[21]	64.87	89.60	96.41
SSRP–ResNet[21]	82.22	96.09	98.74
SSCL[20]	86.71	97.38	99.47
MLMSM(SSRP–VGG)(m1)	70.28	90.63	96.5
MLMSM(SSCL–VGG)(m2)	70.22	90.76	96.68
MLMSM(SSRP–ResNet)(m3)	83.12	96.05	98.87
MLMSM(SSCL–ResNet)(m4)	88	98.09	**99.58**
MLMSM(m1–m2–m3–m4)	**88.49**	**98.24**	99.55

Notes: m1–m4 denotes SSRP–VGG, SSCL–VGG, SSRP–ResNet, and SSCL–ResNet, respectively. The bold values denote the best performance.

Abbreviations: MLMSM(m1–m2–m3–m4), Mutual Learning of Multiple Self-supervised Models (m1, m2, m3, and m4); MLMSM(SSCL–ResNet), Mutual Learning of Multiple Self-supervised Models (Self-supervised Contrastive Learning); MLMSM (SSCL–VGG), Mutual Learning of Multiple Self-supervised Models (Self-supervised Contrastive Learning with VGG-11 backbone); MLMSM(SSRP–ResNet), Mutual Learning of Multiple Self-supervised Models (Self-supervised Rotation Prediction with ResNet backbone); MLMSM(SSRP–VGG), Mutual Learning of Multiple Self-supervised Models (Self-supervised Rotation Prediction with VGG-11 backbone); SRP–AlexNet, Self-supervised Rotation Prediction with AlexNet backbone; SSCL, Self-supervised Contrastive Learning; SSRP–ResNet, Self-supervised Rotation Prediction with ResNet backbone.

network. When AlexNet is substituted with ResNet-50, the *ACC*s rise dramatically but are still a little lower than deep cotraining. The reason is that the rotation angle prediction only learns the orientation of the image, and this pretext task provides limited semantic to the subsequent classifier. As a comparison, classification by SSCL yields even better results than deep cotraining because the pretext task is so complex that it must learn to discover intrinsic characteristic from random image augmentations. After mutual learning, our framework (MLMSM(SSRP–ResNet) and MLMSM(SSCL–ResNet)) achieves further improvements in *top*-1, *top*-3, and *top*-5 *ACC*s over classification based on SSRP and SSCL. Owing to the simpler backbone structure, MLMSM (SSRP–VGG) and MLMSM(SSCL–VGG) did not yield competitive results. However, when we force each of the four models (SSRP–VGG, SSCL–VGG, SSRP–ResNet, and SSCL–ResNet) to learn from the rest models, each model achieves solid performance improvement. In Table 6, the best model after mutual learning (MLMSM(m1–m2–m3–m4)) yields higher *top*-1 and *top*-3 *ACC*s than MLMSM(SSRP–ResNet) and MLMSM(SSCL–ResNet). Table 6 indicates that the mutual learning between two or more SSMs can further improve the performance even if some involved SSMs do not have encouraging performance. We believe this is because mutual learning forces the involved networks to focus on the complementary information between one another such that the correct predictions can be shared.

To achieve fine-grained understanding about the classification result, we compute the *RECALL*s yielded by these methods with respect to each class and show them in Table 7 where the method abbreviations as well as notations R1–R10 are explained under the table. We found that MLMSM(m1–m2–m3–m4) has the best results for five of the 10 classes. MLMSM (SSCL–ResNet) and SSCL yield the best results for two classes, respectively. We found that the

TABLE 7 The *RECALL* yielded by different methods with respect to each class on CIFAR-10

Methods	R1 (%)	R2 (%)	R3 (%)	R4 (%)	R5 (%)	R6 (%)	R7 (%)	R8 (%)	R9 (%)	R10 (%)
Psuedolabel[9]	80.1	88.7	59.3	61.7	63.2	65.1	85.8	78.7	86.2	84.8
Deep cotraining[10]	90.2	93.7	74.9	71.7	85.6	**83**	89.5	87.3	92.7	93.8
SSRP–AlexNet[21]	64.1	71	60.7	32.2	55.9	56.1	75.8	77.8	76.7	78.4
SSRP–ResNet[21]	85.4	93.3	70	65.4	84.1	70.4	90.3	88.2	90.1	85
SSCL[20]	91.7	96.6	81.1	60.7	86	81.9	**92.5**	86.3	94.8	**95.5**
MLMSM(SSRP–VGG)(m1)	77	80.1	54.4	42.2	66.1	65.0	85.2	73.5	78.1	81.2
MLMSM(SSCL–VGG)(m2)	76.8	80.1	56.1	45.2	65.1	65.3	85.8	72.3	75.9	79.6
MLMSM(SSRP–ResNet)(m3)	89.2	89.8	77.4	59	83.1	77.2	89.2	87.9	90.4	88
MLMSM(SSCL–ResNet)(m4)	**92.7**	**96.6**	79.9	69.8	87.7	79.8	91.9	92	95	94.6
MLMSM(m1–m2–m3–m4)	91.1	96	**82.8**	**72.6**	**87.9**	79.4	92.1	**92.1**	**96.5**	94.4

Notes: R1–R10 stands for the *RECALL* with respect to each class. m1–m4 denotes SSRP–VGG, SSCL–VGG, SSRP–ResNet, and SSCL–ResNet, respectively. The bold values denote the best performance.

Abbreviations: MLMSM(m1–m2–m3–m4), Mutual Learning of Multiple Self-supervised Models (m1, m2, m3, and m4); MLMSM(SSCL–ResNet), Mutual Learning of Multiple Self-supervised Models (Self-supervised Contrastive Learning); MLMSM(SSCL–VGG), Mutual Learning of Multiple Self-supervised Models (Self-supervised Contrastive Learning with VGG-11 backbone); MLMSM(SSRP–ResNet), Mutual Learning of Multiple Self-supervised Models (Self-supervised Rotation Prediction with ResNet backbone); MLMSM(SSRP–VGG), Mutual Learning of Multiple Self-supervised Models (Self-supervised Rotation Prediction with VGG-11 backbone); SRP–AlexNet, Self-supervised Rotation Prediction with AlexNet backbone; SSCL, Self-supervised Contrastive Learning; SSRP–ResNet, Self-supervised Rotation Prediction with ResNet backbone.

TABLE 8 The *PRECESION* yielded by different methods with respect to each class on CIFAR-10

Methods	P1 (%)	P2 (%)	P3 (%)	P4 (%)	P5 (%)	P6 (%)	P7 (%)	P8 (%)	P9 (%)	P10 (%)
Psuedolabel[9]	71.77	87.39	70.68	60.49	82.61	70.92	70.04	78.54	81.86	80.99
Deep cotraining[10]	83.67	93.98	**86.09**	73.01	**85.26**	74.71	**90.13**	93.17	92.61	91.33
SSRP–AlexNet[21]	70.83	75.29	50.21	58.44	67.51	58.99	72.54	59.71	74.11	63.74
SSRP–ResNet[21]	83.64	85.76	83.83	70.1	76.66	77.11	84.79	77.92	91.75	91.01
SSCL[20]	88.6	94.34	77.02	**80.08**	83.58	75.76	85.41	**94.73**	95.47	92.45
MLMSM(SSRP–VGG)(m1)	65.87	79.78	67.16	61.88	68.64	58.3	71.12	73.28	82.47	73.22
MLMSM(SSCL–VGG)(m2)	65.81	79.78	66.08	59.79	69.7	58.36	70.39	75.23	83.68	73.43
MLMSM(SSRP–ResNet)(m3)	80.65	91.73	79.55	78.56	81.47	73.04	87.2	80.2	90.85	88
MLMSM(SSCL–ResNet)(m4)	89.31	94.61	80.38	78.25	84	81.26	89.48	90.82	**95.57**	**95.08**
MLMSM(m1–m2–m3–m4)	**91.37**	**95.62**	83.47	76.74	**85.26**	**81.44**	88.81	91.46	95.36	94.49

Notes: P1–P10 stands for the *PRECISION* with respect to each class. m1–m4 denotes SSRP–VGG, SSCL–VGG, SSRP–ResNet, and SSCL–ResNet, respectively. The bold values denote the best performance.

Abbreviations: MLMSM(m1–m2–m3–m4), Mutual Learning of Multiple Self-supervised Models (m1, m2, m3, and m4); MLMSM(SSCL–ResNet), Mutual Learning of Multiple Self-supervised Models (Self-supervised Contrastive Learning); MLMSM(SSCL–VGG), Mutual Learning of Multiple Self-supervised Models (Self-supervised Contrastive Learning with VGG-11 backbone); MLMSM(SSRP–ResNet), Mutual Learning of Multiple Self-supervised Models (Self-supervised Rotation Prediction with ResNet backbone); MLMSM(SSRP–VGG), Mutual Learning of Multiple Self-supervised Models (Self-supervised Rotation Prediction with VGG-11 backbone); SSCL, Self-supervised Contrastive Learning; SSRP–AlexNet, Self-supervised Rotation Prediction with AlexNet backbone; SSRP–ResNet, Self-supervised Rotation Prediction with ResNet backbone.

TABLE 9 The *ACC* of different methods on Caltech-256

Methods	*top*-1 *ACC* (%)	*top*-3 *ACC* (%)	*top*-5 *ACC* (%)
SSRP–ResNet[21]	36.47	51.78	58.59
SSCL[20]	39.17	55.24	62.88
MLMSM(SSRP–ResNet)	39.18	55.65	62.44
MLMSM(SSCL)	**41.45**	**56.97**	**63.94**

Note: The bold values denote the best performance.

Abbreviations: MLMSM(SSCL), Mutual Learning of Multiple Self-supervised Models (Self-supervised Contrastive Learning); MLMSM(SSRP–ResNet), Mutual Learning of Multiple Self-supervised Models (Self-supervised Rotation Prediction with ResNet backbone); SSCL, Self-supervised Contrastive Learning; SSRP–ResNet, Self-supervised Rotation Prediction with ResNet backbone.

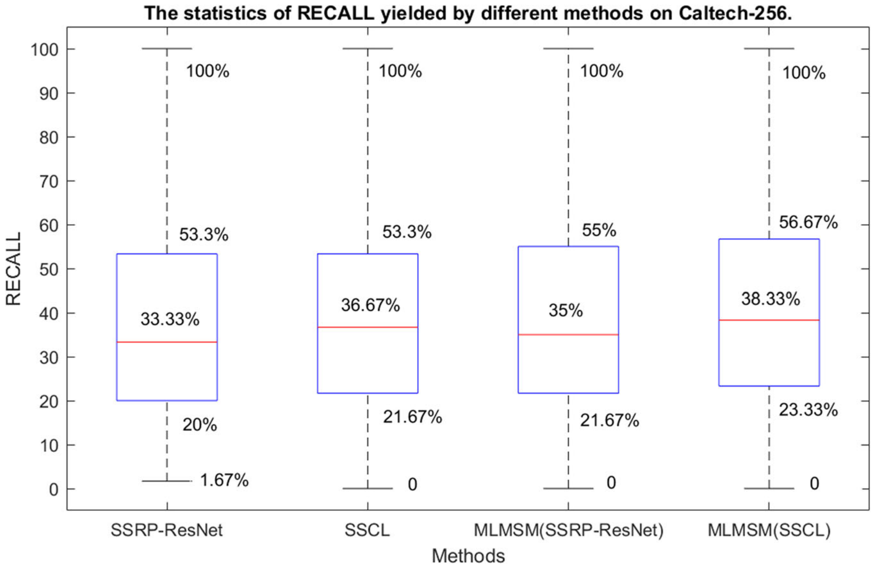

FIGURE 6 The statistics of RECALL yielded by different methods on Caltech-256. Each box describes one method, showing the minimum, maximum, median, upper, and lower quartile of the *RECALL*s over the 257 classes. The meaning of the abbreviations is: MLMSM(SSCL), Mutual Learning of Multiple Self-supervised Models (Self-supervised Contrastive Learning); MLMSM(SSRP–ResNet), Mutual Learning of Multiple Self-supervised Models (Self-supervised Rotation Prediction with ResNet backbone); SSCL, Self-supervised Contrastive Learning; SSRP–ResNet, Self-supervised Rotation Prediction with ResNet backbone

performance of contrastive learning and rotation angle prediction depends on the structure of the backbone network heavily, and this explains why SSRP–AlexNet/SSRP–VGG is inferior to SSRP–ResNet and SSCL–VGG is inferior to SSCL–ResNet. MLMSM(SSCL–ResNet)/MLMSM (SSRP–ResNet) further enhances SSCL/SSRP–ResNet with SSRP–ResNet/SSCL by complementing SSCL/SSRP–ResNet with potentially correct predictions of SSRP–ResNet/SSCL through mutual learning. Similarly, MLMSM(m1–m2–m3–m4) absorbs knowledge from the rest three SSMs and yields the best average *RECALL* for the 10 classes.

We also compute the *PRECISION*s yielded by these methods with respect to each class and show them in Table 8 where the method abbreviations as well as notations P1–P10 are explained under the table. MLMSM(m1–m2–m3–m4) also performs well and achieves the best

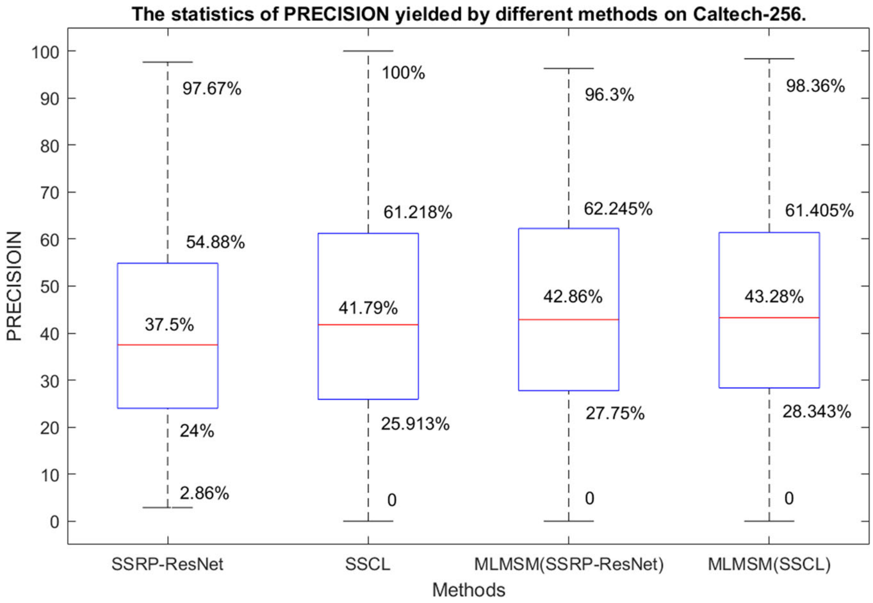

FIGURE 7 The statistics of PRECISION yielded by different methods on Caltech-256. Each box describes one method, showing the minimum, maximum, median, upper, and lower quartile of the *PRECISION*s over the 257 classes. The meaning of the abbreviations is: MLMSM(SSCL), Mutual Learning of Multiple Self-supervised Models (Self-supervised Contrastive Learning); MLMSM(SSRP–ResNet), Mutual Learning of Multiple Self-supervised Models (Self-supervised Rotation Prediction with ResNet backbone); SSCL, Self-supervised Contrastive Learning; SSRP–ResNet, Self-supervised Rotation Prediction with ResNet backbone

result for four of the 10 classes. This is actually predictable based on the discussion about the results of *RECALL*.

Besides CIFAR-10, the *ACC*, *RECALL*, and *PRECISION* are further computed based on Caltech-256 data set. Table 9 demonstrates the *top*-1, *top*-3, and *top*-5 *ACC*s of SSRP–ResNet, SSCL, MLMSM(SSRP–ResNet), and MLMSM(SSCL). On the basis of previous analysis, we think classification based on pseudolabeling and deep cotraining should be inferior to the rest methods, therefore we omit them in the experiment. According to Table 9, MLMSM(SSCL) achieve the best *top*-1, *top*-3, and *top*-5 *ACC* with the improvement of 2.27%, 1.32%, and 1.06%, respectively. Figure 6 demonstrates the statistic of the *RECALL* yielded by these methods on Caltech-256, where the horizontal axis denotes the method, the vertical axis denotes the *RECALL* and the method abbreviations are explained in the figure's caption. For each method, we plot the statistic of the *RECALL* over 257 classes via a box, in which the minimum, median, maximum, upper, and lower quartile of the *RECALLA*s are demonstrated. The MLMSM(SSCL) achieves the highest lower quartile, median, and upper quartile of the *RECALLA*s, which outperform that of the second-best method (SSCL-based classification) by 1.66%, 1.66%, and 3.37%, respectively. Note that the minimum *RECALL* is zero. The reason is that one of the 257 classes denotes the background images, which do not reflect meaningful semantics therefore are prone to be misclassified. The zero minimum pulls down the statistic values of *RECALL*. Similarly, we plot the statistic of the *PRECISION*s yielded by these methods on Caltech-256 in Figure 7 where the horizontal axis denotes the method, the vertical axis denotes the *PRECISION* and the method abbreviations are explained in the figure's caption. The lower quartile, median, and upper quartile of the *RECALLA*s yielded by MLMSM(SSCL) still outperform that of the SSCL-based classification by 2.43%, 1.49%, and 0.187%, respectively. The experiments on Caltech-256 data set also justify the effect of the proposed MLMSM framework.

5 | CONCLUSION

We propose a novel semisupervised framework termed MLMSM, which trains two SSMs, that is, the rotation prediction model and the contrastive learning model, builds classifiers based on the two models and further improves the performance through mutual learning of the two classifiers. Experimental results on CIFAR-10 and Caltech-256 data sets demonstrate the promising results of MLMSM. The key to performance promotion lies in two aspects. First, the SSM should use a carefully designed pretext task to learn comprehensive semantics from the unlabeled data that is useful to subsequent classification. Second, mutual learning enhances each of the classifiers by the complementary information provided by the other such that the correct predictions can be shared between both classifiers. This framework open ended, which means various SSMs can be used. In the future, we aim to seek for better model ensemble approach.

FUNDING INFORMATION

This study was supported by the National Natural Science of China under Grant Nos. 61972361, 61772161, and 61836002, the National Natural Science Foundation for Distinguished Young Scholars of China under Grant No. 62125201, and in part supported by the National Natural Science Foundation of China under Grant No. 62020106007.

CONFLICT OF INTERESTS

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service, and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled "Semisupervised Image Classification by Mutual Learning of Multiple Self-supervised Models."

ORCID

Jian Zhang https://orcid.org/0000-0001-6478-9192
Jianing Yang https://orcid.org/0000-0002-8623-8612
Jun Yu http://orcid.org/0000-0003-1922-7283

REFERENCES

1. Alam F, Ofli F, Imran M, Alam T, Qazi U. Deep learning benchmarks and datasets for social media image classification for disaster response. In: Atzmüller M, Coscia M, Missaoui R, eds. *Proceedings of the 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE; 2020:151-158.
2. Qin Y, Chi M, Liu X, Zhang Y, Zeng Y, Zhao Z. Classification of high resolution urban remote sensing images using deep networks by integration of social media photos. In: *Proceedings of the IGARSS 2018—2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE; 2018:7243-7246.
3. Hoffmann EJ, Werner M, Zhu XX. Building instance classification using social media images. In: *Proceedings of the 2019 Joint Urban Remote Sensing Event (JURSE)*. IEEE; 2019:1-4.
4. Yu J, Rui Y, Tao D. Click prediction for web image reranking using multimodal sparse coding. *IEEE Trans Image Process*. 2014;23(5):2019-2032.

5. Yu J, Rui Y, Chen B. Exploiting click constraints and multi-view features for image re-ranking. *IEEE Trans Multimedia*. 2014;16(1):159-168.
6. Yu J, Tao D, Wang M, Rui Y. Learning to rank using user clicks and visual features for image retrieval. *IEEE Trans Cybern*. 2015;45(4):767-779.
7. Yu J, Tan M, Zhang H, Tao D, Rui Y. Hierarchical deep click feature prediction for fine-grained image recognition. *IEEE Trans Pattern Anal Mach Intell*. 2022;44(2):563-578. doi:10.1109/TPAMI.2019.2932058
8. Yang X, Wang M, Hong R, Tian Q, Rui Y. Enhancing person re-identification in a self-trained subspace. *ACM Trans Multimed Comput Commun Appl*. 2017;13(3):1-23.
9. Lee DH. Pseudo-label: the simple and efficient semisupervised learning method for deep neural networks. In: Dasgupta S, eds. *Proceedings of the 2013 International Conference on Machine Learning (ICML)*. AAAI Press; 2013:1-6.
10. Qiao S, Shen W, Zhang Z, Wang B, Yuille A. Deep co-training for semisupervised image recognition. In: Ferrari V, Hebert M, Sminchisescu C, Weiss Y, eds. *Proceedings of the 2018 European Conference on Computer Vision (ECCV)*. Springer; 2018:135-152.
11. Zhu X, Ghahramani Z. Learning from labeled and unlabeled data with label propagation. *School Comput Sci Carnegie Mellon Univ Tech Rep CMUCALD02107*. 2002;54:2865.
12. Bengio Y, Delalleau O, Roux NL. Label propagation and quadratic criterion. In: Chapelle O, Schölkopf B, Zien A, eds. *Semisupervised Learning*. MIT Press; 2006:193-216. doi:10.7551/mitpress/9780262033589.003.0011
13. Zhang J, Yu J, You J, Tao D, Li N, Cheng J. Data-driven facial animation via semisupervised local patch alignment. *Pattern Recognit*. 2016;57:1-20.
14. Weston J, Ratle F, Mobahi H, Collobert R. Deep learning via semisupervised embedding. In: McCallum A, Roweis S, eds. *Proceedings of the 2008 International Conference on Machine Learning (ICML)*. AAAI Press; 2008:1168-1175.
15. Kipf TN, Welling M. Semisupervised classification with graph convolutional networks. *CoRR*. 2016; abs/1609.02907. Available from: http://arxiv.org/abs/1609.02907
16. Xie Q, Dai Z, Hovy E, Luong T, Le Q. Unsupervised data augmentation for consistency training. In: *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*. Vol 33. MIT Press; 2020: 6256-6268.
17. Laine S, Aila T. Temporal ensembling for semisupervised learning. *CoRR*. 2016; abs/1610.02242. Available from: http://arxiv.org/abs/1610.02242
18. Tarvainen A, Valpola H. Mean teachers are better role models: weight-averaged consistency targets improve semisupervised deep learning results. In: von Luxburg U, Bengio S, Fergus R, Garnett R, Guyon I, Wallach H, Vishwanathan SVN, eds. *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*. MIT Press; 2017:1-10.
19. Cipolla R, Gal Y, Kendall A. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE; 2018:7482-7491.
20. Chen T, Kornblith S, Norouzi M, Hinton GE. A simple framework for contrastive learning of visual representations. In: Daume H, eds. *Proceedings of the 2020 International Conference on Machine Learning (ICML)*. AAAI Press; 2020:1597-1607.
21. Gidaris S, Singh P, Komodakis N. Unsupervised representation learning by predicting image rotations. *CoRR*. 2018; abs/1803.07728. Available from: http://arxiv.org/abs/1803.07728
22. Guo D, Wang H, Wang M. Context-aware graph inference with knowledge distillation for visual dialog. *IEEE Trans Pattern Anal Mach Intell*. 2021:1. doi:10.1109/TPAMI.2021.3085755
23. Wang M, Hong R, Li G, Zha ZJ, Yan S, Chua TS. Event driven web video summarization by tag localization and key-shot identification. *IEEE Trans Multimedia*. 2012;14(4):975-985.
24. Dongdong L, Ziqiu C, Bolu W, Zhe W, Hai Y, Wenli D. Entropy-based hybrid sampling ensemble learning for imbalanced data. *Int J Intell Syst*. 2021;36(7):3039-3067.
25. Cheng J, Zheng J, Yu X. An ensemble framework for interpretable malicious code detection. *Int J Intell Syst*. 2020. doi:10.1002/int.22310
26. Zhang Y, Xiang T, Hospedales TM, Lu H. Deep mutual learning. In: *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE; 2018:4320-4328.

27. Xie Q, Luong MT, Hovy E, Le QV. Self-training with noisy student improves ImageNet classification. In: *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE; 2020:10684-10695.
28. Pham H, Xie Q, Dai Z, Le QV. Meta pseudo labels. *CoRR*. 2020; abs/2003.10580. Available from: https://arxiv.org/abs/2003.10580
29. Chen DD, Wang W, Gao W, Zhou ZH. Tri-net for semisupervised deep learning. In: Lang J, eds. *Proceedings of the 2018 International Joint Conference on Artificial Intelligence, IJCAI*. Morgan Kaufmann; 2018: 2014-2020.
30. Wang M, Li H, Tao D, Lu K, Wu X. Multimodal graph-based reranking for web image search. *IEEE Trans Image Process*. 2012;21(11):4649-4661.
31. Wang M, Liu X, Wu X. Visual classification by l_1-hypergraph modeling. *IEEE Trans Knowl Data Eng*. 2015; 27(9):2564-2574.
32. Zhu X, Ghahramani Z, Lafferty JD. Semisupervised learning using Gaussian fields and harmonic functions. In: Fawcett T, Mishra N, eds. *Proceedings of the 2003 International Conference on Machine Learning (ICML)*. AAAI Press; 2003:912-919.
33. Zhou D, Bousquet O, Lal TN, Weston J, Schölkopf B. Learning with local and global consistency. In: Thrun S, Saul LK, Schölkopf B, eds. *Proceedings of the Neural Information Processing Systems (NIPS)*. MIT Press; 2003:321-328.
34. Tang J, Qu M, Wang M, Zhang M, Yan J, Mei Q. LINE: large-scale information network embedding. In: Gangemi A, Leonardi S, Panconesi A, eds. *Proceedings of the 2015 International Conference on World Wide Web (WWW)*. ACM; 2015:1067-1077.
35. Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE. Neural message passing for quantum chemistry. In: Precup D, Teh YW, eds. *Proceedings of the 2017 International Conference on Machine Learning (ICML)*. Vol 70. AAAI Press; 2017:1263-1272.
36. Doersch C, Gupta A, Efros AA. Unsupervised visual representation learning by context prediction. In: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE; 2015:1422-1430.
37. Zhang R, Isola P, Efros AA. Colorful image colorization. In: Leibe B, Matas J, Sebe N, Welling M, eds. *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer; 2016:649-666.
38. Yu J, Yang X, Gao F, Tao D. Deep multimodal distance metric learning using click constraints for image ranking. *IEEE Trans Cybern*. 2017;47(12):4014-4024.
39. Yang X, Wang M, Tao D. Person re-identification with metric learning using privileged information. *IEEE Trans Image Process*. 2018;27(2):791-805.
40. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. *Commun ACM*. 2017;60(6):84-90.
41. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE; 2016:770-778.
42. Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. *CoRR*. 2015; abs/1503.02531. Available from: http://arxiv.org/abs/1503.02531

How to cite this article: Zhang J, Yang J, Yu J, Fan J. Semisupervised image classification by mutual learning of multiple self-supervised models. *Int J Intell Syst*. 2022;37:3117-3141. doi:10.1002/int.22814