Super-class Mixup for Adjusting Training Data

Shungo Fujii[1], Naoki Okamoto[1], Toshiki Seo[1], Tsubasa Hirakawa[1], Takayoshi Yamashita[1], and Hironobu Fujiyoshi[1]

Chubu University, 1200 Matsumotocho, Kasugai, Aichi, Japan
{drvfs2759@mprg.cs,naok@mprg.cs,seotoshiki@mprg.cs,
hirakawa@mprg.cs,takayoshi@isc,fujiyoshi@isc}.chubu.ac.jp

Abstract. Mixup is one of data augmentation methods for image recognition task, which generate data by mixing two images. Mixup randomly samples two images from training data without considering the similarity of these data and classes. This random sampling generates mixed samples with low similarities, which makes a network training difficult and complicated. In this paper, we propose a mixup considering super-class. Super-class is a superordinate categorization of object classes. The proposed method tends to generate mixed samples with the almost same mixing ratio in the case of the same super-class. In contrast, given two images having different super-classes, we generate samples largely containing one image's data. Consequently, a network can train the features between similar object classes. Furthermore, we apply the proposed method into a mutual learning framework, which would improve the network output used for mutual learning. The experimental results demonstrate that the proposed method improves the recognition accuracy on a single model training and mutual training. And, we analyze the attention maps of networks and show that the proposed method also improves the highlighted region and makes a network correctly focuses on the target object.

Keywords: Mixup · Super-class · Data Augmentation

1 Introduction

Data augmentation is a fundamental method for computer vision tasks, which increases the number of training data and data variations. The classical data augmentation approach is a simple image processing such as transition, resizing, adding noise, and contrast adjustment. Due to the recent development of deep learning-based methods and the requirements of a large number of training samples for enough network training, efficient data augmentation methods have been proposed [1, 2, 4, 11, 16, 17, 19] To generate more efficient training samples, augmentation methods that uses multiple samples have been proposed [17, 16]. Among them, the mixup [17] samples two images from training data and generate a mixed image. The mixup can make the diversity of training samples and improves the image recognition performance.

However, the mixup does not consider the similarity between mixed images. The mixup randomly samples two images from a training set and mixes them with a certain mixing ratio determined under the same conditions for all images. Although we can make a diverse training samples, this diversity affect negative influence on the network training. For example, in case that we use general object recognition dataset such as CIFAR [10] and ImageNet [3] datasets that have wide variety of object classes, random sampling tends to choose images with lower similarity, e.g., plants and fish, rather than those with higher similarity, e.g., different kinds of flowers. In addition to the effect of random sampling, the mixup decides the mixing ratio by using Beta distribution with a single fixed parameter α. In other words, the mixup make mixed samples with the same manner for either lower or higher similarity image pairs. Generating intermediate samples for the higher similarity image pair would effective for learning the relationship between object classes. Meanwhile, the intermediate samples for lower similarity pair would impede to learn such relationship.

In this paper, we propose a super-class mixup, which adjust the mixed images by considering the similarity of object classes. Super-class is a class that classifies each object class by a superordinate object category, which can be defined by WordNet [12], a conceptual dictionary that represents the relationships between things. The proposed method actively generates a mixed image that equally contains the features of both images in case of that these images are categorized in the same super-class. On the other hand, if the super-classes of the images are different, we assume that the images have a low similarity and the proposed method generates a mixed image emphasizing the features of one of them. By using the proposed method, we can focus on learning features between similar classes. Furthermore, we apply the proposed method into the deep mutual learning (DML) [18]. DML uses multiple network and transferring the knowledge, i.e., classification probability obtained from network, as a soft target. Because the proposed method makes a network learn the relationship between object classes, we can improve the soft target from the other network and classification performance. The experimental results show that the proposed method improves the recognition accuracy for training a single model and mutual training. And, we further discuss the effect of the proposed method by using attention maps.

The contributions of this paper are as follows:

- We propose a super-class mixup that considers the similarity between object classes. The proposed method decides the similarity by following the super-class defined by WordNet. The proposed method can learn the relationship of features between similar classes and improve the classification performance.
- The proposed method is useful not only for a single network model training but also for mutual learning framework that uses multiple networks. We can improve the classification probability of the network considering the relationship between classes. The network output is used for the mutual learning as a soft target, which results in the improvement of the classification performance.

– We qualitatively analyze attention maps used for visual explanation. The results show that the proposed method improves the attention maps to capture the target object region.

2 Related Work

2.1 Data augmentation

Data augmentation increase the number of training samples and data variation. It is a fundamental approach to improve recognition accuracy and to prevent overfitting to the training data and is widely used for various computer vision tasks. The classical augmentation is a simple image processing such as transition, resizing, adding noise, and contrast adjustment. In recent years, due to the requirements of the large number of training samples for training deep learning-based method, several augmentation approaches have been proposed. Reinforcement learning has been introduced to decide appropriate augmentation [2]. And, augmentation based on mask processing is also developed [4, 19, 1, 11]. This approach removes the part of an image and use it for training, which is efficient for occlusion and image noise. Cutout [4] is one of mask processing-based augmentation, which removes the part of an image. This is simple and close to the classical image processing approach, but it is effective for improving recognition performance.

Among them, augmentation that uses multiple images to generate a augmented sample is simple and effective approach [17, 16] and widely used in an image recognition task. Mixup [17] is a method of mixing two images and their corresponding labels to generate new mixed data. The mixed image is generated by mixing the entire image with pixel by pixel, and the mixing label represents the mixing ratio of each image. The mixing ratio is decide by the Beta distribution. During the training, the parameter of Beta distribution is fixed, that is, we mix samples with the same conditions even if the object classes of selected two samples are similar or different. This might affect the negative influence on the network training. Our method considers the similarity between the selected images and decide the mixing rate.

2.2 Knowledge distillation and mutual learning

For improving accuracy and shrink the model size retaining the classification accuracy, transferring knowledge from the other network have been developed [8, 18, 6, 13]. This approach trains a network by using the other network output, i.e., classification probability, as an additional supervised label. This additional label is called as a soft-target. This can be categorized into to two approaches. One is the knowledge distillation (KD) [8, 6], which uses two networks. One is a teacher network that is relatively larger pre-trained network model and the other is a smaller student network. KD uses the network output of the teacher network as a soft-target and train a student network in addition to the correct label (hard-target), which improve the performance of the student network.

The other is the deep mutual learning (DML) [18], which is derived from KD. The DML does not use pre-trained model. Instead, each network transfer their knowledge (classification probabilities) as a soft-target mutually. Also, the DML uses the hard-target to train each network. As a result, the DML outperforms the accuracy compared with a single network model training.

The soft-target is depending on the network model. In the above mentioned two approaches, KD might be able to transfer reasonable knowledge because KD uses the pre-trained model as the teacher network. Meanwhile, since DML does not use pre-trained model, the network output at the beginning of training is improper. Using such inappropriate soft-target affects the output of networks and the network performance. In this paper, we apply the proposed method for DML framework. By using the proposed method, network learns the relationship between classes and soft-target, which results in the improvement of accuracy.

3 Proposed Method

We propose a super-class mixup, which adjusts mixed data by considering super-classes that classify object classes in higher categories.

3.1 Preliminaries

Let $D = \{(x_i, y_i)\}_{i=1}^{n}$ be a set of training data, where x_i is a training sample and y_i is the corresponding one-hot encoded label. And, n is the number of training samples. Given the dataset D, we first draw two sets of sample and the label (x_i, y_i) and (x_j, y_j) from D as

$$(x_i, y_i) \sim p(D). \tag{1}$$

Then, mixup [17] generate a augmented sample and label $(\tilde{x}, \tilde{y})$ by mixing two samples, which is defined by

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j \tag{2}$$
$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j, \tag{3}$$

where λ is a mixing ratio which decide the proportion to mix samples.

The mixup probabilistically samples λ from the Beta distribution as

$$\lambda \sim \mathrm{Beta}(\alpha, \alpha), \tag{4}$$

where α is a parameter for the Beta distribution and it adjusts the selection tendency of the mixing ratio. The mixup uses a single fixed value for α throughout the network training[1]. In case of $\alpha \in (0.0, 1.0)$, we tends to select larger or smaller values of λ. This means that the mixup generates a mixed image emphasizing the features of one of them. In contrast, using larger α than 1.0, λ around 0.5 is highly selected, which means that the mixup generates a mixed image that equally contains the features of both images, And, $\alpha = 1.0$ uniformly samples λ.

[1] In [17], $\alpha \in [0.1, 0.4]$ is used for their experiments. In this paper, we discuss the other values of α and the effects of these values in our experiments.

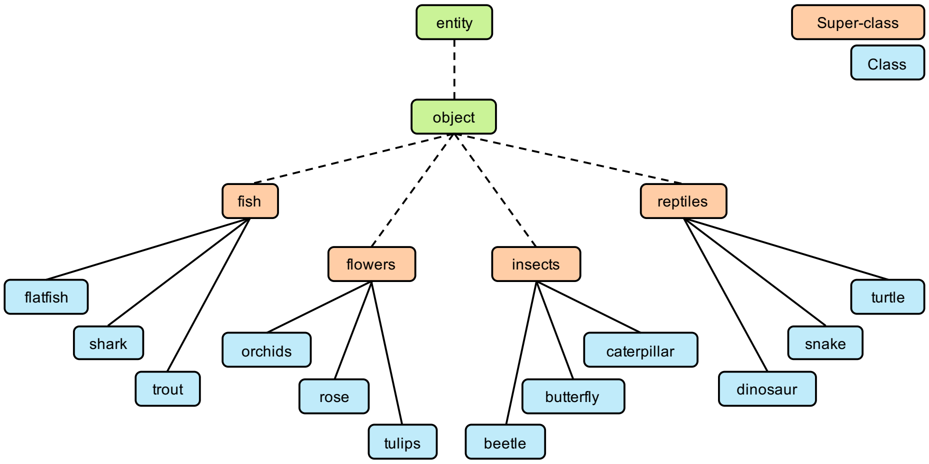

Fig. 1. Example of class structure in WordNet.

3.2 Super-class mixup

As mentioned above, the mixup uses a fixed parameter α throughout the training. However, depending on the similarity between two mixed images and the object classes, appropriate mixing ratio would be different. For the higher similarity image pair, generating intermediate samples would effective for learning the relationship between object classes. The intermediate samples for lower similarity pair would impede to learn such relationship. To overcome the problem and learn the relationship between object classes, we propose a super-class mixup.

Super-class mixup adjusts the mixing ratio considering the similarity of object class between mixed data. As the similarity, we adopt a super-class. The super-class is a class that classifies each object class in a higher category, and it is defined according to WordNet [12] that is a conceptual dictionary. Figure 1 shows an example of class structure in WordNet. We can see that each object class is categorized into the similar object class.

Figure 2 shows the overview of the proposed method. In these figures, “orchids” and “tulips” are the same super-class and “trout” is the different super-class. In this case, the proposed method mixes ‘orchids” and “tulips” class samples with the almost equal proportions. On the other hands, in the case of ”tulip” class and ”trout” classes, we generate a sample with the mixing ratio of one class is higher than the other.

Figure 3 shows an example of mixed images generated by the proposed method. Since the super-classes of the ”orchids” and the ”tulips” are the same, the proposed method selects 0.5 as the intermediate mixing ratio. On the other hand, the super-classes of ”tulips” and ”trout” are different, so the ratio of 0.9 is selected to increase the ratio of ”tulips”, or 0.1 to increase the ratio of ”trout”.

Let $S = \{s_1, \ldots, s_m\}$ is a set of super-class, where m is the number of super-classes. We first prepare the training set with super-class $D_{sc} = \{(x_i, y_i, s_i)\}_{i=1}^{n}$,

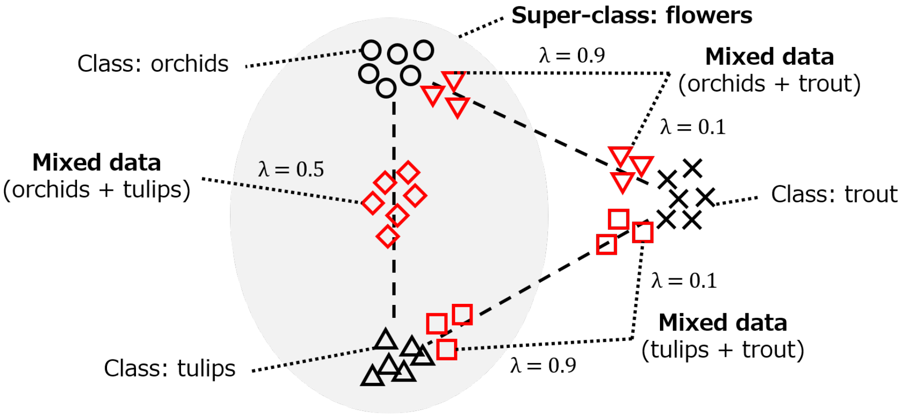

Fig. 2. Overview of adjusting mixed data with super-classes.

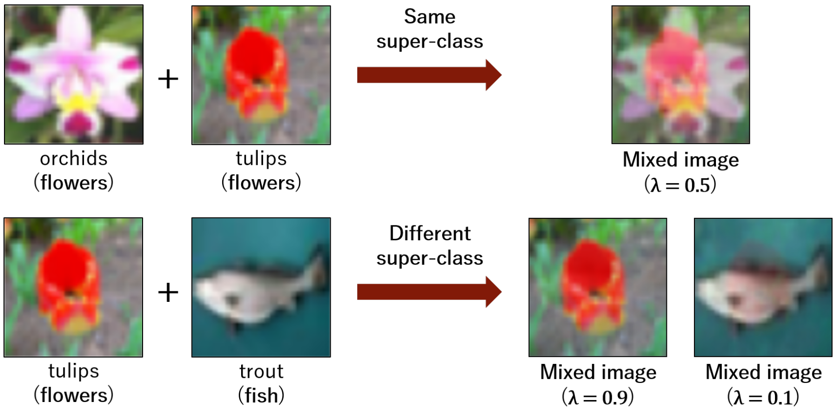

Fig. 3. Example of mixed images generated by the proposed method.

where s_i is the super-class for i-th training sample. In the proposed method, we randomly choose two training samples from D_{sc} as

$$(x_i, y_i, s_i) \sim p(D_{sc}). \tag{5}$$

Given (x_i, y_i, s_i) and (x_j, y_j, s_j), the super-class mixup generates a sample and the label as with the Eqs. (2) and (3). Here, we decides the mixing ratio λ based on the super-class s_i and s_j. In this paper, we propose two approaches for $\lambda(s_i, s_j)$ to select the mixing ratio: i) sampling from predetermined values and ii) sampling from the Beta distribution.

Sampling from predetermined values In this approach, we decide the mixing ratio explicitly. The proposed method samples λ from predetermined values, which is defined by

$$\lambda \sim p(\Lambda), \tag{6}$$

where Λ is a set of predetermined mixing ratio. Based on the super-class, Λ is selected as follows:

$$\Lambda = \begin{cases} \{0.4, 0.5, 0.6\} & (s_i = s_j) \\ \{0.8, 0.9, 1.0\} & (s_i \neq s_j). \end{cases} \tag{7}$$

Sampling from Beta distribution In this approach, we sample λ from the Beta distribution as

$$\lambda \sim \mathrm{Beta}\left(\alpha(s_i, s_j), \alpha(s_i, s_j)\right). \tag{8}$$

The parameter of the Beta distribution $\alpha(s_i, s_j)$ is decided by s_i and s_j, which is defined by

$$\alpha(s_i, s_j) = \begin{cases} 8.0 & (s_i = s_j) \\ 0.2 & (s_i \neq s_j). \end{cases} \tag{9}$$

3.3 Training a single model

In training a single model, we input mixed data using the proposed method and output the prediction probabilities for each class. In other words, it is trained in the same way as the conventional mixup. Since the proposed method trains based on the similarity between the classes defined in WordNet, we can expect to improve the recognition accuracy.

3.4 Training multiple models by deep mutual learning

Deep mutual learning [18] improves the recognition accuracy by using the same input data for multiple models and make their output close to each other. Each model in DML uses the other network output as a label, called as soft-target, in addition to a regular network training with class label, called as hard-target. The loss for hard-target is a cross-entropy loss and the loss for soft-target is calculated by Kullback–Leibler (KL) divergence. The loss function of DML L_{Θ_k} is the sum of the two loss values, which is defined as follows:

$$L_{\Theta_k} = L_{C_k} + \frac{1}{K-1} \sum_{l=1, l \neq k}^{K} D_{KL}(p_l || p_k), \tag{10}$$

where L_{C_k} is the cross-entropy loss function for hard-target, K is the number of models, D_{KL} is the KL-divergence, and p is the prediction probability of each model.

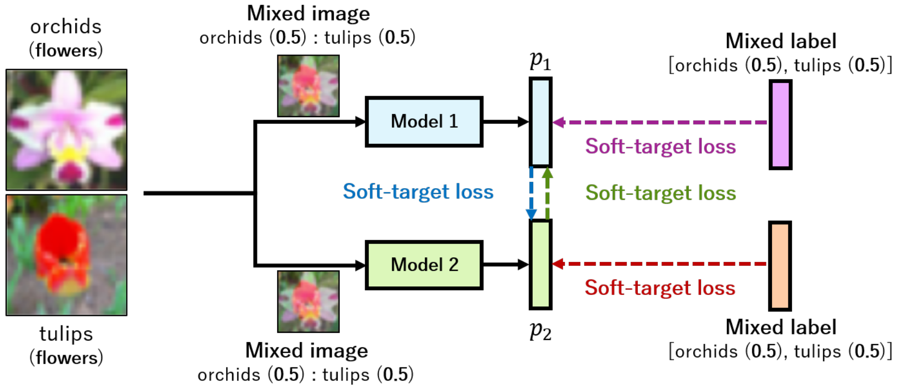

Fig. 4. Overview of the proposed method in DML framework. This shows the case that the super-classes of mixed samples are the same.

Because the prediction probability from a network depends on the model parameters, the network does not always output the desirable prediction probability that we intend. In case that we use a conventional mixup for the DML framework, the mixup generates intermediate mixed data with lower similarity samples because the mixup does not consider the similarity. Such mixed samples causes the complication of network training and decrease the consistency of predicted probability of each object classes. Consequently, the gap of probabilities between each network becomes large.

We apply the proposed method for DML. Because the super-class mixup generates an intended data with the desirable label, we can suppress the generation of such undesirable mixed data. This enables us to train networks efficiently. Figure 4 shows the overview of the proposed method in DML. During the training, we use the same data for the input of multiple networks and the networks predicts object classes. By using the network output, we train the networks so that the output from each network becomes close to each other.

4 Experiment

In this section, we show the results of our evaluation experiments. Specifically, we compare the effectiveness of the proposed method with the conventional method in a single model network training and DML. Then, we further analyse the attention maps as a visual explanation obtained from networks. Finally, we show that the proposed method is equally effective in CutMix.

4.1 Experimental settings

Dataset We use the CIFAR-100 [10] and ImageNet [3] datasets as benchmark dataset.

Table 1. Recognition accuracy in CIFAR-100 [%].

Method		Single	DML	
			Model 1	Model 2
Vanilla		69.64	71.51	71.46
mixup ($\alpha = 1.0$)		70.83	71.37	71.08
Intermediate mixing ratio	Predetermined	61.81	61.16	61.54
	mixup ($\alpha = 8.0$)	64.17	62.73	62.26
Unbalanced mixing ratio	Predetermined	70.98	72.06	71.80
	mixup ($\alpha = 0.2$)	71.39	72.20	72.32
Ours	Predetermined	71.32	72.23	71.93
	Beta dist.	**71.68**	**73.28**	**73.01**

Table 2. Recognition accuracy in ImageNet [%].

Method		Single	DML	
			Model 1	Model 2
Vanilla		73.04	73.55	73.39
mixup ($\alpha = 0.2$)		73.20	**73.85**	73.56
Ours	Predetermined	72.02	73.00	73.15
	Beta dist.	**73.41**	73.73	**73.92**

Network models We use ResNet [7] as the network model. For CIFAR-100 dataset, we use ResNet-20, -32, -44, -56, and -110. For ImageNet dataset, we use ResNet-34.

To visualize attention maps of trained network model, we adopt an attention branch network (ABN) [5]. ABN consists of the backbone network, that extracts features from an image and predicts classification result, and attention branch, that visualizes compute and output attention map. We used ResNet-32 as the backbone network of ABN and CIFAR-100 dataset.

For training each network, we set the mini-batch size as 128 for both dataset. The number of training epochs are 200 for CIFAR-100 and 90 for ImageNet, respectively.

4.2 Comparison of recognition accuracy for each dataset

Here, we compare the performance on CIFAR-100 and ImageNet dataset. As network model, we use ResNet-32[7] for CIFAR-100 and ResNet-34 for ImageNet. Also, as a comparative methods, we evaluate the performance of a mixup using an intermediate or unbalanced mixing ratio regardless of the super-class. In the case of the predetermined, we randomly select 0.4, 0.5, or 0.6 as the intermediate mixing ratio when the super-classes are the same, or 0.8, 0.9, or 1.0 as the unbalanced mixing ratio when the super-classes are different. In the case of the

Table 3. Variation of recognition accuracy with model size (Single model) [%].

Method		ResNet-20	ResNet-32	ResNet-44	ResNet-56	ResNet-110
Vanilla		68.38	69.64	70.88	71.99	73.65
mixup ($\alpha = 1.0$)		68.59	70.83	72.57	73.12	**74.19**
Ours	Predetermined	69.22	71.32	71.96	72.40	73.73
	Beta dist.	**69.73**	**71.68**	**73.01**	**73.49**	74.15

Table 4. Variation of recognition accuracy with model size (DML) [%]. M1 and M2 in the table are the first and second model of DML, respectively.

Method		ResNet-20		ResNet-32		ResNet-44		ResNet-56		ResNet-110	
		M1	M2	M1	M2	M1	M2	M1	M2	M1	M2
Vanilla		70.11	70.35	71.51	71.46	73.07	72.93	73.70	73.91	74.95	74.79
mixup ($\alpha = 1.0$)		69.40	69.68	71.37	71.08	72.10	72.03	73.77	73.51	**75.56**	**75.33**
Ours	Predetermined	69.74	70.02	72.23	71.93	72.35	72.51	73.07	72.85	73.79	73.42
	Beta dist.	**71.24**	**71.19**	**73.28**	**73.01**	**73.77**	**73.58**	**74.54**	**74.06**	75.02	75.17

beta distribution, we set the hyper-parameter α to 8.0 when the super-classes are the same, and set 0.2 when the super-classes are different. The value of α in conventional mixup is 1.0 for CIFAR-100 and 0.2 for ImageNet.

Table 1 shows the recognition accuracy of each method in CIFAR-100, and Table 2 shows the recognition accuracy in ImageNet. From Table 1, the proposed method achieves the highest accuracy in CIFAR-100 for both single model and DML, which is further improved by using beta distribution. Table 2 also shows that the proposed method has the highest accuracy in ImageNet for a single model, and it slightly outperforms the conventional method in DML overall. Therefore, the proposed method is effective for single model and DML. In addition, the results indicates that it is important to include a few data that is mixed with different super-class.

4.3 Comparison of recognition accuracy for different model sizes

We compare the recognition accuracy of different models in CIFAR-100. We use ResNet-20, 32, 44, 56, and 110. The settings for each parameter are the same as in Section 4.2.

Table 3 and Table 4 show the recognition accuracy of each model in single model and DML. From these Tables, the proposed method achieves the highest accuracy for all the models except ResNet-110 in single model and DML. We believe that ResNet-110 can sufficiently improve the generalization performance using mixed data with high diversity by the conventional method. Therefore, the proposed method is effective for lightweight training models.

Table 5. Recognition accuracy of ABN on CIFAR-100 dataset [%].

Method	Single	DML	
		Model 1	Model 2
Vanilla	71.79	74.25	74.12
mixup ($\alpha = 1.0$)	71.70	71.48	71.45
Ours (Beta dist.)	**73.26**	**74.71**	**74.94**

Table 6. Recognition accuracy of CutMix on CIFAR-100 dataset [%].

Method		Single	DML	
			Model 1	Model 2
Vanilla		69.64	71.51	71.46
CutMix ($\alpha = 1.0$)		70.57	72.26	71.92
Ours	Predetermined	72.29	73.82	73.68
	Beta dist.	**73.11**	**74.61**	**74.43**

4.4 Qualitative evaluation of attention map

Next, we qualitatively analyse the attention maps obtained from each network. As mentioned in Sec. 4.1, we use ABN [5] with ResNet-32 backbone as a network model and train the ABN with CIFAR-100 dataset.

Table 5 shows the accuracy of ABN for each methods. In the both of single model training and DML, our method outperforms the other methods. This results show that our method is also effective for the other network model excepting for ResNet.

Figure 5 shows examples of the attention maps obtained from each method. In the top of Fig. 5, the target object of the input image is "boy" in the left part of the image. On the other hand, the attention maps of conventional mixup is widely distributed and does not focuses on the target object correctly. Also, the proposed method focuses around the shoulder although the conventional methods highlights the other regions. In the bottom of Fig. 5, the target object is "apple." The conventional mixup highlights background regions. The proposed method focuses on the three apples correctly. Therefore, the proposed method can capture the features of the recognition target more accurately while considering a wide range of important information.

4.5 Comparison of recognition accuracy in CutMix

Finally, we show the effectiveness of the proposed method in CutMix [16], a derivative of mixup. We compare the recognition accuracy on CIFAR-100. We use ResNet-32. The settings for each parameter are the same as in Section 4.2.

Table 6 shows the recognition accuracy. From Table 6, the proposed method has the highest accuracy in single model and DML. Therefore, the proposed

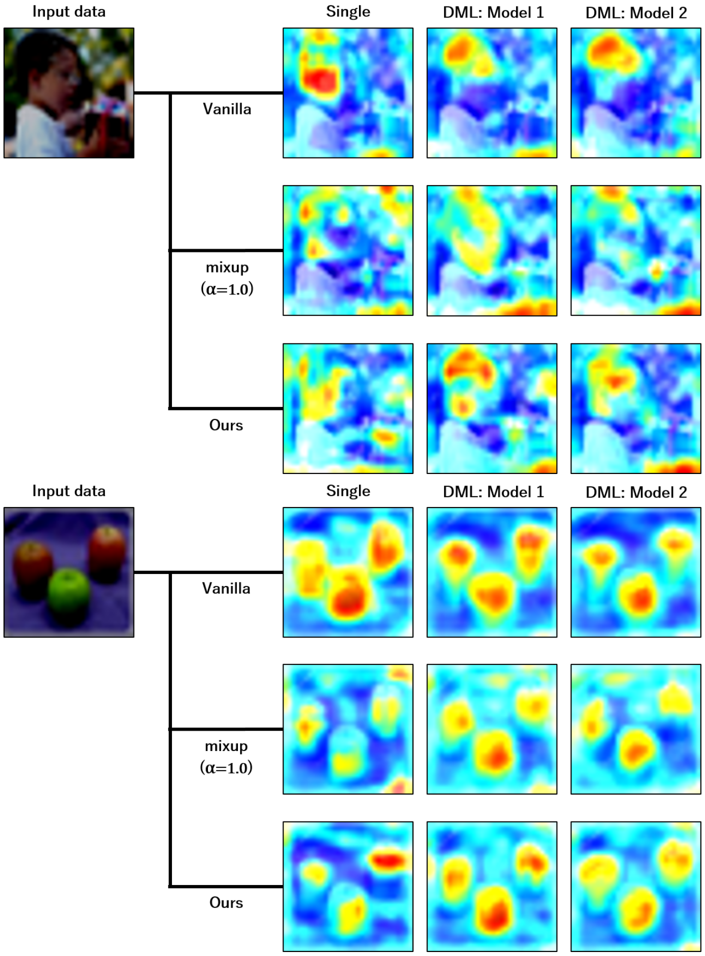

Fig. 5. Examples of attention maps for each method.

method is also effective in improving the recognition accuracy in CutMix. This result suggests that the proposed method may be applied to various data augmentation methods for mixing data. Currently, we are considering using the proposed method in combination with other state-of-the-art data expansion methods (Manifold Mixup [15], SaliencyMix [14], and Puzzle Mix [9]).

5 Conclusion

In this paper, we proposed super-class mixup, a effective data augmentation method considering super-class. The proposed method adjusts the mixed ratio by the similarity between the object classes. The experimental results with

CIFAR-100 and ImageNet datasets show that our method improved the recognition accuracy on a single network model training and deep mutual learning framework. Moreover, we analyzed the attention maps as a visual explanation. As a result, our method improves the highlighted region to the target object correctly.

Our future work includes detailed analysis with respect to the obtained feature spaces, the effect for the improvement of the mis-classified samples. Also, we further extend the proposed method to dynamically decide the parameters during training phase and we will combine the existing another data augmentation methods.

References

1. Chen, P., Liu, S., Zhao, H., Jia, J.: Gridmask data augmentation. arXiv preprint **arXiv:2001.04086** (2020)
2. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation strategies from data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
3. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2009)
4. Devries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint **arXiv:1708.04552** (2017)
5. Fukui, H., Hirakawa, T., Yamashita, T., Fujiyoshi, H.: Attention branch network: Learning of attention mechanism for visual explanation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
6. Furlanello, T., Lipton, Z., Tschannen, M., Itti, L., Anandkumar, A.: Born again neural networks. In: Proceedings of the International Conference on Machine Learning (ICML). Proceedings of Machine Learning Research, vol. 80, pp. 1607–1616 (2018)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
8. Hinton, G.E., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. In: Proceedings of NIPS workshop on Deep Learning and Representation Learning (2014)
9. Kim, J.H., Choo, W., Song, H.O.: Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In: Proceedings of the International Conference on Machine Learning (ICML) (2020)
10. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Tech. rep., University of Tront (2009)
11. Kumar Singh, K., Jae Lee, Y.: Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017)
12. Miller, G.A.: Wordnet: A lexical database for english. Commun. ACM **38**(11), 39–41 (Nov 1995)
13. Mirzadeh, S.I., Farajtabar, M., Li, A., Levine, N., Matsukawa, A., Ghasemzadeh, H.: Improved knowledge distillation via teacher assistant. Proceedings of the AAAI Conference on Artificial Intelligence **34**(04), 5191–5198 (2020)

14. Uddin, A.F.M.S., Monira, M.S., Shin, W., Chung, T., Bae, S.H.: Saliencymix: A saliency guided data augmentation strategy for better regularization. In: International Conference on Learning Representations (2021)
15. Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., Bengio, Y.: Manifold mixup: Better representations by interpolating hidden states. In: Proceedings of the International Conference on Machine Learning (ICML). Proceedings of Machine Learning Research, vol. 97, pp. 6438–6447 (2019)
16. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2019)
17. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond Empirical Risk Minimization. In: International Conference on Learning Representations (2018)
18. Zhang, Y., Xiang, T., Hospedales, T.M., Lu, H.: Deep mutual learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
19. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) (2020)