

# Fully used reliable data and attention consistency for semi-supervised learning

Jui-Hung Chang<sup>a,\*</sup>, Hsiu-Chen Weng<sup>b</sup>

<sup>a</sup> Computer and Network Center, and Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan, R.O.C.

<sup>b</sup> Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan, R.O.C.

## ARTICLE INFO

### Article history:

Received 5 January 2022

Received in revised form 24 March 2022

Accepted 14 April 2022

Available online 25 April 2022

### Keywords:

Deep learning

Semi-supervised learning

Attention consistency

Reliable data

## ABSTRACT

Large labeled datasets represent human labor's costly consumption of resources. Therefore, semi-supervised learning leverages a large amount of unlabeled data to improve the training results in limited labels. Many methods of semi-supervised learning utilize diverse data augmentations to improve model learning and the classification rule from these changes, requiring models to spend a lot of time to adapt to the changes. Besides, reducing the noise in trained unlabeled data is also an issue that is often discussed in semi-supervised learning so that the inference from error predictions can be reduced. It may define that the data, of which the probability predicted from the model is higher than a threshold, as confident and then only train on those high-confidence unlabeled data so that the model avoids the influence from deviation of the error caused by unlabeled data predictions. However, it also leads to the fact that many unlabeled data cannot be effectively used. Thus, this study proposes a semi-supervised framework, including Attention Consistency (AC) and One Supervised (OS) algorithms, which improves efficiency and performance of the model learning by guiding the model to pay attention to classified features and judging whether the model cannot be effectively trained in existing reliable data. This way, the model fully uses unlabeled data to train. The experiment results and comparisons show that similar results can be reached using other methods within a shorter training process. This paper also analyzes the distribution of feature results and proposes a new measurement to find out distribution information.

© 2022 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In recent years, the proposals of powerful GPU hardware technology, enough labeled data, and convolutional neural networks (CNNs) [1] has led to the population of deep neural networks (DNNs) for computer vision task [2]. The growth of DNNs improves the research in various domain applications, such as medical or hyperspectral images [3–5]. The flexibility of DNNs brings great success to machine learning in sufficient data. However, lots of labeled data usually represent time-consuming and costly human labeling effort, especially for data in medical applications that require doctors to label professional information with higher cost.

Unlabeled data is relatively friendly to access so that semi-supervised learning (SSL) [6] is regarded as a more available approach in practice scenarios. SSL is aimed at learning the determining information from limited labeled and unlabeled data, and we can summarize the loss function in recent SSL methods into

two types. A general approach is Entropy Minimization which leads model prediction to meet the label. Pseudo-label [7], a simple and effective method, uses model prediction results of unlabeled data as pseudo labels and the labels are then fed into model training. Another method that often obtains recent state-of-the-art results, such as UDA [8], FixMatch [9] and Noisy Student Training [10], is Consistency Regularization which encourages model to preserve the consistency in the prediction results of differently changed data. Some utilize powerful data augmentation to require the model to be robust to the predictions after perturbations. Some use multiple models to label and train on data processed differently, and then require their results to be consistent, therefore models become stronger after alternative training.

However, most of those powerful methods focus on how to augment for training these data effectively, but a long training time is required for model adapting to learn important features from a variety of data. In addition, while we focus on data transformation, we ignore the key factor that the model based on which features to classify. In SSL study, another important issue is identifying the reliable data. Because there are unlabeled train

\* Corresponding author.

E-mail address: [changrh@mail.ncku.edu.tw](mailto:changrh@mail.ncku.edu.tw) (J.-H. Chang).

data without correct labels in SSL, in order to ensure that the unlabeled data added to training loop are reliable and will not be noisy data affecting the classification, those low-confidence training data will be excluded. Therefore, much research tries to define what the reliable data in training process is, and then we can train on the predictions that are more likely to be helpful for model. For instance, FixMatch uses a threshold to judge the confidence from probability distribution of model output so that the threshold will filter the unconfident data seen as noise. In SSL where labeled data is inherently scarce, restricting available data is easily falling into the situation that much unlabeled data is not effectively used.

To solve the problems above, this work aims to achieve two goals: one is to improve the model's ability on learning object features so as not to only predict more accurately on the objects but also accelerate learning step of distinguishing features. Another goal is to use the unlabeled data as much as possible under the reliable unlabeled data in SSL, because more data usually means better results for the datasets of insufficient labeled data. In this study, we propose a SSL framework that includes Attention Consistency (AC) and One Supervised (OS) to achieve our goals. AC calculates attention through predictions and the weights of model, encouraging the model to pay attention to the feature regions of objects which should be the same as that on the result of flip data. The consistency regularization improves the model ability of classification from features and simultaneously brings interpretability to the model. OS adds a new proposed algorithm to the whole training process for fully using the reliable data that are produced from a confidence threshold. We define the reliable data as high-confidence unlabeled data which will be used to train the model, and high confidence means that the model is more confident in its predictions. While the training results of labeled data tend to be worse than that of the past, OS will switch to another training algorithm to strengthen model's adaptability to more data. The proposed new algorithm first trains the model on the clean high-confidence labeled data, then uses the trained model to predict pseudo labels of all unlabeled data so that we can fully use all these unlabeled data to train model.

This study improves the training efficiency in SSL and achieves better performance on features identity. The following concludes our primary contributions:

- In our study, this approach is the first that uses AC to improve the existing SSL model learning ability for features of data and proposes a new SSL training algorithm that can effectively use all unlabeled data while retaining the characteristics of reliable data. We can easily implement this framework on SSL methods that use strong augmentations and predict the pseudo labels.
- In the experiments, we show that the performance can be greatly improved in short-term training compared to the latest methods such as FixMatch while maintaining accuracy in the long-term. Feature and attention analysis are also conducted to be more explainable for exploring whether the model learns correctly.
- In order to measure the feature distribution of training results, we propose a new metric to assist us in judging the representation of model results, and it is proven effective in our experiments.

The rest of the study is organized as follows. In Section 2, we review related work in different algorithms of semi-supervised learning. Section 3 introduces our mainly proposed method. Section 4 shows the experiment results and analysis. Finally, the conclusions and future work directions are summarized in the Section 5.

## 2. Related work

The development of semi-supervised learning has since begun early Artificial Intelligence (AI) research, such as Self-training [11], Support Vector Machine (SVM) [12], and Graph-based methods [13]. In this section, we explore the technologies used in recently proposed methods, not discussing all of the literatures aforementioned.

### 2.1. Entropy minimization

A common assumption in SSL is that the decision boundary of a classifier should pass through low-density regions of data distribution, and entropy minimization of classification encourages the model predation to be centralized to the classes [14]. Self-training was first proposed to solve unsupervised word sense ambiguous problem in the texts [15]. Recently, a new study [16] evaluates the performance of pre-training and self-training through object detection and semantic segmentation tasks, showing that self-training, which addresses the mismatch between the supervised and self-training objectives, can effectively extract features about the target task instead of learning from pre-trained tasks. In SSL, self-training, a basic pseudo-labeling method [17], includes the concept of low-density assumption [18]. Minimizing entropy can push the class probability away from other classes and encourage the decision boundary to pass through these low-density regions between different classes, therefore, adding entropy regularization to loss function is generally used to SSL methods [19]. Pseudo-label [7] exploits the hard labels of unlabeled data predictions from the confidence filter as pseudo labels used to compute the cross-entropy regularization. The methods of exploring high-confidence predictions of unlabeled data will be discussed in Section 2.3.

However, pseudo-labeling applied individually is not competitive against other consistency regularization [20] so that it is usually used as the part of loss terms. For instance, MixMatch [21] exploits MixUp [22] to increase changes on labeled and unlabeled data and Sharpening which adjusts temperature of average distribution of augmented results to reduce the entropy of categorical predictions. Virtual Adversarial Training (VAT) [23] uses entropy minimization as a component to obtain stronger predictions under the perturbation. FixMatch [9] augments unlabeled data in weakly to produce pseudo labels trained to be consistency regularization with strongly-augmented results.

### 2.2. Consistency regularization

Consistency regularization, widely used in state-of-the-art SSL algorithms, requires the model prediction to be consistent on the perturbations of data. Through the variation in the same data, the model will be trained to learn the features of each data for finding the consistency with them. There are many methods for increasing variability, by a common technology, data augmentation, a variety of differences in data can be easily achieved. A simple intuitive way is to train the model to be label consistency of prediction with these perturbations. The classic  $\Pi$  model defined by Laine and Aila [24] utilizes stochastic augmentation and network with dropout, the concept is similar to [25].  $\Pi$  model evaluates the model outputs of unlabeled data twice in random cases of transformation and dropout and then calculates the mean squared error of the predictions as the unsupervised loss. The advanced method of  $\Pi$  model is Temporal Ensembling [24] that extend from  $\Pi$  model to ensemble temporal prediction results by exponential moving average of record of results as the part of unsupervised loss so that the training enhances the abilities of classification and the robustness for augmentation and

temporal variation. SSLDEC [26] incorporates the random augmentation with deep embedded clustering distribution consistency, and the clustering regularization makes it require less time to tune hyper-parameters. FeatMatch [27] jointly learn on classification and consistency on feature-based augmentation so that it can incorporate information of transformations and class representation.

Another form of consistency is Mean Teacher [28], and there are multiple models' weights. This type of methods iteratively trains and predicts the results, then producing the new weights of models, and the prediction results should be consistent with each other. The models of "Mean Teacher" [28] consist of current parameters of model, student model, and the exponential moving average of model weights, teacher model, and required to be consistent with each prediction so that it can more stably, train and predict the unlabeled data. Noisy Student Training [10] extends the concept of Knowledge Distillation [29] to Knowledge Expansion that makes student model trained to be consistent, and even better than its teacher model through heavy noise. The whole training algorithm of Noisy Student Training incorporating the self-training is first training the teacher model with labeled data, and then iteratively doing that, the teacher model predicts pseudo labels of unlabeled data, a student model trains on the truths and pseudo labels of labeled and unlabeled data including noises, and the teacher network will be replaced by the student network.

One of the key factors of consistency regularization is the augmentation technologies, and more and more strategies of transformations are proposed to improve the model's performance. For instance, MixUp [22] is widely used in SSL, training on the artificial instances constructed from the linear interpolation of two samples. Manifold Mixup [30] utilizes MixUp results from mixing the hidden states and labels or pseudo labels as unreliable data to improve the representations of real instances. Different from these past works which mix the labeled samples and unlabeled samples internally. MixMatch [21] implements MixUp [22] on labeled data with unlabeled data so that training on labeled and unlabeled samples can be better fused. Unsupervised Data Augmentation (UDA) [8] is target-oriented augmentation strategies, Auto Augment [31] for image classification, and combined with Training Signal Annealing (TSA) to avoid over fitting the labeled data so that the model can be robust through consistency regularization. FixMatch [9] trains the consistency on pseudo labels of weakly-augmented and predictions of strongly-augmented, CTAugment [32] and RandAugment [33], unlabeled sample, and it achieves state-of-the-art performance through the simpler regularization.

### 2.3. Reliable data

One of the important issues in SSL is whether unlabeled data are reliable, and we tend to give confident data higher priority to train. For example, Pseudo Label directly utilizes all pseudo labels of unlabeled data to train, and that implies all of unlabeled data reliable and is not realistic. Curriculum learning (CL) [34] is a classic method that progressively trains on easy to hard samples, and it is similar to the human learning that we start in simple general knowledge to challenging states. Many algorithms proposed to design how to judge whether the data are reliable. The study of Hacothen and Weinshall [35] discusses how different pacing functions to improve the performance on deep networks, using two scoring functions, Transfer and Self-taught scoring function, and three pacing functions, Fixed and Varied exponential pacing and Single step pacing, to compare the effect. Shaoyue Song et al. [36] propose an easy-to-hard strategy to solve the task of within-image co-saliency detection. To alleviate the decision

problems of whether the data is easy enough, self-paced learning (SPL) [37] automatically defines the easy training instances by indicators iteratively updated through model training. Lu Jiang et al. [38] considers the diversity neglected in SPL that restriction on only easy samples trained at the beginning leads to a partial understanding of all data. Thus, the defect of only training on only part of data can be seen.

In semi-supervised and unsupervised learning, we often use CL or SPL to solve the problem of how to select reliable data. Progressive unsupervised learning (PUL) [39] proposes a SPL clustering in the unsupervised task of person re-identification (re-ID), and it shows that as the model gets fine-tuned, more complex instances will be covered in reliable area. Paola Cascante-Bonilla et al. [40] proposes pseudo-labeling under curriculum, demonstrating that curriculum labeling the unlabeled data by progressively selecting easy to hard can reach the comparable results with other consistency regularization approaches.

A threshold in FixMatch [9] restricts low confidence unlabeled data which the highest probability is less than the threshold so that it forms a natural curriculum learning. Ryuichiro [41] quantifies the unlabeled data through Kullback-Leibler (KL) divergence of labels' conditional probabilities as trusted or untrusted data and then MixUp them while mixing semi-supervised and noise loss.

### 2.4. Attention mechanism

Attention mechanism [42] aims to find the weights for strengthening the regions where we are concerned about. Since "Attention" [43] has achieved great success in Natural Language Processing (NLP), more research has begun to apply the attention mechanism to various fields. Attention in CNN is usually used to increase the interpretability or enhance the important part for computer vision tasks.

Ronghang Hu et al. [44] proposed a new model making use of compositional reasoning without relying on expert supervision for visual question answering, and the reasoning steps could be explainable through the attentions. Residual Attention Network (RAN) [45] stacks multiple attention modules to capture different levels of attention so that the model can guide learning feature to best performance. Squeeze-and-Excitation Networks (SE-Net) [46] uses squeeze-and-excitation block to perform channel-wise features recalibration as attention mechanism. In the fine-grained tasks, multi-attention convolutional neural network (MA-CNN) [47] produces part attentions from channel features output from CNN, it shows that we can explore the location of attentions to improve classification.

Attention results are also used in consistency regularization, which model can learn the relationships on it from. Visual Attention Consistency [48] utilizes the attentions of augmented data to preserve consistency with transformation. The attention consistency guides the model to learn the key factors of sample classification even if the sample has been transformed. In SSL, attention-based label consistency (ALC) [49] contains the attention regularization but only for prediction labels. The target of ALC is that the data close to each other should share same probability distributions. In addition to introduce a new YSneaker dataset, Semi-Supervised Attention (SSA) [50] proposes a self-attention-based multi-instance CNN framework that calculates the instance-level features after DCNN and combines features with additional attention network to produce bag-level attention representation.

**Table 1**  
Definitions of notations.

Notations	Definitions	Notations	Definitions
$x_k$	A training sample	$\mathbb{R}^d$	Set of $d$ -dimensional real numbers
$p_k$	A one-hot label	$\theta$	Trainable weight of model
$N_s$	Number of labeled data.	$q$	Class prediction distributions
$N_u$	Number of unlabeled data.	$\tau$	A confidence threshold
$\mathcal{X}$	Labeled dataset contains $N_s$ labeled data $(x_k, p_k)$	$\hat{q}$	Maximum of prediction probabilities
$\mathcal{U}$	Unlabeled dataset contains $N_u$ unlabeled data $x_k$	$f$	Feature representations $f \in \mathbb{R}^{C \times H \times W}$
$\mathbf{D}$	Dataset including labeled and unlabeled data $\{\mathcal{X}, \mathcal{U}\}$	$w$	Weight of final fully connected layer $w \in \mathbb{R}^{L \times C}$
$N$	Number of all training data	$\mathbb{L}_n$	Loss result in epoch $n$
$M$	Attention heatmaps before max pooling	$\bar{\mathbb{L}}_n$	Average pass 5 losses of epoch $n$
$z$	Representation from max pooling $z \in \mathbb{R}^{C \times L}$	$c$	Center of 2-dimensional representations $z$
$N_c$	Number of classes		

### 3. Methodology

This chapter explains the workflow and algorithm of the mainly proposed SSL method which contains Attention Consistency (AC) and One Supervised (OS) and how to combine our method with SSL training progress. Besides, a new metric proposed in the study will be illustrated.

#### 3.1. Definition

This section defines the notations used in this paper and explains their definitions in Table 1.

#### 3.2. Background

For the consistency and confidence of semi-supervised learning methods, we define as follow. Let  $\mathcal{X} = \{(x_k, p_k) : k \in (1, 2, \dots, N_s)\}$  be the  $N_s$  labeled data and  $\mathcal{U} = \{x_k : k \in (1, 2, \dots, N_u)\}$  be the  $N_u$  unlabeled data, where  $x_k$  and  $u_k$  are training examples of labeled and unlabeled data and  $p_k$  is the one-hot labels of labeled data.

For the input  $x$ , the  $y$  class predicted distribution  $p_{\text{model}}(y|x; \theta)$  is produced from deep convolutional neural network (DCNNs) with the model parameter  $\theta$ . In these methods, the loss  $\mathcal{L}$  usually consists of two terms: the supervised loss  $\mathcal{L}_s$  for labeled data and the unsupervised loss  $\mathcal{L}_u$  applied to unlabeled data. Specifically, the supervised loss function  $\mathcal{L}_s$  is computed the following:

$$\mathcal{L}_s = \frac{1}{N_s} \sum_{x, p \in \mathcal{X}} H(p, p_{\text{model}}(y|x; \theta)) \quad (1)$$

where  $H(p, q)$  is the cross-entropy between the probability distributions  $p$  and  $q$ . The unsupervised loss function can be summarized as follows:

$$\mathcal{L}_u = \frac{1}{N_u} \sum_{u \in \mathcal{U}} h(p, p_{\text{model}}(y|x; \theta)) I(\max(q) \geq \tau) \quad (2)$$

Where  $h(p, q)$  denotes the algorithm of different methods.  $I(\max(q) \geq \tau)$  represents indicator function of whether the unlabeled data to be training model is reliable, where  $q = p_{\text{model}}(y|u; \theta)$  is predicted distribution result and  $\tau$  is the confidence threshold. Pseudo-label and FixMatch can be respectively rewritten by following Eq. (3) and Eq. (4):

$$\mathcal{L}_{up} = \frac{1}{N_u} \sum_{u \in \mathcal{U}} H(\hat{q}, q) I(\max(q) \geq \tau) \quad (3)$$

$$\mathcal{L}_{uf} = \frac{1}{N_u} \sum_{u \in \mathcal{U}} H(\hat{q}, p_{\text{model}}(y|\mathcal{A}(u); \theta)) I(\max(q) \geq \tau) \quad (4)$$

where  $\hat{q} = \arg \max(q)$  denotes the pseudo-label from class distribution.  $\mathcal{A}(u)$  Indicates a strongly-augmented unlabeled data  $u$ . Pseudo-label unsupervised loss function encourages predicted class distribution  $q$  to be closed to pseudo-label computed based

on the maximum predicted probability of itself. Unsupervised loss function of FixMatch shown in Fig. 1 is different from Pseudo-label in that cross-entropy enforces prediction of the strongly-augmented image  $\mathcal{A}(u)$  to be same as the pseudo-label of weakly-augmented image  $u$ . For the artificial pseudo-label, a threshold is implemented to require the highest-class probability of  $q$  to higher than a predefined value to ensure the reliability of unlabeled data.

#### 3.3. Attention Consistency (AC)

##### 3.3.1. Class Activation Map (CAM)

AC regularization is inspired by a supervised multi-label image classification, Visual Attention Consistency [48]. The approach can improve the model's capability to focus attention on decisive regions of image classification. In this paper, Visual Attention Consistency is used as the consistency regularization in SSL for improving model feature identity. To get the label-relevant regions, Class Activation Map (CAM) [51] is applied to extract heatmap as attention. The last of deep convolutional layers outputs the feature maps of input image to a Global Average Pooling (GAP) that feeds the pooling results of the features into the classifier layer. The dimension of the feature representations  $f$  is  $C \times H \times W$ , where  $C, H, W$  denote the number of channels, height, and width of feature in pixels. CAM computes attention heatmaps by separately adding classification weights to all feature channels and then summing all weighted feature channels. The weights  $w \in \mathbb{R}^{L \times C}$  are taken from the final fully connected (FC) layer which the feature is fed into for classification. Therefore, the attention heatmaps for label  $l$  can be represented using the following equation:

$$M_l = \sum_{k=1}^C w_l^k f_k \quad (5)$$

Where  $w_l^k$  indicates the weight of the FC layer for channel  $k$  to class  $l$  and  $f_k$  denotes the features of channel  $k$  after deep convolutional layers. To perform product on all feature channels with different classes for the channel, we expand feature and weight and then multiply channel-wise. As shown in Fig. 2, the dimension of feature maps is translated to  $1 \times C \times H \times W$  and the shape of FC weights expanded into  $L \times C \times 1 \times 1$ , and then, we perform the channel-wise multiplication and sum the output results, the shape of which is  $L \times C \times H \times W$ , so that we get all classes of summed heatmaps  $M_l$  in shape of  $L \times H \times W$ .

##### 3.3.2. Attention consistency regularization

In this section, we describe how to implement CAM to achieve AC and use it to improve existing algorithms. We show the process of attention consistency regularization in Fig. 3. Assuming the process of CAM can be represented as  $M = A(f)$  and  $T(\cdot)$  is the horizontal flip operation, we input the images  $x$  and flipped



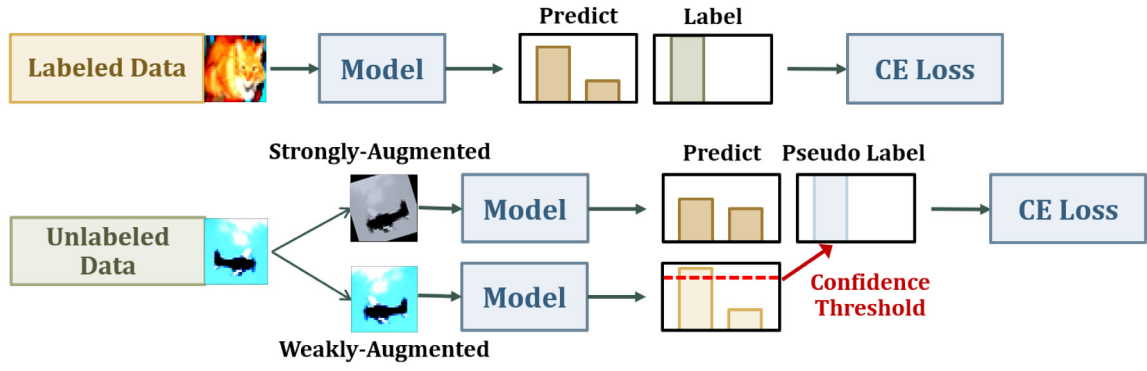


Fig. 1. Diagram of FixMatch.

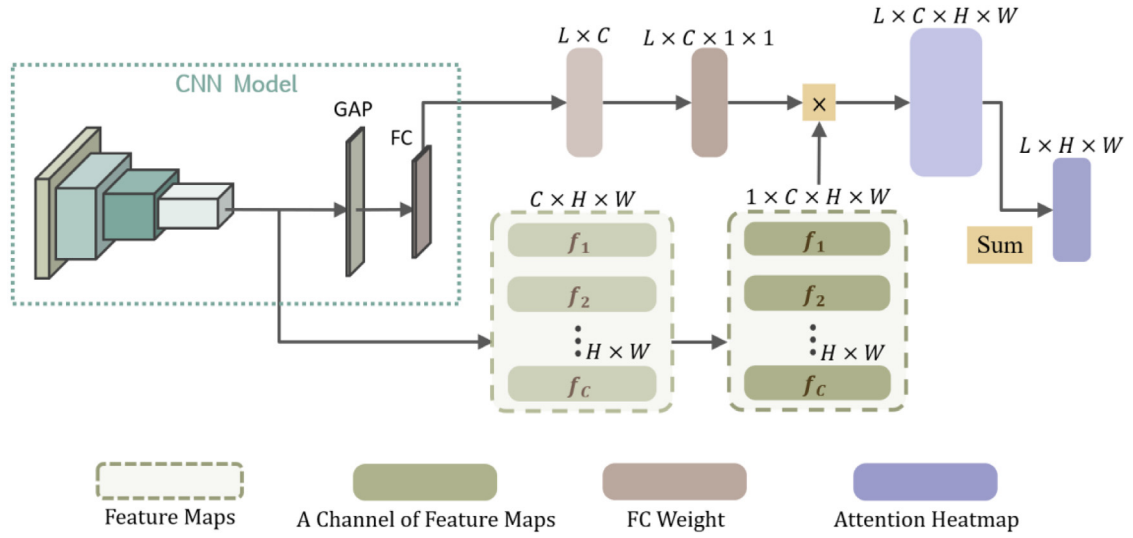


Fig. 2. Features translate to attention heatmaps.

image  $T(x)$  into the deep convolutional model. For images  $x$ , we get the attention heatmaps from DCNNs and the last FC weight from a classifier result, and the later will be applied to calculate the prediction consistency for target task. The former will further conduct the Mean Square Error (MSE) loss between the features of original image and the horizontal flipped results of the flipped image features to achieve consistency under the perturbation, which can be represented using following formula:

$$\mathcal{L}_a = \frac{1}{NL} \sum_{n \in N} \sum_{l \in L} \|\mathcal{N}_{layer}(A(x_n)) - \mathcal{N}_{layer}(T(A(T(x_n))))\|_2 \quad (6)$$

Where  $x_n$  is the image  $n$  of  $N$  data, in which number of total images  $N = N_s + N_u$  and  $\mathcal{N}_{layer}(\cdot)$  is performing Layer Normalization [52] on every channel of feature heatmaps. Therefore, the final loss is the sum of entropy minimization and heatmaps consistency loss.

In order to improve model ability of learning feature, we combine the attention consistency regularization and those consistency regularization algorithms in semi-supervised learning. The attention heatmaps of labeled and unlabeled data will separately be attracted from model for the regularization. The illustration of FixMatch combined with AC is shown in Fig. 4. The attention consistency loss, Eq. (6), will be added to original cross-entropy (CE) loss of classification in supervised and unsupervised losses. For labeled data, the processing, as previous Fig. 3, the images are flipped horizontally to calculate AC. And for unlabeled data, the image is compared with its flipped version for MSE loss, and then the masks on probability threshold of predictions are

implemented on CE and MSE unsupervised losses. Specifically, we use the weakly-augmented image as the computed data in the FixMatch algorithm. After applying the reliable threshold and averaging the results, we will sum two unsupervised losses as labeled data loss and then add labeled data loss as the final loss  $\mathcal{L}$  to train the model. We will solve the following equations:

$$\mathcal{L}_s = \frac{1}{N_s} \sum_{x, p \in \mathcal{X}} H(p, p_{model}(y|x; \theta)) + \mathcal{L}_a(\mathcal{X}) \quad (7)$$

$$\mathcal{L}_u = \frac{1}{N_u} \sum_{u \in \mathcal{U}} I(\max(q) \geq \tau) (h(p, p_{model}(y|x; \theta)) + mse(I_n, T(I_n))) \quad (8)$$

$$\mathcal{L} = \mathcal{L}_s + \mathcal{L}_u \quad (9)$$

### 3.4. One Supervised (OS)

This section introduces the details of new algorithm added to the training process for improving the usage of reliable unlabeled data. The new algorithm OS can be applied to the methods that use the confidence mechanism. The indicator  $I(\max(q) \geq \tau)$  based on the condition judging whether maximum probability is over the threshold will be used to ignore low-confidence predictions, and OS adds addition condition for another environment where the model is trained on not limited to reliable data without modifying the original confidence method. An overview of OS algorithm different from the general training method is shown in

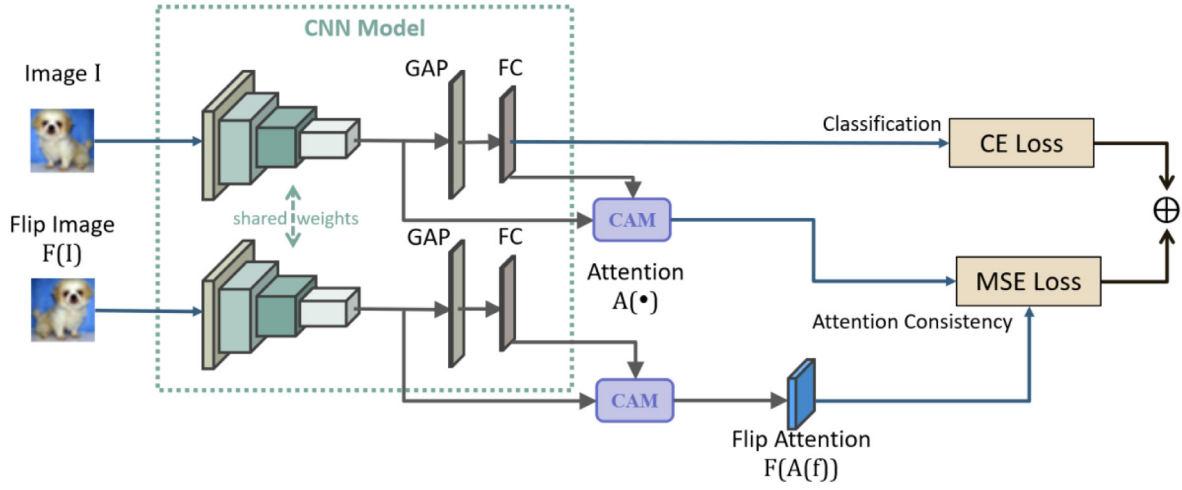


Fig. 3. Attention consistency regularization.

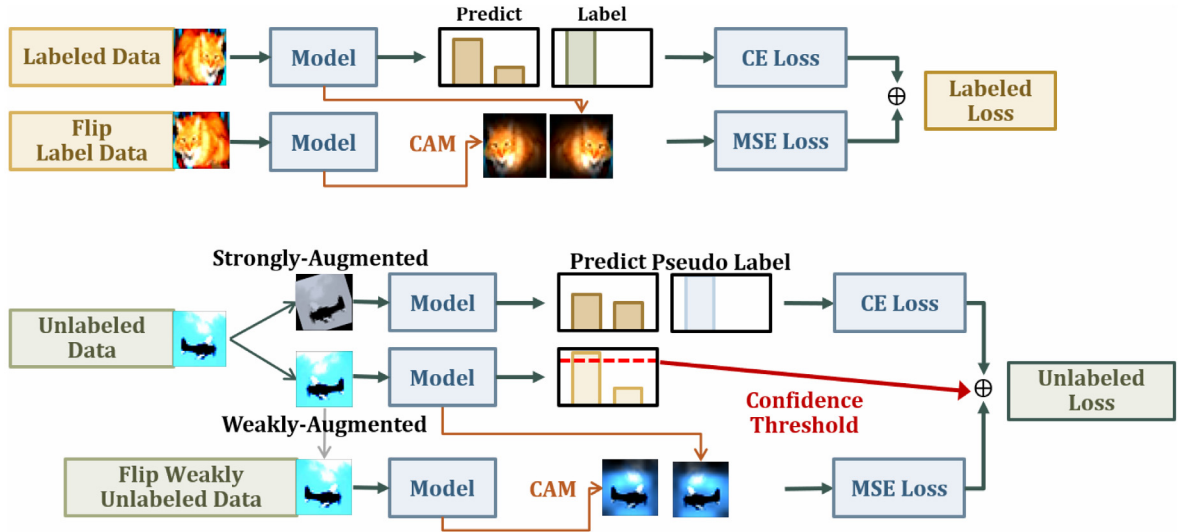


Fig. 4. FixMatch with Attention Consistency.

Fig. 5, where an additional decision is used to include more cases. The process of OS is mainly to exploit the dependable model which is trained by confident labeled data to predict pseudo labels of all unlabeled data which are expected to be more precise. The complete unlabeled data will train the shared weights model so that the model can learn more variability of data. Through the mechanism, while the model cannot learn more from these reliable data, OS request model to be trained on all unlabeled data so that it can detect more reliable data for training.

Assuming that the loss result of labeled data in epoch  $n$  is  $\mathbb{L}_n = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \ell(x)$ , where  $\ell(\cdot)$  indicates the supervised loss of the algorithm, and the average past 5 losses of labeled data can be represented as  $\bar{\mathbb{L}}_n = \frac{1}{5} \sum_{k=1}^5 \mathbb{L}_{n-k}$ , the training process of OS is shown in Algorithm 1. Algorithm 1 illustrates the framework to implement on other algorithms. In this study, when OS is combined with AC algorithm, the supervised and unsupervised losses in “One Supervised” branch that OS status is true will be added to the attention regularization so that it can also fully utilize the complete unsupervised attention losses.

We can analyze the complexity of Algorithm 1. In SSL, the time complexity calculates the times of training all  $N$  data in  $n$  epochs. The OS algorithm adds an additional condition in each epoch end and switches the training process, where model would

train on same data. Besides, The AC method use another flip data to calculate the attention consistency results, thus it will train  $2N$  data with  $O(n)$  complexity. Therefore, our algorithm would not increase the complexity of  $O(n)$ . In our methods OS and AC, the model could learn more efficiently so that we use less  $n$  to achieve same results.

### 3.5. Feature metrics

In this section, we introduce a new metric to measure the feature distribution. The study focuses on whether the model is learning well on these uncertain pseudo labels. Therefore, AC is used to enhance model's learning ability, and we utilize the max pooled features to be transferred to the FC layer for classification. The features are reduced to two dimensions for visualization so that we can observe the class distribution of data. It is usually used in clustering tasks to project data to low dimension for clustering the features. A general concept used in previous SSL graph-based method [53] encourages inter-class separation and intra-class compactness, as shown in Fig. 6, and we use the concept to understand the distribution results.

There are measures for data distribution in statics, such as central tendency [54]. However, as far as we are concerned, there

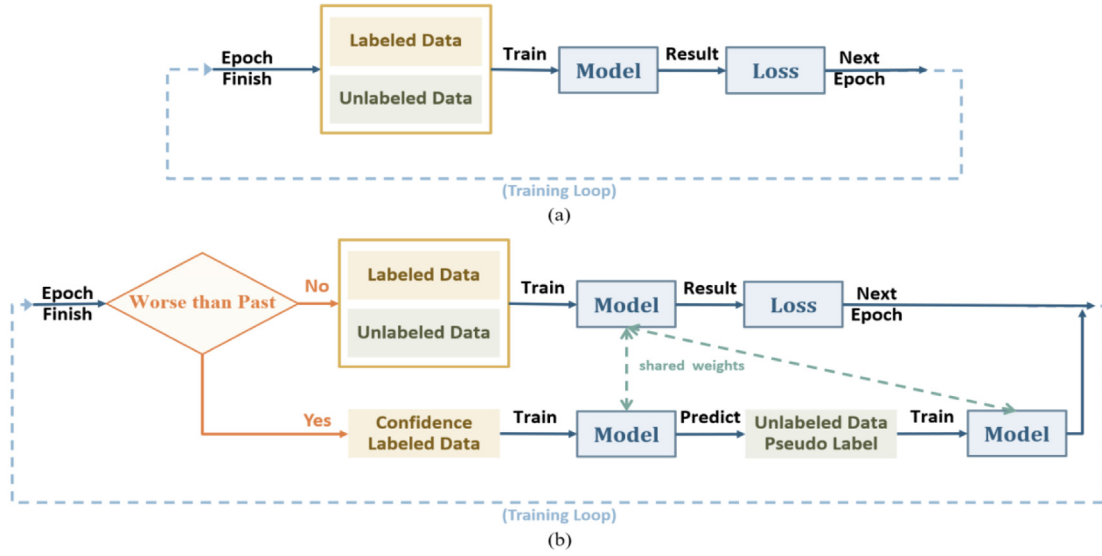


Fig. 5. Difference on general training with additional OS.

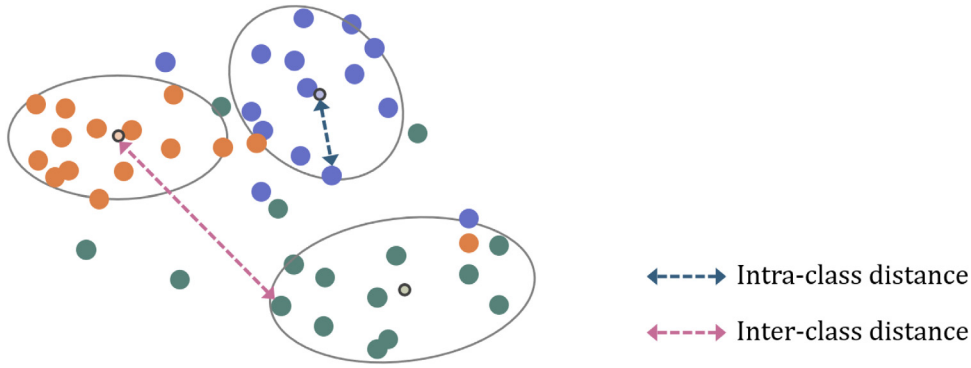


Fig. 6. Illustration of intra- and inter-class.

are no related literatures to measure the data distribution in classification or clustering problems, so we defined a new metric *score* to measurement the features as follows:

$$\text{score} = \sum_{i=1}^N \left( \frac{1}{\|\phi(z_{i,j}) - c_j\|_2} - \frac{1}{N_c - 1} \sum_{k \neq j} \frac{1}{\|\phi(z_{i,j}) - c_k\|_2} \right) \quad (10)$$

Where  $z_{i,j} = F(x_i; \theta)$  is the representation of the image  $x_i$  from max pooling results with the label  $j$ , and  $\phi(\cdot)$  indicates the t-SNE operation to two dimensions, and  $c_j$  is the mean of representations  $\{\phi(z_{i,j})\}_{y=j}$  in class  $j$ .  $N_c$  denotes the number of classes. *score* aims to measure the distributions of representations in the principle that samples belong to same class should be close to each other and samples from different classes should be far away from each other. Therefore, the measurement computes the distance of sample with the class center that it belongs to and other centers of inter-class. There is a higher score in minimizing the intra-class distance and maximizing the average distance of other classes. The result analysis will be discussed in Section 4.7.

## 4. Experiments

In this chapter, we introduce the settings of the experiments and evaluate the performance. Sections 4.1 and 4.2 separately elaborates on the datasets and parameter settings. Section 4.3

compares our methods with state-of-the-arts. Section 4.4 gives more detail of the training process and the advantages of our approach. Sections 4.5 and 4.6 discusses other properties of training results. Section 4.7 shows the data distribution results and presents a new perspective from the feature distributions to analyze training results. Section 4.8 compares the attention results of model after training.

### 4.1. Datasets

This study selects several renowned image datasets for comparison of computer vision task, and the details of the datasets are presented in Table 2. The fundamental MNIST consists of black and white images and 10 handwritten digits for classification. The color transformation cannot be implemented on it, so we only experiment on Pseudo-label algorithm for comparison. SVHN is also a digits' dataset but a color cropped numbers of street houses. CIFAR-10 and CIFAR-100 are standard color datasets of 10 classes and 100 classes respectively with small images of objects.

### 4.2. Implementation details

In this study, we perform experiments of image classification tasks on the renowned datasets mentioned in Section 4.1. Residual Attention Network (RAN) [45] is mainly used for fitting the Attention Consistency (AC) algorithm, and Wide ResNet (WRN) [55] is also used for comparison with FixMatch. According

**Algorithm 1:** The whole training process with One Supervised (OS)

---

**Input:** Training dataset  $\mathbf{D} = \{\mathcal{X}, \mathcal{U}\}, \theta$ .

Initialize model weights  $\theta$  and  $w$  with random parameters

Initialize list of last five losses  $\ell_5 := \{\}$

Initialize OS status  $Flag := False$

**for**  $e := 0$  **to** epochs **do**

**if**  $Flag$  **then**

$Flag := False$

    Predict pseudo labels of unlabeled data  $\hat{\mathcal{P}} := \arg \max(p_{model}(y|\mathcal{U}; \theta))$

    Compute unlabeled cross-entropy loss  $\mathcal{L}_u := H(\hat{\mathcal{P}}, p_{model}(y|\mathcal{U}; \theta))$

    Update model parameters  $\theta$  through  $\mathcal{L}_u$  minimization

**else**

    Initialize total label loss  $\mathbb{L}_e := 0$

**for**  $b := 0$  **to** batch size **do**

      Compute label cross-entropy loss  $\mathcal{L}_s := H(\mathcal{P}, p_{model}(y|\mathcal{X}; \theta))$

      Compute unlabeled loss  $\mathcal{L}_u$  according to algorithm

      Update model parameters  $\theta$  through  $\mathcal{L}_s$  and  $\mathcal{L}_u$  minimization

$\mathbb{L}_e := \mathbb{L}_e + \mathcal{L}_s$

**end for**

    Average the supervised loss in epoch  $e$   $\mathbb{L}_e = \frac{1}{|\mathcal{X}|} \mathbb{L}_e$

**if**  $|\ell_5| \geq 5$  **then**

      Compute average past 5 losses of labeled data  $\bar{\mathbb{L}}_n = \frac{1}{5} \sum_{k=1}^5 \mathbb{L}_{n-k}$

**if**  $\mathbb{L}_e > \bar{\mathbb{L}}_n$  **then**

$Flag := True$

        Clear the losses record of  $\ell_5$

**else**

        Add  $\mathbb{L}_e$  to the last five losses list  $\ell_5$

**if**  $|\ell_5| > 5$  **then**

          Delete the oldest record in  $\ell_5$

**else**

          Add  $\mathbb{L}_e$  to the last five losses list  $\ell_5$

**end**

**end**

**end**

**end**

---

**Table 2**  
Information of datasets.

Datasets	Description	Data shape	Training data
MNIST	Handwritten digits	$28 \times 28$	60,000
SVHN	Color images of street house numbers	$32 \times 32 \times 3$	73,257
CIFAR-10	10 classes of color images	$32 \times 32 \times 3$	50,000
CIFAR-100	100 classes of color images	$32 \times 32 \times 3$	50,000

to FixMatch, we use WRN-28-2 for 10 classes datasets, MNIST, SVHN, and CIFAR-10, and WRN-28-8 for CIFAR-100. The random horizontal flip, random crop, and normalization for standard deviation and mean are applied in all datasets, and the Rand Augment is implemented on color image datasets for strong augmentation, including SVNH, CIFAR-10, and CIFAR-100. The batch size of

experiments is 64. For CIFAR-100 dataset, the batch size ratio of unlabeled and labeled data is set to 7 according to the ablation study on FixMatch. Based on hardware considerations, the batch size in the experiment containing the unlabeled ratio is adjusted to 32 and  $32 \times 7$  for trainable. We use the stochastic gradient descent optimizer with an initial learning rate of 0.03, and the



**Table 3**  
Comparisons with other algorithms on different datasets.

Algorithms	SVHN		CIFAR-10		CIFAR-100	
	250 labels	1000 labels	250 labels	4000 labels	2500 labels	10000 labels
Pseudo-label	79.79	90.06	50.22	83.91	42.62	63.79
$\Pi$ model	81.06	92.46	45.74	85.99	42.75	62.12
MixMatch	96.02	96.5	88.95	93.58	40.06	71.69
UDA	94.31	97.54	91.18	95.12	66.87	75.5
FixMatch	97.52	97.72	94.93	95.74	71.71	77.4
Ours	95.94	95.9	89.85	95.63	65.32	70.51

scheduled step in short-term training, different from FixMatch linearly decreasing to 0.1. The weight decay is separately set to 0.003 for RAN and 0.005 for WRN. We evaluate the models by the exponential moving average of model's parameters with 0.999 decay. The confidence threshold  $\tau$  is set to 0.95. For the One Supervised (OS) algorithm, we compare the supervised loss with past 5 epochs results. And we will evaluate the accuracy of training results with baseline methods, Pseudo-label and FixMatch.

#### 4.3. Accuracy comparisons with baseline methods

Our method can be implemented on various computer vision problems, including general image classification FixMatch or hyperspectral application [56]. In this section, we compare our method to existing baseline methods that only use a single model and are considered in FixMatch, including Pseudo-label [7],  $\Pi$  model [24], Mean Teacher [28], MixMatch [21], and UDA [8]. These methods' results are based on FixMatch. We observe that the whole training progress of FixMatch is up to about 10212 epochs for 46000 unlabeled data in the provided code, because of the batch size ratio 64 and  $64 \times 7$  and total  $2^{20}$  steps. The comparisons of long-term training process on color image datasets with different numbers of labeled data are shown in Table 3, and the best result of each dataset will be highlighted in bold. For training long progress, we implement our algorithm AC and Oson FixMatch with WRN to train 5000 epochs on CIFAR-10 and CIFAR-100 and 250 epochs on SVHN. In SVHN and 4000 labeled data of CIFAR-10 datasets, we can achieve the results close to FixMatch. Our results are competitive to the MixMatch performance.

#### 4.4. Ablation study

The efficiency of training is also an important issue, so this section compares our methods with the implemented algorithm, even under more severe situations and shows our competitiveness on the performance.

##### 4.4.1. Comprehensive comparisons with pseudo-label

We implement our algorithm, AC and OS, on Pseudo-label and compare the performance results shown in Table 4. Due to black and white dataset that cannot transform by strongly color augmentations, the comparisons on MNIST are only implemented on Pseudo-label, and the basic MNIST dataset, which is too easy for supervised learning even with only 300 labeled data, experimented in 40 labeled data. As shown in Table 4, it is seen that we improved the original Pseudo-label on all datasets. In MNIST, although the result of RAN model is even worse than the origin, the performance of RAN combined with AC and OS algorithm can be better than 88.94%. In CIFAR-10 in 4000 labels and CIFAR-100 in 10000 labels datasets, our methods used not only WRN but also RAN model structures, achieving higher accuracy than original Pseudo-label, and the results do not use the complex RAN, even better in this simple case.

##### 4.4.2. Comprehensive comparisons with FixMatch

Our algorithm is mainly applied on FixMatch and compared with it on color image datasets, including SVHN, CIFAR-10, and CIFAR-100. The task of comparisons is to shorten training process to about one fortieth of the original from 10212 to 250 epoch. We observed that the FixMatch training takes too long such that  $2^{10}$  steps fixed in one epoch and the 7 unlabeled data ratio which has batch size of unlabeled data is 7 times larger than labeled data with batch size 64, which results show that the 1 epoch in FixMatch is equivalent to about 10 epochs (9.973 epochs) in general definition. The compared results of one-fortieth process are shown in Table 5, highlighting the best performances in bold.

In the easier task SVHN, there are no obvious gaps for our algorithm and original FixMatch method, but FixMatch with OS and AC is still better than the result of only FixMatch in less labeled data (250 labels). In CIFAR-100 dataset, the original FixMatch cannot effectively achieve its performance in long-term training, but ours can improve the original accuracies from 48.94% to 53.13% in 2500 labels and from 67.17% to 68.81% in 10000 labels. The performance in CIFAR-10 is obviously improved by our algorithm, and the result 95.39% in 4000 labeled data is even similar to the complete training process result 95.74% of FixMatch. The result of the comparison can be seen that our method is effective in improving model, identifying the features and performance on training process by attention and effective use of unlabeled data.

#### 4.5. Epoch analysis

The comparisons of accuracy changes in each epoch, aims to analyze the efficiency of training process for each method. For more detailed comparison, we attempted optimizing the original FixMatch by adjusting the hyper parameters. Finally, the accuracy of original FixMatch in 250 epochs increases to 93.06% through modifying the cosine learning rate decay scheduler to linear decay to 0.001 and canceling the unlabeled batch size ratio. We compare the test data results of original FixMatch in 5000 epochs and optimized original FixMatch in 250 epochs with our method, and the test accuracy curves are drawn in Fig. 7. The curve can be seen that the original FixMatch rises slowly but steadily and finally, it achieves 95.81% in our experiment in epoch 4427. However, the cost seems to be too high in comparison with the effective short term training process. In our method, we achieved 95% accuracy in epoch 250, contrast to original FixMatch reaching in epoch 2811 and only 90.02% in epoch 292.

We further analyze the time spent on training and show the comparison in Table 6. The WRN is more streamlined and powerful so that it can spend the least time to achieve acceptable results. Due to more complex model structure, we can find that not only our method, but also original in RAN model spends more time than the same process in WRN, but ours reach better result. Especially, it took five times the time of training for FixMatch to achieve the result we arrived in short-term training. The lengthy training time of the original FixMatch is not practical in real application.

**Table 4**  
Ablation on Pseudo-label.

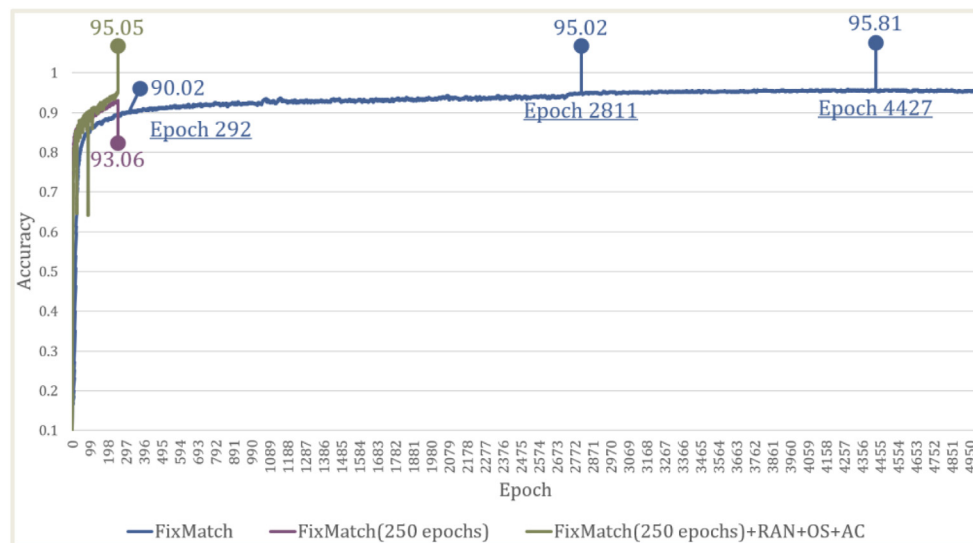
Algorithms	MNIST 40 labels	CIFAR-10 4000 labels	CIFAR-100 10000 labels
Pseudo-label	88.94	83.91	63.79
Pseudo-label + RAN	83.09	84.44	61.16
Pseudo-label + Ours	88.73	<b>86.74</b>	<b>64.95</b>
Pseudo-label + RAN + Ours	<b>89.1</b>	86.13	64.91

**Table 5**  
Ablation on FixMatch with only one-fortieth of training.

Algorithms	SVHN		CIFAR-10		CIFAR-100	
	250 labels	1000 labels	250 labels	4000 labels	2500 labels	10000 labels
FixMatch	95.25	95.91	77.78	91.01	48.94	68.76
FixMatch + RAN	94.7	<b>96.47</b>	79.72	93.18	43.72	61.67
FixMatch + Ours	<b>95.94</b>	95.66	83.97	93.25	<b>53.65</b>	<b>70.92</b>
FixMatch + RAN + Ours	94.93	95.9	<b>85.43</b>	<b>95.39</b>	50.86	68.81

**Table 6**  
Time comparison on training process.

Method	Accuracy	Cost time (sec)
FixMatch (5,000 epochs)	95.81	658,596
FixMatch (250 epochs)	93.06	35,151
FixMatch + RAN (250 epochs)	93.18	92,735
FixMatch + Ours (250 epochs)	93.25	39,011
FixMatch + RAN + Ours (250 epochs)	95.39	116,192

**Fig. 7.** Comparison of accuracy curve for CIFAR-10 with 4000 labels.

#### 4.6. Data usage analysis

In this section, the efficiency of unlabeled data usage will be discussed. We explore the number of the reliable data, the predicted probability which should be above the threshold, in the end of training. The ablation of reliable data results in different datasets is shown in Table 7, and we compare FixMatch with our method in 250 epochs. Analyzing with Table 5 can be found that more labeled data can lead to training more reliable data and higher accuracy. The results, where OS and AC algorithm is implemented in WRN, have more unlabeled data included for training. The part of the results using RAN model achieves more reliable data. However, the usage of these reliable data does not reflect on the performance of classification. Instead, it is possible that our method finds more precise reliable data so that the accuracy is improved.





#### 4.7. Feature analysis

Reliable data are important in SSL because of the instability of unlabeled data. We are concerned about the feature distributions to understand the training results. Therefore, we analyze the accuracy and feature results of train and test data separately, and the results of accuracy and metrics, and the experiments on short-term process are conducted with five different random seeds and obtained the average results to ensure stability. All training data feature comparisons on FixMatch with our methods in CIFAR-10 4000 labels are shown in Table 8, and the feature distribution is the t-SNE result from the model's max pool output. First, we can analyze the long-term and short-term performance of original FixMatch in Table 8 (a). In human observation, FixMatch in the long-term (5000 epochs) training process learns well and can clearly distinguish the features of different classes for training data, and FixMatch in limited training (250 epochs) has

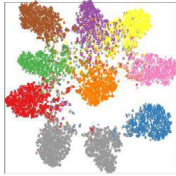



**Table 7**  
Ablation of unlabeled data usage in different datasets.

Algorithms	SVHN		CIFAR-10		CIFAR-100	
	250 labels	1000 labels	250 labels	4000 labels	2500 labels	10000 labels
FixMatch	64,187	64,331	38,190	40,093	16,320	16,470
FixMatch + RAN	44,791	52,625	36,244	42,417	14,467	24,203
FixMatch + Ours	<b>64,608</b>	<b>65,059</b>	38,118	41,421	<b>22,156</b>	<b>28,751</b>
FixMatch + RAN + Ours	47,790	58,257	<b>44,289</b>	<b>43,857</b>	18,178	22,337

**Table 8**  
Comparison analysis of train data in CIFAR-10 4000 labels.

Method	(a)		(b)	
	FixMatch (250 epochs)	FixMatch (5000 epochs)	FixMatch (RAN) (250 epochs)	FixMatch (RAN) (Ours) (250 epochs)
Train Accuracy	91.6%	96.09%	94.79%	95.09%
Metrics Result	17.4412	18.7452	19.0625	19.2396
Feature Distribution				

**Table 9**  
Comparison analysis of test data in CIFAR-10 4000 labels.

Method	(a)		(b)	
	FixMatch (250 epochs)	FixMatch (5000 epochs)	FixMatch (RAN) (250 epochs)	FixMatch (RAN) (Ours) (250 epochs)
Test Accuracy	90.06%	94.59%	93.51%	94.45%
Metrics Result	18.511	21.2555	22.1394	22.1234
Feature Distribution				

the worst distribution. Through the figures of feature distribution, our method shows good performance in feature classification of training data. The metrics proposed in Section 3.5 added to analysis, proved the rationality according to positive correlation with training accuracy. Observing Table 8(b), FixMatch using RAN cannot obviously improve the accuracy and feature distinguish ability, and our method performs well in short-term training.

Then, we also analyze the test data results and in Table 9, it can be found that test distribution result of ours has the best score (21.5764) through our algorithm. The performance of original FixMatch in Table 9 (a) which has the best classification accuracy is slightly worse in feature distribution, the reason is probably that it depends too much on consistency regularization. Ours in Table 9 (b) also distinguish features well on test cases, and it shows that our algorithm effectively improves the model's ability in feature classification.

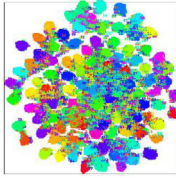
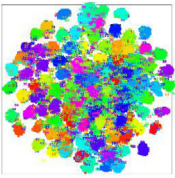
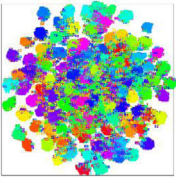
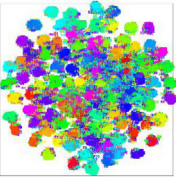
For CIFAR-100 datasets, we also compute the metrics and analyze with classification task results in 10000 labels for comparison, which are shown in Tables 10 and 11. Due to 100 classes in the datasets, the figures cannot clearly distinguish the class distributions in two dimensions so that the metrics can be used to assist the researchers in distribution analysis. In results of train data, the metrics values match the trend of classification results, and our method has trained better than the original cases, which is also found in metrics. The comparison results show that the

FixMatch using WRN in Table 10(a) and Table 11(a) are better than that using RAN in Table 10(b) and Table 11(b) for CINFAR-100 dataset, and our methods can bring the improvements on accuracy and feature distribution regardless of the model.

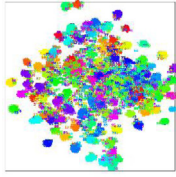
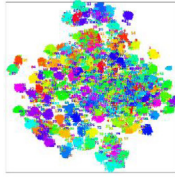
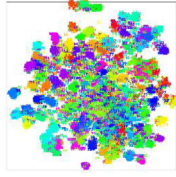

#### 4.8. Attention analysis

The AC algorithm in our method aims to utilize the attention to improve model performance, and this section discusses the effect of attention in data after training. To visualize the attention results, we show the results of attention heatmaps extracted from the model. The heatmaps are first performed Layer Normalization for normalizing the distribution of attention, then unsampled to the original data size through bilinear interpolation, and finally summed on the classes channels and normalized again for weighting the original image. We show the attentions in various labels of CIFAR-10 test data that model has not seen with different methods in Table 12. Row (a) to row (d) are separately the image of label horse, ship, airplane, and cat. The original FixMatch in long-term training can clearly detect the objects after 5000 epochs. The original in short-term training is obviously observed that it cannot precisely distinguish the region of object, such as Table 12(a) horse image, it cannot pay attention to the hind leg, and Table 12(c)(d) also cannot focus on the complete characteristics of cat face and airplane tail. The FixMatch which


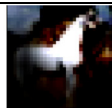
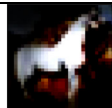


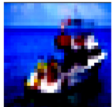
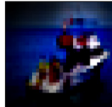
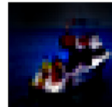
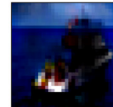
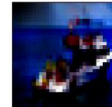






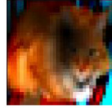



**Table 10**  
Comparison analysis of CIFAR-100 10000 labels and 250 epochs.

Method	(a)		(b)	
	FixMatch (250 epochs)	FixMatch (Ours) (250 epochs)	FixMatch (RAN) (250 epochs)	FixMatch (RAN) (Ours) (250 epochs)
Train Accuracy	75.42%	76.08%	71.55%	73.63%
Train Metrics Result	40.0331	42.6113	33.9662	39.367
Feature Distribution				

**Table 11**  
Comparison analysis of CIFAR-100 10000 labels and 250 epochs.

Method	(a)		(b)	
	FixMatch (250 epochs)	FixMatch (Ours) (250 epochs)	FixMatch (RAN) (250 epochs)	FixMatch (RAN) (Ours) (250 epochs)
Test Accuracy	68.15%	69.57%	63.58%	66.71%
Metrics Result	32.521	33.7232	28.48.9	33.4466
Feature Distribution				

**Table 12**  
Attention comparisons of test data in CIFAR-10 4000 labels.

Label	Original Image	FixMatch (250 epochs)	FixMatch (5000 epochs)	FixMatch (RAN) (250 epochs)	FixMatch (RAN) (Ours) (250 epochs)
(a) horse					
(b) ship					
(c) airplane					
(d) cat					

used RAN does not work well on features identity even if its accuracy is higher than original FixMatch used WRN, but the implements that our OS and AC added into cannot only achieve better performance, but also clearly focus on the objects.

We also do ablation study for the attention results in our method and show it in Table 13. It can be found that the results using the AC algorithm can perform well on attention to improve the classification. The result of only AC on RAN cannot precisely distinguish the shape of object and the background which is not required, such as the region under the horse and cat body. FixMatch with OS and AC in original WRN is very specific on


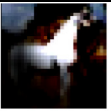



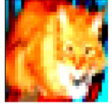



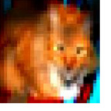
region of target center to ignore the small but important details, such as horse head and cat limbs. The combination method of our algorithms and RAN structure focuses on the shape of object, such as the horse's curve, while retaining the key detail.

## 5. Conclusions and future work

In recent years, many powerful semi-supervised learning methods were proposed to solve the problem of high labeling cost. However, their consistency regularization with strong augmentations leads to longer training time to adapt to complex



**Table 13**  
Attention ablations of test data in CIFAR-10 4000 labels and 250 epochs.

Label	Original Image	Original FixMatch	FixMatch Ours	FixMatch (RAN) AC	FixMatch (RAN) Ours
(a) horse					
(b) cat					

transformations, and the entropy minimization loses much information of unlabeled data for ensuring the reliable data. In this paper, Attention Consistency (AC) and One Supervised (OS) are proposed to alleviate these problems, and we use AC to improve the model's ability to feature identity and fully use all unlabeled data at the right time through the OS algorithm. The comprehensive comparisons show that we have competition on the performance in limited training process. For understanding the model's learning status, a new metric is proposed to measure the feature distributions, and it shows that the metrics results match the accuracy of classification.

The experiment results show our advantages on efficiency of model training, but we are still unable to surpass the state-of-the-arts in long-term training process. In the future, more issue will face the cost problems including not only labor cost but also training efficiency. Our future works is that we will try to reinforce the connection between general consistency regularization and attention and automatically update OS status for more stable and lasting results.

### CRediT authorship contribution statement

**Jui-Hung Chang:** Writing, Research, Experiment content planning. **Hsiu-Chen Weng:** Writing – original draft, Programs, Experiments, Statistical analyses.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Acknowledgments

We thank the anonymous reviewers for their constructive comments. This research work was supported in part by the Ministry of Science and Technology of Taiwan, ROC. (MOST 110-2221-E-006-181- and MOST 110-2634-F-006 -022 -).

### References

- [1] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: *ICLR*, 2015.
- [2] W. Rawat, Z. Wang, Deep convolutional neural networks for image classification: A comprehensive review, *Neural Comput.* 29 (9) (2017) 2352–2449, [http://dx.doi.org/10.1162/neco\\_a\\_00990](http://dx.doi.org/10.1162/neco_a_00990).
- [3] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, J. Chanussot, Graph convolutional networks for hyperspectral image classification, *IEEE Trans. Geosci. Remote Sens.* 59 (7) (2021) 5966–5978, <http://dx.doi.org/10.1109/TGRS.2020.3015157>.
- [4] D. Hong, et al., SpectralFormer: Rethinking hyperspectral image classification with transformers, *IEEE Trans. Geosci. Remote Sens.* 60 (2022) 5518615, <http://dx.doi.org/10.1109/TGRS.2021.3130716>, 1–15.
- [5] D. Hong, et al., More diverse means better: Multimodal deep learning meets remote-sensing imagery classification, *IEEE Trans. Geosci. Remote Sens.* 59 (5) (2021) 4340–4354, <http://dx.doi.org/10.1109/TGRS.2020.3016820>.
- [6] O. Chapelle, B. Scholkopf, A. Zien, Semi-supervised learning, *IEEE Trans. Neural Netw.* 20 (3) (2009) 542, <http://dx.doi.org/10.1109/TNN.2009.2015974>.
- [7] D.H. Lee, Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks, in: *Workshop on Challenges in Representation Learning*, Vol. 3, no. 2, *ICML*, 2013, p. 896.
- [8] Q. Xie, Z. Dai, E. Hovy, M.T. Luong, Q.V. Le, Unsupervised data augmentation for consistency training, 2019, arXiv preprint [arXiv:1904.12848](https://arxiv.org/abs/1904.12848).
- [9] K. Sohn, D. Berthelot, C.L. Li, Z. Zhang, N. Carlini, E.D. Cubuk, A. Kurakin, H. Zhang, C. Raffel, Fixmatch: Simplifying semi-supervised learning with consistency and confidence, *Adv. Neural Inf. Process. Syst.* (2020).
- [10] Q. Xie, M.T. Luong, E. Hovy, Q.V. Le, Self-training with noisy student improves imagenet classification, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10687–10698, <http://dx.doi.org/10.1109/CVPR42600.2020.01070>.
- [11] C. Rosenberg, M. Hebert, H. Schneiderman, Semi-supervised self-training of object detection models, in: *Proceedings of the Seventh IEEE Workshops on Application of Computer Vision*, 2005, <http://dx.doi.org/10.1109/ACVMOT.2005.107>, 2005.
- [12] K. Bennett, A. Demiriz, Semi-supervised support vector machines, *Adv. Neural Inf. Process. Syst.* (1999) 368–374.
- [13] Y. Chong, Y. Ding, Q. Yan, S. Pan, Graph-based semi-supervised learning: A review, *Neurocomputing* 408 (2020) 216–230, <http://dx.doi.org/10.1016/j.neucom.2019.12.130>.
- [14] O. Chapelle, A. Zien, Semi-supervised classification by low density separation, in: *International Workshop on Artificial Intelligence and Statistics*, 2005, pp. 57–64.
- [15] D. Yarowsky, Unsupervised word sense disambiguation rivaling supervised methods, in: *33rd Annual Meeting of the Association for Computational Linguistics*, 1995, pp. 189–196, <http://dx.doi.org/10.3115/981658.981684>.
- [16] B. Zoph, G. Ghiasi, T.Y. Lin, Y. Cui, H. Liu, E.D. Cubuk, Q.V. Le, Rethinking pre-training and self-training, *Adv. Neural Inf. Process. Syst.* (2020).
- [17] I. Triguero, S. García, F. Herrera, Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study, *Knowl. Inf. Syst.* 42 (2) (2015) 245–284, <http://dx.doi.org/10.1007/s10115-013-0706-y>.
- [18] J.E. Van Engelen, H.H. Hoos, A survey on semi-supervised learning, *Mach. Learn.* 109 (2) (2020) 373–440, <http://dx.doi.org/10.1007/s10994-019-05855-6>.
- [19] Y. Grandvalet, Y. Bengio, Semi-supervised learning by entropy minimization, *CAP* 367 (2005) 281–296.
- [20] A. Oliver, A. Odena, C. Raffel, E.D. Cubuk, I.J. Goodfellow, Realistic evaluation of deep semi-supervised learning algorithms, *Adv. Neural Inf. Process. Syst.* (2018) 3235–3246.
- [21] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, C. Raffel, Mixmatch: A holistic approach to semi-supervised learning, *Adv. Neural Inf. Process. Syst.* (2019) 5050–5060.
- [22] H. Zhang, M. Cisse, Y.N. Dauphin, D. Lopez-Paz, Mixup: Beyond empirical risk minimization, 2017, arXiv preprint [arXiv:1710.09412](https://arxiv.org/abs/1710.09412).
- [23] T. Miyato, S.I. Maeda, M. Koyama, S. Ishii, Virtual adversarial training: a regularization method for supervised and semi-supervised learning, *IEEE Trans. Pattern Anal. Mach. Intell.* 41 (8) (2018) 1979–1993, <http://dx.doi.org/10.1109/TPAMI.2018.2858821>.
- [24] S. Laine, T. Aila, Temporal ensembling for semi-supervised learning, in: *International Conference on Learning Representations*, 2017.
- [25] M. Sajjadi, M. Javanmardi, T. Tasdizen, Regularization with stochastic transformations and perturbations for deep semi-supervised learning, *Adv. Neural Inf. Process. Syst.* 29 (2016) 1163–1171.



- [26] J. Enguehard, P. O'Halloran, A. Gholipour, Semi-supervised learning with deep embedded clustering for image classification and segmentation, *IEEE Access* 7 (2019) 11093–11104, <http://dx.doi.org/10.1109/ACCESS.2019.2891970>.
- [27] C.W. Kuo, C.Y. Ma, J.B. Huang, Z. Kira, Featmatch: Feature-based augmentation for semi-supervised learning, in: *European Conference on Computer Vision*, Springer, Cham, 2020, pp. 479–495, [http://dx.doi.org/10.1007/978-3-030-58523-5\\_28](http://dx.doi.org/10.1007/978-3-030-58523-5_28).
- [28] A. Tarvainen, H. Valpola, Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, *Adv. Neural Inf. Process. Syst.* (2017).
- [29] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, 2015, arXiv preprint [arXiv:1503.02531](https://arxiv.org/abs/1503.02531).
- [30] V. Verma, A. Lamb, C. Beckham, A. Najafi, A. Courville, I. Mitliagkas, Y. Bengio, Manifold mixup: learning better representations by interpolating hidden states, 2018.
- [31] E.D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, Q.V. Le, Autoaugment: Learning augmentation policies from data, in: *The IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, Jun 2019, <http://dx.doi.org/10.1109/CVPR.2019.00020>.
- [32] D. Berthelot, N. Carlini, E.D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, C. Raffel, Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring, in: *Eighth International Conference on Learning Representations*, 2019.
- [33] E.D. Cubuk, B. Zoph, J. Shlens, Q.V. Le, Randaugment: Practical automated data augmentation with a reduced search space, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 702–703, <http://dx.doi.org/10.1109/CVPRW50498.2020.00359>.
- [34] Y. Bengio, J. Louradour, R. Collobert, J. Weston, Curriculum learning, in: *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009, pp. 41–48, <http://dx.doi.org/10.1145/1553374.1553380>.
- [35] G. Hachohen, D. Weinshall, On the power of curriculum learning in training deep networks, in: *International Conference on Machine Learning*, 2019, pp. 2535–2544.
- [36] S. Song, H. Yu, Z. Miao, D. Guo, W. Ke, C. Ma, S. Wang, An easy-to-hard learning strategy for within-image co-saliency detection, *Neurocomputing* 358 (2019) 166–176, <http://dx.doi.org/10.1016/j.neucom.2019.05.009>.
- [37] M. Kumar, M. BKumar, B. Packer, D. Koller, Self-paced learning for latent variable models, *Adv. Neural Inf. Process. Syst.* 23 (2010) 1189–1197.
- [38] L. Jiang, D. Meng, S.I. Yu, Z. Lan, S. Shan, A. Hauptmann, Self-paced learning with diversity, *Adv. Neural Inf. Process. Syst.* 27 (2014) 2078–2086.
- [39] H. Fan, L. Zheng, C. Yan, Y. Yang, Unsupervised person re-identification: Clustering and fine-tuning, *ACM Trans. Multimedia Comput., Commun. Appl. (TOMM)* 14 (4) (2018) 1–18, <http://dx.doi.org/10.1145/3243316>.
- [40] P. Cascante-Bonilla, F. Tan, Y. Qi, V. Ordóñez, Curriculum labeling: Re-visiting pseudo-labeling for semi-supervised learning, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35, no. 8, 2021, pp. 6912–6920.
- [41] R. Hataya, H. Nakayama, Unifying semi-supervised and robust learning by mixup, in: *ICLR 2019 Workshop on Learning from Unlabeled Data*, 2019.
- [42] W. Ocasio, Attention to attention, *Organ. Sci.* 22 (5) (2011) 1286–1296.
- [43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, *Adv. Neural Inf. Process. Syst.* (2017) 5998–6008.
- [44] R. Hu, J. Andreas, T. Darrell, K. Saenko, Explainable neural computation via stack neural module networks, in: *Proceedings of the European Conference on Computer Vision, ECCV*, 2018, pp. 53–69, [http://dx.doi.org/10.1007/978-3-030-01234-2\\_4](http://dx.doi.org/10.1007/978-3-030-01234-2_4).
- [45] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, X. Tang, Residual attention network for image classification, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3156–3164, <http://dx.doi.org/10.1109/CVPR.2017.683>.
- [46] J. JHu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2018, pp. 7121–7141.
- [47] H. Zheng, J. Fu, T. Mei, J. Luo, Learning multi-attention convolutional neural network for fine-grained image recognition, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5209–5217, <http://dx.doi.org/10.1109/ICCV.2017.557>.
- [48] H. Guo, K. Zheng, X. Fan, H. Yu, S. Wang, Visual attention consistency under image transforms for multi-label image classification, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 729–739, <http://dx.doi.org/10.1109/CVPR.2019.00082>.
- [49] J. Chen, M. Yang, J. Ling, Attention-based label consistency for semi-supervised deep learning based image classification, *Neurocomputing* 453 (2021) 731–741, <http://dx.doi.org/10.1016/j.neucom.2020.06.133>.
- [50] Y. Yang, N. Zhu, Y. Wu, J. Cao, D. Zhan, H. Xiong, A semi-supervised attention model for identifying authentic sneakers, *Big Data Min. Anal.* 3 (1) (2019) 29–40, <http://dx.doi.org/10.26599/BDMA.2019.9020017>.
- [51] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2921–2929, <http://dx.doi.org/10.1109/CVPR.2016.319>.
- [52] J.L. Ba, J.R. Kiros, G.E. Hinton, Layer normalization, 2016, arXiv preprint [arXiv:1607.06450](https://arxiv.org/abs/1607.06450).
- [53] Z. Zhang, M. Zhao, T.W. Chow, Graph based constrained semi-supervised learning framework via label propagation over adaptive neighborhood, *IEEE Trans. Knowl. Data Eng.* 27 (9) (2013) 2362–2376, <http://dx.doi.org/10.1109/TKDE.2013.182>.
- [54] S. Manikandan, Measures of central tendency: Median and mode, *J. Pharmacol. Pharmacother.* 2 (3) (2011) 214–215, <http://dx.doi.org/10.4103/0976-500X.83300>.
- [55] S. Zagoruyko, N. Komodakis, Wide residual networks, in: *Proceedings of the British Machine Vision Conference, BMVC*, 2016, <http://dx.doi.org/10.5244/C.30.87>.
- [56] D. Hong, N. Yokoya, J. Chanussot, X.X. Zhu, An augmented linear mixing model to address spectral variability for hyperspectral unmixing, *IEEE Trans. Image Process.* 28 (4) (2019) 1923–1938, <http://dx.doi.org/10.1109/TIP.2018.2878958>.