Deep ensemble learning by diverse knowledge distillation for fine-grained object classification

Naoki Okamoto, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi

Chubu University, Kasugai, Aichi, Japan
{naok,hirakawa}@mprg.cs.chubu.ac.jp,{takayoshi,fujiyoshi}@isc.chubu.ac.jp

Abstract. Ensemble of networks with bidirectional knowledge distillation does not significantly improve on the performance of ensemble of networks without bidirectional knowledge distillation. We think that this is because there is a relationship between the knowledge in knowledge distillation and the individuality of networks in the ensemble. In this paper, we propose a knowledge distillation for ensemble by optimizing the elements of knowledge distillation as hyperparameters. The proposed method uses graphs to represent diverse knowledge distillations. It automatically designs the knowledge distillation for the optimal ensemble by optimizing the graph structure to maximize the ensemble accuracy. Graph optimization and evaluation experiments using Stanford Dogs, Stanford Cars, CUB-200-2011, CIFAR-10, and CIFAR-100 show that the proposed method achieves higher ensemble accuracy than conventional ensembles.

Keywords: ensemble learning, knowledge distillation

1 Introduction

Deep learning models trained under the same conditions, such as network architecture and dataset, produce variations in accuracy and different errors due to random factors such as network initial values and mini-batches. Ensemble and knowledge distillation improve the performance by using multiple networks with different weight parameters for training and inference.

Ensemble performs inference using multiple trained networks. It performs inference on the basis of the average of the output of each network for the input samples and thus improves the performance compared with an inference using a single network. It is also effective against problems such as adversarial attack and out-of-distribution detection due to the nature of using multiple networks [15,6,5,21,26]. It is computationally more expensive than a single network, so methods for constructing parameter-efficient ensembles have been proposed[33,34,28,16,26].

Knowledge distillation is a training method where a network shares the knowledge acquired through training with other networks to reduce parameters or improve network performance. There are two types of knowledge distillation:

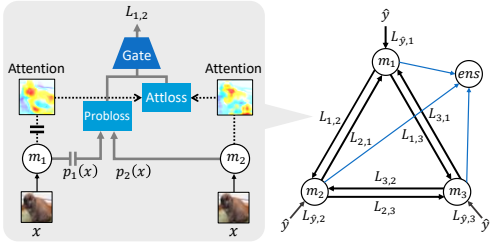


Fig. 1: Ensemble learning with diverse knowledge distillation by graph representation. Loss calculation shows knowledge distillation from m_1 to m_2.

unidirectional [10] and bidirectional [38]. A typical method for unidirectional knowledge distillation is knowledge distillation (KD) [10]. KD uses probability distributions as knowledge of the network and trains an untrained network with shallow layers using a trained network with deep layers. A typical method for bidirectional knowledge distillation is deep mutual learning (DML) [38]. DML is a method of mutual distillation using multiple untrained networks. Various distillation methods have been proposed depending on the combination of the networks and the type of knowledge [27,20,35,31,25,1,36,30,37,3]. Minami et al. [22] introduced a graph representation for knowledge distillation to unify the existing methods.

In this paper, we propose knowledge distillation for ensemble. The critical factors of the proposed method are knowledge distillation that promotes diversity among networks and automatic design of the distillation method. We consider knowledge distillation that separates knowledge between networks, in addition to conventional knowledge distillation. The direct separation of probability distributions as knowledge may degrade the performance of the network. Therefore, we perform diverse knowledge distillation from two types of knowledge: probability distributions and attention maps. The proposed method represents the diverse knowledge distillation in a graph [22], as shown in Fig. 1. We define the combination of loss design of knowledge distillation as a hyperparameter and automatically design complex knowledge distillation, which is difficult to design manually by hyperparameter search.

Our contributions are as follows.

- We investigate the relationship between the difference of probability distributions and the effect of ensemble and find a positive correlation.
- We perform knowledge distillation to promote diversity among networks for ensembles. We design a loss of proximity and loss of separation of knowledge using probability distribution and attention map and weight the loss values using gates to achieve diverse knowledge distillation.
- We use graphs to represent diverse knowledge distillation and automatically design appropriate knowledge distillations by optimizing the graph structure. The networks of the optimized graph improve the ensemble accuracy, and each network is a diverse model with a different attention map.

2 Related work

In this section, we introduce ensemble and knowledge distillation, which are methods for using multiple networks.

2.1 Ensemble

Ensemble is one of the oldest machine learning methods. Ensemble in deep learning is a simple method of averaging probability distributions or logits, which are the outputs of networks with different weight parameters. It is known that the ensemble accuracy improves depending on the number of networks and that the ensemble accuracy ceases to improve after exceeding a certain number of networks. Ensemble is also effective against problems such as adversarial attack and out-of-distribution detection due to it using multiple networks [6,15,5,21,26].

The training and inference cost of ensemble increases with the number of networks. In knowledge distillation [28,16], a single network can achieve the same performance as ensemble by training the network to approach the probability distribution by the ensemble. Batch ensemble[33] and hyperparameter ensemble [34] prevent increasing the parameters by sharing some of them and reduce the training and inference costs.

2.2 Knowledge distillation

Knowledge distillation is a training method where a network shares the knowledge acquired through training with other networks to reduce parameters or improve network performance. There are two types of knowledge distillation: unidirectional and bidirectional.

Unidirectional knowledge distillation uses a teacher network, which is a trained network, and a student network, which is an untrained network. The student network trains the outputs of the teacher network as pseudo-labels in addition to the labels. Hinton et al.[10] proposed KD, which trains the student network with small parameters using the probability distribution of the teacher network with large parameters. KD is effective even for teacher and student networks with the same number of parameters[8]. There is also a two-stage knowledge distillation using three networks[23].

Bidirectional knowledge distillation trains multiple student networks at the same time, using the probability distributions of the student networks as pseudo-labels. The first bidirectional knowledge distillation, DML, was proposed by Zhang et al. [38]. In DML, the accuracy of the network increases with the number of networks.

On-the-Fly Native Ensemble (ONE) [16] is knowledge distillation using ensemble. ONE reduces the number of parameters by using a multi-branch network and produces a training effect similar to that of DML by using an ensemble of multi-branches as pseudo-labels.

A variety of knowledge has been proposed, such as probability distributions, feature maps, attention maps, and relationships between samples [27,20,35,31,25]

Table 1: Correlation coefficient for each dataset.

Dataset	Correlation coefficient
Stanford Dogs	0.237
Stanford Cars	0.499
CUB-200-2011	0.322
CIFAR-10	0.386
CIFAR-100	0.325

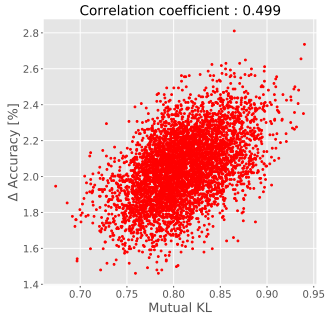


Fig. 2: Relationship on Stanford Cars.

[1,36,30,37,3]. Minami et al. [22] introduced a graph representation for knowledge distillation to unify the existing methods. Knowledge distillation is also effective in a variety of problem settings [4,19,29,26].

3 Investigating the relationship between ensemble and knowledge distillation

We think that there is a correlation between the differences of probability distributions and ensemble accuracy because of the relationship between bidirectional distillation and ensemble. In this section, we investigate the relationship between the difference in probability distributions and ensemble accuracy and verify the change in ensemble accuracy caused by knowledge distillation on the Stanford Dogs dataset [12].

3.1 Relationship between bidirectional knowledge distillation and ensemble

We investigate the relationship between KL-divergence, a loss design of DML[38], and the accuracy improvement by ensemble of two models. KL-divergence is a measure of the difference in distribution. However, it is an asymmetric measure, so the analysis defines the measure of difference as

$$\text{Mutual KL} = \frac{1}{2}(KL(p_1 \parallel p_2) + KL(p_2 \parallel p_1)), \tag{1}$$

where p_1 and p_2 are the probability distributions of networks 1 and 2, respectively. The accuracy improvement by ensemble is defined as

$$\Delta\text{Accuracy} = \text{ACC}_{ens} - \frac{1}{2}(\text{ACC}_1 + \text{ACC}_2), \tag{2}$$

where ACC_{ens} is the ensemble accuracy, ACC_1 is the accuracy of network 1, and ACC_2 is the accuracy of network 2. The datasets are Stanford Dogs [12],

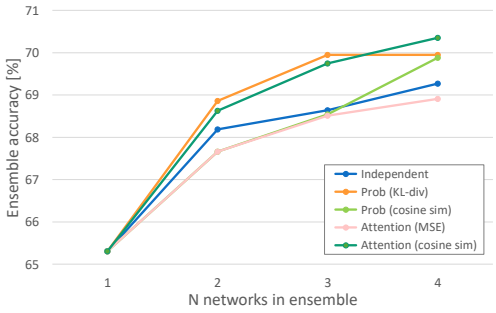


Fig. 3: Ensemble accuracy of loss design for various numbers of networks.

Stanford Cars [13], Caltech-UCSD Birds-200-2011 (CUB-200-2011) [32], CIFAR-10 [14], and CIFAR-100 [14]. The network is ResNet [9], and 100 trained networks are prepared. ResNet-20 is used for CIFAR datasets, and ResNet-18 is used for other datasets. Ensemble is constructed by selecting two networks out of 100, and evaluation is performed on all combinations (4,950 pairs).

Table 1 shows the correlation coefficients for each dataset, and Fig. 2 shows the evaluation results for Stanford Cars. There is a weak positive correlation between the difference of probability distributions and the improvement of accuracy by ensemble. Therefore, it is expected that the ensemble effect can be improved by training to have diversity in the probability distributions between networks.

3.2 Bidirectional knowledge distillation to promote diversity for ensemble

On the basis of analysis trends, we consider knowledge distillation that separates knowledge to improve the ensemble accuracy. Therefore, we investigate the effect on the ensemble by training to bring knowledge closer and training to separate knowledge. We use two types of knowledge: the probability distribution that is the final output of the network, and the attention map that represents the information in the middle layer. The loss design for probability distributions uses KL-divergence to bring the probability distributions closer together and cosine similarity to separate the probability distributions. The loss design for the attention map uses mean square error to bring the attention map closer together and cosine similarity to separate the attention maps. We use ResNet-18 [9] as the network and Stanford Dogs [12] as the dataset. The attention map is created from the output of ResBlock4 using Attention Transfer [37].

Fig. 3 shows the results of the evaluation of each loss design with the number of networks used in the ensemble from 1 to 4. Here, Independent is the ensemble accuracy without knowledge distillation. The ensemble accuracy is improved by training to bring the probability distributions closer together when the number of networks is small and training to separate the attention maps when the number of networks is large.

4 Proposed Method

We propose an ensemble learning method using knowledge distillation. From the trend in Sec. 3, it is difficult to intentionally design a knowledge distillation method for each number of networks. Therefore, we propose to automatically design an effective ensemble learning method. We use the graph representation in the knowledge transfer graph [22] and optimize the loss design of diverse knowledge distillation as a hyperparameter of the graph by hyperparameter search. We consider various ensemble learning methods by optimizing the structure of the graph to maximize ensemble accuracy. We show that using the automatically designed ensemble learning methods improves the ensemble accuracy and that each network is prompted to a specific attention strategy by the combination of the selected knowledge distillations regardless of the dataset.

4.1 Designing for loss of knowledge distillation to promote diversity

We perform knowledge distillation to promote diversity among networks for ensemble. In this paper, we use probability distributions and attention maps as knowledge, and design loss of bringing knowledge closer and loss of separating knowledge. To train as a minimization problem, we use different loss designs for bringing knowledge closer and separating knowledge. We refer to the destination of knowledge distillation as the target network t and the source of knowledge as the source network s.

Loss design for the probability distribution When the probability distribution is brought closer together, KL-divergence is used, and when it is separated, cosine similarity is used. The loss function using KL-divergence is defined as

$$KL(p_s(x) \parallel p_t(x)) = \sum_{c=1}^{C} p_s^c(x) \log \frac{p_s^c(x)}{p_t^c(x)}, \tag{3}$$

$$L_p = KL(p_s(x) \parallel p_t(x)), \tag{4}$$

where C is the number of the classes, x is the input sample, $p_s(x)$ is the probability distribution of the source network, and $p_t(x)$ is the probability distribution of the target network. The loss function using cosine similarity is defined as

$$L_p = \frac{p_s(x)}{\parallel p_s(x) \parallel_2} \cdot \frac{p_t(x)}{\parallel p_t(x) \parallel_2}. \tag{5}$$

Loss design for the attention map The attention map responds strongly to regions in the input sample that is useful for training. The size of the target object varies from sample to sample, so the similarity may be high even though the map responds strongly to different parts of the target object. Therefore, we crop the attention map. The attention map of the source network is cropped to a square centered on the position with the highest value, and the attention map of the target network is cropped to the same position as the source network.

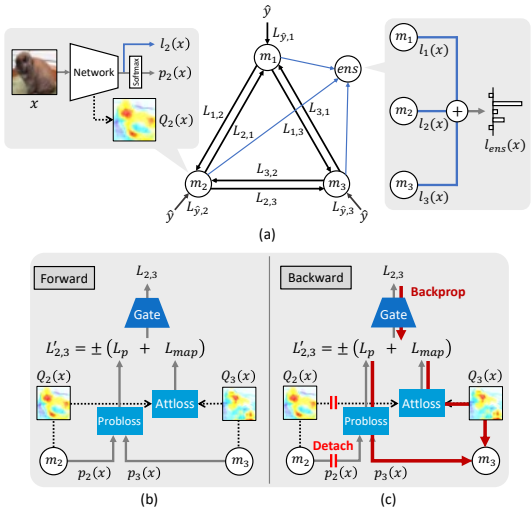


Fig. 4: (a) Ensemble learning with diverse knowledge distillation by graph representation. (b) Loss calculation shows knowledge transfer from m_2 to m_3. (c) Calculated loss gradient information is only propagated in m_3.

Cropping is performed at multiple sizes, and the average of the similarities at each size is used as the similarity of the attention map. When the attention map is brought closer together, the mean square error is used, and when it is separated, cosine similarity is used. The loss function using mean squared error is defined as

$$L_{map} = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{Q_s^k(x)}{\| Q_s^k(x) \|_2} - \frac{Q_t^k(x)}{\| Q_t^k(x) \|_2}\right)^2, \tag{6}$$

where K is the number of crops, Q_s is the attention map of the source network and Q_t is the attention map of the target network. The loss function using cosine similarity is defined as

$$L_{map} = \frac{1}{K}\sum_{k=1}^{K}\frac{Q_s^k(x)}{\| Q_s^k(x) \|_2} \cdot \frac{Q_t^k(x)}{\| Q_t^k(x) \|_2}. \tag{7}$$

Introducing Gate We control knowledge distillation by weighting the above loss values of the probability distribution and the attention map using gates. In this paper, we consider four types of gates: through, cutoff, linear, and correct. The through gate passes through the loss value of each input sample as it is and is defined as

$$G_{s,t}^{Though}(a) = a. \tag{8}$$

The cutoff gate does not execute loss calculation and is defined as

$$G_{s,t}^{Cutoff}(a) = 0. \tag{9}$$

The linear gate changes the weights linearly with training time and is defined as

$$G_{s,t}^{Linear}(a) = \frac{k}{k_{end}}a, \tag{10}$$

where k is the number of the current iterations and k_{end} is the total number of iterations at the end of the training. The correct gate passes only the samples that the source network answered correctly and is defined as

$$G_{s,t}^{Correct}(a) = \begin{cases} a & y_s = \hat{y} \\ 0 & y_s \neq \hat{y} \end{cases}, \tag{11}$$

where y_s is the output of the source network and $\hat{y}$ is a label.

4.2 Graph representation and optimization of graph structures

A diverse knowledge distillation by the losses in Sec. 4.1 is represented by a graph [22], and an appropriate knowledge distillation is automatically designed by optimizing the graph structure.

Graph representation for ensemble We use a graph representation [22] in a knowledge transfer graph to automatically design knowledge distillation. The ensemble learning using knowledge distillation by the graph representation is shown in Fig. 4. The graph consists of nodes and edges. Nodes define the network node that represents the network and the ensemble node that performs ensemble. Edges represent loss calculation. Edges between network nodes represent knowledge distillation. Edges between the network node and the label represent cross-entropy loss using the output of the node and the label.

The ensemble node performs ensemble by using the outputs of all the network nodes. The process in an ensemble node is defined as

$$l_{ens} = \frac{1}{M}\sum_{m=1}^{M} l_m(x). \tag{12}$$

where M is the number of network nodes, l_m is the logits of the network node, and x is the input sample.

Knowledge distillation between nodes Edges between network nodes perform knowledge distillation between nodes. Fig. 4b and 4c show the process of loss processing at the edge from node m_2 to node m_3. First, the edge computes the loss of knowledge distillation of the probability distribution and attention map as shown in Fig. 4b. The loss calculation of knowledge distillation at the edge is defined as

$$L_{s,t}^{'} = L_p(x) + L_{map}(x). \tag{13}$$

The final loss of knowledge distillation is then applied to the gate. The loss of knowledge distillation applied to the gate is defined as

$$L_{s,t} = \frac{1}{N}\sum_{n=1}^{N} G_{s,t}(L_{s,t}^{'}(x_n)), \tag{14}$$

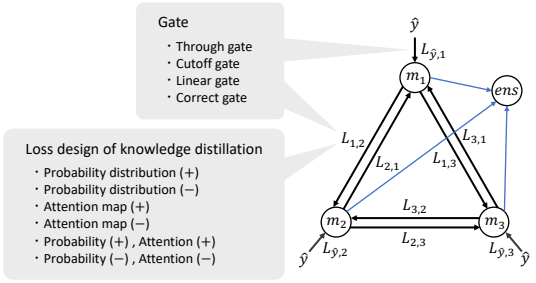


Fig. 5: Hyperparameters in graph structures for ensemble.

where N is the number of the input sample, and $G_{s,t}(\cdot)$ is one of the four types of gate. The gradient of the loss of knowledge distillation changes the network that propagates the gradient depending on the edge direction. Fig. 4c shows the gradient flow at the edge from node m_2 to node m_3. In this case, knowledge distillation from node m_2 to node m_3 is performed by cutting the computational graph of node m_2 to propagate the gradient only to node m_3.

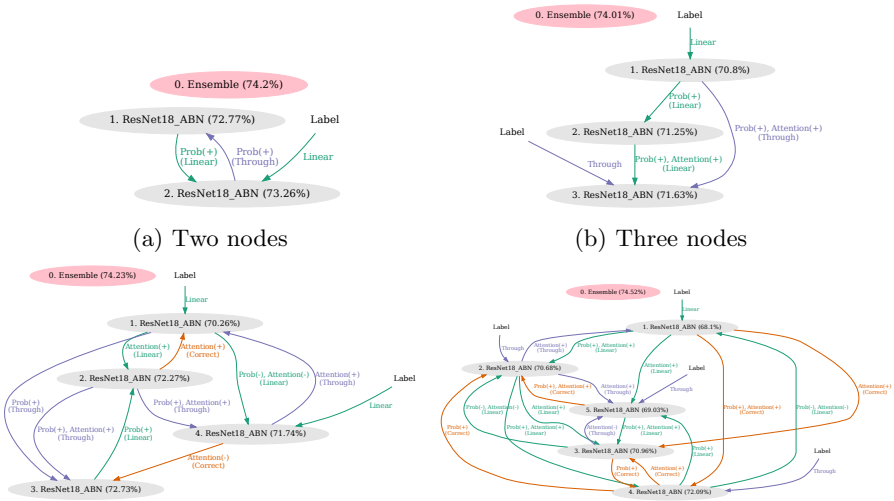
The loss calculation is performed for each edge, and the final loss of the network node is defined as

$$L_t = G_{hard,t}(L_{hard}) + \sum_{s=1,s\neq t}^{M} L_{s,t}, \tag{15}$$

where $G_{hard,t}$ is the gate, and L_{hard} is cross-entropy loss using the output of the network node and the label $\hat{y}$.

Hyperparameter search Fig. 5 shows the hyperparameters of the graph structure. The hyperparameters of the graph are the loss design of the edges between the network nodes and the gate of each edge. There are six loss designs: bring the probability distribution closer to that of the other edge (Eq.4), separate the probability distribution (Eq.5), bring the attention map closer to that of the other edge (Eq.6), separate the attention map (Eq.7), bring the probability distribution and attention map closer to those of the other edge at the same time (Eqs.4 and 6), and separate the probability distribution and attention map at the same time (Eqs.5 and 7). The network to be used as the network node is fixed to that determined before optimization.

The optimization of the graph structure uses random search and the asynchronous successive halving algorithm (ASHA) [18]. The combination of hyperparameters is determined randomly, and the graph evaluates the ensemble node at $1, 2, 4, 8 \cdots 2^k$ epochs. If the accuracy of the ensemble node is less than the median accuracy at the same epoch in the past, the training is terminated and the next graph is trained.



(a) Two nodes (b) Three nodes

(c) Four nodes (d) Five nodes

Fig. 6: Graph optimized on Stanford Dogs. Red node represents ensemble node, gray node represents network node, and "Label" represents supervised labels. At each edge, selected loss design and gate are shown, exclusive of cutoff gate. Accuracy in parentheses is the result of one of five trials.

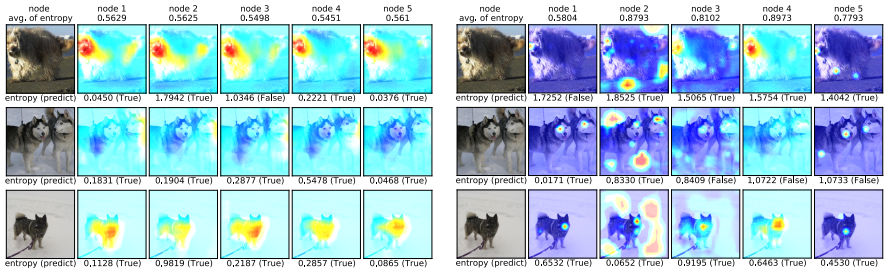
5 Experiments

We evaluate the proposed method. In Sec. 5.2, we visualize the optimized graph structure. In Sec. 5.3, we compare the proposed method with the conventional method. In Sec. 5.4, we evaluate the generalizability of the graph structure on various datasets. In Sec. 5.5, we evaluate the performance of knowledge distillation from the optimized ensemble graph into a single network.

5.1 Experimental setting

Datasets We used Stanford Dogs [12], Stanford Cars [13], Caltech-UCSD Birds-200-2011 (CUB-200-2011) [32], CIFAR-10 [14], and CIFAR-100 [14]. Stanford Dogs, Stanford Cars, and CUB-200-2011 belong to the fine-grained object classification task. CIFAR-10 and CIFAR-100 belong to the general object classification task. When optimizing the graphs, we used part of the training data for training and the rest for evaluation. We used 40,000 images for CIFAR and half of the training data for other datasets. For the comparative evaluation discussed in Sec.5.3 and 5.4, the original training data and testing data were used.

Networks We used ResNet [9] and attention branch network (ABN) [7] based on ResNet. When training the CIFAR dataset, we used ResNet-20 and ABN based on ResNet-20. When training the other dataset, we used ResNet-18 and



(a) Independent (b) Ours

Fig. 7: Attention map of ABN in individually training and the optimized graph with five nodes (Fig. 6d). Bottom of map shows prediction results and entropy of probability distribution.

ABN based on ResNet-18. The attention map of ResNet is created from the output of ResBlock4 by Attention Transfer [37]. ABN creates an attention map on the basis of the class activation map [39] and weights the attention map to the feature map by using the attention mechanism.

Implementation details The training conditions were the same for all experiments. The optimization algorithms were stochastic gradient descent (SGD) and momentum. The initial learning rate was 0.1, momentum was 0.9, coefficient of weight decay was 0.0001, batch size was 16, and number of epochs was 300. The learning rate was decayed by a factor of 10 at 150 and 225 epochs. The attention map of ResNet is cropped to 3×3, 5×5, and 7×7 for loss calculation. The attention map of ABN is cropped to 3×3, 7×7, and 11×11 for loss calculation. In the optimization of the graph, we tried 6,000 combinations of hyperparameters. We used PyTorch [24] as a framework for deep learning and Optuna [2] as a framework for hyperparameter search. For the optimization of a graph, we used 90 Quadro P5000 servers. Each result represents the mean and standard deviation of five trials.

5.2 Visualization of optimized graphs

Fig. 6 shows the graphs of two to five nodes optimized on Stanford Dogs. With two nodes, we obtained a graph that is an extension of DML. With three nodes, we obtained a graph that combines the conventional knowledge distillation methods of KD and TA. With four and five nodes, we obtained graphs with a mixture of loss designs that are brought closer together and loss designs that are separated.

Fig. 7b shows the attention map of ABN with the five nodes. Each node has a different focus of attention. Looking at the average entropy, nodes 1 and 5, which focus on a single point on the dog's head, have low entropy. Nodes 2, 3, and 4, which focus on the whole image or background, have higher entropy

Table 2: Comparison of the accuracy on Stanford Dogs [%].

Method	No. of nodes	ResNet-18		ABN	
		Node	Ensemble	Node	Ensemble
Independent	2	65.31 ± 0.16	68.19 ± 0.20	68.13 ± 0.16	70.90 ± 0.19
DML	2	67.55 ± 0.27	68.86 ± 0.25	69.91 ± 0.46	71.45 ± 0.52
ONE(B×2)	1	67.96 ± 0.41	68.53 ± 0.39	69.38 ± 0.38	69.81 ± 0.33
Ours	2	**71.38 ± 0.08**	**72.41 ± 0.20**	**72.77 ± 0.23**	**73.86 ± 0.26**
Independent	3	65.08 ± 0.23	68.64 ± 0.38	68.04 ± 0.28	71.41 ± 0.34
DML	3	68.66 ± 0.34	69.95 ± 0.39	70.50 ± 0.26	72.08 ± 0.42
ONE(B×3)	1	68.49 ± 0.60	68.94 ± 0.56	69.96 ± 0.47	70.44 ± 0.44
Ours	3	**69.58 ± 0.15**	**71.87 ± 0.33**	**70.95 ± 0.16**	**73.41 ± 0.30**
Independent	4	65.29 ± 0.35	69.27 ± 0.49	68.30 ± 0.27	72.06 ± 0.53
DML	4	68.83 ± 0.44	69.95 ± 0.58	**71.50 ± 0.31**	72.87 ± 0.29
ONE(B×4)	1	68.48 ± 0.32	68.85 ± 0.37	70.16 ± 0.47	70.54 ± 0.54
Ours	4	**70.34 ± 0.12**	**72.71 ± 0.13**	71.46 ± 0.22	**74.16 ± 0.22**
Independent	5	65.00 ± 0.24	69.47 ± 0.13	68.24 ± 0.26	72.32 ± 0.18
DML	5	**68.77 ± 0.17**	69.94 ± 0.20	**71.15 ± 0.28**	72.50 ± 0.16
ONE(B×5)	1	68.51 ± 0.18	68.95 ± 0.24	70.59 ± 0.28	70.89 ± 0.14
Ours	5	52.28 ± 0.87	**71.35 ± 0.48**	70.23 ± 0.33	**74.14 ± 0.50**

than nodes 1 and 5. This means that inferences are made on the basis of the importance of different locations and the state of attention affects probability distribution.

Fig. 7a shows the attention map in the ensemble method using individually trained networks. Compared with the optimized graph, the average entropy of the ensemble method using individually trained networks is lower. This is because the attention regions are almost the same among the networks even though they are trained individually.

5.3 Comparison with conventional methods

Table 2 shows the average and ensemble accuracy of the nodes of the proposed and conventional methods on Stanford Dogs. "Ours" is the result of the optimized graph, "Independent" is the result of the individually trained network, "DML" is the result of the network with DML [38], and "ONE" is the result of the multi-branch network with ONE [16]. "ONE(B×2)" is the result of the two-branch network. The ensemble accuracy of "Ours" was higher than those of "Independent," "DML," and "ONE." Comparing "Independent" and "DML," we can see that the improvement in ensemble accuracy was smaller than the improvement in node accuracy. With "Ours," compared with "DML," ensemble accuracy also improved as network accuracy improved. Therefore, we can say that "Ours" obtained the graph that generates more diversity by training.

Fig. 8 shows the comparison results with ABN and different base networks. The vertical axis is accuracy, and the horizontal axis is the total number of parameters. In Stanford Dogs, the accuracy of the single network and "Independent" varied with the number of parameters. "Ours" shows that ensemble with high parameter efficiency can be constructed by mutual learning with diversity

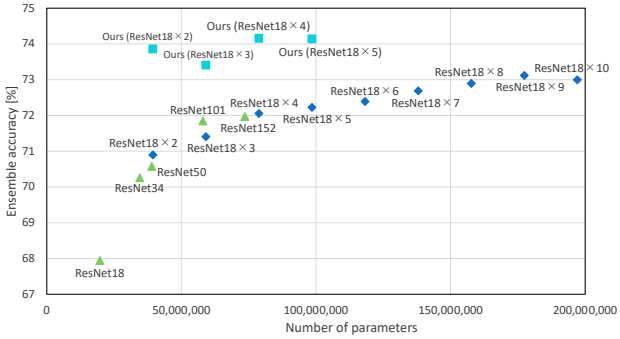


Fig. 8: Relationship between number of parameters and accuracy in Stanford Dogs. Green shows single network, blue shows "Independent," and light blue shows "Ours."

Table 3: Ensemble accuracy of reused two-node graphs optimized on another dataset [%].

Method	Training Graph	Optimizing Graph	Ensemble
Independent	CUB-200-2011	-	65.26
Ours	CUB-200-2011	Stanford Dogs	**72.06**
Ours	CUB-200-2011	CUB-200-2011	69.81
Independent	Stanford Cars	-	88.49
Ours	Stanford Cars	Stanford Dogs	**89.76**
Ours	Stanford Cars	Stanford Cars	89.44
Independent	CIFAR-100	-	73.16
Ours	CIFAR-100	Stanford Dogs	72.19
Ours	CIFAR-100	CIFAR-100	**74.18**
Independent	CIFAR-10	-	93.99
Ours	CIFAR-10	Stanford Dogs	93.87
Ours	CIFAR-10	CIFAR-100	**94.37**
Ours	CIFAR-10	CIFAR-10	94.15

without changing the network structure. When the number of networks is increased, ensemble accuracy reaches a ceiling of around 73%. This shows that the proposed method achieved an accuracy that exceeds the limit of a conventional method.

5.4 Generalizability of graphs

We evaluate the optimized graph in Stanford Dogs on a variety of datasets. Table 3 shows the ensemble accuracy of the two-node graph. On the dataset of the fine-grained object classification, the graph optimized by Stanford Dogs has better accuracy than Independent. On CIFAR-10, the graph optimized by CIFAR-100 has better accuracy than Independent. We believe that there is generalizability

Table 4: Accuracy of single network by knowledge distillation [%].

Method	Teacher	Studet
DML	67.97	69.68
KTG	71.71	72.71
SLA	-	69.36
FRSKD	-	71.42
Ours	72.60	**72.94**

in the graph structure when the problem set is the same, and that optimization has resulted in a graph structure that corresponds to the problem set.

5.5 Knowledge Distillation from ensemble learning

We evaluate the performance of knowledge distillation from the optimized ensemble graph into a single network. We use ResNet-18 as a student network and DML [38] KTG [22] as a teacher network for knowledge distillation on the Stanford Dogs dataset. We also compare with the state of the art of knowledge distillation, such as SLA [17] and FRSKD [11], which are self-distillation methods. The table 4 shows the accuracy of teachers and students trained with each method. "Ours" means knowledge distillation using the ensemble of two networks trained by the graph of Fig. 6a as a teacher. From the tabale 4, we see that the accuracy of the student network by "Ours" is higher than that of the conventional methods. This is because the ensemble of two networks trained by the graph has diversity for representing dark knowledge to make suitable knowledge transfer.

6 Conclusion and Future Work

This paper proposed a knowledge distillation for ensemble. We investigated loss design for ensemble to promote diversity among the networks and automatically designed knowledge distillation for ensemble by graph representation. Experimental results on five different datasets showed that the proposed method increased the accuracy. The optimization of the graph structure was evaluated on 6,000 randomly determined pairs using the asynchronous successive halving algorithm (ASHA). The number of combinations of graph structures increases in proportion to the number of nodes. Therefore, increasing the number of combinations to be evaluated may result in a better graph structure. Our future work will include introducing Bayesian optimization and fine-tuning graph structures.

Acknowledgements This paper is based on results obtained from a project, JPNP18002, commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

References

1. Ahn, S., Hu, S.X., Damianou, A., Lawrence, N.D., Dai, Z.: Variational information distillation for knowledge transfer. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9163–9171 (2019)
2. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2019)
3. Chattopadhay, A., Sarkar, A., Howlader, P., Balasubramanian, V.N.: Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In: IEEE Winter Conference on Applications of Computer Vision (2018)
4. Chen, G., Choi, W., Yu, X., Han, T., Chandraker, M.: Learning efficient object detection models with knowledge distillation. In: Advances in Neural Information Processing Systems. pp. 742–751 (2017)
5. Dabouei, A., Soleymani, S., Taherkhani, F., Dawson, J., Nasrabadi, N.M.: Exploiting joint robustness to adversarial perturbations. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)
6. Dvornik, N., Schmid, C., Mairal, J.: Diversity with cooperation: Ensemble methods for few-shot classification. In: IEEE/CVF International Conference on Computer Vision (2019)
7. Fukui, H., Hirakawa, T., Yamashita, T., Fujiyoshi, H.: Attention branch network: Learning of attention mechanism for visual explanation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)
8. Furlanello, T., Lipton, Z., Tschannen, M., Itti, L., Anandkumar, A.: Born again neural networks. In: International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 1607–1616 (2018)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
10. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. In: Neural Information Processing Systems Deep Learning and Representation Learning Workshop (2015)
11. Ji, M., Shin, S., Hwang, S., Park, G., Moon, I.C.: Refine myself by teaching myself: Feature refinement via self-knowledge distillation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10664–10673 (2021)
12. Khosla, A., Jayadevaprakash, N., Yao, B., Fei-Fei, L.: Novel dataset for fine-grained image categorization. In: First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition (2011)
13. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: 4th International IEEE Workshop on 3D Representation and Recognition. Sydney, Australia (2013)
14. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Tech. rep., Citeseer (2009)
15. Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. In: International Conference on Learning Representations (2017)
16. Lan, X., Zhu, X., Gong, S.: Knowledge distillation by on-the-fly native ensemble. In: Advances in Neural Information Processing Systems. pp. 7527–7537 (2018)
17. Lee, H., Hwang, S.J., Shin, J.: Self-supervised label augmentation via input transformations. In: International Conference on Machine Learning. pp. 5714–5724 (2020)

18. Li, L., Jamieson, K., Rostamizadeh, A., Gonina, E., Ben-tzur, J., Hardt, M., Recht, B., Talwalkar, A.: A system for massively parallel hyperparameter tuning. In: Dhillon, I.S., Papailiopoulos, D.S., Sze, V. (eds.) Proceedings of Machine Learning and Systems. vol. 2, pp. 230–246 (2020)
19. Liu, Y., Chen, K., Liu, C., Qin, Z., Luo, Z., Wang, J.: Structured knowledge distillation for semantic segmentation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)
20. Liu, Y., Cao, J., Li, B., Yuan, C., Hu, W., Li, Y., Duan, Y.: Knowledge distillation via instance relationship graph. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)
21. Malinin, A., Mlodozeniec, B., Gales, M.: Ensemble distribution distillation. In: International Conference on Learning Representations (2020)
22. Minami, S., Hirakawa, T., Yamashita, T., Fujiyoshi, H.: Knowledge Transfer Graph For Deep Collaborative Learning. In: Asian Conference on Computer Vision (2020)
23. Mirzadeh, S.I., Farajtabar, M., Li, A., Ghasemzadeh, H.: Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher. In: Association for the Advancement of Artificial Intelligence (2020)
24. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019)
25. Peng, B., Jin, X., Liu, J., Li, D., Wu, Y., Liu, Y., Zhou, S., Zhang, Z.: Correlation congruence for knowledge distillation. In: IEEE/CVF International Conference on Computer Vision (2019)
26. Radosavovic, I., Dollár, P., Girshick, R., Gkioxari, G., He, K.: Data distillation: Towards omni-supervised learning. In: IEEE Conference on Computer Vision and Pattern Recognition (2018)
27. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets. In: International Conference on Learning Representations (2015)
28. Song, G., Chai, W.: Collaborative learning for deep neural networks. In: Advances in Neural Information Processing Systems. pp. 1837–1846 (2018)
29. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In: Advances in Neural Information Processing Systems (2017)
30. Tian, Y., Krishnan, D., Isola, P.: Contrastive representation distillation. In: International Conference on Learning Representations (2020)
31. Tung, F., Mori, G.: Similarity-preserving knowledge distillation. In: IEEE/CVF International Conference on Computer Vision (2019)
32. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset. Tech. Rep. CNS-TR-2011-001, California Institute of Technology (2011)
33. Wen, Y., Tran, D., Ba, J.: Batchensemble: an alternative approach to efficient ensemble and lifelong learning. In: International Conference on Learning Representations (2020)
34. Wenzel, F., Snoek, J., Tran, D., Jenatton, R.: Hyperparameter ensembles for robustness and uncertainty quantification. In: Larochelle, H., Ranzato, M., Hadsell,

R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 6514–6527. Curran Associates, Inc. (2020)
35. Yim, J., Joo, D., Bae, J., Kim, J.: A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 4133–4141 (2017)
36. Yu, L., Yazici, V.O., Liu, X., Weijer, J.v.d., Cheng, Y., Ramisa, A.: Learning metrics from teachers: Compact networks for image embedding. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)
37. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In: International Conference on Learning Representations (2017)
38. Zhang, Y., Xiang, T., Hospedales, T.M., Lu, H.: Deep mutual learning. In: IEEE Conference on Computer Vision and Pattern Recognition (2018)
39. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)