Sigma J Eng Nat Sci, Vol. 42, No. 3, pp. 862–874, June, 2024

Sigma Journal of Engineering and Natural Sciences
Web page info: https://sigma.yildiz.edu.tr
DOI: 10.14744/sigma.2024.00000

Research Article

Iterative ensemble pseudo-labeling for convolutional neural networks

Serdar YILDIZ[1,2,*], Mehmet Fatih AMASYALI[1]

[1]*Department of Computer Engineering, Yildiz Technical University, Istanbul, 34220, Türkiye*
[2]*BİLGEM, TÜBİTAK, Kocaeli, 41470, Türkiye*

ARTICLE INFO

Article history
Received: 18 July 2022
Revised: 03 December 2022
Accepted: 04 January 2023

Keywords:
Ensemble Learning; Pseudo Labeling; Semi-Supervised Learning; STL-10

ABSTRACT

As is well known, the quantity of labeled samples determines the success of a convolutional neural network (CNN). However, creating the labeled dataset is a difficult and time-consuming process. In contrast, unlabeled data is cheap and easy to access. Semi-supervised methods incorporate unlabeled data into the training process, which allows the model to learn from unlabeled data as well. We propose a semi-supervised method based on the ensemble approach and the pseudo-labeling method. By balancing the unlabeled dataset with the labeled dataset during training, both the decision diversity between base-learner models and the individual success of base-learner models are high in our proposed training strategy. We show that using multiple CNN models can result in both higher success and a more robust model than training a single CNN model. For inference, we propose using both stacking and voting methodologies. We have shown that the most successful algorithm for the stacking approach is the Support Vector Machine (SVM). In experiments, we use the STL-10 dataset to evaluate models, and we increased accuracy by 15.9% over training using only labeled data. Since we propose a training method based on cross-entropy loss, it can be implemented combined with state-of-the-art algorithms.

Cite this article as: Yıldız S, Amasyalı MF. Iterative ensemble pseudo-labeling for convolutional neural networks. Sigma J Eng Nat Sci 2024;42(3):862–874.

INTRODUCTION

With the developments in artificial intelligence, many applications in the field of image and signal processing have reached the human level, especially studies using convolutional neural networks [1-3]. The publication of large data sets has a great impact on the emergence of these successful studies. Computer vision research has gained momentum, particularly after the public release of the ImageNet dataset. The ImageNet [4] dataset, which contains millions of image samples for 1000 classes, has been the basis for many studies, and it has been shown that very successful models can be produced with convolutional neural networks in the case of a very large number of samples.

Today, deep learning methods are applied in a wide variety of fields, such as medical image analysis [5], object classification [1], word recognition [6], etc. Almost all state-of-the-art models in deep learning applications are trained using massive amounts of data. However, in many

***Corresponding author.**
*E-mail address: serdar.yildiz@std.yildiz.edu.tr

This paper was recommended for publication in revised form by Editor in Chief Ahmet Selim Dalkilic

Published by Yıldız Technical University Press, İstanbul, Turkey

real-world applications, annotated datasets containing millions of images are not possible.

Generating annotated datasets is generally expensive, time-consuming, and hard. In particular, it requires data labeling by experts in special fields such as medical image analysis. For these reasons, sufficient data cannot always be obtained in task-specific studies [7]. However, unlabeled data can be obtained in large numbers and much more easily than labeled data. Due to the small amount of labeled data and the large amount of unlabeled data, unlabeled data was also tried to be included in the training process.

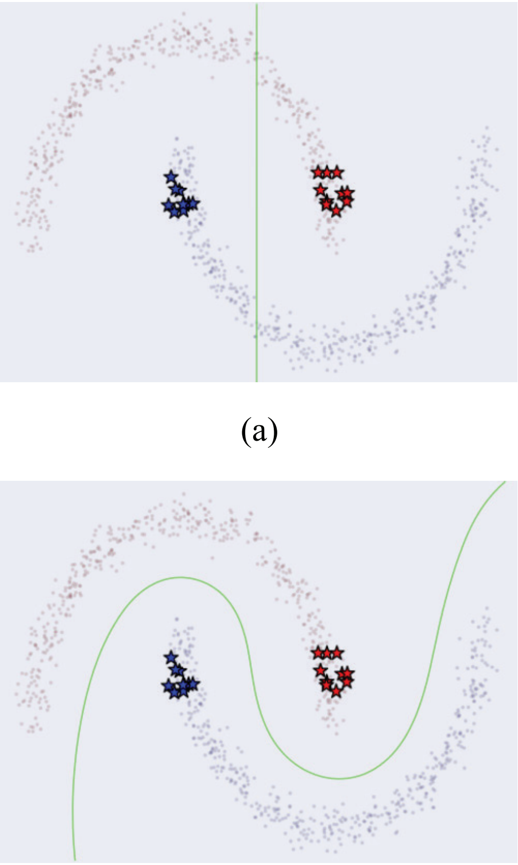
In supervised learning, the model is trained using only labeled data. The limitation of supervised learning approaches is that they can only learn from labeled datasets. If the labeled dataset is very small and does not accurately represent the data distribution, generalization performance may be poor.

Semi-supervised learning (SSL) techniques that include both unlabeled and labeled samples in the training process have been developed. By including large numbers of unlabeled data in the training, the generalization performance of the model can be increased with a small number of labeled data. Since the labeled dataset can only represent a part of the real-world data distribution, the generalization performance of the model trained with only the labeled dataset is low. As shown in Figure 1, including the unlabeled dataset in the training leads to more accurate generalization.

There are some assumptions to include unlabeled data in the training process [8, 9]. The smoothness assumption implies that samples that are close to each other in the feature space should be in the same class. The low-density assumption accepts that the decision line should be located where the sample density is low, and the manifold assumption assumes that samples belonging to the same class with a high-dimensional input space will converge on each other when converted to a lower-dimensional subspace.

State-of-the-art techniques in several challenges are based on the ensemble learning approach [6, 10-12]. Ensemble learning is a machine learning approach that aims to achieve better results by combining multiple base learner models that attempt to solve the same problem[13]. By combining the base learner decisions, the ensemble model will be more successful if the base learner decisions are more diverse and accurate [14]. In fact, accuracy and diversity are two opposing qualities of base learners. In general, when accuracy is high, diversity is low, and vice versa when diversity is high. The ensemble model performs the best when the two factors are optimum. For this reason, while training base learner models, diversity should be considered as well as accuracy.

The pseudo-labeling strategy [15] is used in the majority of current semi-supervised learning research. Interpolation Consistency Training [16] (ICT), MixMatch [17], ReMixMatch [18], DivideMix [19], FixMatch [20], etc. studies are generally aimed at applying the regularization methods they recommend for samples with a confidence score above 95%. However, the success of the methods decreases if successful pseudo-labels are not produced. As a result, the focus of this study was on generating more successful pseudo-labels.


(a)

(b)

Figure 1. Supervised and Semi-Supervised Learning Decision Boundary Comparison (a) Supervised Learning, (b) Semi-Supervised Learning.

Our motivation is to apply the ensemble learning methodology in the pseudo-labeling approach, which is the most fundamental method in semi-supervised learning research, to add more accurate pseudo-labels in training in each iteration and to improve model performance by utilizing less labeled data. We compared the proposed algorithm to research in the literature that apply more than one model and produce more successful pseudo-labels for a more fair comparison.

In this study, we propose iterative ensemble pseudo-labeling for convolutional neural networks that produce more successful pseudo-labels using the ensemble method. We created an approach that makes CNN models more tolerant of changes and more successful on the test set using a minimal number of labeled data. This technique, which applies with all image dataset, reduces the requirement for a labeled dataset in order to produce a robust convolutional neural network model.

We demonstrated that the base-learner models' decisions should be combined with voting to produce pseudo-labels at the end of each iteration, and that the base-learner models should be combined with the SVM algorithm at the end of training. We have shown that the proposed method is robust to operations and noises such as horizontal flip, vertical flip rotation, random crop, erasing, and Gaussian blur. To our knowledge, it is the first study to consider decision diversity among base-learner models when training base learner models in studies combining ensemble convolutional neural networks and semi-supervised learning.

The remainder of this paper is organized as follows. We detailed the method we proposed in the Methodology section, the dataset and model we used in the Experimental Setup section, the results we obtained, and their comparison with previous research in the Results section. Finally, the ablations are in the Ablation Study section.

Related Work

Pseudo-Labeling [15] is a fundamental semi-supervised learning strategy that employs generated labels obtained from model predictions. During the fine-tuning phase, the network is trained with both labeled and unlabeled data at the same time, and unlabeled samples are labeled with the class having the highest predicted probability. Training the model with a pseudo-labeled sample pushes the decision boundary to low-density areas. It forces the model to produce predictions with high confidence. This is referred to as Entropy Regularization [21].

The fundamental problem of pseudo-labeling techniques is that they are prone to confirmation bias by generating high confidence scores for samples that the model mistakenly predicts [22]. Training the model using incorrectly generated pseudo-labels reduces its success.

The Noisy Student [23] method is an improvement of the pseudo-label method. Initially, student model training is performed using labeled data. Pseudo labels are created for unlabeled samples using the student model, and a model that is the same or larger than the student model is trained with labeled and unlabeled samples. It is tried to increase the success of the pseudo-label by applying augmentation to the dataset and applying the dropout and stochastic depth regularization methods for the student model.

π Model [24] method is based on consistency regularization. An input image is given to the model by applying two random augmentations, and it is trained to produce the same result on these two inputs. To reduce the prediction difference between augmented images, a loss function such as L1 or L2 loss is used. Input images can be labeled or unlabeled. In this way, the robustness of the model is improved.

As in π Model, there is a training strategy in the Mean Teacher [25] method to give the same result to different augmented variants of the same input. The point where the Mean Teacher method differs is that instead of getting results from the same model, it makes predictions using the weights obtained using the exponential moving average of the same model. With exponential moving average, the model produces more accurate pseudo labels, which makes it more successful.

In the Co-training [26] method, two models are trained for labeled samples using two different distributions of a dataset. By combining the decisions of the models, pseudo-labels are produced for the unlabeled samples, and pseudo-labeled data is included in the training set. The training process continues iteratively until there is no change. Since the initial models are trained with labeled datasets from different distributions, decision diversity between models will be high, and they will produce more accurate pseudo-labels.

Tri-training [27] is a method that involves training three models at the same time. Three classification models collaborate to generate pseudo-labels for unlabeled data. For each model, the training process begins with a subset of the labeled dataset. After the three models have been trained, pseudo-label generation is performed on the unlabeled samples. If two models determine the same decision for the unlabeled sample, the sample is added to the training set of the other model with the pseudo-label. The training process continues until the decision of the model remains the same.

Tri-training with disagreement [28] is an improved version of the tri-training method. The models' training is maintained by only using samples whose decisions differ from those of the other models. As a result of having a more dominating training set for weak points, the model is expected to provide more successful results on the test set. At the same time, since it is more data-efficient than tri-training, the training time is reduced.

Ghosh et al. [29] begin the training process for the three models with subsets of the labeled dataset, as in the tri-training technique. It then combines the three models' decisions to generate pseudo-labels for all unlabeled samples. By combining the samples selected from the unlabeled dataset and the labeled dataset, the training of the models is continued by performing subsampling for this new dataset. Since different subsets of the labeled dataset are used in each iteration, the models eventually see all of the labeled samples. As a result, the decision diversity between the models is not considered.

The pseudo-labeling strategy is used in the large majority of semi-supervised learning research. The research's principal purpose is to develop more successful pseudo-labels. The reason is that a more successful model can be obtained by using more successfully generated pseudo-labels. In Tri-training, Tri-training with disagreement, and Ghosh et al. [29] studies, it has been suggested to use ensemble methods in order to produce more successful pseudo-labels. While Tri-training and Tri-training with disagreement provide decision diversity by using random subsets and random initialization, decision diversity comes from using different models in Ghosh et al. [29] However, when the training is finished in these studies, the diversity among the decisions of the base learner models diminishes due to the iterative process. In our study, based on the balance of the labeled and unlabeled dataset, we have created a training model that will both preserve the diversity of decisions between base learner models and increase the success of the ensemble model. In this way, we achieved more successful results than other studies.

MATERIALS AND METHODS

In semi-supervised learning, training data consists of two parts. Assume $D = \{D_L , D_U\}$ is the total of training data and $D = \{\boldsymbol{X}, \boldsymbol{Y}\} = \{(x_i , y_i)\}_{i=1}^{N_L}$ denotes labeled training dataset with inputs $x_i \epsilon \mathbb{R}^D$, where x_i, D, N_L represent an input sample, input dimension, and the size of the labeled training dataset, respectively, and y_i symbolizes the target class label $y_i \epsilon \{1,2,...,\boldsymbol{C}\}$ where $\boldsymbol{C}$ is the number of classes. Similarly, the unlabeled training dataset is denoted by $D_U = \{\widehat{\boldsymbol{X}}\} = \{(\widehat{x_\imath})\}_{i=1}^{N_U}$.

Pseudo labeling is the technique of predicting labels for unlabeled data using a model that is trained with labeled data. Pseudo-labeling can be represented as:

$$P(D_U) = \{\widehat{\boldsymbol{X}}, \widehat{\boldsymbol{Y}}\} = \{(\widehat{x_\imath}, \widehat{y_\imath})\}_{i=1}^{N_U} \qquad (1)$$

where $\hat{y}_i$ is pseudo-label of a sample of $\hat{x}_i, \hat{y}_i \epsilon \{1, 2, \ldots, C\}$, and $P(.)$ is proposed pseudo-labeling technique. To clarify, the basic goal of pseudo-labeling is to $D_U = \{\hat{X}\}$ to convert to $\widehat{D}_U = \{\hat{X}, \hat{Y}\}$, and $\hat{Y}$ must accurately match the ground-truth label for model performance. As a solution, we propose applying iterative and ensemble technique to produce pseudo-labels for the unlabeled dataset.

In the iterative approach, instead of creating a pseudo-label for the entire unlabeled dataset, the number of K samples with the lowest entropy [30] value of the probability distributions obtained by combining the decisions of the base learner models was selected. The number of K is increased by K in the next iteration to obtain a larger number of training samples. At each new iteration, a pseudo-label is obtained for the entire unlabeled dataset. This ensures that a few incorrectly labeled selected samples are more likely to be correctly labeled. In this way, the training set includes more reliable pseudo-labels and agreed-upon samples.

In the first iteration, a randomly selected subset of the labeled dataset is used to train each base learner. The selected labeled dataset is in the training set of the base learner until the last iteration. Let D_{L_i} is selected subset of labeled dataset,

$$\left\{D_{L_1} \cup D_{L_2} \cup \ldots \cup D_{L_{N_{BL}}}\right\} = D_L \qquad (2)$$

where $|D_{L_i}| < N_L$, $i \epsilon \{1, \ldots, N_{BL}\}$, and N_{RL} is number of base-learner. In order to increase the diversity between the decisions of the base learner models, the training is performed with the labeled subset of training data that did not fully overlap with each other. After each base-learner has been trained with the provided training dataset, the pseudo-label generation process is performed.

To produce pseudo-labels by combining the decisions of the base learner models, the probability vectors are produced by base-learner models for each unlabeled sample. For an unlabeled example, probability vector of the ensemble model:

$$\widehat{y_\imath} = softmax(\sum_{l=1}^{N_{BL}} \mathfrak{h}_l(\widehat{x_\imath})) \qquad (3)$$

where $\mathfrak{h}_l$ denotes the probability vector of a base learner. At each iteration, the probability vectors of the ensemble model are obtained for the entire unlabeled dataset and the entropy value of each probability vector is calculated. Entropy value of sample $\hat{x}_i$:

$$H(\widehat{y_\imath}) = -\sum_{c=0}^{C} \widehat{y_\imath}^{c} \log{(\widehat{y_\imath}^{c})} \qquad (4)$$

where $\widehat{y_\imath}^{c}$ is probability of class c. The N_{select} unlabeled samples with the lowest entropy value are selected and a sample of $\hat{x}_i$ labelled with $\hat{\boldsymbol{y}}_{\boldsymbol{i}} = argmax(\widehat{y_\imath})$. As a result, the new iteration has more reliable pseudo-labels and agreed-upon samples thanks to ensemble model.

To train a base-learner with labeled and unlabeled dataset, the training dataset contains the same number of labeled and unlabeled data. Assume the training dataset for base-learner i in iteration J is $D_{T_i}^{j} = \{\mathfrak{D}_{L_i}^{j}, D_U^{j}\}$ with seleted unlabeled dataset D_U^{j}, where $\mathfrak{D}_{L_i}^{j}$ is augmented labeled dataset which contains duplicated samples of D_{L_i} to obtain $|\mathfrak{D}_{L_i}^{j}| \approx |D_U^{j}|$. Since the same unlabeled dataset is used for all base-learners in iteration J, randomly selected labeled dataset samples (D_{L_i}) are duplicated, and added to the training dataset in order to ensure decision diversity among the base-learner models. All train steps are given in Algorithm 1.

In terms of optimization, it is costly to randomly initialize the base learner model while progressing to the new iteration. However, if the models are continued with the

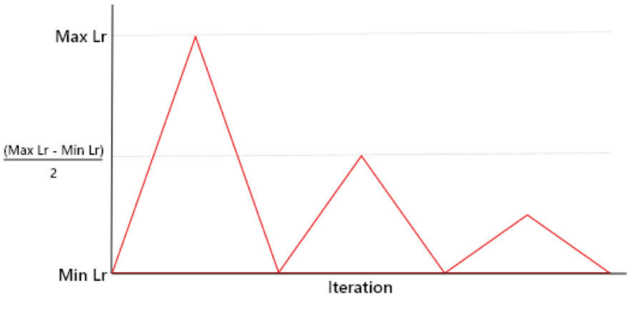


Figure 2. Learning Rate Scheduler.

Algorithm 1. Base Learner Models Training

Input: D_L as the labeled training dataset, D_U as unlabeled dataset, N_{iter} as number of iteration, N_{select} how many samples will be selected in the first iteration

Output: Arr_{BL} as an array of base learner models

```
$Arr_{BL} \leftarrow [\,]$
$for\ i \leftarrow 1\ to\ N_{BL}\ do$
    $D_{L_i} \leftarrow getSubset(D_L)$
    $BL_i \leftarrow getInitModel()$
    $BL_i \leftarrow trainBaseLearner(BL_i, D_{L_i})$
    $Arr_{BL} \leftarrow Arr_{BL} \cup \{BL_i\}$
$end\ for$
$for\ j \leftarrow 1\ to\ N_{iter}\ do$
    $D_U^j \leftarrow getPseudoLabel(Arr_{BL}, D_U, N_{select})$
    $for\ i \leftarrow 1\ to\ N_{BL}\ do$
        $\mathfrak{D}_{L_i}^j \leftarrow bootstrapping(D_{L_i}, N_{select})$
        $D_{T_i}^j \leftarrow \{\mathfrak{D}_{L_i}^j, D_U^j\}$
        $BL_i \leftarrow trainBaseLearner(BL_i, D_{T_i}^j)$
        $Arr_{BL}[i] \leftarrow BL_i$
    $end\ for$
    $N_{select} \leftarrow N_{select}\ x\ (j+1)$
$end\ for$
```

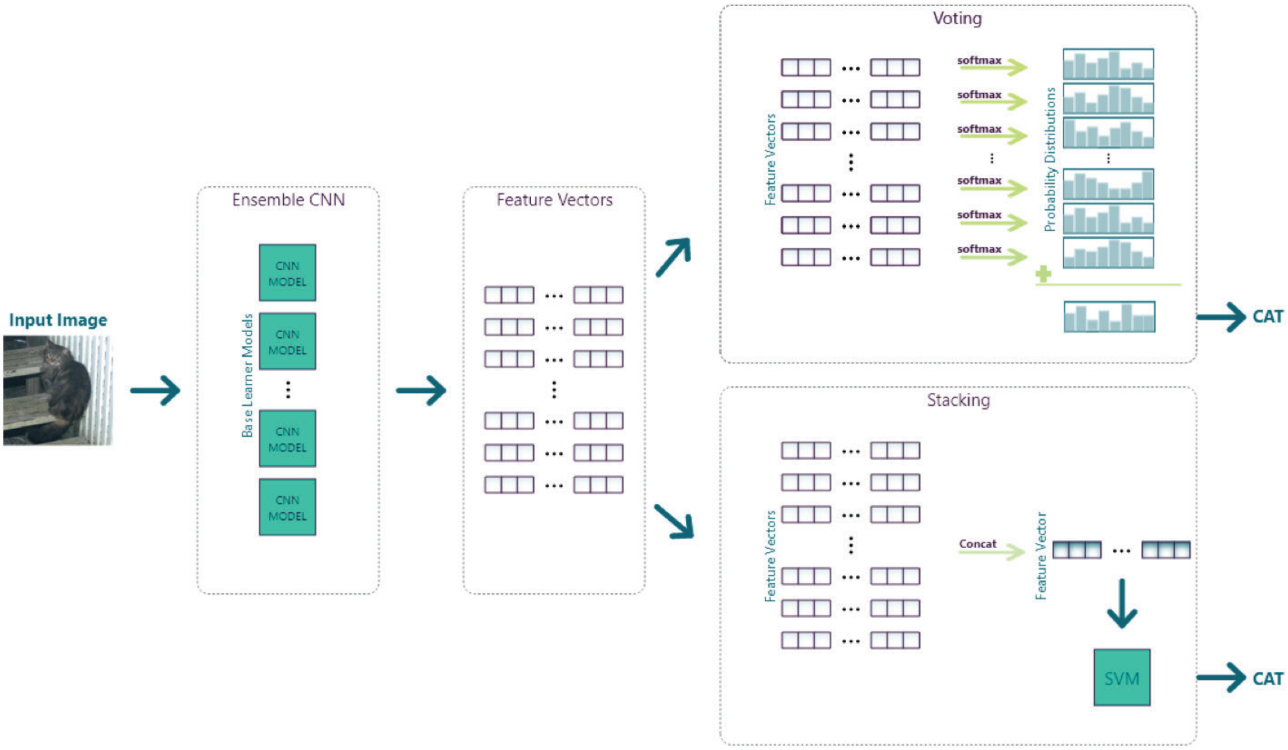


Figure 3. Method Inference Structure.

parameters that generated the pseudo-labels, it is hard to move to a more optimized point in the parameter space. To solve this dilemma, we propose to use the triangular learning rate scheduler approach [31], which reduces the maximum learning rate at the end of two epochs to the midpoint between the minimum learning rate and the previous maximum learning rate (Figure 2).

In inference, we apply two approaches shown in Figure 3. First, feature vectors are obtained from all base learner models for the input image. If it is desired to infer with the voting approach, the softmax function is applied to the feature vectors. The obtained probability distributions are summed and the most dominant class is returned as the estimation. With the addition of probability vectors, base learner models that produce high-score predictions become dominant. If it is desired to infer with the stacking approach, the feature vectors concatenation operation is applied. The resulting new feature vector is given to a trained machine learning method and expected to produce a class prediction. In this study, the SVM algorithm was chosen as the machine learning model.

As in the pseudo-label producing process, the probability vectors obtained from the base learner are summed in the voting process, and the class with the highest probability in the obtained probability vector is returned as the result. In the stacking process, a feature vector is obtained by concatenating the feature vectors obtained from the base learner models without applying the softmax function. Let is K_i the feature vector which is output of *i*. base learner model,

$$\widehat{y_\iota} = SVM([\boldsymbol{K_1}, \boldsymbol{K_2}, \ldots, \boldsymbol{K}\_N_{BL}]) \tag{5}$$

where $\widehat{y_\iota}$ is the probability vector of ensemble model. With the help of the Support Vector Machine(SVM) [32] algorithm, the concatenated feature vector is returned to the probability vector. The SVM model is trained with the entire set of labeled training data D_L. In this way, the SVM model learns the response produced by the base learner models for samples that have never been used in the training of the base learner models, and the model becomes more robust.

As the number of iterations increases, the diversity among the base learner decisions decreases, so the probability vectors obtained in voting are similar to each other, but even though the decisions in stacking are similar, the success of the ensemble is preserved at lower decision diversity as different feature vectors are produced. This situation is examined in detail in Ablation Section.

EXPERIMENTAL SETUP

Dataset

We preferred to verify the proposed method on the STL-10 dataset [33]. Because, in addition tobeing one of the most popular datasets in semi-supervised learning studies [18, 17, 20, 16], it is one of the few datasets created specifically for this field.

The STL-10 dataset contains 96x96 RGB images for 10 classes (airplane, bird, car, cat, deer, dog, horse, monkey, ship, truck). It has 500 training samples and 800 test samples for each class, with a total of 5,000 training and 8,000 test samples. Apart from this, there are 100,000 unlabeled samples. The unlabeled dataset has a different domain than the labeled dataset and class distribution of the unlabeled dataset is unbalanced. For this reason, it is more difficult to achieve high successes compared to other datasets.

Base Learner Model Training Setup

Dense Convolutional Network [34] (DenseNet) model was used as the base learner model. DenseNet has several advantages over other models. DenseNet minimizes the gradient vanishing problem, enables feature reuse, and significantly reduces the number of parameters [34]. The obtained feature vectors are generated by concatenating all layers before the current connection, rather than by summing in DenseNet architecture.

In this study, 10 DenseNet121 base learner models with 7.6M parameters were used. In this way, the study's scalability was ensured. TheAdamW [35] optimization technique is utilized, with the beta values are 0.9 and 0.999 and the weight decay parameter of 0.01. The maximum and minimum learning rates are 0.1 and 0.0001, respectively. In the first iteration, each base learner is trained 40 epochs, in all other iterations 20 epochs are trained.

Machine Learning Models Training Setup

The outputs of the base learner model are transformed into a feature vector in the stacking approach to obtain class predictions from the machine learning model. The machine learning models used here are Decision tree [36], random forest [37], K-NN [38] Multi-layer perceptron and SVM algorithms.

In the Decision tree algorithm, Gini is used as the measurement metric. Each node is set to split into two nodes and no pruning is applied. In the random forest algorithm, the number of trees in the forest is set to 100. The Gini metric is also used in this algorithm. In the K-NN algorithm, the decision is made according to the distance to the 5 nearest neighbors. In the multi-layer perceptron algorithm, the hidden layer size is set to 100. ReLU was used as the activation function. In SVM algorithm, the radial basis function kernel is used and the regularization parameter is set to 1.

Related Works Training Setup

For a fair comparison, the same configuration in our proposed study was used for the tri-training and tri-training with disagreement methods. Densenet121 was used as base learner models. The AdamW optimizer is used, with the beta values are 0.9 and 0.999 and the weight decay parameter of 0.01. The triangular learning rate scheduler

Table 1. STL-10: Model Accuracy (%)

	Single Model	Base-Learner best (mean±std)	SVM	Voting
0. iter	66.725	66.25(65.26±0.73)	70.300	70.650
1. iter	73.237	73.29(72.69±0.31)	75.487	75.150
2. iter	76.775	76.35(75.89±0.22)	78.238	77.913
3. iter	77.375	77.66(77.37±0.22)	79.150	79.138
4. iter	78.687	78.79(78.38±0.25)	80.025	79.588
5. iter	79.100	79.21(78.95±0.15)	80.625	79.888
6. iter	79.825	79.64(79.28±0.20)	80.850	**80.050**
7. iter	79.950	79.79(79.48±0.17)	**81.163**	79.963

approach was used. For Ghosh et al. [29], the results they reported in their study were used.

RESULTS AND DISCUSSION

In this section, the results of the proposed method for the classification task are examined. In terms of robustness and accuracy, the study was assessed from two perspectives. As can be seen in Table 1, a maximum success of 81.163% was achieved.

To compare the effectiveness of the suggested method, a single CNN model was trained with all training data and the same training structure. The only difference between the single model and the base learner models is that the entire labeled dataset is used in the training of the single model, while subsets of the labeled dataset are used in the base learners. The remaining hyperparameters, such as iteration number, epoch number, optimizer, and learning rate, are all set to the same value. In this way, it was compared with the approach of training a model using the entire training dataset, which is widely used in the literature.

It can be shown that the success of single model is higher than the success of mean base learner in Figure 4. The reason for this can be explained by the higher number of labeled samples used in the single model. As the number of iterations increases, the information exchange between base learner models increases (a pseudo-label is produced at the end of each iteration by the joint decision of the base learners), and the mean base learner success and the success of the single model converge. At the same time, as it progresses towards the final iterations, the success of the voting ensemble model converges on the mean success of the base learner and the success of the single model. This is due to the fact that as the number of iterations increases, the diversity of decisions among the base learner models decreases. This brings the success of the voting ensemble model closer to the success of the base learner models. However, in the stacking method using the SVM algorithm, the success of the ensemble continues to increase at the same rate since the stacking method produces a feature vector from the base learner models rather than a probability vector. Although each base learner model makes similar decisions for similar samples, since the labeled dataset used in their training does not overlap, they make these

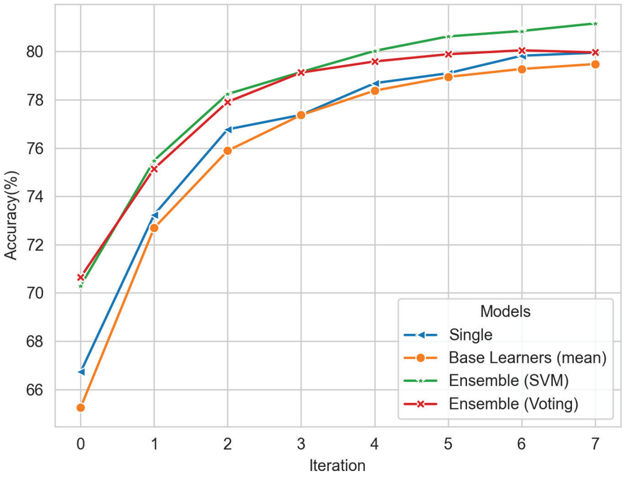


Figure 4. Model Accuracy Comparison.

Table 2. Metrics on STL-10 Dataset

	Accuracy (%)	Precision(%)	Recall(%)	F1(%)
Tri-training	76.788	77.603	76.787	76.811
Tri-training with disagreement	77.150	77.317	77.150	77.164
Ghosh et al. [29], 2021	78.88	-	-	-
Ours	**81.163**	**81.739**	**81.162**	**81.077**

Table 3. Machine Learning Models Metrics on STL-10 Dataset

	Accuracy	Precision	Recall	F1
Decision Tree	79.038	80.127	79.038	79.100
Multi-layer Perceptron	80.450	80.971	80.450	80.374
K-Neighbors	80.612	81.562	80.612	80.528
Random Forest	80.950	81.278	80.950	80.937
SVM	***81.163***	***81.739***	***81.162***	***81.077***

decisions using different spatial features. This situation is explained in detail in Ablation Section.

Table 2 compares the accuracy of the method we suggest to other methodologies. It shows that the method we proposed is ~2% more accurate than the closest one. At the same time, the most successful results were obtained for precision, recall, and F1.

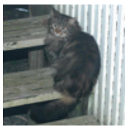
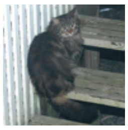
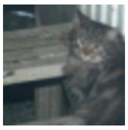
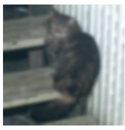
It has been proposed in the Tri-training, Tritraining with disagreement, and Ghosh et al. [29] research to use ensemble approaches to develop more successful pseudo-labels. While tri-training and tritraining with disagreement provide decision diversity via the use of random subsets and random initialization, Ghosh et al. [29] provide decision diversity through the use of different models. Based on the balance of the labeled and unlabeled datasets, we designed a training model that would retain the diversity of decisions amongst base learner models while increasing the performance of the ensemble model. We were able to get better findings than other studies in this manner.

When cumulatively labeled data is combined with unlabeled data, the amount of unlabeled data in the training set increases and becomes dominant. Since the same unlabeled dataset is used in the training set of each base learner, the decision diversity of base learners will decrease. As a result, the success of the ensemble will also decrease. To avoid this, in our study, the labeled sub-dataset selected for each base learner is balanced with the added unlabeled dataset by bootstrapping. The balancing method ensures that decision diversity among base learner models remains constant and that the ensemble model's success keeps rising.

Another achievement of this study is that a more robust model was obtained compared to the single model training approach commonly used in the literature. In Table 4, the outputs of the base learner models and the ensemble model outputs are given for a sample input. Each base learner model can be considered a single model in this case.

While all base learner models and ensemble models produce accurate results in the initial test picture, it can be observed that the base learner decisions have changed as a result of the image manipulations, but the ensemble model continues to produce correct results. Especially as a result of vertical flip operation, although 4 out of 10 base learners

Table 4. Base Learners and Ensemble Model Outputs

	Original	Horizontal Flip	Vertical Flip	Rotation	Random Crop	Erasing	Gaussian Blur
Base Learners							
1	cat	cat	bird	horse	monkey	bird	Cat
2	cat	monkey	bird	bird	monkey	cat	Monkey
3	cat	cat	bird	dog	dog	cat	Monkey
4	cat	monkey	cat	dog	cat	bird	Cat
5	cat	monkey	bird	cat	cat	bird	Cat
6	cat	cat	monkey	cat	cat	cat	Cat
7	cat	monkey	cat	cat	monkey	cat	Cat
8	cat	cat	monkey	ship	monkey	cat	Cat
9	cat	cat	monkey	cat	cat	bird	Monkey
10	cat	monkey	cat	cat	cat	cat	Monkey
Ensemble (Voting)	cat	cat	cat	cat	cat	cat	Cat

give the result ’bird’, the ensemble model gives the correct result. This is because, instead of each model casting a single vote, the voting procedure assigns a vote to each class in proportion to the probability value. In this way, even if the class with the highest probability value for the decision of 4 base learners is wrong, when all base learner decisions are combined, the minority but high probability value base learner decisions dominate and the ensemble model is correctly classified.

ABLATION STUDY

In this section, we will provide a better understanding of the fact that our proposed method is more successful than the other approachs. We show why stacking is more effective than voting and why the proposed method is more robust.

Stacking Vs Voting

The class distributions of the concatenated feature vector for stacking are seen to be more discrete sets in the feature space than the feature vector obtained by summing probability vectors for voting in Figure 5. Giving the softmax function the feature vectors generated by each base learner model causes information loss in voting. Since the feature vector obtained from each base learner model is processed within itself, the softmax function cannot capture a relationship between base learner models. However, when we concatenate the feature vectors instead of giving them to the softmax function, the feature vectors given in Figure 5a are obtained. The samples represented by the concatenated feature vector are more separable from each other by using decision lines in the feature space.

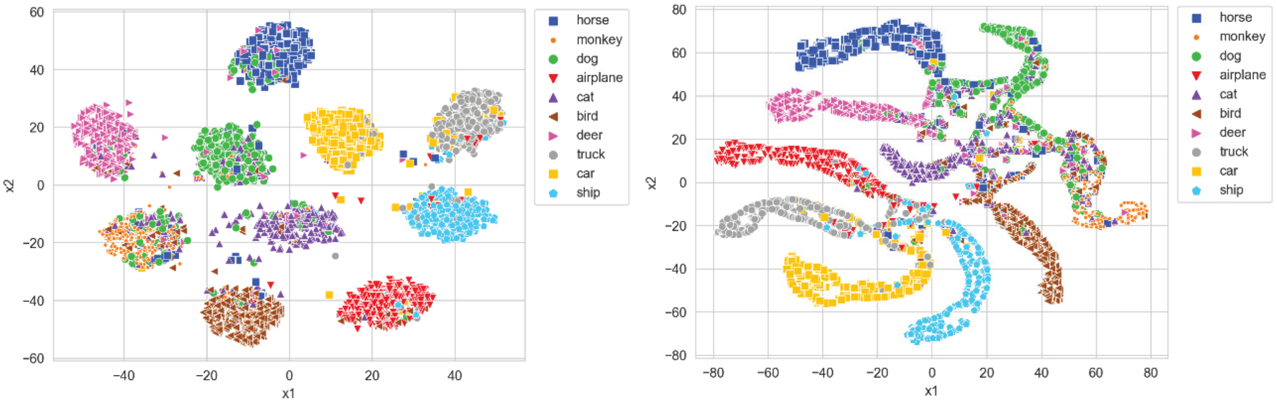
The concatenation approach’s challenge is to find the class probability value corresponding to the obtained feature vector values. To transform a feature vector to a probability vector, the SVM algorithm is used. In this way, base learner model decisions are combined without losing information. The SVM algorithm, which is trained using the entire training dataset, learns the results produced by the base learner models for samples from different distributions and produces ensemble result by weighing the results of base learner models. As shown in Table 3, Decision tree [36], random forest [37], K-NN [38], Multi-layer perceptron algorithms have also been tried, but the SVM algorithm is the most successful among them.

Effects of First Iteration Result

One of the biggest factors affecting the success of the ensemble is the accuracy obtained in the 0th iteration. The 0th iteration is when base learner models are trained using only a subset of the labeled dataset. With the base learner models trained in this iteration, the first pseudo-labels were obtained. The success of the first iteration’s pseudo labels (whether correctly labeled or not) is directly related to the success of the 0th iteration. Because a more successful model generates more successful pseudo labels, the model with the most success in the 0th iteration is also the model with the most success at the end of the 1st iteration. The same can be said for the 2nd and 3rd, the 3rd and 4th, and so on. For this reason, the success of the 0th iteration is directly related to the success of the model obtained in the last iteration. This is clearly seen in Figure 6. The only distinction between training the two models is that one of the first epoch numbers is 40 and the other is 20. While the success difference between the two models in the 0th iteration starts as 2%, the difference continues as the iteration progresses.

Diversity of Decisions

Convolution is not an invariance process [40, 41]. As a result of operations such as rotation, translation, illumination, it produces different results for the same input image. However, the model is expected to produce the same result for a transformed input sample. To accomplish this, various



(a)

(b)

Figure 5. Ensemble Model Feature Visualization Using t-SNE [39] for Test Samples (a) Concatenation of Feature Vectors (Stacking), (b) Sum of Probability Distributions (Voting).

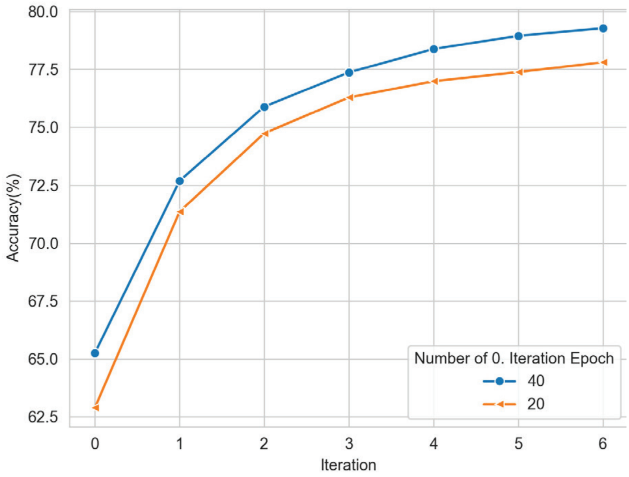


Figure 6. Mean Accuracy of Base Learner Models Trained with Different Epoch Numbers in Iteration 0.

augment states of the training samples are added to the training set during model training. The model gereralizes for the domain of the dataset in which it was trained. But in the real world, the model cannot generalize to all cases as the dataset contains subdomains. However, in our proposed method, since each base learner model is trained for different subsets of the labeled training set, they can generalize more effectively the domain of the subset on which they have been trained. Base learner models trained with train datasets that are not exactly the same (containing different train samples as well as different augmented versions) produce different feature vectors for the same test sample. Base learner models that make decisions based on different features in the same test sample; for example, one cannot decide (producing very close probabilities or producing false results with high confidence) for the processed samples such as rotation, but another can. The accuracy and robustness of the ensemble model is increased by choosing the base learner model that produces the correct result for the test sample given in the joint decision-making phase. In this way, base learner models trained with different train samples produce different feature vectors for the same input image and thus behave more robustly to operations such as rotation and translation.

Using GradCAM [42] the feature vector of the same layer (denseblock4) was drawn on the input image for 10 base learners. As can be seen in Figure 7, each base learner generates different feature vectors for the same input.

In the proposed training algorithm, pseudo-labeled samples are generated at the end of each iteration as a result of the joint decision of all base learner models, and each of the base learner model is then trained with the combination of labeled samples and pseudo-labeled samples. Pseudo-labeled samples generated by joint decision and included in training improve the success of base learner models while reducing decision variety. Kappa [43] calculates the degree of agreement between two decisions. A high kappa value indicates that the two decisions are similar, whereas a low number indicates that the two decisions are unlike each other. The kappa value between the two baselearner models and the average errors are represented as points on the kappa-error graph. The Kappa-Error graph is given in Figure 8. The average

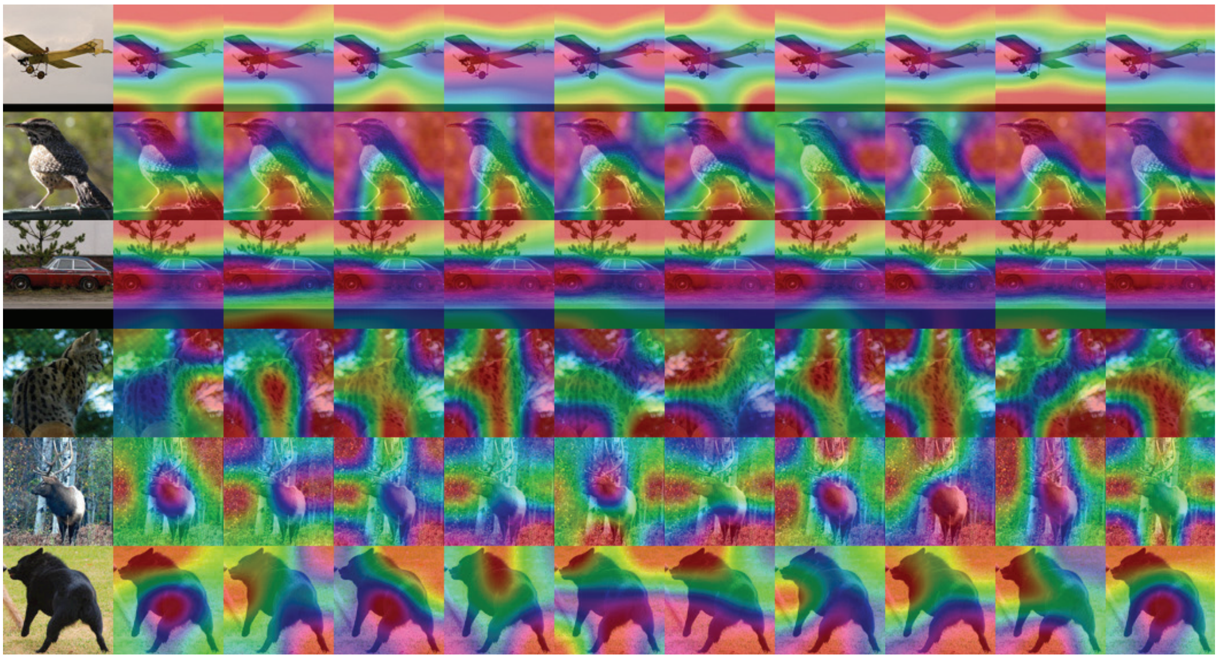


Figure 7. GradCAM : Input Images Feature Maps.

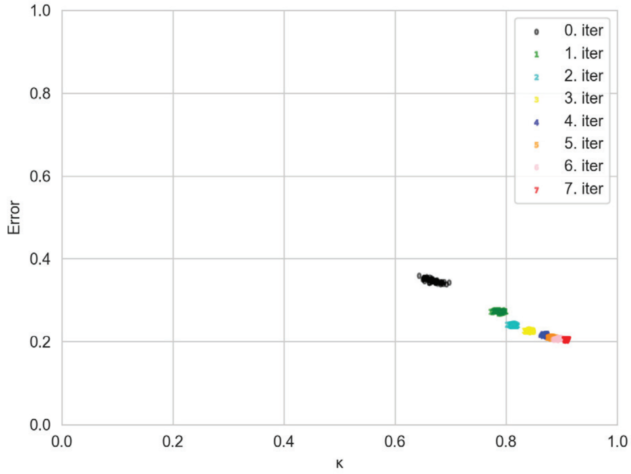


Figure 8. Kappa-Error Graph.

kappa value among base learners increases as the number of iterations increases, as shown in the Kappa-Error graph. As decision variety declines, the decisions produced for the test samples begin to overlap, and because all base learner models make similar decisions, combining the decisions does not significantly boost the success of the ensemble model over the performance of the base learner model.

Limitations

In the ensemble learning approach, each base learner model must be trained independently from other models. For this reason, in each iteration, training is carried out as much as the number of base learner models. The base learner model multiplied by the quantity of iterations represents the total performance of training operations.

In our experiments, we progressed 8 iterations with the 10 base learner model. This means that 80 training were carried out. It can be said that it requires 80 times more processing power than a single model training. However, this value depends on the number of base learners and the number of iterations. It is seen in Table 1 that more successful results can be obtained from the single model with fewer iterations.

Future Works

In this study, it has been investigated that more successful pseudo-labels can be obtained by using ensemble CNN and it is compared with the studies that only examine this. Based on this study, consistency regularization techniques such as Mixup and ensemble CNN approach should be combined in future studies. In this way, successful pseudo-labels, which are the prerequisites of consistency regularization techniques, are produced and it is predicted that higher success will be achieved.

A solution to the imbalance class distribution problem can be produced by training base learner models with a class specificity. For this reason, experiments with imbalance datasets have been added to future studies.

CONCLUSION

In this study, we propose iterative ensemble pseudo-labeling for convolutional neural networks to improve model success and robustness. We show that an ensemble model, which is created by training more than one CNN model, can produce a more successful and robust model. The central idea is to improve the ensemble model's robustness and success with fewer data by producing more accurate pseudo labels while preserving the diversity of decisions among base learner models. As the number of iterations increases, the diversity of decisions decreases as the information exchange between base learner models increases. However, since each base learner model is trained with different subsets of the labeled dataset, they generate different feature vectors, ensuring the ensemble's success with the stacking method. In experiments, we evaluate models using the STL-10 dataset and achieve an accuracy of 81.16 percent.

AUTHORSHIP CONTRIBUTIONS

Authors equally contributed to this work.

DATA AVAILABILITY STATEMENT

The authors confirm that the data that supports the findings of this study are available within the article. Raw data that support the finding of this study are available from the corresponding author, upon reasonable request.

CONFLICT OF INTEREST

The author declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

ETHICS

There are no ethical issues with the publication of this manuscript.

REFERENCES

[1] Liu Z, Mao H, Wu CY, Feichtenhofer C, Darrell T, Xie S. A convnet for the 2020s. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2022. p. 11976–11986. [CrossRef]

[2] Brock A, De S, Smith SL, Simonyan K. High-performance large-scale image recognition without normalization. In: International Conference on Machine Learning, PMLR; 2021. p. 1059–1071.

[3] Zhang H, Wu C, Zhang Z, Zhu Y, Lin H, Zhang Z, et al. Resnest: Split-attention networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2022. p. 2736–2746. [CrossRef]

[4] Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition; 2009. p. 248–255. [CrossRef]

[5] Memis A, Varlı S, Bilgili F. Semantic segmentation of the multiform proximal femur and femoral head bones with the deep convolutional neural networks in low quality mri sections acquired in different mri protocols. Comput Med Imaging Graph 2020;81:101715. [CrossRef]

[6] Fang S, Xie H, Wang Y, Mao Z, Zhang Y. Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021. p. 7098–7107. [CrossRef]

[7] Yıldız S, Aydemir O, Memis A, Varlı S. A turnaround control system to automatically detect and monitor the time stamps of ground service actions in airports: A deep learning and computer vision based approach. Eng Appl Artif Intell 2022;114:105032. [CrossRef]

[8] Van Engelen JE, Hoos HH. A survey on semisupervised learning. Mach Learn 2020;109:373–440. [CrossRef]

[9] Yang X, Song Z, King I, Xu Z. A survey on deep semi-supervised learning. IEEE Trans Knowl Data Eng 2022;35:8934–8954. [CrossRef]

[10] Chen T, Guestrin C. Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining; 2016. p. 785–794. [CrossRef]

[11] Amasyali MF, Ersoy OK. Classifier ensembles with the extended space forest. IEEE Trans Knowl Data Eng 2014;26:549–562. [CrossRef]

[12] Yildiz S, Aydemır O, Yilmaz İ, Say A, Varli S. Customer churn analysis. In: 2020 28th Signal Processing and Communications Applications Conference (SIU), IEEE; 2020. p. 1–4. [CrossRef]

[13] Polikar R. Ensemble learning. In: Ensemble machine learning. Springer; 2012. p. 1–34. [CrossRef]

[14] Dong X, Yu Z, Cao W, Shi Y, Ma Q. A survey on ensemble learning. Front Comput Sci 2020;14:241–258. [CrossRef]

[15] Lee DH. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on challenges in representation learning, ICML; 2013. p. 896.

[16] Verma V, Kawaguchi K, Lamb A, Kannala J, Solin A, Bengio Y, et al. Interpolation consistency training for semi-supervised learning. Neural Netw 2022;145:90–106. [CrossRef]

[17] Berthelot D, Carlini N, Goodfellow I, Papernot N, Oliver A, Raffel CA. Mixmatch: A holistic approach to semi-supervised learning. Adv Neural Inf Process Syst 2019;32.

[18] Berthelot D, Carlini N, Cubuk ED, Kurakin A, Sohn K, Zhang H, et al. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. arXiv Prepr arXiv:1911.09785; 2019.

[19] Li J, Socher R, Hoi SC. Dividemix: Learning with noisy labels as semi-supervised learning. In: International Conference on Learning Representations; 2019.

[20] Sohn K, Berthelot D, Carlini N, Zhang Z, Zhang H, Raffel CA, et al. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. Adv Neural Inf Process Syst 2020;33:596–608.

[21] Grandvalet Y, Bengio Y. Semi-supervised learning by entropy minimization. In: Saul L, Weiss Y, Bottou L, editors. Advances in Neural Information Processing Systems. MIT Press; 2004.

[22] Arazo E, Ortego D, Albert P, O'Connor NE, McGuinness K. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In: 2020 International Joint Conference on Neural Networks (IJCNN), IEEE; 2020. p. 1–8. [CrossRef]

[23] Xie Q, Luong MT, Hovy E, Le QV. Self-training with noisy student improves imagenet classification. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020. p. 10687–10698. [CrossRef]

[24] Sajjadi M, Javanmardi M, Tasdizen T. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. Adv Neural Inf Process Syst 2016;29.

[25] Tarvainen A, Valpola H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. Adv Neural Inf Process Syst 2017;30.

[26] Blum A, Mitchell T. Combining labeled and unlabeled data with co-training. In: Proceedings of the eleventh annual conference on Computational learning theory; 1998. p. 92–100. [CrossRef]

[27] Zhou ZH, Li M. Tri-training: exploiting unlabeled data using three classifiers. IEEE Trans Knowl Data Eng 2005;17:1529–1541. [CrossRef]

[28] Søgaard A. Simple semi-supervised training of part-of-speech taggers. In: Proceedings of the ACL 2010 Conference Short Papers; 2010. p. 205–208.

[29] Ghosh S, Kumar S, Verma J, Kumar A. Self training with ensemble of teacher models. arXiv Prepr arXiv:2107.08211; 2021.

[30] Grandvalet Y, Bengio Y. Semi-supervised learning by entropy minimization. Adv Neural Inf Process Syst 2004;17.

[31] Smith LN. Cyclical learning rates for training neural networks. In: 2017 IEEE winter conference on applications of computer vision (WACV), IEEE; 2017. p. 464–472. [CrossRef]

[32] Hearst M, Dumais S, Osuna E, Platt J, Scholkopf B. Support vector machines. IEEE Intell Syst Appl 1998;13:18–28. [CrossRef]

[33] Coates A, Ng A, Lee H. An analysis of single-layer networks in unsupervised feature learning. In: Gordon G, Dunson D, Dudík M, editors. Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, PMLR, Fort Lauderdale, FL, USA; 2011. p. 215−223.

[34] Huang G, Liu Z, Weinberger KQ. Densely connected convolutional networks. CoRR abs/1608.06993; 2016. [CrossRef]

[35] Loshchilov I, Hutter F. Decoupled weight decay regularization. In: International Conference on Learning Representations; 2018.

[36] Quinlan JR. Learning decision tree classifiers. ACM Comput Surv. 1996;28:71−72. [CrossRef]

[37] Rigatti SJ. Random forest. J Insur Med 2017;47:31−39. [CrossRef]

[38] Peterson LE. K-nearest neighbor. Scholarpedia 2009;4:1883. [CrossRef]

[39] Van der Maaten L, Hinton G. Visualizing data using t-sne. J Mach Learn Res 2008;9:1−27.

[40] Jaderberg M, Simonyan K, Zisserman A, et al. Spatial transformer networks. Adv Neural Inf Process Syst 2015;28.

[41] Kayhan OS, Gemert JC. On translation invariance in cnns: Convolutional layers can exploit absolute spatial location. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. p. 14274−14285.

[42] Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision; 2017. p. 618−626. [CrossRef]

[43] McHugh ML. Interrater reliability: the kappa statistic. Biochem Med 2012;22:276−282. [CrossRef]